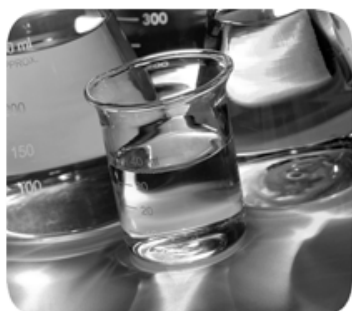


# Manuel de référence des Automates

## Logix 5000 – Instructions

1756 ControlLogix, 1756 GuardLogix, 1769 CompactLogix, 1769 Compact GuardLogix, 1789 SoftLogix, 5069 CompactLogix, Emulate 5570



# Informations aux utilisateurs importantes

Lisez ce document et les documents répertoriés dans la section de la documentation supplémentaire concernant l'installation, la configuration et l'utilisation de cet équipement avant d'installer, de configurer, de faire fonctionner ou d'entretenir ce produit. Les utilisateurs sont tenus de se familiariser avec les instructions d'installation et de câblage, outre les conditions stipulées dans tous les codes, lois et normes en vigueur.

Les activités, notamment l'installation, les réglages, la mise en service, l'utilisation, le montage, le démontage et la maintenance, doivent être effectuées par du personnel dûment formé, conformément au code de pratique en vigueur. Si cet équipement est utilisé d'une manière qui n'est pas indiquée par le fabricant, la protection fournie par l'équipement peut être compromise.

Rockwell Automation, Inc. ne sera en aucun cas responsable des dommages indirects ou consécutifs, résultant de l'utilisation de cet équipement.

Les exemples et diagrammes de ce manuel ne sont inclus qu'à des fins d'illustration. En raison des nombreuses variables et conditions associées à une installation particulière, Rockwell Automation, Inc. ne peut pas assumer la responsabilité d'une utilisation réelle reposant sur les exemples et les diagrammes.

Rockwell Automation, Inc. n'assume aucune responsabilité en matière de brevet concernant l'utilisation des informations, circuits, de l'équipement ou des logiciels décrits dans ce manuel.

La reproduction de ce manuel, tout ou en partie, sans l'autorisation écrite de Rockwell Automation, Inc., est interdite.

Lorsque cela est nécessaire, nous utilisons dans ce manuel des notes, visant à vous tenir informé de la sécurité.



**AVERTISSEMENT** : indique des informations sur les pratiques ou circonstances pouvant provoquer une explosion dans un environnement dangereux, pouvant entraîner des blessures ou la mort, des dégâts matériels ou pertes économiques.



**ATTENTION** : indique des informations concernant des pratiques ou circonstances pouvant entraîner des blessures ou la mort, ainsi que des dégâts matériels ou des pertes économiques. Les mentions Attention vous aident à découvrir un danger, à en éviter un et à en découvrir les conséquences

---

**Important** : Indique des informations essentielles à l'utilisation et à la connaissance du produit.

---

Des étiquettes spécifiques de précautions peuvent également se trouver dessus ou à l'intérieur de l'équipement.



**RISQUE DE CHOC** : des étiquettes peuvent être apposées sur l'équipement ou à l'intérieur de celui-ci, par exemple un variateur ou un moteur, pour alerter chacun de la présence possible d'une tension dangereuse.



**RISQUE DE BRÛLURE** : des étiquettes peuvent être apposées sur l'équipement ou à l'intérieur de celui-ci, par exemple un variateur ou un moteur, pour alerter chacun que des surfaces peuvent atteindre des températures dangereuses.

---



**RISQUE D'ARC ÉLECTRIQUE** : des étiquettes peuvent être apposées sur ou à l'intérieur de l'équipement, par exemple un centre de commande de moteur, pour avertir les gens d'un risque d'arc électrique. Un arc électrique causera des blessures graves ou la mort. Portez un Équipement de protection individuelle (PPE). Respectez TOUTES les obligations réglementaires relatives aux méthodes de travail en toute sécurité et à l'équipement de protection individuelle (PPE).

---

Allen-Bradley, Rockwell Software, Rockwell Automation et TechConnect sont des marques déposées de Rockwell Automation, Inc.

Les marques n'appartenant pas à Rockwell Automation, sont détenues par leurs sociétés respectives.



## Résumé des modifications

---

Ce manuel contient de nouvelles informations ainsi que des mises à jour. Utilisez ces tableaux de référence pour trouver les informations qui ont été modifiées.

### Modifications globales

Aucune pour cette version.

### Fonctionnalités nouvelles ou améliorées

Ce tableau contient une liste des rubriques modifiées dans cette version, la raison de la modification et une liaison vers la rubrique qui contient les informations modifiées.

Nom de rubrique	Raison
<a href="#">Opération d'alarme définie (ASO)</a> sur la <a href="#">page 69</a>	Nouvelle instruction d'alarme
<a href="#">Instructions d'alarme</a> sur la <a href="#">page 27</a>	Ajout de l'instruction Opération d'alarme définie (ASO) à la rubrique.
<a href="#">Examiner si fermé (XIC)</a> sur la <a href="#">page 74</a>	Ajout de nouveaux types de données
<a href="#">Examiner si ouvert (XIO)</a> sur la <a href="#">page 76</a>	Ajout de nouveaux types de données
<a href="#">Activation de sortie (OTE)</a> sur la <a href="#">page 93</a>	Ajout de nouveaux types de données
<a href="#">Verrouillage de sortie (OTL)</a> sur la <a href="#">page 95</a>	Ajout de nouveaux types de données
<a href="#">Déverrouillage de sortie (OTU)</a> sur la <a href="#">page 97</a>	Ajout de nouveaux types de données
<a href="#">Instructions de comparaison</a> sur la <a href="#">page 297</a>	Ajout de nouvelles illustrations graphiques de fonctions de diagramme de bloc fonctionnel.
<a href="#">Égal à (EQU)</a> sur la <a href="#">page 302</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Supérieur à (GRT)</a> sur la <a href="#">page 310</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Supérieur ou égal à (GEQ)</a> sur la <a href="#">page 319</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Inférieur à (LES)</a> sur la <a href="#">page 327</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Inférieur ou égal à (LEQ)</a> sur la <a href="#">page 336</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.

Nom de rubrique	Raison
<a href="#">Limite (LIM)</a> sur la <a href="#">page 345</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Masque égal à (MEQ)</a> sur la <a href="#">page 354</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Différent de (Fonction NEQ)</a> sur la <a href="#">page 363</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Valeur absolue (ABS)</a> sur la <a href="#">page 374</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Addition (ADD)</a> sur la <a href="#">page 380</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Calcul (CPT)</a> sur la <a href="#">page 387</a>	Ajout de nouveaux types de données
<a href="#">Division (DIV)</a> sur la <a href="#">page 391</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Modulo (MOD)</a> sur la <a href="#">page 397</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Multiplication (MUL)</a> sur la <a href="#">page 404</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Négation (NEG)</a> sur la <a href="#">page 411</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Racine carrée (SQR/SQRT)</a> sur la <a href="#">page 417</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Soustraction (SUB)</a> sur la <a href="#">page 424</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">ET booléen (BAND)</a> sur la <a href="#">page 460</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">OU exclusif booléen (BXOR)</a> sur la <a href="#">page 466</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">NON booléen (BNOT)</a> sur la <a href="#">page 470</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">OU booléen (BOR)</a> sur la <a href="#">page 474</a>	Ajout de nouveaux types de données et d'un nouveau langage de fonctions de diagramme de bloc fonctionnel.
<a href="#">Recherche et comparaison de fichier (FSC)</a> sur la <a href="#">page 533</a>	Modification du bit .POS vers .POS dans la section Description. Suppression du tableau Opérateurs valides et remplacement de celui-ci par un lien vers la rubrique Opérateurs valides.

Nom de rubrique	Raison
<a href="#">Arithmétique et logique des fichiers (FAL)</a> sur la <a href="#">page 509</a>	Suppression du tableau Opérateurs valides et remplacement de celui-ci par un lien vers la rubrique Opérateurs valides.
<a href="#">Opérateurs valides</a> sur la <a href="#">page 371</a>	Mise à jour du tableau pour inclure Autorisé dans les colonnes et les lignes pour les instructions applicables.
<a href="#">Pour (FOR)</a> sur la <a href="#">page 671</a>	Mise à jour de la description des fins de boucles.
<a href="#">Proportionnel, intégral et dérivé (PID)</a> sur la <a href="#">page 705</a>	Mise à jour de la description mnémotechnique .CTL pour le bit .CA pour l'action de contrôle (0=inverse (SP-PV) ; 1=direct (PV-SP)).
<a href="#">Validation de licence (LV)</a> sur la <a href="#">page 881</a>	Nouvelle instruction.
<a href="#">Attributs communs</a> sur la <a href="#">page 885</a>	Ajout d'un lien à la rubrique Types de données élémentaires.
<a href="#">Valeurs immédiates</a> sur la <a href="#">page 888</a>	Ajout des tableaux Valeurs immédiates entières et Valeurs immédiates à virgule flottante.
<a href="#">Conversions de données</a> sur la <a href="#">page 889</a>	Modification des types de données Optimaux en types de données intermédiaires et inclusion des types de données étendues USINT, INT, UINT, UDINT, ULINT et LREAL. Dans la section Conversion de SINT ou INT vers DINT, ajout de la conversion DINT vers LINT. Inclusion des données de conversion pour 32 et 64 bits.
<a href="#">Types de données élémentaires</a> sur la <a href="#">page 893</a>	Modification du titre de rubrique de Types de données en Types de données élémentaires. Ajout de LINT, USINT, UINT, UDINT, ULINT, REAL et LREAL.
<a href="#">Types de données LINT</a> sur la <a href="#">page 896</a>	Ajout d'une liste d'automates applicables prenant en charge les types de données LINT utilisés dans les instructions.
<a href="#">Valeurs de virgules flottantes</a> sur la <a href="#">page 896</a>	Ajout d'une liste d'automates applicables. Ajout de la description de l'étiquette LREAL.
<a href="#">Indexer via des tableaux</a> sur la <a href="#">page 899</a>	Ajout de deux nouvelles astuces expliquant que Logix Designer autorise les indices qui sont uniquement des étiquettes de types de données étendues. Explication également de l'utilisation de tous les types de données élémentaires entiers disponibles comme index d'indice.
<a href="#">Adressage de bits</a> sur la <a href="#">page 900</a>	Ajout de nouvelles définitions.
<a href="#">FOR_DO</a> sur la <a href="#">page 935</a>	Mise à jour de la description des fins de boucles.





## Localisateur d'instruction

Utiliser ce localisateur pour trouver le manuel d'instructions applicable de chaque instruction pour les automates Logix5000.

<b>Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003</b>	<b>Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006</b>	<b>Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002</b>
Valeur absolue (ABS)	Alarme (ALM)	Contrôle coordonné piloté par le maître (MDCC)
Addition (ADD)	Attacher à la phase d'équipement (PATT)	Réglages d'axe d'application de mouvement (MAAT)
Alarme analogique (ALMA)	Attacher à la séquence d'équipement (SATT)	Diagnostics de raccordement d'application de mouvement (MAHD)
Toujours faux (AFI)	Contrôle coordonné (CC)	Came de sortie d'armement de mouvement (MAOC)
Cosinus d'arc (ACS, ACOS)	Basculeur D (DFF)	Registration d'armement de mouvement (MAR)
Sinus d'arc (ASN, ASIN)	Temps mort (DEDT)	Surveillance d'armement de mouvement (MAW)
Tangente d'arc (ATN, ATAN)	Dérivée (DERV)	Réinitialisation de défaut d'axe de mouvement (MAFR)
Caractères ASCII dans le tampon (ACB)	Détacher de la phase d'équipement (PDET)	Transmission d'axe de mouvement (MAG)
Effacement ASCII dans le tampon (ACL)	Détacher de la séquence d'équipement (SDET)	Prise d'origine d'axe de mouvement (MAH)
Lignes d'échange ASCII (AHL)	Dispositif à 3 états discret (D3SD)	Mouvement d'axe en marche par à-coups (MAJ)
Lecture ASCII (ARD)	Dispositif à 2 états discret (D2SD)	Mouvement d'axe de mouvement (MAM)
Ligne de lecture ASCII (ARL)	PID amélioré (PIDE)	Came de position d'axe de mouvement (MAPC)
Test ASCII pour la ligne de tampon (ABL)	Sélection améliorée (ESEL)	Arrêt d'axe de mouvement (MAS)
Écriture ASCII (AWT)	Échec de suppression de phase d'équipement (PCLF)	Came de temps d'axe de mouvement (MATC)
Ajout d'écriture ASCII (AWA)	Commande de phase d'équipement (PCMD)	Arrêt d'axe de mouvement (MASD)
Répartition champ de bits (BTD)	Requête externe de phase d'équipement (PXRQ)	Réinitialisation d'arrêt d'axe de mouvement (MASR)
Répartition de champ de bits avec cible (BTDT)	Échec de phase d'équipement (PFL)	Profil de came de calcul de mouvement (MCCP)
Décalage binaire à gauche (BSL)	Nouveaux paramètres de phase d'équipement (PRNP)	Mouvement du chemin coordonné de mouvement (MCPM)
Décalage binaire à droite (BSR)	Commande d'outrepassement de phase d'équipement (POVR)	Valeurs esclaves de calcul de mouvement (MCSV)

<b>Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003</b>	<b>Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006</b>	<b>Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002</b>
ET au niveau du bit (AND)	Pause de phase d'équipement (PPD)	Transformation coordonnée de mouvement avec orientation (MCTO)
NON au niveau du bit (NOT)	Identificateur de séquence d'attribution de séquence d'équipement (SASI)	Position de transformation de calcul de mouvement (MCTP)
OU au niveau du bit (OR)	Échec de suppression de séquence d'équipement (SCLF)	Position de transformation de calcul de mouvement avec orientation (MCTPO)
Booléen ET (BAND)	Commande de séquence d'équipement (SCMD)	Dynamiques de changement de mouvement (MCD)
Booléen exclusif OU (BXOR)	Outrepassement de séquence d'équipement (SOVR)	Dynamiques de changement coordonnée de mouvement (MCCD)
Booléen NOT (BNOT)	Générateur de fonction (FGEN)	Mouvement circulaire coordonné de mouvement (MCCM)
Booléen OR (BOR)	Filtre passe haut (HPF)	Mouvement linéaire coordonné de mouvement (MCLM)
Rupture (BRK)	Limite haute/basse (HLL)	Arrêt coordonné de mouvement (MCSD)
Points d'arrêt (BPT)	Intégrateur (INTG)	Réinitialisation d'arrêt coordonné de mouvement (MCSR)
Effacer (CLR)	Contrôle de modèle interne (IMC)	Arrêt coordonné du mouvement (MCS)
Comparer (CMP)	Basculeur JK (JKFF)	Transformation coordonnée de mouvement (MCT)
Conversion en BCD (TOD)	Avance-retard (LDLG)	Commande directe de mouvement arrêté (MDF)
Convertir en entier (FRD)	Filtre passe bas (LPF)	Commande directe de mouvement activé (MDO)
Copie de fichier (COP), Copie synchrone de fichier (CPS)	Capture maximum (MAXC)	Démarrage direct de mouvement (MDS)
Cosinus (COS)	Capture minimum (MINC)	Came de sortie de désarmement de mouvement (MDOC)
Calcul (CPT)	Contrôle multivariable modulaire (MMC)	Registration de désarmement de mouvement (MDR)
Comptage dégressif (CTD)	Moyenne de mouvement (MAVE)	Surveillance de désarmement de mouvement (MDW)
Comptage progressif (CTU)	Déviations standard de mouvement (MSTD)	Arrêt de groupe de mouvement (MGSD)
Comptage progressif/dégressif (CTUD)	Multiplexeur (MUX)	Réinitialisation d'arrêt de groupe de mouvement (MGSR)
Transition de données (DTR)	Filtre réjecteur (NTCH)	Arrêt de groupe de mouvement (MGS)
Degrés (DEG)	État de phase terminé (PSC)	Position de strobe de groupe de mouvement (MGSP)

<b>Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003</b>	<b>Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006</b>	<b>Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002</b>
Détection de diagnostic (DDT)	Position proportionnelle (POSP)	Position de redéfinition de mouvement (MRP)
Alarme numérique (ALMD)	Proportionnel + Intégral (PI)	Réglage d'axe d'exécution de mouvement (MRAT)
DINT vers chaîne (DTOS)	Multiplicateur d'impulsion (PMUL)	Diagnostics de raccordement d'exécution de mouvement (MRHD)
Division (DIV)	Rampe/Stabilisation (RMPS)	Désactivation de mouvement asservi (MSF)
Fin de transition (EOT)	Limiteur de variation (RLIM)	Activation de mouvement asservi (MSO)
Égal à (EQU)	Réinitialisation dominante (RESD)	
Arithmétique de fichier (FAL)	Échelle (SCL)	
Comparaison de fichier de bits (FBC)	Courbe en S (SCRV)	
Chargement de FIFO (FFL)	Automate de deuxième ordre (SOC)	
Déchargement de FIFO (FFU)	Avance-retard de deuxième ordre (LDL2)	
Moyenne de fichier (AVE)	Sélection (SEL)	
Écart standard de fichier (STD)	Négatif sélectionné (SNEG)	
Remplissage de fichier (FLL)	Additionneur sélectionné (SSUM)	
Tri de fichier (SRT)	Réglage Dominant (SETD)	
Rechercher une chaîne (FIND)	Temps proportionnel de Split Range (SRTP)	
Pour (FOR)	Totalisateur (TOT)	
Recherche et comparaison de fichier (FSC)	Totalisateur progressif/dégressif (UPDN)	
Lire la valeur système (GSV) et Définir la valeur système (SST)		
Supérieur ou égal à (GEQ)		
Supérieur à (GRT)		
Insertion d'une chaîne (INSERT)		
Sortie immédiate (IOT)		
Saut vers une étiquette (JMP) et Étiquette (LBL)		
Saut vers sous-routine (JSR), Sous-routine (SBR), et Retour (RET)		
Saut vers routine externe (JXR)		
Inférieur à (LES)		

Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003	Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006	Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002
Inférieur ou égal à (LEQ)		
Chargement de LIFO (LFL)		
Déchargement de LIFO (LFU)		
Validation de license (LV)		
Limite (LIM)		
Logarithme décimal (LOG)		
Minuscule (LOWER)		
Mouvement masqué (MVM)		
Mouvement masqué avec cible (MVMT)		
Réinitialisation de contrôle maître (MCR)		
Égal à masqué (MEQ)		
Message (MSG)		
Chaîne médiane (MID)		
Modulo (MOD)		
Mouvement (MOV)		
Multiplier (MUL)		
Logarithme naturel (LN)		
Négation (NEG)		
Différent de (NEQ)		
Pas d'opération (NOP)		
Impulsion (ONS)		
Impulsion sur front descendant (OSF)		
Impulsion sur front descendant avec entrée (OSFI)		
Impulsion sur front montant (OSR)		
Impulsion sur front montant avec entrée (OSRI)		
Activation de sortie (OTE)		
Verrouillage de sortie (OTL)		
Déverrouillage de sortie (OTU)		
Proportionnel, intégral et dérivé (PID)		
Radian (RAD)		
Real vers chaîne (RTOS)		
Réinitialisation (RES)		
Réinitialisation SFC (SFR)		

<b>Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003</b>	<b>Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006</b>	<b>Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002</b>
Retour (RET)		
Temporisateur de rémanence activé (RTO)		
Temporisateur de rémanence activé avec réinitialisation (RTOR)		
Pause SFC (SFP)		
Taille en éléments (SIZE)		
Entrée de séquenceur (SQI)		
Chargement de séquenceur (SQL)		
Sortie de séquenceur (SQO)		
Sinus (SIN)		
Racine carrée (SQR/SQRT)		
Concaténation de chaîne (CONCAT)		
Effacement de chaîne (DELETE)		
Chaîne vers DINT (STOD)		
Chaîne vers REAL (STOR)		
Permutation d'octet (SWPB)		
Soustraction (SUB)		
Tangente (TAN)		
Temporisateur au déclenchement (TOF)		
Temporisateur au déclenchement avec réinitialisation (TOFR)		
Temporisateur à l'enclenchement (TON)		
Temporisateur à l'enclenchement avec réinitialisation (TONR)		
Fin temporaire (TND)		
Points de suivi (TPT)		
Tâche de déclenchement d'événement (EVENT)		
Troncation (TRN)		
Instruction inconnue (UNK)		
Majuscules (UPPER)		
Désactivation d'interruption par l'utilisateur (UID)/Activation d'interruption par l'utilisateur (UIE)		
X à la puissance Y (XPY)		
Examiner si fermé (XIC)		

## Localisateur d'instruction

---

<b>Manuel de référence des Automates Logix5000™ – Instructions, 1756-RM003</b>	<b>Manuel de référence des instructions de séquence de phase des variateurs /équipements et contrôle avancé du processus de l'automate LOGIX 5000, 1756-RM006</b>	<b>Logix5000 Controllers Motion Instructions Reference Manual MOTION-RM002</b>
Examiner si ouvert (XIO)		
Exclusif sur bits (XOR)		

<b>Préface</b>	Environnement studio 5000 ..... 23
	Ressources supplémentaires ..... 24
	Mentions légales ..... 24
 <b>Chapitre 1</b>	
<b>Instructions d'alarme</b>	Instructions d'alarme ..... 27
	Alarme analogique (ALMA) ..... 28
	Alarme numérique (ALMD) ..... 56
	Opération d'ensemble d'alarmes (ASO) ..... 69
 <b>Chapitre 2</b>	
<b>Instructions de bit</b>	Instructions de bit ..... 73
	Examiner si fermé (XIC) ..... 74
	Examiner si ouvert (XIO) ..... 76
	Impulsion (ONS) ..... 78
	Impulsion sur front descendant (OSF) ..... 80
	Impulsion sur front descendant avec entrée (OSFI) ..... 83
	Impulsion sur front montant (OSR) ..... 86
	Impulsion sur front montant avec entrée (OSRI) ..... 90
	Activation de sortie (OTE) ..... 93
	Verrouillage de sortie (OTL) ..... 95
	Déverrouillage de sortie (OTU) ..... 97
 <b>Chapitre 3</b>	
<b>Instructions de temporisateur et de compteur</b>	Instructions de temporisateur et de compteur ..... 101
	Comptage dégressif (CTD) ..... 102
	Comptage progressif (CTU) ..... 107
	Comptage progressif/dégressif (CTUD) ..... 112
	Réinitialisation (RES) ..... 117
	Temporisateur de rémanence activé (RTO) ..... 120
	Temporisateur de rémanence activé avec réinitialisation (RTOR) ..... 125
	Temporisateur au déclenchement (TOF) ..... 130
	Temporisateur au déclenchement avec réinitialisation (TOFR) ..... 135
	Temporisateur à l'enclenchement (TON) ..... 140
	Temporisateur à l'enclenchement avec réinitialisation (TONR) ..... 145
 <b>Chapitre 4</b>	
<b>Entrée/Sortie</b>	Instructions d'entrée/sortie ..... 151
	Message (MSG) ..... 152

Exemples de configuration MSG .....	162
Types et codes des défauts majeurs .....	163
Types et codes des défauts mineurs.....	169
Codes d'erreur des messages .....	172
Codes d'erreur .....	172
Codes d'erreur étendus.....	174
Codes d'erreur PLC et SLC (.ERR) .....	176
Codes d'erreur des blocs-transferts .....	178
Spécifier les détails de communication .....	179
Spécification de messages SLC .....	189
Spécification de messages de bloc-transfert.....	189
Lire la valeur système (GSV) et Définir la valeur système (SSV) .....	190
Sortie immédiate (IOT).....	194
Accéder aux valeurs système.....	198
Déterminer les informations relatives à la mémoire de l'automate .....	198
Codes d'état de DeviceNet .....	201
Obtenir et définir une valeur système .....	204
Exemple de programmation GSV/SSV .....	206
Objets GSV/SSV .....	210
Accès à l'objet AddOnInstructionDefinition.....	211
Accès à l'objet ALARMBUFFER .....	212
Accès à l'objet Axis .....	215
Accès à l'objet Controller.....	226
Accès à l'objet ControllerDevice .....	228
Accès à l'objet CoordinateSystem .....	230
Accès à l'objet MotionGroup.....	232
Accès à l'objet Message .....	234
Accès à l'objet CST .....	235
Accès à l'objet Datalog.....	236
Accès à l'objet DF1 .....	238
Accès à l'objet FaultLog.....	241
Accès à l'objet HardwareStatus.....	242
Accès à l'objet Message .....	244
Accès à l'objet Module.....	244
Accès à l'objet Routine .....	247
Accès à l'objet Redondance.....	247
Accès à l'objet Programme .....	252
Accéder à l'objet Safety.....	252
Accès à l'objet SerialPort .....	254
Accès à l'objet Tâche.....	255
Accès à l'objet TimeSynchronize.....	257
Accès à l'objet WallClockTime.....	262
Objets de Sécurité GSV/SSV .....	263
Contrôle des indicateurs d'état.....	269
Sélection du type de message.....	269
Défauts du Module : 16#0000 - 16#00ff.....	271
Défauts du Module : 16#0100 - 16#01ff.....	273



Défauts du Module : 16#0200 - 16#02ff.....	279
Défauts du Module : 16#0300 - 16#03ff.....	280
Défauts du Module : 16#0800 - 16#08ff.....	283
Défauts du Module : 16#fd00 - 16#fdff .....	283
Défauts du Module : 16#fe00 - 16#feff.....	284
Défauts du Module : 16#ff00 - 16#ffff .....	287
Spécification de messages CIP .....	288
Spécification de messages PLC-3 .....	293
Spécification de messages PLC-5 .....	294
Spécification de messages PLC-2 .....	295

## Chapitre 5

### Instructions de comparaison

Instructions de comparaison.....	297
Comparaison (CMP) .....	298
Égal à (EQU).....	302
Supérieur à (GRT) .....	310
Supérieur ou égal à (GEQ) .....	319
Inférieur à (LES).....	327
Inférieur ou égal à (LEQ).....	336
Limite (LIM).....	345
Masque égal à (MEQ).....	354
Différent de (NEQ) .....	363
Opérateurs valides .....	371
Qu'est-ce qu'un remplissage avec des zéros ?.....	372

## Chapitre 6

### Instructions de calcul/Math

Instructions de calcul/math .....	373
Valeur absolue (ABS) .....	374
Addition (ADD) .....	380
Calcul (CPT) .....	387
Division (DIV) .....	391
Modulo (MOD).....	397
Multiplication (MUL) .....	404
Négation (NEG).....	411
Racine carrée (SQR/SQRT) .....	417
Soustraction (SUB).....	424
Fonctions FBD .....	430
Débordement de fonction .....	431

## Chapitre 7

### Instructions de mouvement/logique

Instructions de mouvement/logique .....	433
Répartition champ de bits (BTD) .....	435
Répartition champ de bits avec cible (BTDT) .....	438
ET au niveau du bit (AND) .....	443
Ou exclusif au niveau du bit (XOR) .....	447
Non au niveau du bit (NOT) .....	452
Ou au niveau du bit (OR) .....	456
ET booléen (BAND) .....	460
OU exclusif booléen (BXOR) .....	466
NON booléen (BNOT) .....	470
OU booléen (BOR) .....	474
Effacer (CLR) .....	479
Transfert avec masque (MVM) .....	481
Transfert masqué avec cible (MVMT) .....	485
Transfert (MOV) .....	490
Permutation d'octet (SWPB) .....	493

## Chapitre 8

### Instructions de diverse/tableau (fichier)

Tableau (Fichier)/Instructions diverses .....	499
Copie de fichier (COP), Copie synchrone de fichier (CPS) .....	500
Arithmétique et logique des fichiers (FAL) .....	509
Moyenne de fichier (AVE) .....	526
Remplissage de fichier (FLL) .....	530
Recherche et comparaison de fichier (FSC) .....	533
Tri sur fichier (SRT) .....	547
Écart standard de fichier (STD) .....	552
Taille en éléments (SIZE) .....	557
Mode Tous .....	562
Organigramme du mode Tous (FSC) .....	563
Mode Numérique .....	563
Organigramme du mode Numérique (FSC) .....	565
Mode Incrémentiel .....	566
Organigramme du mode Incrémentiel (FSC) .....	567
Étiquette de tableau .....	567
Écart standard .....	568

## Chapitre 9

### Instructions de décalage/tableau (fichier)

Instructions de décalage/tableau (fichier) .....	569
Décalage binaire à gauche (BSL) .....	570
Décalage binaire à droite (BSR) .....	574
Chargement de FIFO (FFL) .....	579
Déchargement de FIFO (FFU) .....	586

Chargement de LIFO (LFL).....	593
Déchargement de LIFO (LFU).....	600

## Chapitre 10

### Instructions de séquenceur

instructions de séquences.....	609
Entrée séquenceur (SQI) .....	610
Chargement de séquenceur (SQL) .....	614
Sortie séquenceur (SQO).....	618

## Chapitre 11

### Instructions de contrôle du programme

Instructions de contrôle du programme .....	624
Toujours faux (AFI) .....	626
Fin de transition (EOT) .....	627
Saut vers sous-programme externe (JXR).....	630
Saut vers une étiquette (JMP) et Etiquette (LBL).....	633
Saut vers sous-routine (JSR), Sous-routine (SBR), et Retour (RET) .....	636
Relais de contrôle maître (MCR).....	646
Organigramme MCR (faux) .....	649
Pas d'opération (NOP) .....	650
Pause SFC (SFP).....	651
Réinitialisation SFC (SFR) .....	654
Fin temporaire (TND).....	657
Déclenchement de tâche événementielle (EVENT) .....	659
Désactivation d'interruption par l'utilisateur (UID)/Activation d'interruption par l'utilisateur (UIE) .....	664
Instruction inconnue (UNK).....	667

## Chapitre 12

### Instructions de fin/rupture

Instructions de fin/rupture .....	669
Rupture (BRK) .....	669
Pour (FOR) .....	671
Saut vers sous-routine (JSR), Sous-routine (SBR), et Retour (RET) .....	675

## Chapitre 13

### Instructions spéciales

Instructions spéciales.....	685
Transition de données (DTR).....	686
Détection de diagnostic (DDT).....	689
Comparaison de fichier de bits (FBC) .....	697
Proportionnel, intégral et dérivé (PID) .....	705
Utilisation des instructions PID .....	712

	Anti-saturation d'intégrale et transfert sans à-coup de manuel à automatique (PID).....	716
	Redémarrage sans à-coup (PID) .....	717
	Boucles en cascade (PID).....	718
	Contrôle d'un rapport (PID).....	719
	Lissage des dérivées (PID).....	720
	Anticipation ou polarisation de la sortie (PID) .....	720
	Temporisation de l'instruction PID .....	721
	Réglage de la plage morte (PID) .....	725
	Utilisation de la limitation de sortie (PID).....	726
	 <b>Chapitre 14</b>	
<b>Instructions trigonométriques</b>	Instructions trigonométriques.....	728
	Cosinus d'arc (ACS, ACOS) .....	729
	Sinus d'arc (ASN, ASIN).....	733
	Tangente d'arc (ATN, ATAN).....	737
	Cosinus (COS) .....	741
	Sinus (SIN) .....	744
	Tangente (TAN) .....	748
	 <b>Chapitre 15</b>	
<b>Mathématiques avancées</b>	Instructions mathématiques avancées.....	753
	Logarithme décimal (LOG) .....	754
	Logarithme naturel (LN).....	758
	X à la puissance Y (XPY) .....	762
	 <b>Chapitre 16</b>	
<b>Instructions de conversion mathématique</b>	Instructions de conversion mathématique.....	767
	Conversion en BCD (TOD) .....	768
	Conversion en nombre entier (FRD).....	772
	Degrés (DEG) .....	775
	Radians (RAD) .....	779
	Troncation (TRN).....	783
	 <b>Chapitre 17</b>	
<b>Instructions pour port série ASCII</b>	Instructions pour port série ASCII .....	789
	Caractères ASCII dans la mémoire tampon (ACB) .....	791
	Effacement ASCII de la mémoire tampon (ACL) .....	795
	Lignes d'échange ASCII (AHL).....	798
	Lecture ASCII (ARD) .....	803
	Ligne de lecture ASCII (ARL) .....	807

Test ASCII pour ligne de mémoire tampon (ABL) .....	813
Écriture ASCII (AWT) .....	817
Ajout d'écriture ASCII (AWA) .....	823
Types de chaînes .....	828
Codes d'erreur ASCII.....	829

## Chapitre 18

### Instructions de chaîne ASCII

Instructions de chaîne ASCII.....	831
Rechercher une chaîne (FIND).....	832
Insertion d'une chaîne (INSERT).....	835
Chaîne médiane (MID) .....	838
Concaténation de chaîne (CONCAT) .....	842
Effacement de chaîne (DELETE).....	847

## Chapitre 19

### Instructions de conversion ASCII

Instructions de conversion ASCII .....	851
DINT vers chaîne (DTOS).....	852
Minuscule (LOWER) .....	855
REAL vers chaîne (RTOS).....	858
Chaîne vers DINT (STOD) .....	860
Chaîne vers REAL (STOR) .....	863
Haut de casse (UPPER) .....	867

## Chapitre 20

### Instructions de débogage

Instructions de débogage .....	871
Points d'interruption (BPT).....	872
Points de suivi (TPT) .....	876

## Chapitre 21

### Instructions Licence

Validation de licence (LV).....	881
---------------------------------	-----

## Chapitre 22

### Attributs courants des instructions générales

Attributs communs.....	885
Indicateurs d'état mathématique .....	885
Valeurs immédiates .....	888
Conversions de données .....	889
Types de données élémentaires.....	893
Types de données LINT .....	896
Valeurs de virgules flottantes .....	896
Indexer via des tableaux.....	899

Adressage de bits.....	900
------------------------	-----

## Chapitre 23

### Attributs du bloc fonctionnel

Sélectionner les éléments de bloc fonctionnel .....	901
Verrouillage des données .....	902
Ordre d'exécution .....	903
Réponses du bloc fonctionnel aux conditions de débordement .....	908
Les modes de temporisation.....	908
Contrôle Programme/Opérateur .....	912

## Chapitre 24

### Programmation de texte structuré

Syntaxe du texte structuré .....	917
Composants du texte structuré : Commentaires .....	919
Composants du texte structuré : Affectations.....	920
Spécifier une affectation non rémanente.....	921
Affecte un caractère ASCII à un membre de données chaîne .....	922
Composants du texte structuré : Expressions.....	923
Utiliser les opérateurs et les fonctions arithmétiques.....	924
Utiliser les opérateurs de bits .....	925
Utiliser les opérateurs logiques .....	926
Utiliser les opérateurs relationnels.....	927
Composants du texte structuré : Instructions.....	928
Composants du texte structuré : Constructions.....	930
Valeurs littérales des chaînes de caractères .....	930
Types de chaînes .....	932
CASE_OF .....	932
FOR_DO.....	935
IF_THEN.....	938
REPEAT_UNTIL.....	941
WHILE_DO .....	943
Attributs de texte structuré.....	946

### Indice

Ce manuel fournit des détails à un programmeur sur le jeux d'instructions général, le mouvement, le processus et le variateur pour un automate Logix.

Si vous concevez, définissez ou résolvez les problèmes des applications de la sécurité pour les automates GuardLogix, reportez-vous au [GuardLogix Safety Application Instruction Set Safety Reference Manual](#), publication [1756-RM095](#).

Ce manuel fait partie d'un ensemble de manuels complémentaires qui indiquent les procédures communes concernant la programmation et le fonctionnement des automates LOGIX 5000.

Pour obtenir la liste complète des manuels de procédures communes, reportez-vous au document [LOGIX 5000 Controllers Common Procedures Programming Manual](#), publication [1756-PM001](#).

Le terme automate LOGIX 5000 réfère à tout automate basé sur le système d'exploitation LOGIX 5000.

## Environnement studio 5000

Le Studio 5000 Automation Engineering & Design Environment® combine des éléments d'ingénierie et de conception dans un environnement commun. Le premier élément est l'application Studio 5000 Logix Designer®. L'application Logix Designer est un relookage du logiciel RSLogix 5000® et continuera d'être le produit de programmation des automates LOGIX 5000™ pour des solutions discrètes, de processus, de lots, de mouvement, de sécurité et les applications basées sur disque.



L'environnement Studio 5000® est le fondement pour l'avenir de Rockwell Automation® outils de conception d'ingénierie et capacités.

L'environnement Studio 5000 est l'emplacement qu'il faut pour que les ingénieurs concepteurs développent tous les éléments de leur système de commande.

## Ressources supplémentaires

Ces documents contiennent des informations supplémentaires concernant les produits Rockwell Automation.

Ressource	Description
<a href="#">Industrial Automation Wiring and Grounding Guidelines</a> , publication 1770-4.1	Fournit des recommandations générales pour installer un système industriel de Rockwell Automation.
Site Web pour les certifications de produits, disponible à l'adresse <a href="http://ab.rockwellautomation.com">http://ab.rockwellautomation.com</a>	Fournit les déclarations de conformité, les certificats et autres détails de certification.

Consultez ou téléchargez les publications à l'adresse <http://www.rockwellautomation.com/literature> . Pour commander des copies papier de documentation technique, contactez votre distributeur local Rockwell Automation ou votre représentant commercial.

## Mentions légales

### Mention de copyright

Copyright © 2018 Rockwell Automation Technologies, Inc. Tous droits réservés. Imprimé aux États-Unis.

Ce document et les produits Rockwell Software qui l'accompagnent sont protégés par copyright de Rockwell Automation Technologies, Inc. Toute reproduction et/ou distribution sans autorisation écrite préalable de Rockwell Automation Technologies, Inc. est strictement interdite. Reportez-vous au contrat de licence pour plus d'informations.

### Accord de licence utilisateur (EULA)

Vous pouvez afficher le Contrat de licence utilisateur final (« EULA ») Rockwell Automation en ouvrant le fichier License.rtf situé dans le dossier d'installation de votre produit sur votre disque dur.

### Licences open source

Le logiciel inclus dans ce produit contient un logiciel protégé par copyright soumis à une ou plusieurs licences open source. Des copies de ces licences sont fournies avec le logiciel. Le code source correspondant pour les progiciels open source inclus dans ce produit est disponible sur leurs sites Web respectifs.

Vous pouvez également obtenir le code source correspondant complet en prenant contact avec Rockwell Automation au moyen du formulaire de contact disponible sur le site Web de Rockwell Automation :

<http://www.rockwellautomation.com/global/about-us/contact/contact.page>

Veillez indiquer « Open Source » dans le texte de la requête.



Une liste complète de tous les logiciels open source utilisés dans ce produit et de leurs licences correspondantes se trouve [dans le dossier OPENSOURCE](#) inclus dans les notes de version. L'emplacement d'installation de ces licences est C:\Program Files (x86)\Common Files\Rockwell\Help\*<Produit>*\ReleaseNotes\OPENSOURCE\index.htm.

### Marques

Allen-Bradley, ControlBus, ControlFLASH, Compact GuardLogix, Compact I/O, ControlLogix, CompactLogix, DCM, DH+, Data Highway Plus, DriveLogix, DPI, DriveTools, Explorer, FactoryTalk, FactoryTalk Administration Console, FactoryTalk Alarms and Events, FactoryTalk Batch, FactoryTalk Directory, FactoryTalk Security, FactoryTalk Services Platform, FactoryTalk View, FactoryTalk View SE, FLEX Ex, FlexLogix, FLEX I/O, Guard I/O, High Performance Drive, Integrated Architecture, Kinetix, Logix5000, LOGIX 5000, Logix5550, MicroLogix, DeviceNet, EtherNet/IP, PLC-2, PLC-3, PLC-5, PanelBuilder, PowerFlex, PhaseManager, POINT I/O, PowerFlex, Rockwell Automation, RSBizWare, Rockwell Software, RSEmulate, Historian, RSFieldbus, RSLinx, RSLogix, RSNetWorx for DeviceNet, RSNetWorx for EtherNet/IP, RSMACC, RSView, RSView32, Rockwell Software Studio 5000 Automation Engineering & Design Environment, Studio 5000 View Designer, SCANport, SLC, SoftLogix, SMC Flex, Studio 5000, Ultra 100, Ultra 200, VersaView, WINtelligent, XM, SequenceManager sont des marques de Rockwell Automation, Inc.

Tout logo, logiciel ou matériel Rockwell Automation non mentionné ici est également une marque, déposée ou pas, de Rockwell Automation, Inc.

### Autres marques commerciales

CmFAS Assistant, CmDongle, CmStick, CodeMeter, CodeMeter Control Center et WIBU sont des marques commerciales de WIBU-SYSTEMS AG aux Etats-Unis et/ou dans d'autres pays.

Toutes les autres marques sont la propriété de leurs détenteurs respectifs et sont reconnues ici comme telles.

### Garantie

Ce produit est garanti conformément à la licence de produit. Les performances du produit peuvent être affectées par la configuration du système, par l'application en cours d'exécution, par le contrôle de l'opérateur, par la maintenance et par d'autres facteurs connexes. Rockwell Automation n'est pas responsable de ces facteurs intermédiaires. Les instructions de ce document ne comprennent pas tous les détails ou variations de l'équipement, de la procédure ou du processus décrit et ne fournissent pas de conseils pour répondre à chaque urgence possible pendant

l'installation, le fonctionnement ou la maintenance. L'implémentation de ce produit peut varier en fonction des utilisateurs.

Ce document est à jour au moment de la sortie du produit ; toutefois, le logiciel qui l'accompagne peut avoir été modifié depuis sa sortie. Rockwell Automation, Inc. se réserve le droit de modifier les informations contenues dans ce document ou le logiciel à tout moment sans notification préalable. Il est de votre responsabilité d'obtenir les informations les plus récentes disponibles de la part de Rockwell lors de l'installation ou de l'utilisation de ce produit.

### **Respect de l'environnement**

Rockwell Automation conserve les informations actuelles sur le produit relatives à l'environnement sur son site Web :

<http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page>

### **Contacter Rockwell**

Téléphone du service client — 1.440.646.3434

Aide en ligne — <http://www.rockwellautomation.com/support/>

# Instructions d'alarme

## Instructions d'alarme

Utilisez les instructions d'alarme pour surveiller et contrôler les conditions d'alarme.

Les instructions basées sur Logix intègrent les alarmes entre les applications RSVIEW® SE et les automates LOGIX 5000™.

### Instructions disponibles

#### Diagramme à relais

<a href="#">ALMD</a>	<a href="#">ALMA</a>	<a href="#">ASO</a>
----------------------	----------------------	---------------------

#### Bloc fonctionnel

<a href="#">ALMD</a>	<a href="#">ALMA</a>
----------------------	----------------------

#### Texte structuré

<a href="#">ALMD</a>	<a href="#">ALMA</a>	<a href="#">ASO</a>
----------------------	----------------------	---------------------

Si :	Utilisez l'élément suivant :
Fourniture d'une alarme pour toute valeur booléenne discrète pour une logique à relais, un bloc fonctionnel ou du texte structuré,	Instruction Alarme numérique (ALMD).
Fourniture d'une alarme de niveau et de fréquence de changement pour tout signal analogique pour une logique à relais, un bloc fonctionnel, un diagramme et du texte structuré,	Instruction Alarme analogique (ALMA).
Génération d'une opération spécifiée pour toutes les conditions d'alarme de l'ensemble d'alarmes spécifié,	Instruction Opération d'ensemble d'alarmes (ASO).

**Voir aussi**

[Instructions de diverse/tableau \(fichier\)](#) sur la [page 499](#)

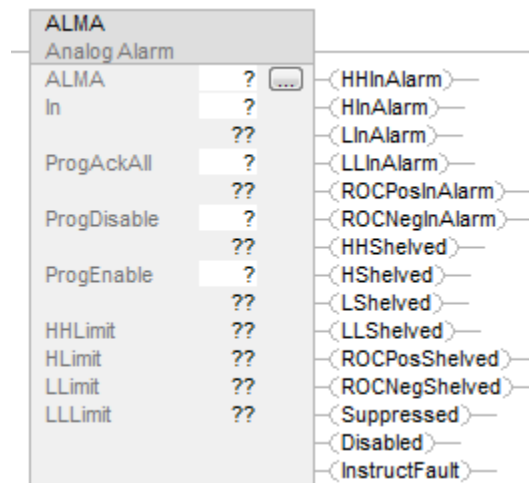
[Instructions de conversion ASCII](#) sur la [page 851](#)

**Alarme analogique (ALMA)**

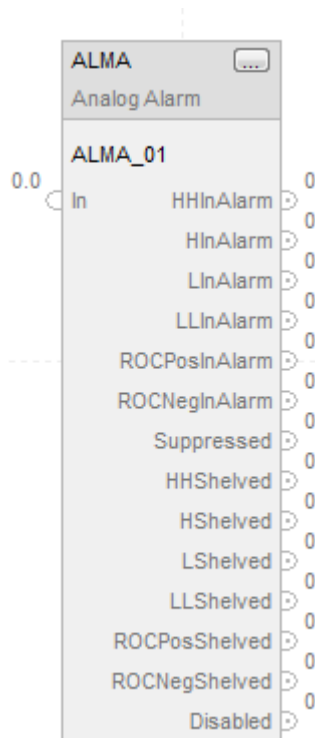
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction ALMA fournit le niveau et la fréquence de changement des alarmes pour un signal analogique.

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

ALMA (ALMA,In,ProgAckAll,ProgDisable,ProgEnable)

**Opérandes**

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
ALMA	ALARM_ANALOG	Structure	Structure ALMA
In	REAL DINT INT SINT	Étiquette immédiate	Valeur d'entrée de l'alarme qui est comparée aux limites de l'alarme afin de détecter sa condition.
ProgAckAll	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, confirme toutes les conditions d'alarme qui nécessitent une confirmation.
ProgDisable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, désactive l'alarme (n'outrepasse pas les commandes d'activation).
ProgEnable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, active l'alarme (est prioritaire sur les commandes d'activation).

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
ALMA tag	ALARM_ANALOG	structure	Structure ALMA

**Texte structuré**

Opérande	Type (Type)	Format	Description (Description)
ALMA	ALARM_ANALOG	Structure	Structure ALMA
In	REAL DINT INT SINT	Étiquette immédiate	Valeur d'entrée de l'alarme qui est comparée aux limites de l'alarme afin de détecter sa condition.
ProgAckAll	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, confirme toutes les conditions d'alarme qui nécessitent une confirmation.
ProgDisable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, désactive l'alarme (n'outrepasse pas les commandes d'activation).
ProgEnable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, active l'alarme (est prioritaire sur les commandes d'activation).

Reportez-vous à Syntaxe du texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Structure ALMA**

**Paramètres d'entrée**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Diagramme à relais : Correspond à l'état d'échelon. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Texte structuré : Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie. Bloc fonctionnel : Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.

Paramètre d'entrée	Type de donnée	Description (Description)
In	REAL	Valeur d'entrée de l'alarme qui est comparée aux limites de l'alarme afin de détecter sa condition. Valeur par défaut = 0,0. Diagramme à relais : copié à partir de l'opérande d'instruction. Texte structuré : copié à partir de l'opérande d'instruction.
InFault	BOOL	Indicateur de mauvaise santé pour l'entrée. L'application utilisateur peut définir InFault pour indiquer que le signal d'entrée comporte une erreur. Si défini, l'instruction définit InFaulted (Status.1). Si défini sur faux, l'instruction définit sur faux InFaulted (Status.1). Dans les deux cas, l'instruction continue d'évaluer In pour des conditions d'alarme. La valeur par défaut est fausse (bonne santé).
HHEnabled	BOOL	Détection de condition d'alarme haute haute. Définir sur vrai pour activer la détection de la condition d'alarme haute haute. Définir sur faux si vous souhaitez que la détection ne soit pas disponible pour la condition d'alarme haute haute. La valeur par défaut est définie.
HEnabled	BOOL	Détection de condition d'alarme haute. Définir sur vrai pour activer la détection de la condition d'alarme haute. Définir sur faux pour que la détection ne soit pas disponible pour la condition d'alarme haute. La valeur par défaut est définie.
LEnabled	BOOL	Détection de condition d'alarme basse. Définir sur vrai pour activer la détection de la condition d'alarme basse. Définir sur faux pour que la détection ne soit pas disponible pour la condition d'alarme basse. La valeur par défaut est définie.
LLEnabled	BOOL	Détection de condition d'alarme basse basse. Définir sur vrai pour activer la détection de la condition d'alarme basse basse. Définir sur faux pour que la détection ne soit pas disponible pour la condition d'alarme basse basse. La valeur par défaut est définie.
AckRequired	BOOL	Indique si une confirmation d'alarme est requise. Si définir sur vrai, une confirmation est requise. Si définir sur faux, aucune confirmation n'est requise et HHAcked, HAcked, LAcked, LLAcked, ROCPosAcked et ROCNegAcked sont toujours définis sur vrai. La valeur par défaut est vraie.
ProgAckAll	BOOL	Définir sur vrai par le programme utilisateur pour confirmer toutes les conditions d'alarme. Prend effet uniquement si une condition d'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse. Diagramme à relais : Copié à partir de l'opérande d'instruction. Texte structuré : Copié à partir de l'opérande d'instruction.

Paramètre d'entrée	Type de donnée	Description (Description)
OperAckAll	BOOL	Définir sur vrai par l'interface opérateur pour confirmer toutes les conditions d'alarme. Prend effet uniquement si une condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
HHProgAck	BOOL	Confirmation d'alarme haute haute par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition d'alarme haute haute. Prend effet uniquement si la condition d'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse.
HHOperAck	BOOL	Confirmation d'alarme haute haute par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition d'alarme haute haute. Prend effet uniquement si la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
HProgAck	BOOL	Confirmation d'alarme haute par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition d'alarme haute. Prend effet uniquement si la condition d'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse.
HOperAck	BOOL	Confirmation d'alarme haute par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition d'alarme haute. Prend effet uniquement si la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
LProgAck	BOOL	Confirmation d'alarme basse par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition d'alarme basse. Prend effet uniquement si la condition d'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse.
LOperAck	BOOL	Confirmation d'alarme basse par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition d'alarme basse. Prend effet uniquement si la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
LLProgAck	BOOL	Confirmation de l'alarme basse basse par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition d'alarme basse basse. Prend effet uniquement si la condition d'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse.



Paramètre d'entrée	Type de donnée	Description (Description)
LLOperAck	BOOL	Confirmation d'alarme basse basse par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition d'alarme basse basse. Prend effet uniquement si la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ROCPoSProgAck	BOOL	Confirmation de fréquence de changement positive par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition de fréquence de changement positive. Requiert une transition de faux à vrai lorsque la condition d'alarme n'est pas confirmée. La valeur par défaut est fausse.
ROCPoSOperAck	BOOL	Confirmation de la fréquence de changement positive par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition de fréquence de changement positive. Requiert une transition de faux à vrai lorsque la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ROCNegProgAck	BOOL	Confirmation de fréquence de changement négative par le programme. Définir sur vrai par le programme utilisateur pour confirmer une condition de fréquence de changement négative. Requiert une transition de faux à vrai lorsque la condition d'alarme n'est pas confirmée. La valeur par défaut est fausse.
ROCNegOperAck	BOOL	Confirmation de fréquence de changement négative par l'opérateur. Définir sur vrai par l'interface opérateur pour confirmer une condition de fréquence de changement négative. Requiert une transition de faux à vrai lorsque la condition d'alarme n'est pas confirmée. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ProgSuppress	BOOL	Définir sur vrai par le programme utilisateur pour supprimer l'alarme. La valeur par défaut est mise à zéro.
OperSuppress	BOOL	Définir sur vrai par l'interface opérateur pour supprimer l'alarme. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ProgUnsuppress	BOOL	Définir sur vrai par le programme utilisateur pour annuler la suppression de l'alarme. Est prioritaire sur les commandes de suppression. La valeur par défaut est fausse.
OperUnsuppress	BOOL	Définir sur vrai par l'interface opérateur pour annuler la suppression de l'alarme. Est prioritaire sur les commandes de suppression. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.

Paramètre d'entrée	Type de donnée	Description (Description)
HHOperShelve	BOOL	Réservation haute-haute par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition haute-haute. Requiert une transition de faux à vrai. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La réservation d'une alarme reporte son traitement. Cette opération est semblable à la suppression d'une alarme, si ce n'est que la réservation est limitée dans le temps. Si une alarme est confirmée pendant qu'elle est réservée, elle conserve cet état même si elle redevient active. Sa confirmation est annulée lorsque la réservation arrive à son terme.
HOperShelve	BOOL	Réservation haute par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition haute. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation.
LOperShelve	BOOL	Réservation basse par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition basse. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation.
LLOperShelve	BOOL	Réservation basse-basse par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition basse-basse. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation.
ROCPoSOperShelve	BOOL	Réservation de fréquence de changement positive par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition de fréquence de changement positive. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation.

Paramètre d'entrée	Type de donnée	Description (Description)
ROCNegOperShelve	BOOL	Réservation de fréquence de changement négative par l'opérateur. Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau une condition de fréquence de changement négative. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation.
ProgUnshelveAll	BOOL	Définir sur vrai par le programme utilisateur pour annuler la réservation de toutes les conditions de cette alarme. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.
HHOperUnshelve	BOOL	Annulation de réservation haute-haute par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition haute-haute. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.
HOperUnshelve	BOOL	Annulation de réservation haute par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition haute. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.
LOperUnshelve	BOOL	Annulation de réservation basse par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition basse. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.
LLOperUnshelve	BOOL	Annulation de réservation basse-basse par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition basse-basse. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.

Paramètre d'entrée	Type de donnée	Description (Description)
ROCPosOperUnshelve	BOOL	Annulation de réservation de fréquence de changement positive par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition de fréquence de changement positive. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies, les commandes de réservation sont prioritaires sur les commandes d'annulation de réservation. La valeur par défaut est fausse.
ROCNegOperUnshelve	BOOL	Annulation de réservation de fréquence de changement négative par l'opérateur. Définir sur vrai par l'interface opérateur pour annuler la réservation d'une condition de fréquence de changement négative. L'instruction d'alarme définit ce paramètre sur faux. Si une réservation et une annulation de réservation sont définies sur vrai, les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La valeur par défaut est fausse.
ProgDisable	BOOL	Copié à partir de l'opérande d'instruction.
OperDisable	BOOL	Définir sur vrai par l'interface opérateur pour désactiver l'alarme. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ProgEnable	BOOL	Copié à partir de l'opérande d'instruction.
OperEnable	BOOL	Définir sur vrai par l'interface opérateur pour activer l'alarme. Est prioritaire sur la commande de désactivation. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
AlarmCountReset	BOOL	Définir sur vrai par l'interface opérateur pour réinitialiser le comptage d'alarme pour toutes les conditions. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
HHMinDurationEnable	BOOL	Activation de la durée minimale haute-haute. Définir sur vrai pour activer le temporisateur de durée minimale lors de la détection de la condition haute-haute. La valeur par défaut est vraie.
HMinDurationEnable	BOOL	Activation de la durée minimale haute. Définir sur vrai pour activer le temporisateur de durée minimale lors de la détection de la condition haute. La valeur par défaut est vraie.
LMinDurationEnable	BOOL	Activation de la durée minimale basse. Définir sur vrai pour activer le temporisateur de durée minimale lors de la détection de la condition basse. La valeur par défaut est vraie.

Paramètre d'entrée	Type de donnée	Description (Description)
LLMinDurationEnable	BOOL	Activation de la durée minimale basse-basse. Définir sur vrai pour activer le temporisateur de durée minimale lors de la détection de la condition basse-basse. La valeur par défaut est vraie.
HHLimit	REAL	Limite d'alarme haute haute. Valide = HLimit < HHLimit < valeur positive à virgule flottante maximale. Valeur par défaut = 0,0.
HHSeverity	DINT	Sévérité de la condition d'alarme haute haute. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
HLimit	REAL	Limite d'alarme haute. Valide = LLimit < HLimit < HHLimit. Valeur par défaut = 0,0.
HSeverity	DINT	Sévérité de la condition d'alarme haute. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
LLimit	REAL	Limite d'alarme basse. Valide = LLLimit < LLimit < HLimit. Valeur par défaut = 0,0.
LSeverity	DINT	Sévérité de la condition d'alarme basse. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
LLLimit	REAL	Limite d'alarme basse basse. Valide = valeur négative à virgule flottante maximale < LLLimit < LLimit. Valeur par défaut = 0,0.
LLSeverity	DINT	Sévérité de la condition d'alarme basse basse. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.

Paramètre d'entrée	Type de donnée	Description (Description)
MinDurationPRE	DINT	Durée prédéfinie (en millisecondes) pendant laquelle une condition de niveau d'alarme doit rester définie sur vrai avant que la condition ne soit marquée comme InAlarm et qu'une notification d'alarme ne soit envoyée aux clients. L'automate collecte les données d'alarme dès que la condition d'alarme est détectée, ce qui évite la perte de données jusqu'à ce que la condition de durée minimale soit remplie. Ne s'applique pas aux conditions de fréquence de changement ou aux conditions pour lesquelles la détection de la durée minimale est désactivée. MinDurationPRE s'applique uniquement à la première course de niveau normal dans les deux directions. Par exemple, une fois la condition haute expirée, la condition haute haute devient immédiatement active, tandis que la condition basse attend le période délai d'expiration . Valide = de 0 à 2 147 483 647. Valeur par défaut = 0.
ShelveDuration	DINT	Durée (en minutes) de réservation d'une alarme. La durée minimale est d'une minute. La durée maximale est définie par MaxShelveDuration.
MaxShelveDuration	DINT	Durée maximale (en minutes) de réservation d'une alarme.
Deadband	REAL	Plage morte relative à la détection du retour à la normale des niveaux d'alarme haute haute, haute, basse et basse basse. Une Deadband d'une valeur différente de zéro peut réduire les vibrations de condition d'alarme si la valeur In varie sans s'éloigner du seuil de condition d'alarme. La valeur Deadband n'affecte pas la transition vers l'état InAlarm (actif). Avant qu'une condition de niveau active revienne à un état inactif (normal), la valeur In doit : diminuer pour atteindre une valeur inférieure au seuil moins la plage morte (pour les conditions haute et haute haute), OU augmenter pour atteindre une valeur supérieure au seuil plus la plage morte (pour les conditions basse et basse basse). La plage morte ne sert pas à régler la mesure de durée minimale. Valide = 0 = Deadband < la plage de la première alarme basse activée à la première alarme haute activée. Valeur par défaut = 0,0.
ROCPoSLimit	REAL	Limite d'une fréquence de changement croissante, en unités par seconde. La détection est activée pour toutes les valeurs > 0,0 si ROCPeriod est également > 0,0. Valide = de 0,0 à la valeur à virgule flottante maximale. Valeur par défaut = 0,0.

Paramètre d'entrée	Type de donnée	Description (Description)
ROCPoSSeverity	DINT	Sévérité de la condition de fréquence de changement croissante. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
ROCNegLimit	REAL	Limite d'une fréquence de changement décroissante, en unités par seconde. La détection est activée pour toutes les valeurs > 0,0 si ROCPeriod est également > 0,0. Valide = de 0,0 à la valeur à virgule flottante maximale. Valeur par défaut = 0,0.
ROCNegSeverity	DINT	Sévérité de la condition de fréquence de changement décroissante. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
ROCPeriod	REAL	Durée en secondes pour le calcul (intervalle d'échantillonnage) de la valeur de fréquence de changement. Chaque fois que l'intervalle d'échantillonnage expire, un nouvel échantillon d'In est stocké et ROC est recalculée. Plutôt qu'un bit d'activation comme les autres conditions d'alarme analogique, la détection de fréquence de changement est activée en définissant une valeur non égale à zéro dans la ROCPeriod. Valide = de 0,0 à 32 767,0 Valeur par défaut = 0,0.

### Paramètres de sortie

Ces paramètres de sortie sont communs à la logique à relais.

Paramètre de sortie	Type de donnée	Description (Description)
AnyInAlarmUnack	BOOL	État confirmé et actif d'une alarme. Définir sur vrai si une condition d'alarme est détectée et non confirmée. Définir sur faux si toutes les conditions d'alarme sont inactives, confirmées ou les deux.
HHInAlarm	BOOL	État de condition d'alarme haute haute. Définir sur vrai si une condition haute haute est active. Définir sur faux s'il n'existe aucune condition haute haute.
HInAlarm	BOOL	État de condition d'alarme haute. Définir sur vrai si une condition haute est active. Définir sur faux s'il n'existe aucune condition haute.
LInAlarm	BOOL	État de condition d'alarme basse. Définir sur vrai si une condition basse est active. Définir sur faux s'il n'existe aucune condition basse.

Paramètre de sortie	Type de donnée	Description (Description)
LLInAlarm	BOOL	État de condition d'alarme basse basse. Définir sur vrai si une condition basse basse est active. Définir sur faux s'il n'existe aucune condition basse basse.
ROCPosInAlarm	BOOL	État de condition d'alarme de fréquence de changement positive. Définir sur vrai s'il existe une condition de fréquence de changement positive. Définir sur faux s'il n'existe aucune condition de fréquence de changement positive.
ROCNegInAlarm	BOOL	État de condition d'alarme de fréquence de changement négative. Définir sur vrai s'il existe une condition de fréquence de changement négative. Définir sur faux s'il n'existe aucune condition de fréquence de changement négative.
ROC	REAL	Fréquence de changement calculée de la valeur In. Cette valeur est mise à jour lorsque l'instruction est scrutée à la suite de chaque valeur ROCPeriod écoulée. La valeur de la ROC est utilisée pour évaluer les conditions ROCPosInAlarm et ROCNegInAlarm. $ROC = (\text{échantillon en cours de In} - \text{échantillon précédent de In}) / ROCPeriod$
HHAcked	BOOL	État confirmé de condition haute haute. Définir sur vrai si une condition haute haute est confirmée. Toujours définir sur vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition haute haute n'est pas confirmée.
HAcked	BOOL	État confirmé de condition haute. Définir sur vrai si une condition haute est confirmée. Toujours définir sur vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition haute n'est pas confirmée.
LAcked	BOOL	État confirmé de condition basse. Définir sur vrai si une condition basse est confirmée. Toujours définir sur vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition basse n'est pas confirmée.
LLAcked	BOOL	État confirmé de condition basse basse. Définir sur vrai si une condition basse basse est confirmée. Il est toujours vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition basse basse n'est pas confirmée.
ROCPosAcked	BOOL	État confirmé de la condition de fréquence de changement positive. Définir sur vrai si une condition de fréquence de changement positive est confirmée. Il est toujours vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition de fréquence de changement positive n'est pas confirmée.
ROCNegAcked	BOOL	État confirmé de condition de fréquence de changement négative. Définir sur vrai si une condition de fréquence de changement négative est confirmée. Toujours définir sur vrai lorsque AckRequired est défini sur faux. Définir sur faux si une condition de fréquence de changement négative n'est pas confirmée.



Paramètre de sortie	Type de donnée	Description (Description)
HHInAlarmUnack	BOOL	État actif et non confirmé de condition haute haute. Définir sur vrai si la condition haute haute est active (HHInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition haute haute est inactive, confirmée ou les deux.
HInAlarmUnack	BOOL	État actif et non confirmé de condition haute. Définir sur vrai si la condition haute est active (HInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition haute est inactive, confirmée ou les deux.
LInAlarmUnack	BOOL	État actif et non confirmé de condition basse. Définir sur vrai si la condition basse est active (LInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition basse est inactive, confirmée ou les deux.
LLInAlarmUnack	BOOL	État actif et non confirmé de condition basse basse. Définir sur vrai si la condition basse basse est active (LLInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition basse basse est inactive, confirmée ou les deux.
ROCPosInAlarmUnack	BOOL	État actif et non confirmé de condition de fréquence de changement positive. Définir sur vrai si la condition de fréquence de changement positive est active (ROCPosInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition de fréquence de changement positive est inactive, confirmée ou les deux.
ROCNegInAlarmUnack	BOOL	État actif et non confirmé de condition de fréquence de changement négative. Définir sur vrai si la condition de fréquence de changement négative est active (ROCNegInAlarm est défini sur vrai) et non confirmée. Définir sur faux si la condition de fréquence de changement négative est inactive, confirmée ou les deux.
Suppressed	BOOL	État supprimé de l'alarme. Définir sur vrai si l'alarme est supprimée. Définir sur faux si l'alarme n'est pas supprimée.
HHShelved	BOOL	État de réservation de condition haute-haute. Définir sur vrai si une condition haute-haute est réservée. Définir sur faux si la réservation de la condition haute-haute est annulée.
HShelved	BOOL	État de réservation de condition haute. Définir sur vrai si une condition haute est réservée. Définir sur faux si la réservation de la condition haute est annulée.
LShelved	BOOL	État de réservation de condition basse. Définir sur vrai si une condition basse est réservée. Définir sur faux si la réservation de la condition basse est annulée.
LLShelved	BOOL	État de réservation de condition basse-basse. Définir sur vrai si une condition basse-basse est réservée. Définir sur faux si la réservation de la condition basse-basse est annulée.
ROCPosShelved	BOOL	État de réservation de condition de fréquence de changement positive. Définir sur vrai si une condition de fréquence de changement positive est réservée. Définir sur faux si la réservation de la condition de fréquence de changement positive est annulée.

Paramètre de sortie	Type de donnée	Description (Description)
ROCNegShelved	BOOL	État de réservation de condition de fréquence de changement négative. Définir sur vrai si une condition de fréquence de changement négative est réservée. Définir sur faux si la réservation de la condition de fréquence de changement négative est annulée.
Désactivé	BOOL	État désactivé de l'alarme. Définir sur vrai lorsque l'alarme n'est pas disponible (désactivée). Définir sur faux si l'alarme est activée.
Demandée	BOOL	Le bit commissionné n'est pas utilisé.
MinDurationACC	DINT	Non utilisé. La valeur est toujours 0.
HHInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition haute haute pour la transition la plus récente vers l'état actif.
HHAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme haute haute a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
HInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition haute pour la transition la plus récente vers l'état actif.
HAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme haute a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
LInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition basse pour la transition la plus récente vers l'état actif.
LAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme basse a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
LLInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition basse basse pour la transition la plus récente vers l'état actif.
LLAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme basse basse a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
ROCPoSInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition de fréquence de changement positive pour la transition la plus récente vers l'état actif.
ROCPoSInAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme de fréquence de changement positive a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
ROCNegInAlarmTime	LINT	Horodatage si l'instruction ALMA détecte que la valeur In dépasse la limite de condition de fréquence de changement négative pour la transition la plus récente vers l'état actif.
ROCNegAlarmCount	DINT	Indique le nombre de fois où la condition d'alarme de fréquence de changement négative a été activée. Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.

Paramètre de sortie	Type de donnée	Description (Description)																														
AckTime	LINT	Horodatage de la confirmation de condition la plus récente. Si l'alarme ne nécessite pas de confirmation, cet horodatage correspond à l'heure de la condition d'alarme la plus récente.																														
RetToNormalTime	LINT	Horodatage de l'alarme revenant à un état normal.																														
AlarmCountResetTime	LINT	Horodatage indiquant lorsque le comptage d'alarme a été réinitialisé.																														
ShelveTime	LINT	L'horodatage indique quand a eu lieu la dernière réservation d'une condition d'alarme. Définir par l'automate lorsqu'une condition d'alarme est réservée. Les conditions d'alarme peuvent être réservées et leur réservation être annulée à de nombreuses reprises. À chaque réservation d'une condition d'alarme, l'horodatage est défini sur l'heure actuelle.																														
UnshelveTime	LINT	Horodatage indiquant quand toutes les conditions d'alarme ne seront plus réservées. Cette valeur est définie uniquement si aucune condition d'alarme n'est encore réservée. L'horodatage correspond à la somme de la période ShelveDuration et de l'heure actuelle. Si la réservation d'une condition d'alarme est annulée par programmation ou par un opérateur et si aucune autre condition d'alarme n'est réservée, l'horodatage est défini sur l'heure actuelle.																														
État (Status)	DINT	<p>Indicateurs d'état associés :</p> <table border="0"> <thead> <tr> <th>Indicateur d'état</th> <th>Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570</th> <th>Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580</th> </tr> </thead> <tbody> <tr> <td>Status.0 = InstructFault</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.1 = InFaulted</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.2 = SeverityInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.3 = AlarmLimitsInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.4 = DeadbandInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.5 = ROCPosLimitInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.6 = ROCNegLimitInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.7 = ROCPeriodInv</td> <td>X</td> <td>X</td> </tr> <tr> <td>Status.8 = Débordement</td> <td>-</td> <td>X</td> </tr> </tbody> </table>	Indicateur d'état	Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Status.0 = InstructFault	X	X	Status.1 = InFaulted	X	X	Status.2 = SeverityInv	X	X	Status.3 = AlarmLimitsInv	X	X	Status.4 = DeadbandInv	X	X	Status.5 = ROCPosLimitInv	X	X	Status.6 = ROCNegLimitInv	X	X	Status.7 = ROCPeriodInv	X	X	Status.8 = Débordement	-	X
Indicateur d'état	Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580																														
Status.0 = InstructFault	X	X																														
Status.1 = InFaulted	X	X																														
Status.2 = SeverityInv	X	X																														
Status.3 = AlarmLimitsInv	X	X																														
Status.4 = DeadbandInv	X	X																														
Status.5 = ROCPosLimitInv	X	X																														
Status.6 = ROCNegLimitInv	X	X																														
Status.7 = ROCPeriodInv	X	X																														
Status.8 = Débordement	-	X																														

Paramètre de sortie	Type de donnée	Description (Description)
InstructFault (Status.0)	BOOL	Des conditions d'erreur d'instruction existent. Il ne s'agit pas d'une erreur d'automate mineure ou majeure. Contrôlez les bits d'état restants pour déterminer la cause du problème.
InFaulted (Status.1)	BOOL	Le programme utilisateur a défini InFault pour indiquer la mauvaise qualité des données d'entrée. L'alarme continue d'évaluer In pour les conditions d'alarme.
SeverityInv (Status.2)	BOOL	La configuration de sévérité de l'alarme est invalide. Si la sévérité est <1, l'instruction utilise Sévérité = 1. Si la sévérité est >1 000, l'instruction utilise Sévérité = 1 000.
AlarmLimitsInv (Status.3)	BOOL	La configuration de limite d'alarme est invalide (LLimit < LLLimit, par exemple). Si ce paramètre n'est pas valide, l'instruction efface tous les bits actifs des conditions de niveau. Aucune nouvelle condition de niveau ne sera détectée tant que le défaut n'aura pas été corrigé.
DeadbandInv (Status.4)	BOOL	La configuration de plage morte est invalide. Si la valeur est invalide, l'instruction utilise Deadband = 0,0. Valide = 0 = Deadband < la plage de la première alarme basse activée à la première alarme haute activée.
ROCPosLimitInv (Status.5)	BOOL	Limite de fréquence de changement positive invalide. Si la valeur est invalide, l'instruction utilise ROCPosLimit = 0,0 et la détection de fréquence de changement positive n'est pas disponible.
ROCNegLimitInv (Status.6)	BOOL	Limite de fréquence de changement négative invalide. Si la valeur est invalide, l'instruction utilise ROCNegLimit = 0,0 et la détection de fréquence de changement négative n'est pas disponible.
ROCPeriodInv (Status.7)	BOOL	Période de fréquence de changement invalide. Si la valeur est invalide, l'instruction utilise ROCPeriod = 0,0 et la détection de fréquence de changement n'est pas disponible.
Overflow (Status.8)	BOOL	Le bit de Overflow est défini sur vrai lorsqu'une condition de débordement est détectée. Le bit de Overflow est défini sur faux lorsque la condition de débordement a été corrigée. S'applique à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement.

### Connecter un bouton à l'étiquette OperShelve

L'instruction d'alarme ne traite que l'étiquette OperShelve lorsqu'elle transforme de zéro à définie pour empêcher une réservation non désirée de l'alarme. Par exemple, si un opérateur appuie sur un bouton-poussoir pour réserver l'alarme alors que l'étiquette ProgUnshelve est définie, l'alarme n'est pas réservée, car ProgUnshelve est prioritaire. Pour réserver l'alarme, l'opérateur peut relâcher le bouton-poussoir, puis appuyer de nouveau dessus. Une fois que ProgUnshelve est mis à zéro.

### Affecter les indicateurs d'état mathématique

Automates	Indicateurs d'état mathématique affectés
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La valeur d'entrée est INF ou NAN pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570 uniquement.	4	4

Reportez-vous à la section Indicateurs d'état mathématique.

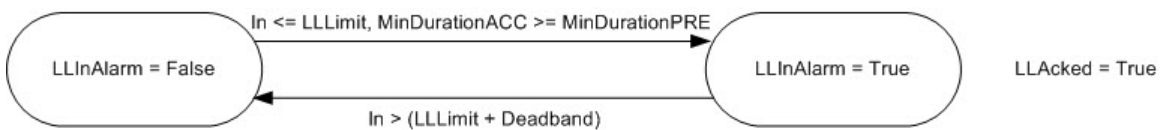
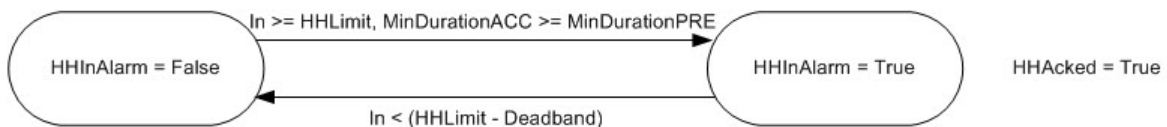
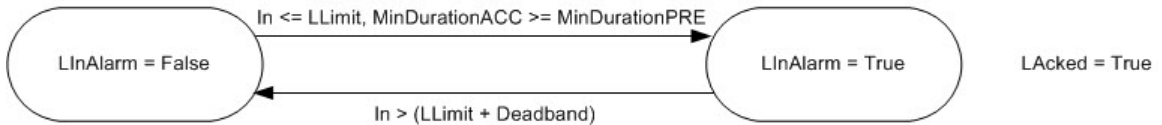
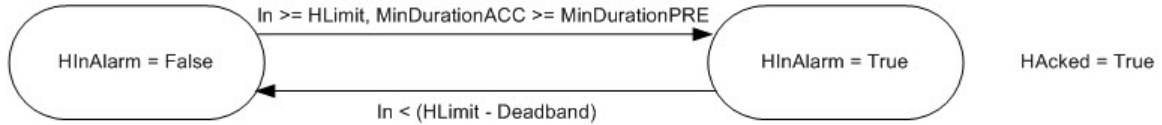
### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

### Diagrammes d'état d'alarme analogique

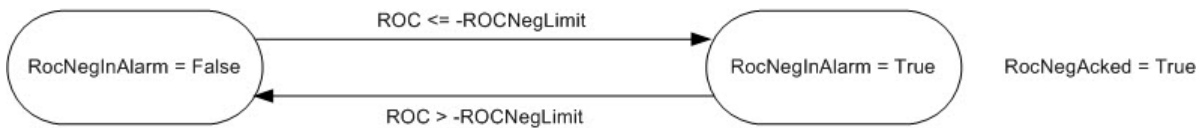
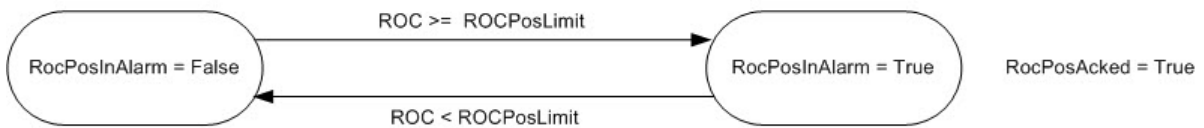
Les illustrations suivantes montrent de quelle manière une alarme analogique répond à la modification des conditions d'alarme et aux commandes de l'opérateur.

**AckRequired = False**

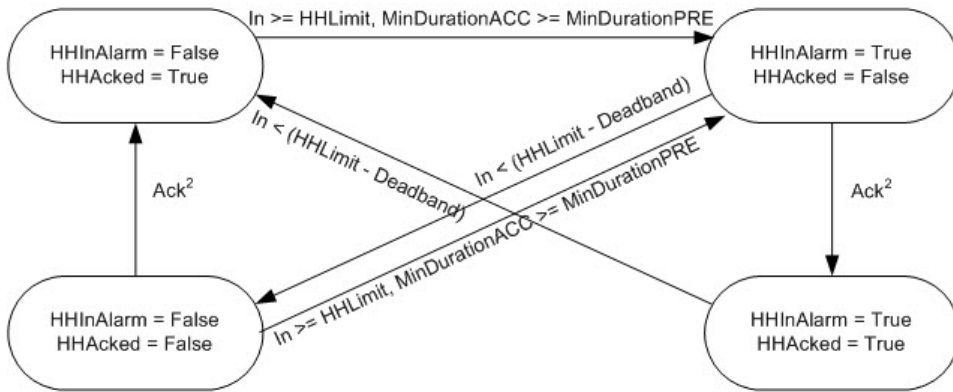


$$ROC = \frac{In(\text{Current Sample}) - In(\text{Previous Sample})}{ROCPeriod}$$

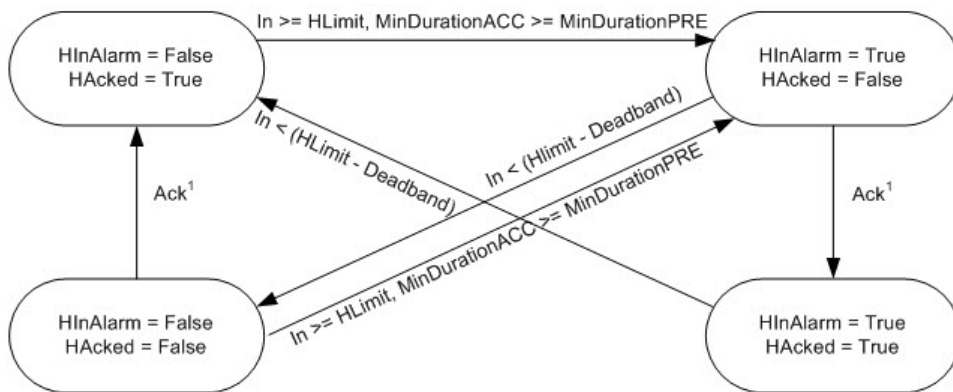
Where a new sample is collected on the next scan after the ROCPeriod has elapsed.



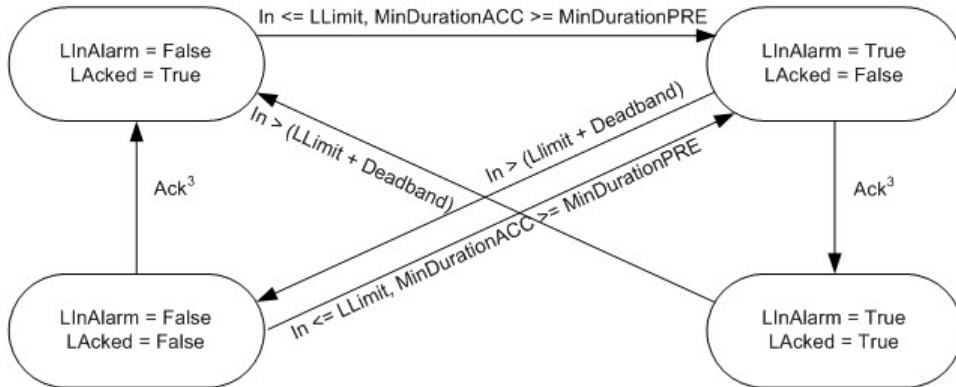
**AckRequired = True**



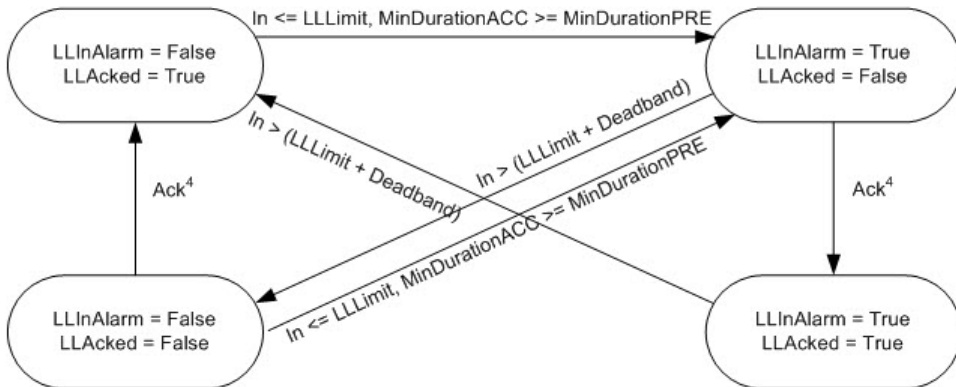
<sup>2</sup> HH alarm condition can be acked by several different ways: HHProgAck, HHOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)



<sup>1</sup> H alarm condition can be acked by several different ways: HProgAck, HOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

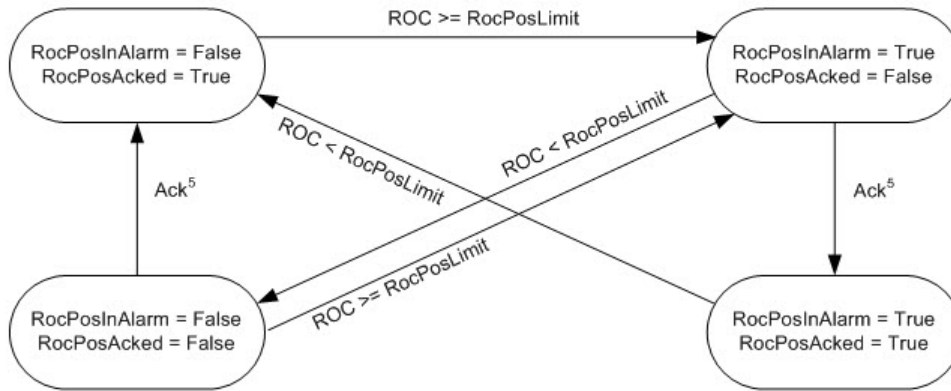


<sup>3</sup> L alarm condition can be acked by several different ways: LProgAck, LOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

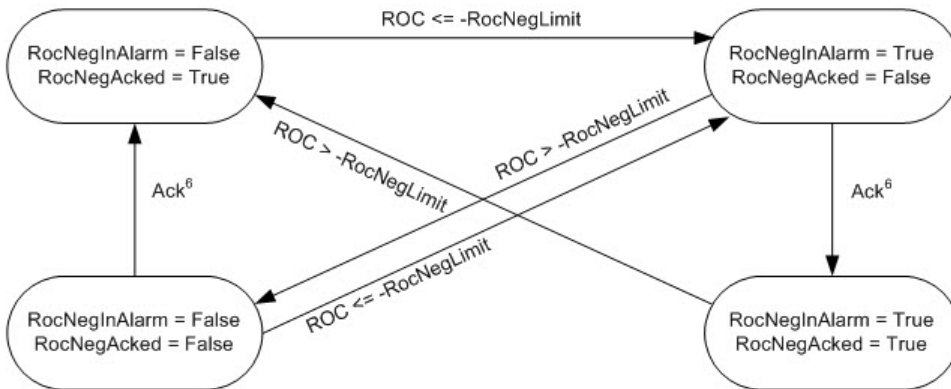


<sup>4</sup> LL alarm condition can be acked by several different ways: LLProgAck, LLOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)





<sup>5</sup> RocPos alarm condition can be acked by several different ways: RocPosProgAck, RocPosOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

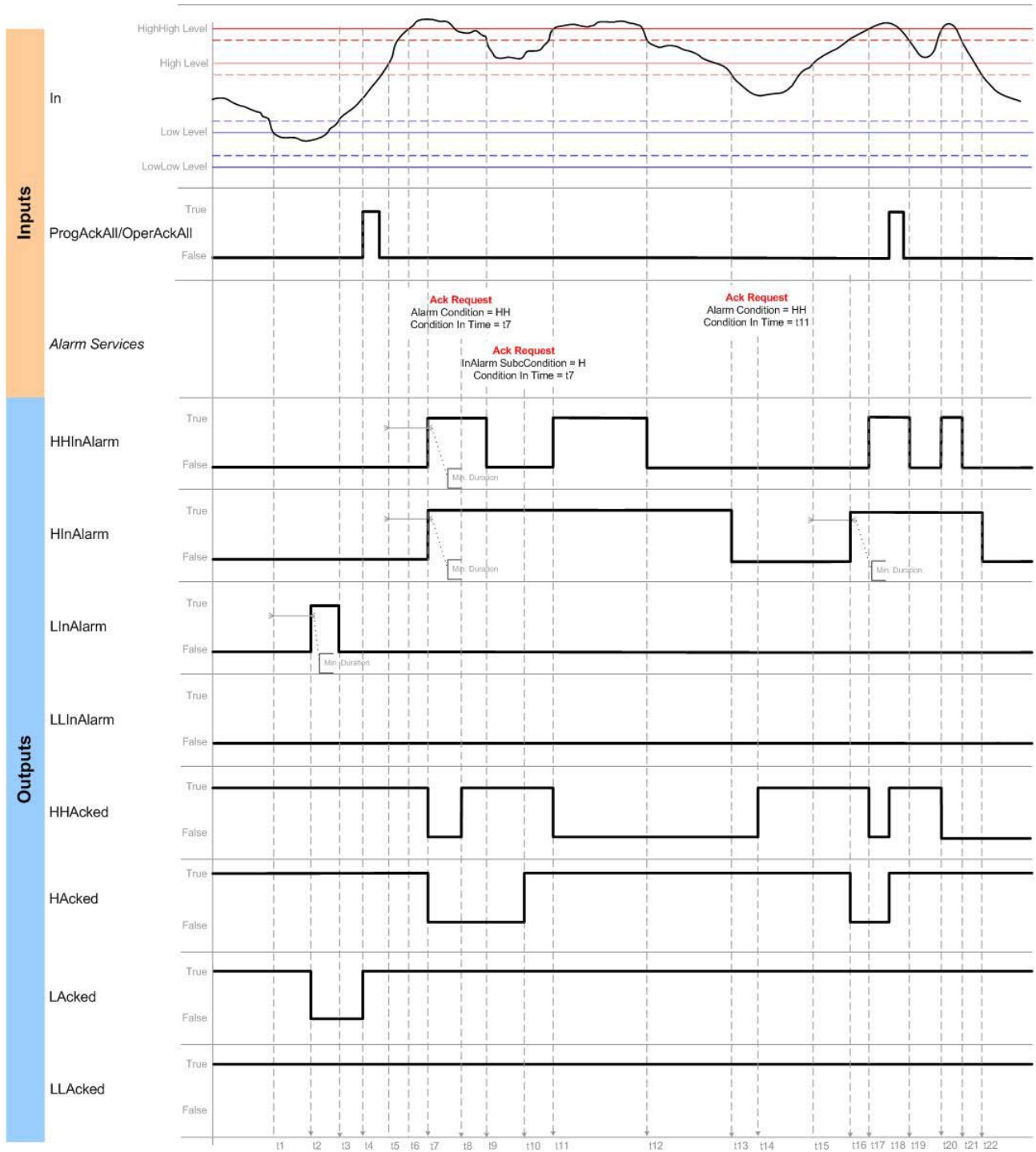


<sup>6</sup> RocNeg alarm condition can be acked by several different ways: RocNegProgAck, RocNegOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

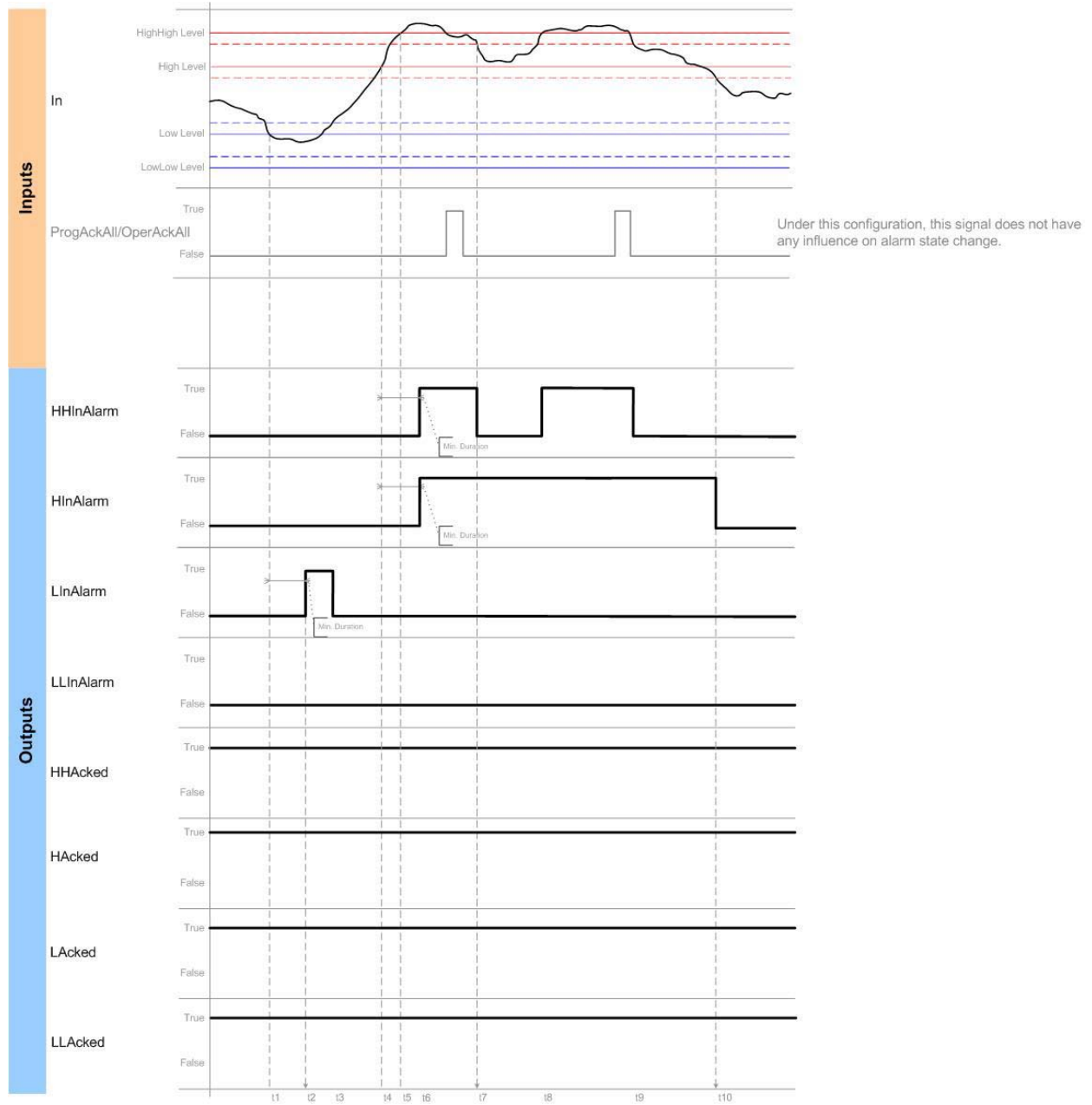
### Diagrammes temporels d'alarme analogique

Ces diagrammes temporels montrent la séquence des opérations d'alarme analogique.

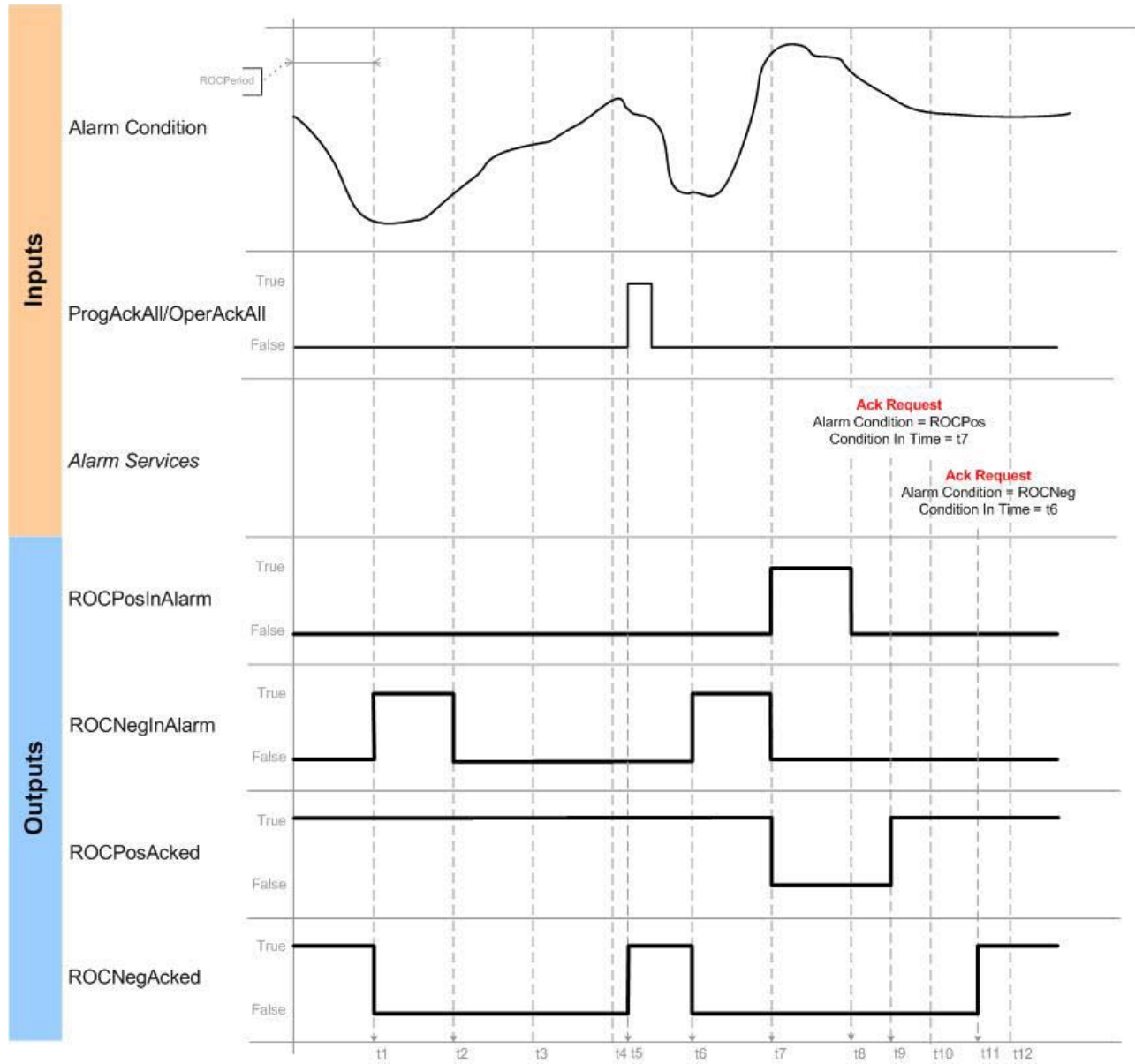
### Confirmation du comportement des conditions de niveau



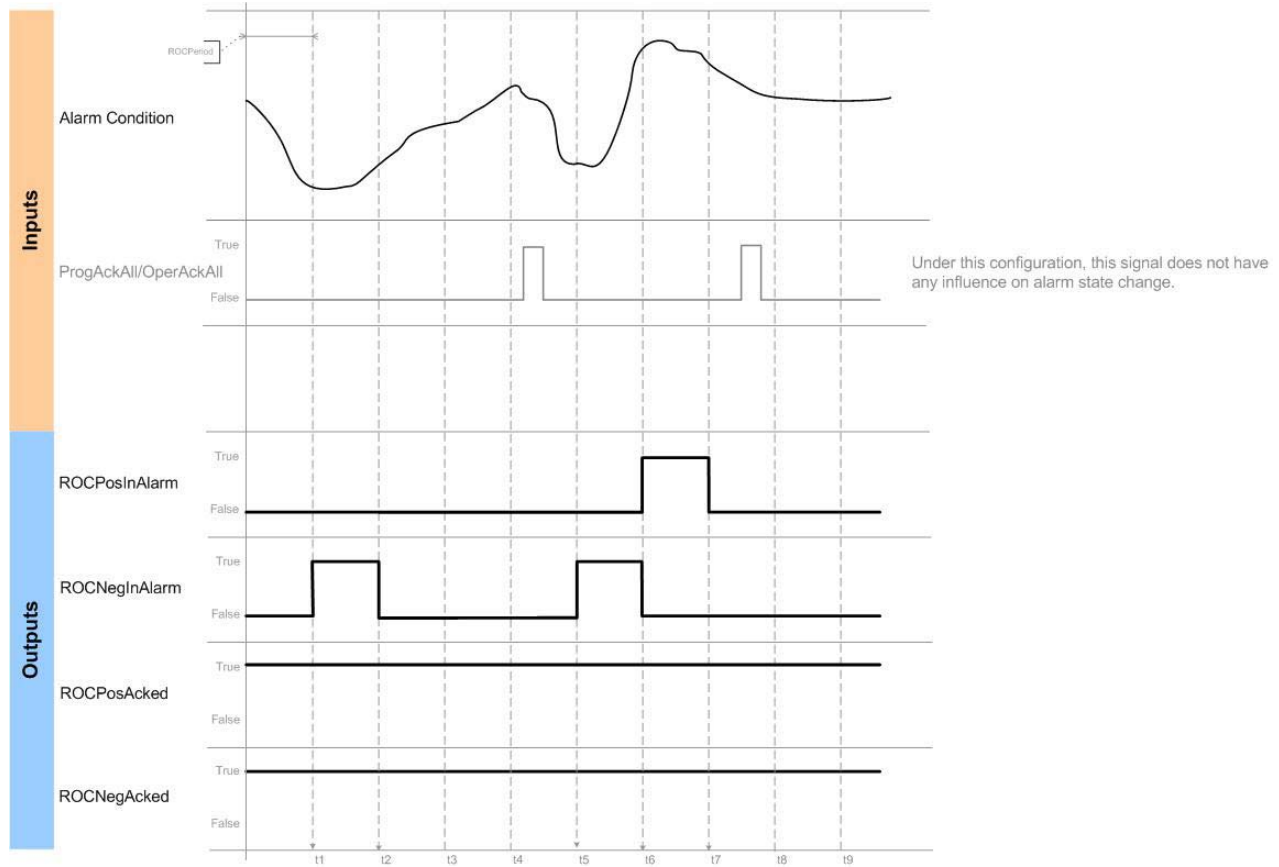
**Aucune confirmation du comportement des conditions de niveau**



Confirmation du comportement des conditions ROC



**Aucune confirmation du comportement des conditions ROC**



**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	La condition de sortie d'échelon est définie sur faux. Tous les paramètres de structure ALMA sont mis à zéro Toutes les conditions d'alarme sont confirmées. Toutes les demandes d'opérateurs sont mises à zéro. Tous les horodatages sont mis à zéro. Tous les indicateurs de remise sont mis à zéro
Condition d'entrée d'échelon est fausse	La condition de sortie d'échelon est définie sur faux.
Condition d'entrée d'échelon est vraie	La condition de sortie d'échelon est définie sur vrai L'instruction s'exécute
Post-scrutation	La condition de sortie d'échelon est définie sur faux

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	Tag.EnableOut est défini sur false (faux). Tous les paramètres de structure ALMA sont mis à zéro Toutes les conditions d'alarme sont confirmées. Toutes les demandes d'opérateurs sont mises à zéro. Tous les horodatages sont mis à zéro. Tous les indicateurs de remise sont mis à zéro
Tag.EnableIn est faux	Tag.EnableOut est défini sur faux
Tag.EnableIn est vrai	L'instruction s'exécute Tag.EnableOut est défini sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Tag.EnableOut est défini sur faux

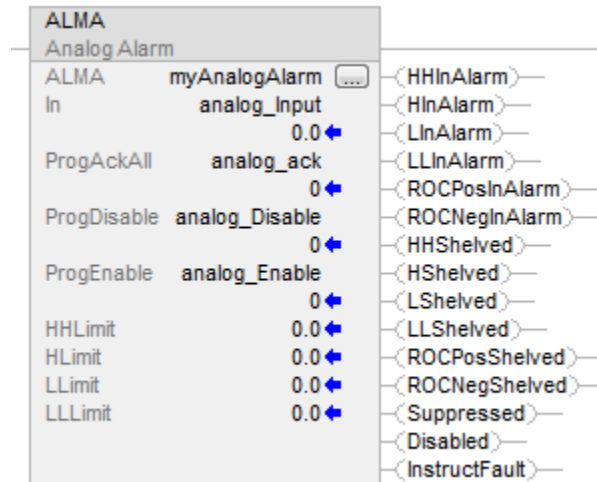
**Texte structuré**

Dans Texte Structuré, EnableIn est toujours vrai lors d'une scrutation normale. Par conséquent, si l'instruction est dans le chemin d'accès de commande activé par la logique, elle sera exécutée.

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

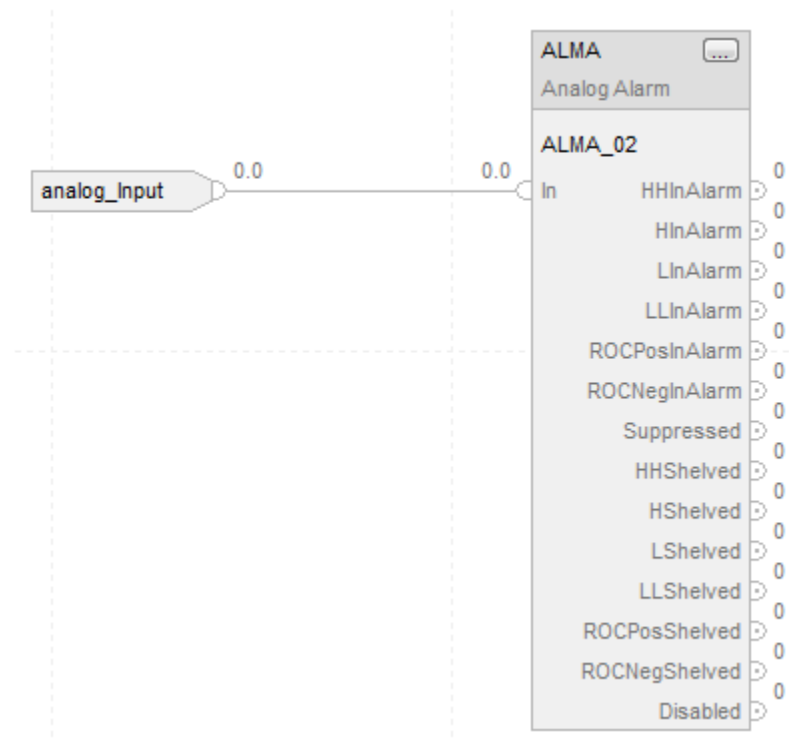
### Exemples

#### Diagramme à relais



#### Bloc fonctionnel

L'exemple ci-dessous montre une instruction ALMA dans un bloc fonctionnel. Dans cet exemple, les conditions d'alarme du transmetteur de niveau du réservoir 32 (Tank32LT) sont surveillées. L'étiquette Tank32LevelAck peut être utilisée pour confirmer toutes les conditions de cette alarme.



### Texte structuré

Dans cet exemple, les conditions d'alarme du transmetteur de niveau du réservoir 32 (Tank32LT) sont surveillées. L'étiquette Tank32LevelAck peut être utilisée pour confirmer toutes les conditions de cette alarme.

```
ALMA ( Tank32Level , Tank32LT , Tank32LevelAck , 0 , 0 ) ;
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Indexer via des tableaux](#) sur la [page 899](#)

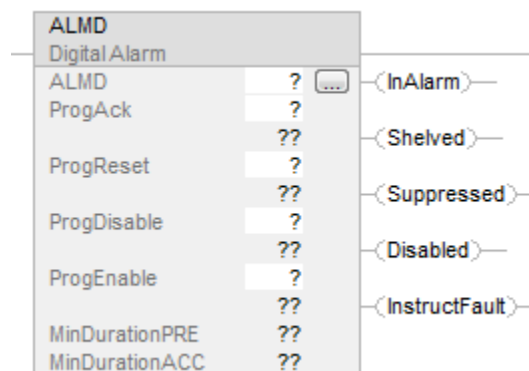
## Alarme numérique (ALMD)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction ALMD fournit des alarmes pour toute valeur booléenne discrète.

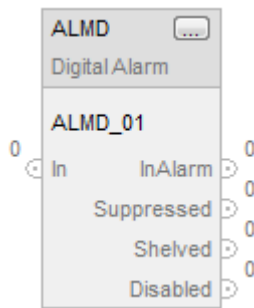
### Langages disponibles

### Diagramme à relais





**Bloc fonctionnel**



**Texte structuré**

ALMD (ALMD, In, ProgAck, ProgReset, ProgDisable, ProgEnable)

**Opérandes**

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
ALMD tag	ALARM_DIGITAL	Structure	Structure ALMD
ProgAck	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, confirme l'alarme (si une confirmation est requise).
ProgReset	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, réinitialise l'alarme (si une confirmation est requise).
ProgDisable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, désactive l'alarme (n'outrepasse pas les commandes d'activation).
ProgEnable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, active l'alarme (est prioritaire sur les commandes d'activation).
MinDurationPRE	DINT	Immédiate	Spécifie pendant combien de temps la condition d'alarme doit être satisfaite avant d'être signalée (millisecondes).
MinDurationACC	DINT	Immédiate	Indique la valeur actuelle du totalisateur pour le temporisateur de MinDuration.

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
ALMD tag	ALARM_DIGITAL	structure	Structure ALMD

**Texte structuré**

Opérande	Type (Type)	Format	Description (Description)
ALMD tag	ALARM_DIGITAL	Structure	Structure ALMD
ProgAck	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, confirme l'alarme (si une confirmation est requise).

ProgReset	BOOL	Étiquette immédiate	Lors d'une transition de faux à vrai, réinitialise l'alarme (si une confirmation est requise).
ProgDisable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, désactive l'alarme (n'outrepasse pas les commandes d'activation).
ProgEnable	BOOL	Étiquette immédiate	Lorsque définir sur vrai, active l'alarme (est prioritaire sur les commandes d'activation).
MinDurationPRE	DINT	Immédiate	Spécifie pendant combien de temps la condition d'alarme doit être satisfaite avant d'être signalée (millisecondes).
MinDurationACC	DINT	Immédiate	Indique la valeur actuelle du totalisateur pour le temporisateur de MinDuration.

Reportez-vous à Syntaxe du texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Structure ALMD

#### Paramètres d'entrée

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Diagramme à relais : Correspond à l'état d'échelon. N'affecte pas le traitement. Bloc fonctionnel : Si définir sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est vraie. Texte structuré : Aucun effet. L'instruction s'exécute toujours.
In	BOOL	Entrée du signal numérique dans l'instruction. La valeur par défaut est fausse. Diagramme à relais : Suit la condition d'échelon. Définir vrai si la condition d'échelon est vraie. Définir sur faux si la condition d'échelon est fausse. Texte structuré : copié à partir de l'opérande d'instruction.
InFault	BOOL	Indicateur de mauvaise santé pour l'entrée. L'application utilisateur peut définir InFault pour indiquer que le signal d'entrée comporte une erreur. Si défini, l'instruction définit InFaulted (Status.1). Si défini sur faux, l'instruction définit InFaulted (Status.1) sur faux. Dans les deux cas, l'instruction continue d'évaluer In pour des conditions d'alarme. La valeur par défaut est fausse (bonne santé).
Condition	BOOL	Indique le mode d'activation de l'alarme. Si Condition est défini sur vrai, la condition d'alarme est activée si In est défini sur vrai. Si Condition est défini sur faux, la condition d'alarme est activée si In est défini sur faux. La valeur par défaut est vraie.
AckRequired	BOOL	Indique si une confirmation d'alarme est requise. Si définir sur vrai, une confirmation est requise. Si ce paramètre est défini sur faux, une confirmation n'est pas requise et Acked est toujours défini sur vrai. La valeur par défaut est vraie.

Paramètre d'entrée	Type de donnée	Description (Description)
Latched	BOOL	Indique si l'alarme est verrouillée. Les alarmes verrouillées restent InAlarm lorsque la condition d'alarme devient faux, jusqu'à ce qu'une commande de réinitialisation soit reçue. Si définir sur vrai, l'alarme est verrouillée. Si définir sur faux, l'alarme est déverrouillée. La valeur par défaut est fausse. Une alarme verrouillée ne peut être réinitialisée que si la condition d'alarme est définie sur faux.
ProgAck	BOOL	Définir sur vrai par le programme utilisateur pour confirmer l'alarme. Prend effet uniquement si l'alarme n'est pas confirmée. Requiert une transition de faux à vrai. La valeur par défaut est fausse. Diagramme à relais : Copié à partir de l'opérande d'instruction. Texte structuré : Copié à partir de l'opérande d'instruction.
OperAck	BOOL	Définir sur vrai par l'interface opérateur pour confirmer l'alarme. Prend effet uniquement si l'alarme n'est pas confirmée. L'instruction efface ce paramètre. La valeur par défaut est fausse.
ProgReset	BOOL	Définir sur vrai par le programme utilisateur pour réinitialiser l'alarme verrouillée. Prend effet uniquement si l'alarme verrouillée est associée à InAlarm et si la condition d'alarme est définie sur faux. Requiert une transition de faux à vrai. La valeur par défaut est fausse. Diagramme à relais : Copié à partir de l'opérande d'instruction. Texte structuré : Copié à partir de l'opérande d'instruction.
OperReset	BOOL	Définir sur vrai par l'interface opérateur pour réinitialiser l'alarme verrouillée. Prend effet uniquement si l'alarme verrouillée est associée à InAlarm et si la condition d'alarme est définie sur faux. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ProgSuppress	BOOL	Définir sur vrai par le programme utilisateur pour supprimer l'alarme. La valeur par défaut est fausse.
OperSuppress	BOOL	Définir sur vrai par l'interface opérateur pour supprimer l'alarme. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
ProgUnsuppress	BOOL	Définir sur vrai par le programme utilisateur pour annuler la suppression de l'alarme. Est prioritaire sur les commandes de suppression. La valeur par défaut est fausse.
OperUnsuppress	BOOL	Définir sur vrai par l'interface opérateur pour annuler la suppression de l'alarme. Est prioritaire sur les commandes de suppression. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.

Paramètre d'entrée	Type de donnée	Description (Description)
OperShelve	BOOL	Définir sur vrai par l'interface opérateur pour réserver ou réserver à nouveau de l'alarme. Requiert une transition de faux dans une scrutation de programme à vrai dans la scrutation de programme suivante. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse. Les commandes d'annulation de réservation sont prioritaires sur les commandes de réservation. La réservation d'une alarme reporte son traitement. Cette opération est semblable à la suppression d'une alarme, si ce n'est que la réservation est limitée dans le temps. Si une alarme est confirmée pendant qu'elle est réservée, elle conserve cet état même si elle redevient active. Sa confirmation est annulée lorsque la réservation arrive à son terme, à condition que l'alarme soit encore active à ce moment donné.
ProgUnshelve	BOOL	Définir sur vrai par le programme utilisateur pour annuler la réservation de l'alarme. Est prioritaire sur les commandes de réservation. La valeur par défaut est fausse. Pour obtenir des informations supplémentaires sur la réservation des alarmes, reportez-vous à la description du paramètre OperShelve.
OperUnshelve	BOOL	Définir sur vrai par l'interface opérateur pour annuler la réservation de l'alarme. L'instruction d'alarme définit ce paramètre sur faux. Est prioritaire sur les commandes de réservation. La valeur par défaut est mise à zéro. Pour obtenir des informations supplémentaires sur la réservation des alarmes, reportez-vous à la description du paramètre OperShelve.
ProgDisable	BOOL	Définir sur vrai par le programme utilisateur pour désactiver l'alarme. La valeur par défaut est fausse. Diagramme à relais : Copié à partir de l'opérande d'instruction. Texte structuré : Copié à partir de l'opérande d'instruction.
OperDisable	BOOL	Définir sur vrai par l'interface opérateur pour désactiver l'alarme. L'instruction d'alarme définit ce paramètre sur vrai. La valeur par défaut est fausse.
ProgEnable	BOOL	Définir sur vrai par le programme utilisateur pour activer l'alarme. Est prioritaire sur une commande de désactivation. La valeur par défaut est fausse. Diagramme à relais : Copié à partir de l'opérande d'instruction. Texte structuré : Copié à partir de l'opérande d'instruction.
OperEnable	BOOL	Définir sur vrai par l'interface opérateur pour activer l'alarme. Est prioritaire sur la commande de désactivation. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
AlarmCountReset	BOOL	Définir sur vrai par l'interface opérateur pour réinitialiser le comptage d'alarme à zéro. L'instruction d'alarme définit ce paramètre sur faux. La valeur par défaut est fausse.
UseProgTime	BOOL	Spécifie si l'horloge de l'automate ou la valeur ProgTime est utilisée pour horodater les événements de changement d'état des alarmes. Si définir sur vrai, la valeur ProgTime fournit l'horodatage. Si définir sur faux, l'horloge de l'automate fournit l'horodatage. La valeur par défaut est fausse.

Paramètre d'entrée	Type de donnée	Description (Description)
ProgTime	LINT	Si UseProgTime est défini sur vrai, cette valeur est utilisée pour fournir la valeur d'horodatage pour tous les événements. Cela permet à l'application d'appliquer des horodatages obtenus à partir de la source de l'alarme, comme un module d'entrée de séquence des événements.
Severity	DINT	Sévérité de l'alarme. Cela n'affecte pas le traitement des alarmes par l'automate, mais est utile pour les fonctions de tri et de filtre de l'abonné de l'alarme. Valide = de 1 à 1 000 (1 000 = la plus sévère ; 1 = la moins sévère). Valeur par défaut = 500.
MinDurationPRE	DINT	Durée minimale prédéfinie (en millisecondes) pendant laquelle une condition d'alarme doit être définie sur vrai avant que l'alarme ne soit marquée comme InAlarm et qu'une notification d'alarme ne soit envoyée aux clients. L'automate collecte les données d'alarme dès que la condition d'alarme est détectée, ce qui évite la perte de données jusqu'à ce que la condition de durée minimale soit remplie. Valide = de 0 à 2 147 483 647. Valeur par défaut = 0.
ShelveDuration	DINT	Durée en minutes de réservation d'une alarme. La réservation d'une alarme reporte son traitement. Cette opération est semblable à la suppression d'une alarme, si ce n'est que la réservation est limitée dans le temps. Si une alarme est confirmée pendant qu'elle est réservée, elle conserve cet état même si elle redevient active. Sa confirmation est annulée lorsque la réservation arrive à son terme (à condition que l'alarme soit encore active à ce moment donné). La durée minimale est d'une minute. La durée maximale est définie par MaxShelveDuration.
MaxShelveDuration	DINT	Durée maximale en minutes pendant laquelle une alarme peut être réservée. Pour obtenir des informations supplémentaires sur la réservation des alarmes, reportez-vous à la description du paramètre ShelveDuration.

### Paramètres de sortie

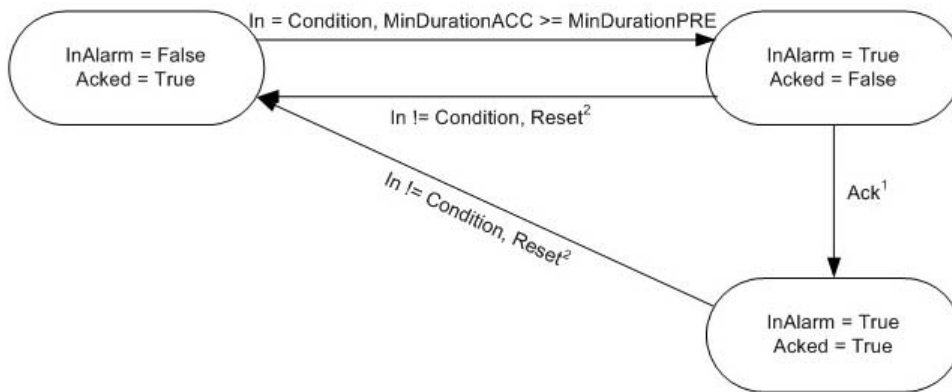
Paramètre de sortie	Type Type (Type)	Description (Description)
EnableOut	BOOL	Activer la sortie.
InAlarm	BOOL	État actif d'alarme. Définir sur vrai si l'alarme est active. Définir sur faux si l'alarme n'est pas active (état normal).
Acked	BOOL	État confirmé de l'alarme. Définir sur vrai si l'alarme est confirmée. Définir sur faux si l'alarme n'est pas confirmée. Acked est toujours défini sur vrai lorsque AckRequired est défini sur faux.
InAlarmUnack	BOOL	État confirmé et actif d'une alarme. Définir sur vrai si l'alarme est active (InAlarm est défini sur vrai) et non confirmée (Acked est défini sur faux). Définir sur faux si l'alarme est inactive, confirmée ou les deux.
Suppressed	BOOL	État supprimé de l'alarme. Définir sur vrai si l'alarme est supprimée. Définir sur faux si l'alarme n'est pas supprimée.

Shelved	BOOL	État de réservation de l'alarme. Définir sur vrai si l'alarme est réservée. Définir sur faux si l'alarme n'est pas réservée. La réservation d'une alarme reporte son traitement. Cette opération est semblable à la suppression d'une alarme, si ce n'est que la réservation est limitée dans le temps. Si une alarme est confirmée pendant qu'elle est réservée, elle conserve cet état même si elle redevient active. Sa confirmation est annulée lorsque la réservation arrive à son terme.
Désactivé	BOOL	État désactivé de l'alarme. Définir sur vrai si l'alarme n'est pas activée. Définir sur faux si l'alarme est activée.
Demandée	BOOL	État de demande de l'alarme. Définir sur vrai lorsque l'alarme est commissionnée. Définir sur faux lorsque l'alarme est décommissionnée. Actuellement toujours définir sur vrai.
MinDurationACC	DINT	Non utilisé. La valeur est toujours 0.
AlarmCount	DINT	Nombre de fois où l'alarme a été activée (InAlarm est défini). Si la valeur maximale est atteinte, le compteur limite la valeur à la valeur maximale.
InAlarmTime	LINT	Horodatage de la détection de l'alarme.
AckTime	LINT	Horodatage de la confirmation de l'alarme. Si l'alarme ne nécessite pas de confirmation, cet horodatage correspond à l'heure de l'alarme.
RetToNormalTime	LINT	Horodatage de l'alarme revenant à un état normal.
AlarmCountResetTime	LINT	Horodatage indiquant lorsque le comptage d'alarme a été réinitialisé.
ShelveTime	LINT	Horodatage indiquant quand a eu lieu la dernière réservation de l'alarme. Cette valeur est définie par l'automate lorsqu'une alarme est réservée. L'alarme peut être réservée et sa réservation peut être annulée à de nombreuses reprises. À chaque réservation d'une alarme, l'horodatage est défini sur l'heure actuelle. Pour obtenir des informations supplémentaires sur la réservation des alarmes, reportez-vous à la description du paramètre Shelved.
UnshelveTime	LIN	Horodatage indiquant quand la réservation de l'alarme sera annulée. Cette valeur est définie à chaque réservation de l'alarme (même si celle-ci a déjà été réservée). Cet horodatage est obtenu en ajoutant la valeur ShelveDuration à l'heure actuelle. Si la réservation de l'alarme est annulée par programmation ou par un opérateur, cette valeur est définie sur l'heure actuelle. Pour obtenir des informations supplémentaires sur la réservation des alarmes, reportez-vous à la description du paramètre Shelved.
État (Status)	DINT	Indicateurs d'état associés : Status.0 = InstructFault Status.1 = InFaulted Status.2 = SeverityInv
InstructFault (Status.0)	BOOL	Des conditions d'erreur d'instruction existent. Il ne s'agit pas d'une erreur d'automate mineure ou majeure. Contrôlez les bits d'état restants pour déterminer la cause du problème.
InFaulted (Status.1)	BOOL	Le programme utilisateur a défini InFault pour indiquer la mauvaise qualité des données d'entrée. L'alarme continue d'évaluer In pour la condition d'alarme.
SeverityInv (Status.2)	BOOL	Configuration de sévérité de l'alarme. Si la sévérité est <1, l'instruction utilise Sévérité = 1. Si la sévérité est >1 000, l'instruction utilise Sévérité = 1 000.

### Diagrammes d'état d'alarmes numériques

**Acknowledgement Required, Latched**

**AckRequired = True, Latched = True**

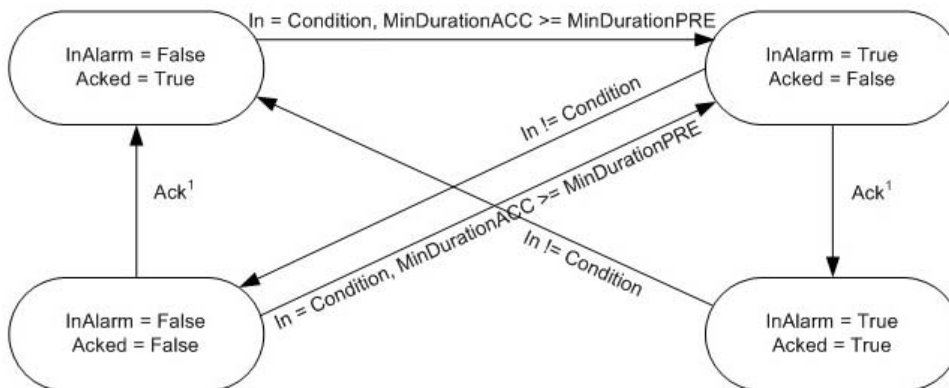


<sup>1</sup> Alarm can be acked by several different ways: ProgAck, OperAck, clients (RSLogix 5000, RSview)

<sup>2</sup> Alarm can be reset by several different ways: ProgReset, OperReset, clients (RSLogix 5000, RSview)

**Acknowledgement Required, Not Latched**

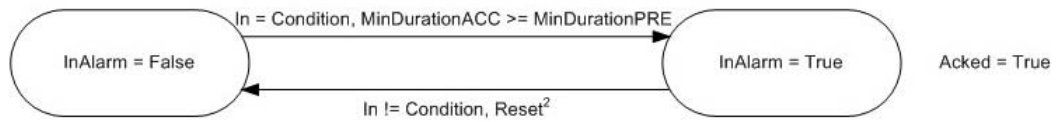
**AckRequired = True, Latched = False**



<sup>1</sup> Alarm can be acked by several different ways: ProgAck, OperAck, clients (RSLogix 5000, RSview)

**Acknowledgement Not Required, Latched**

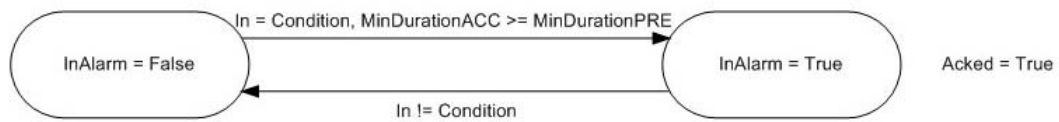
**AckRequired = False, Latched = True**



<sup>2</sup> Alarm can be reset by several different ways: ProgReset, OperReset, clients (RSLogix 5000, RSview)

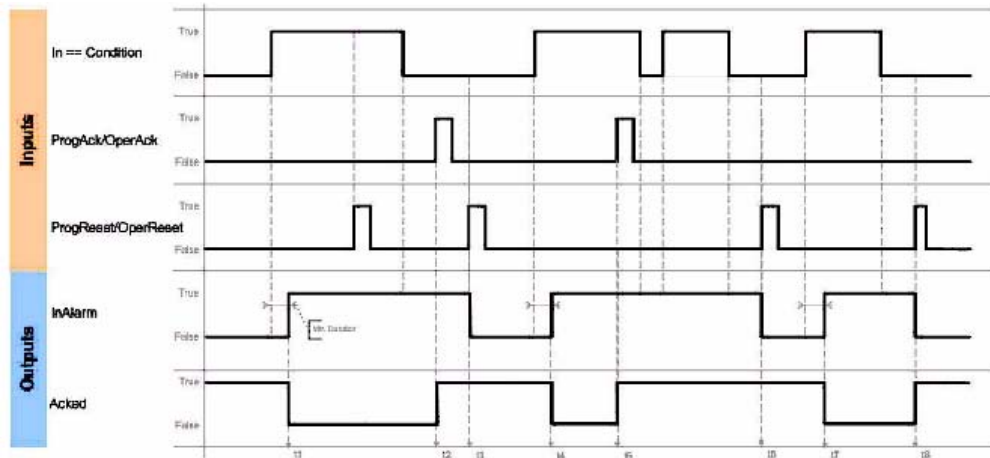
**Acknowledgement Not Required, Not Latched**

**AckRequired = False, Latched = False**



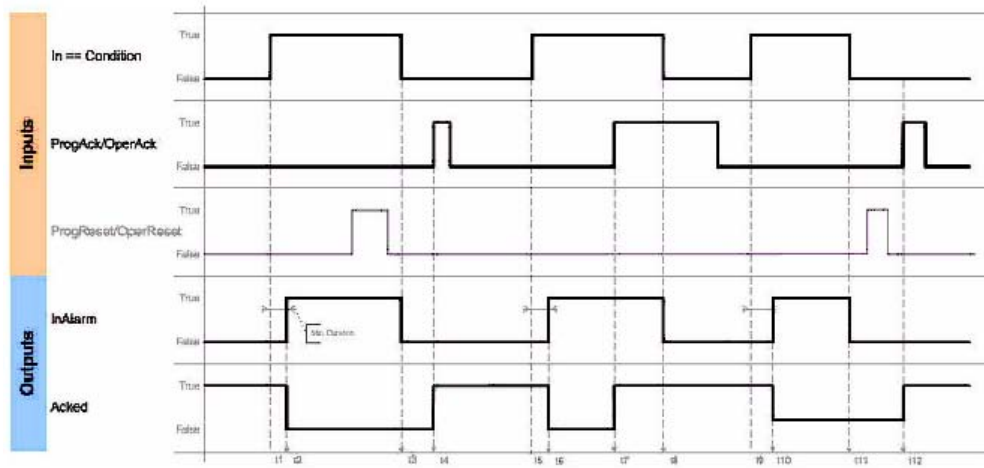
**Diagrammes temporels d'alarme numérique**

**Confirmation d'alarme ALMD requise et verrouillée**

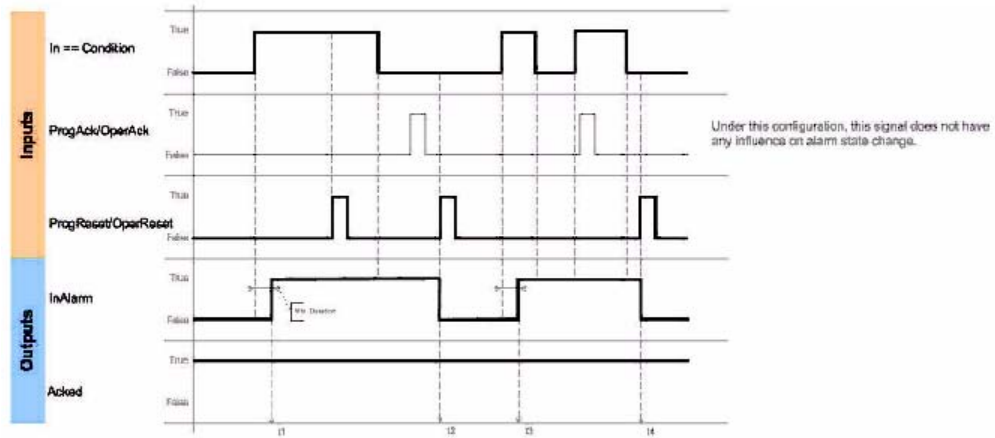




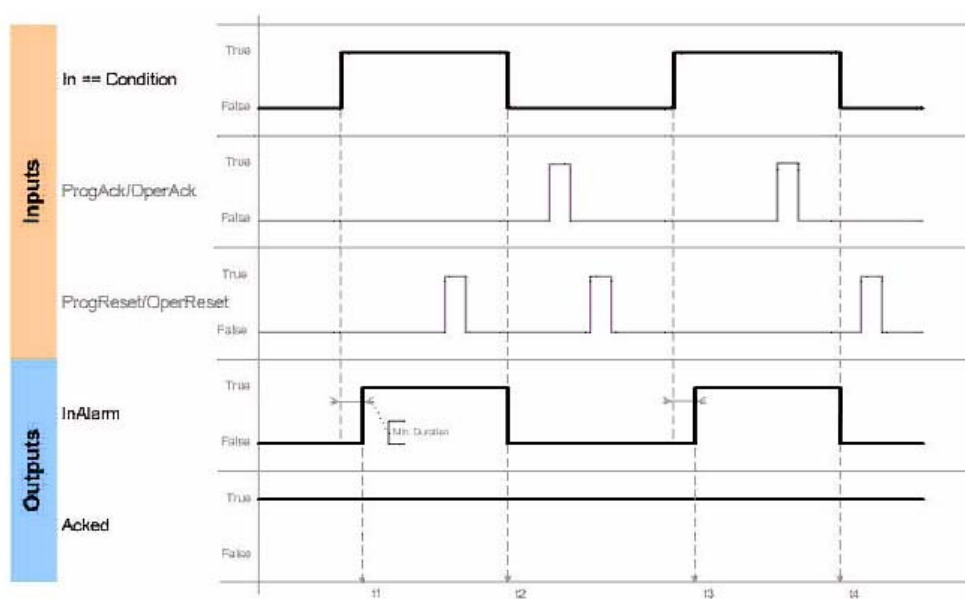
### Confirmation d'alarme ALMD requise et non verrouillée



### Confirmation d'alarme ALMD non requise et verrouillée



### Confirmation d'alarme ALMD non requise et non verrouillée



### Connecter un bouton à l'étiquette OperShelve

Pour éviter toute nouvelle réservation indésirable de l'alarme, l'instruction d'alarme traite l'étiquette OperShelve uniquement si une transition de faux à vrai a lieu entre une scrutation de programme et la scrutation suivante. Si un opérateur appuie sur un bouton-poussoir pour réserver l'alarme alors que l'étiquette ProgUnshelve est définie sur vrai, l'alarme n'est pas réservée, car ProgUnshelve est prioritaire. Cependant, puisque les scrutations de programme durent quelques millisecondes, l'opérateur peut maintenir le bouton-poussoir enfoncé afin que l'étiquette OperShelve reste définie sur vrai pour plusieurs scrutations de programme, même si l'étiquette ProgUnshelve a été définie sur faux. Cela signifie que l'alarme n'est pas réservée.

Pour réserver l'alarme, l'opérateur peut relâcher le bouton, puis appuyer de nouveau dessus.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	EnableOut est défini sur faux La sortie InAlarm est définie sur faux La sortie Shelved est définie sur faux La sortie Acked est définie sur vrai. Toutes les conditions d'alarme sont confirmées. Toutes les demandes d'opérateurs sont mises à zéro. Tous les horodatages sont mis à zéro.
Condition d'entrée d'échelon est fausse	Échelon est défini sur faux. Le paramètre In est défini sur faux L'instruction s'exécute.
Condition d'entrée d'échelon est vraie	Échelon est défini sur vrai. Le paramètre In est défini sur vrai L'instruction s'exécute.
Post-scrutation	Le bit d'échelon est défini sur faux.

### Bloc fonctionnel

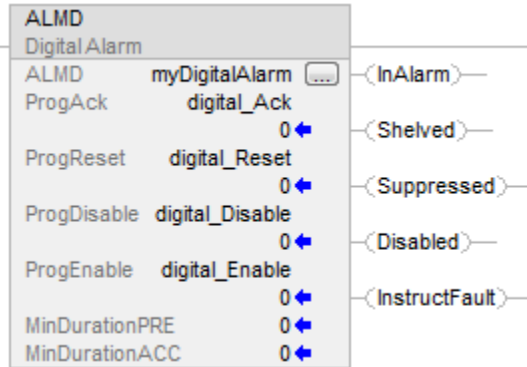
Condition/État	Action entreprise
Pré-scrutation	Tag.EnableOut est défini sur false (faux). La sortie InAlarm est définie sur faux La sortie Shelved est définie sur faux La sortie Acked est définie sur vrai Toutes les demandes d'opérateurs sont mises à zéro. Tous les horodatages sont mis à zéro.
Tag.EnableIn est faux	Tag.EnableOut est défini sur faux
Tag.EnableIn est vrai	L'instruction s'exécute Tag.EnableOut est défini sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Tag.EnableOut est défini sur false (faux).

### Texte structuré

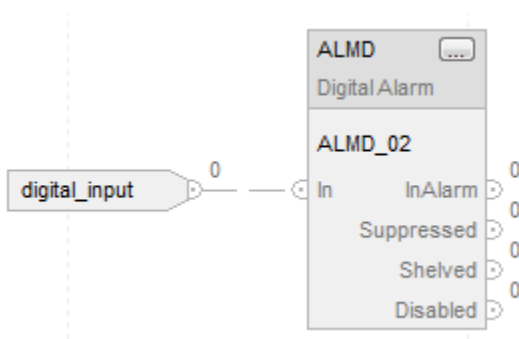
Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

L'exemple ci-dessous montre une instruction ALMD dans du texte structuré. Dans cet exemple, deux signaux de défaillance de moteur sont combinés de telle sorte que si l'un d'eux apparaît, une alarme de défaut moteur est activée. L'étiquette Motor101Ack peut être utilisée pour confirmer l'alarme.

```
Motor101FaultConditions := Motor101Overtemp OR Motor101FailToStart;

ALMD(Motor101Fault, Motor101FaultConditions, Motor101Ack, 0,
0, 0 );
```

**Voir aussi**

- [Syntaxe du texte structuré](#) sur la [page 917](#)
- [Indicateurs d'état mathématique](#) sur la [page 885](#)
- [Indexer via des tableaux](#) sur la [page 899](#)

## Opération d'ensemble d'alarmes (ASO)

Ces informations s'appliquent aux automates Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction Opération d'ensemble d'alarmes génère une opération spécifiée pour toutes les conditions d'alarme de l'ensemble d'alarmes spécifié. L'instruction Opération d'ensemble d'alarmes s'utilise pour initier une exécution asynchrone d'une opération d'alarme pour toutes les conditions d'alarme de l'ensemble d'alarmes spécifié. L'instruction effectue des itérations dans les conditions d'alarme de l'ensemble d'alarmes spécifié et définit un indicateur interne demandant l'exécution de l'opération pour chacune des conditions. Les indicateurs internes ont les mêmes fonction et priorité que les bits Progxxx accessibles de l'utilisateur existant et seront traités pour toutes les conditions d'alarme de l'ensemble d'alarmes spécifié lors de la prochaine évaluation périodique de chaque condition d'alarme de l'ensemble.

### Langages disponibles

#### Diagramme à relais

ASO	
Alarm Set Operation	
Alarm Set	?
Alarm Set Control	?
Operation	?

#### Diagramme de bloc fonctionnel

Cette instruction n'est pas disponible dans le diagramme de blocs fonctionnels.

#### Texte structuré

ASO (Ensemble d'alarmes, contrôle de l'ensemble d'alarmes, opération)

#### Opérandes

- 
- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Le même indicateur (ALARM\_SET\_CONTROL) est utilisé comme paramètre pour plusieurs invocations d'instructions.
  - Le membre de structure .LastState est modifié par un programme d'application utilisateur.
- 



**ATTENTION :** la structure Contrôle de l'ensemble d'alarmes contient des informations sur l'état interne. En cas de modification d'un des opérandes de l'instruction en mode Exécution, il faut accepter les éditions en attente et faire passer le mode de l'automate de Programme à Exécution pour que les modifications prennent effet.

---

Vous trouverez dans le tableau suivant l'opérande utilisé pour configurer l'instruction.

Opérande	Type de donnée	Format	Description (Description)
Ensemble d'alarmes	ALARM_SET	AlarmSet	La structure ALARM_SET représente des conditions d'alarme qui sont exploitées par cette instruction.
Contrôle de l'ensemble d'alarmes	ALARM_SET_CONTROL	étiquette	Ce type de données contient trois indicateurs BOOL : <ul style="list-style-type: none"> <li>• EnableIn</li> <li>• EnableOut</li> <li>• LastState</li> </ul> L'instruction réagit au front (transition de .EnableIn de faux à vrai) au lieu du niveau. EnableOut est toujours défini sur .EnableIn. La demande pour effectuer l'opération de l'instruction a la même priorité que les indicateurs ProgXXX.
Fonctionnement		immediate	Cet opérande peut être sélectionné dans la liste ou entré sous forme de valeur entière : <ul style="list-style-type: none"> <li>0 - Confirmer</li> <li>1 - Remettre à zéro</li> <li>2 - Activer</li> <li>3 - Désactiver</li> <li>4 - Annuler la réservation</li> <li>5 - Supprimer</li> <li>6 - Annuler la suppression</li> <li>7 - Réinitialiser le nombre d'alarmes</li> </ul>

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution**

Condition/État	Action entreprise
Pré-scrutation	L'instruction efface toutes les membres de structure ALARM_SET.
Condition d'entrée d'échelon est fausse	L'instruction efface les membres de structure .EnableOut et .LastState.
Condition d'entrée d'échelon est vraie	Si .LastState est faux, alors l'instruction initie l'opération et définit le membre de structure .LastState sur vrai. Le membre de structure .EnableOut est toujours défini sur vrai.
Post-scrutation	L'instruction efface toutes les membres de structure ALARM_SET.

## Fonctionnement

L'instruction Opération d'ensemble d'alarmes initie une exécution asynchrone de l'une des opérations d'alarme suivantes de l'ensemble d'alarmes spécifié :

- Confirmer
- Réinitialiser
- Activation
- Désactiver
- Annuler la réservation
- Supprimer
- Annuler la suppression
- Réinitialiser le nombre d'alarmes

L'instruction effectue des itérations dans les conditions d'alarme qui sont incluses dans l'ensemble d'alarmes spécifié ou dans les ensembles d'alarmes imbriqués pour définir un indicateur interne représentant la demande pour effectuer l'opération requise sur une condition d'alarme particulière. L'opération est initiée pour toutes les conditions d'alarme qui sont exécutées à plusieurs reprises par l'instruction, avec les exceptions suivantes :

- Conditions d'alarme qui ne sont pas configurées pour prendre en charge les opérations d'alarme
- Conditions d'alarme qui sont configurées comme non utilisées

Lorsqu'une opération d'alarme est initiée pour une condition d'alarme particulière par l'instruction, l'opération est effectuée lors de la prochaine évaluation périodique de la condition d'alarme.

Lorsque l'instruction est appelée à plusieurs reprises pour le même ensemble d'alarmes afin d'initier des opérations d'alarme contradictoires, la dernière opération demandée est toujours appliquée à toutes les conditions d'alarme du jeu d'alarmes. Les opérations d'alarme initiées pour le jeu d'alarmes peuvent être appliquées aux conditions avant l'exécution de la dernière opération demandée.

Lorsqu'une condition d'alarme est évaluée périodiquement, les demandes pour effectuer des opérations d'alarme particulières ont la même priorité que les demandes pour effectuer des opérations d'alarme initiées via des indicateurs Progxxx accessibles par l'utilisateur. Cela signifie que si une demande pour effectuer une opération d'alarme est générée par l'instruction, elle est alors traitée comme si l'indicateur Progxxx était défini et que les mêmes règles utilisées pour résoudre des demandes contradictoires spécifiées pour les indicateurs ProgXXX

étaient utilisées pour résoudre les conflits entre les demandes de l'instruction et celles effectuées via les indicateurs Progxxx.

L'instruction Opération d'ensemble d'alarmes initie l'opération d'alarme demandée seulement lorsqu'elle détecte la transition de la valeur .EnableIn de faux à vrai. Afin de détecter la transition, le membre de structure .LastState est utilisé pour stocker la valeur .EnableIn de l'exécution précédente de l'instruction. Reportez-vous à la section Exécution ci-dessus.

**Astuce :** Si l'ensemble d'alarmes fourni comme paramètre d'instruction contient un trop grand nombre de conditions d'alarme, le temps d'exécution de l'instruction ASO peut augmenter considérablement.

#### Voir aussi

[Instructions d'alarme](#) sur la [page 27](#)

[Indexer via des tableaux](#) sur la [page 899](#)



# Instructions de bit

## Instructions de bit

Utilisez les instructions des bits (relais-type) pour surveiller et contrôler l'état des bits, tels que les bits d'entrée ou les bits du mot de commande du temporisateur.

### Instructions disponibles

#### Diagramme à relais

<a href="#">XIC</a>	<a href="#">XIO</a>	<a href="#">OTE</a>	<a href="#">OTL</a>	<a href="#">OTU</a>	<a href="#">ONS</a>	<a href="#">OSR</a>	<a href="#">OSF</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

#### Bloc fonctionnel et Texte structuré

<a href="#">OSRI</a>	<a href="#">OSFI</a>
----------------------	----------------------

Si vous voulez :	Utilisez cette instruction :
Activer les sorties lorsqu'un bit est défini	XIC
Activer les sorties lorsqu'un bit est mis à zéro	XIO
définir un bit	OTE
définir un bit (rémanent)	OTL
mettre un bit à zéro (rémanent)	OTU
Activer les sorties pour une scrutation chaque fois qu'un échelon passe à l'état vrai	ONS
définir un bit pour une scrutation chaque fois qu'un échelon passe à l'état vrai	OSR
définir un bit pour une scrutation chaque fois qu'un échelon passe à l'état faux	OSF
définir un bit pour une scrutation chaque fois que le bit d'entrée est défini dans le bloc fonctionnel	OSRI
définir un bit pour une scrutation chaque fois que le bit d'entrée est mis à zéro dans le bloc fonctionnel	OSFI

**Voir aussi**

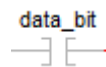
[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de calcul/Math](#) sur la [page 373](#)

**Examiner si fermé (XIC)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction XIC examine le bit de données pour définir ou mettre la condition d'échelon à zéro.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes****Diagramme à relais**

Opérande	Type de données	Format	Description
Bit de données	BOOL	étiquette	Bit à tester. Il existe divers mode d'adressage d'opérande possibles pour le bit de données, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

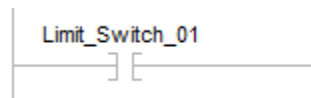
**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Si DataBit est vrai, la condition de sortie d'échelon est définie sur vrai. Si DataBit est faux, la condition de sortie d'échelon est définie sur faux.
Post-scrutation	N/A

**Exemple 1**

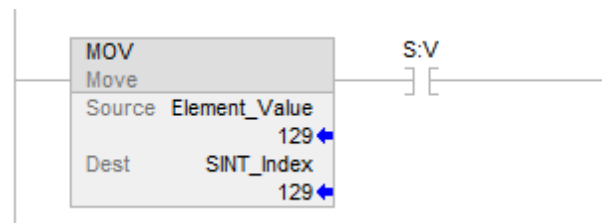
**Diagramme à relais**



Si Limit\_Switch\_1 est vrai, l'instruction suivante est activée.

**Exemple 2**

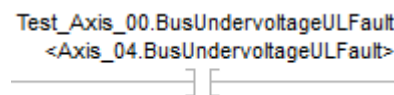
**Diagramme à relais**



Si S:V est vrai (généralisé par MOV), l'instruction suivante est activée.

**Exemple 3**

**Diagramme à relais**



Nombre LINT d'accès XIC

Axis\_04 est une étiquette AXIS\_CIP\_DRIVE.

Test\_Axis\_00 est un alias pour Axis\_04.

Le type AXIS\_CIP\_DRIVE possède un membre LINT appelé CIPAxisFaults.

BusUndervoltageULFault est un membre des bits de CIPAxisFaults.

Test\_Axis\_00.BusUndervoltageULFault est le bit 34 de CIPAxisFaults. Le bit 34 a la valeur 0x40000000.

Si Test\_Axis\_00.BusUndervoltageULFault est vrai, cela active l'instruction suivante.

#### **Voir aussi**

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## **Examiner si ouvert (XIO)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction XIO examine le bit de données pour définir ou mettre la condition d'échelon à zéro.

#### **Langages disponibles**

#### **Diagramme à relais**

data\_bit  
A relay diagram symbol consisting of a horizontal line with a vertical line crossing it, and a diagonal line extending from the intersection to the right.

#### **Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### **Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

### Diagramme à relais

Opérande	Type de données	Format	Description
Bit de données	BOOL	étiquette	Bit à tester. Il existe divers mode d'adressage d'opérande possibles pour le bit de données, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

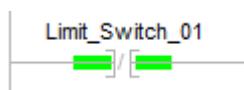
### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Si Bit de données est vrai, la condition de sortie d'échelon est définie sur faux. Si Bit de données est faux, la condition de sortie d'échelon est définie sur vrai.
Post-scrutation	N/A

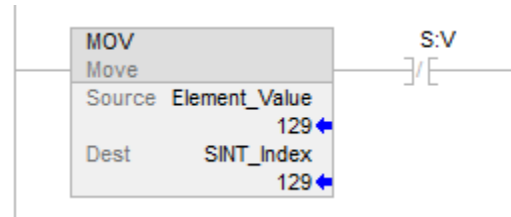
### Exemples

#### Exemple 1

### Diagramme à relais



Si Limit\_Switch\_01 est faux, l'instruction suivante est activée.

**Exemple 2****Diagramme à relais**

Si S:V est faux, cela active l'instruction suivante.

**Voir aussi**

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

**Impulsion (ONS)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction ONS effectue le reste de l'échelon vrai chaque fois que la condition d'entrée d'échelon passe de l'état faux à vrai.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

## Diagramme à relais

Opérande	Type de données	Format	Description
Storage bit	BOOL	étiquette	Bit de stockage interne. Conserve la condition d'entrée d'échelon de la dernière fois que l'instruction a été exécutée. Il existe divers mode d'adressage d'opérande possibles pour le bit de stockage, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

## Affecte les indicateurs d'état mathématique

Non

## Défauts majeurs/mineurs

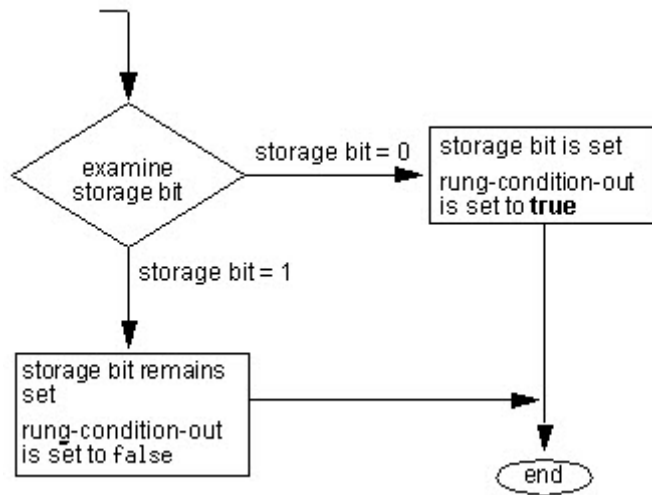
Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

## Exécution

## Diagramme à relais

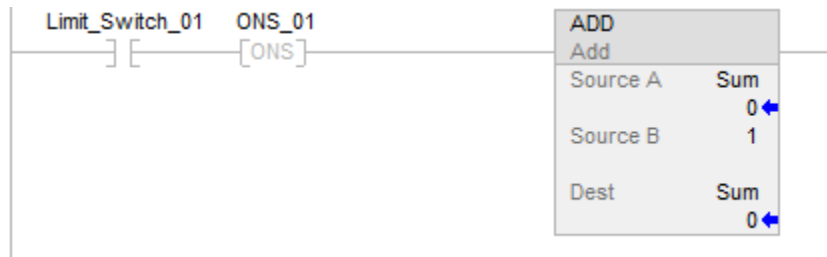
Condition/État	Action entreprise
Pré-scrutation	Le bit de stockage est défini sur vrai pour empêcher un déclenchement non valide pendant la première scrutation.
Condition d'entrée d'échelon est fausse	Le bit de stockage est défini sur faux, la condition de sortie d'échelon est définie sur faux.
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme ONS (vrai).
Post-scrutation	N/A

**Organigramme ONS (Vrai)**



**Exemple**

**Diagramme à relais**



Dans cet exemple, la somme est incrémentée chaque fois que limit\_switch\_1 passe de l'état faux à vrai.

**Voir aussi**

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

**Impulsion sur front descendant (OSF)**

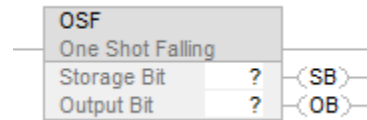
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.



L'instruction OSF définit le bit de sortie pour une scrutation lorsque la condition d'entrée d'échelon passe de l'état vrai à faux.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

#### Diagramme à relais

Opérande	Type de données	Format	Description
Storage Bit	BOOL	étiquette	Stocke la condition d'entrée d'échelon de la dernière exécution de l'instruction. Il existe divers mode d'adressage d'opérande possibles pour le bit de stockage, reportez-vous à la section Adressage de bits pour en voir des exemples.
Output Bit	BOOL	étiquette	Bit à modifier.

#### Affecte les indicateurs d'état mathématique

Non

**Défauts majeurs/mineurs**

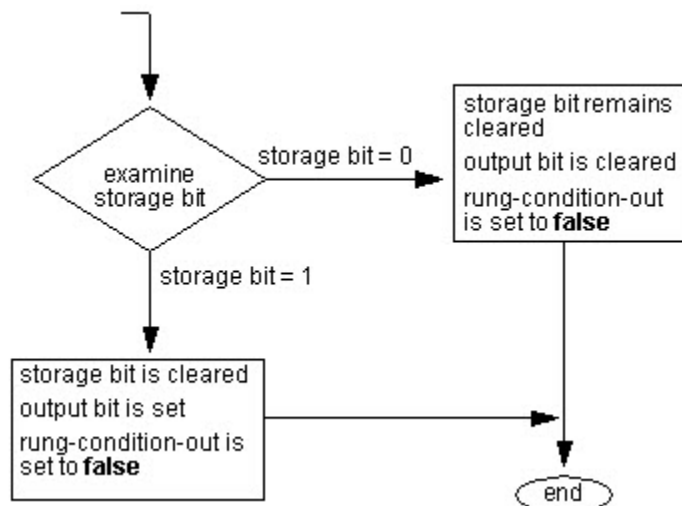
Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit de stockage est défini sur faux pour empêcher un déclenchement non valide pendant la première scrutation du programme. Le bit de sortie est défini sur faux.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Reportez-vous à l'organigramme OSF (faux).
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Le bit de stockage est défini sur vrai. Le bit de sortie est défini sur faux.
Post-scrutation	N/A

**Organigramme OSF (faux)**



## Exemple

### Diagramme à relais



Cet exemple montre comment OSF peut être utilisé pour qu'une ou plusieurs instructions soient déclenchées sur un front. Chaque fois que Limit\_Switch\_01 passe de l'état vrai à faux, OSF définit Output\_bit\_02 sur vrai. Toutes les instructions conditionnées par Output\_bit\_02 seront activées et, comme Output\_bit\_02 n'est à l'état vrai que pour une scrutation, se s'exécuteront qu'une seule fois par transition.

### Voir aussi

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Impulsion sur front descendant avec entrée (OSFI)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction OSFI définit OutputBit pour un cycle d'exécution lorsqu'InputBit passe de l'état faux à vrai.

### Langages disponibles

### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

**Bloc fonctionnel**



**Texte structuré**

OSFI(OSFI\_tag)

**Opérandes**

**Texte structuré**

Opérande	Type	Format	Description
OSFI tag	FBD_ONESHOT	Structure	Structure OSFI

Reportez-vous à la section *Syntaxe du texte structuré* pour plus d'informations sur les défauts liés aux opérandes

**Bloc fonctionnel**

Opérande	Type	Format	Description
OSFI tag	FBD_ONESHOT	Structure	Structure OSFI

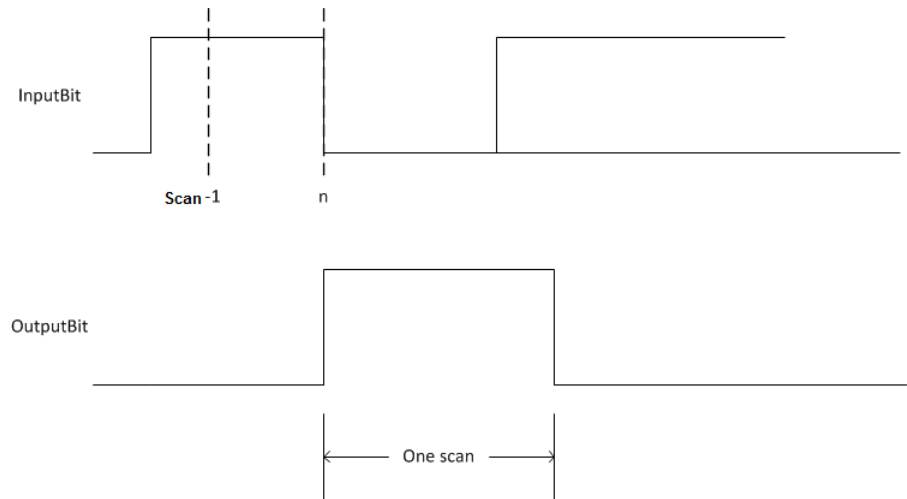
**Structure FBD\_ONESHOT**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
InputBit	BOOL	Bit d'entrée.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
OutputBit	BOOL	Bit de sortie

### Description

Si InputBit est faux et avait la valeur vraie la dernière fois que l'instruction a été scrutée, OutputBit est défini, sinon OutputBit est mis à zéro.



### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	L'historique d'InputBit précédent est mis à zéro pour demander une transition de l'état vrai à faux d'InputBit.
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

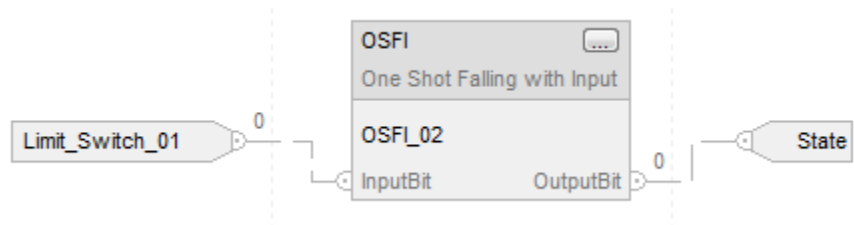
**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemple**

Lorsque limit\_switch1 passe de l'état défini à mis à zéro, l'instruction OSFI définit OutputBit pour une scrutation.

**Bloc fonctionnel**



**Texte structuré**

```
OSFI_01.InputBit := limit_switch1;
OSFI(OSFI_01);
Output_state := OSFI_01.OutputBit;
```

**Voir aussi**

[Instructions de bit](#) sur la [page 73](#)

[OSF](#) sur la [page 80](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

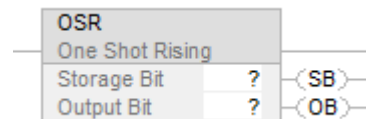
**Impulsion sur front montant (OSR)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction OSR définit le bit de sortie pour une scrutation lorsque la condition d'entrée d'échelon passe de l'état faux à vrai.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

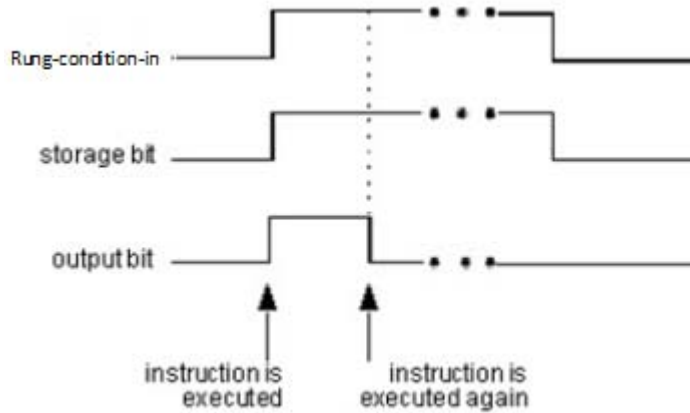
- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

### Diagramme à relais

Opérande	Type de données	Format	Description
Storage Bit	BOOL	étiquette	Stocke la condition d'entrée d'échelon de la dernière exécution de l'instruction. Il existe divers mode d'adressage d'opérande possibles pour le bit de stockage, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.
Output Bit	BOOL	étiquette	Bit à modifier.

**Description**



**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

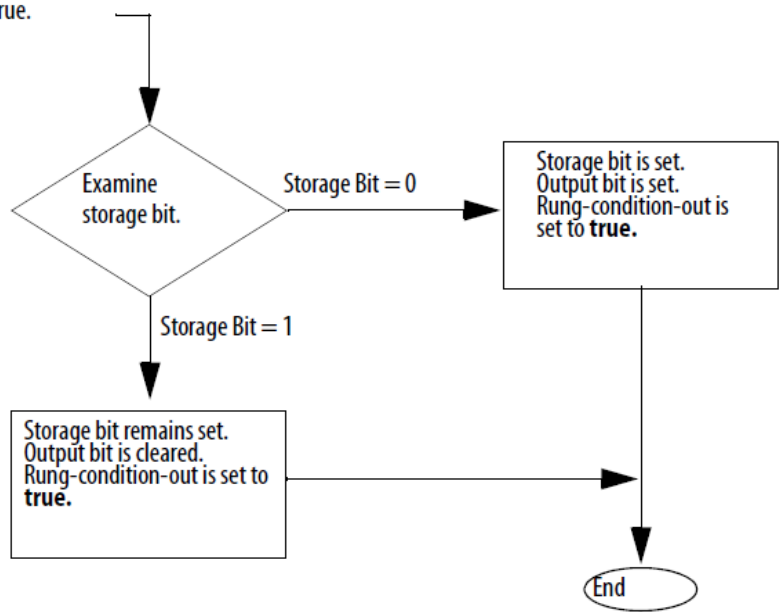
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit de stockage est défini sur vrai pour empêcher un déclenchement non valide pendant la première scrutation du programme. Le bit de sortie est défini sur faux.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Le bit de stockage est défini sur faux. Le bit de sortie est défini sur faux.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme OSR (vrai).
Post-scrutation	N/A



**Organigramme OSR (vrai)**

Rung-condition-in is true.



**Exemple**

**Diagramme à relais**



Cet exemple montre comment OSR peut être utilisé pour qu'une ou plusieurs instructions soient déclenchées sur un front. Chaque fois que Limit\_Switch\_01 passe de l'état faux à vrai, OSR définit Output\_bit\_02 sur vrai. Toutes les instructions conditionnées par Output\_bit\_02 seront activées et, comme Output\_bit\_02 n'est à l'état vrai que pour une scrutation, se s'exécuteront qu'une seule fois par transition.

**Voir aussi**

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Impulsion sur front montant avec entrée (OSRI)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

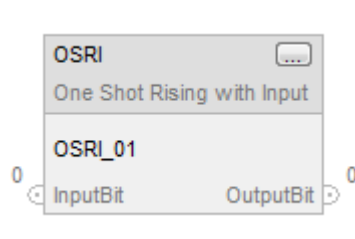
L'instruction OSRI définit le bit de sortie pour un cycle d'exécution lorsque bit d'entrée passe de zéro à défini.

### Langages disponibles

### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

### Bloc fonctionnel



### Texte structuré

OSRI(OSRI\_tag);

### Opérandes

### Texte structuré

Opérande	Type	Format	Description
OSRI tag	FBD_ONESHOT	Structure	Structure OSRI

### Bloc fonctionnel

Opérande	Type	Format	Description
OSRI tag	FBD_ONESHOT	Structure	Structure OSRI

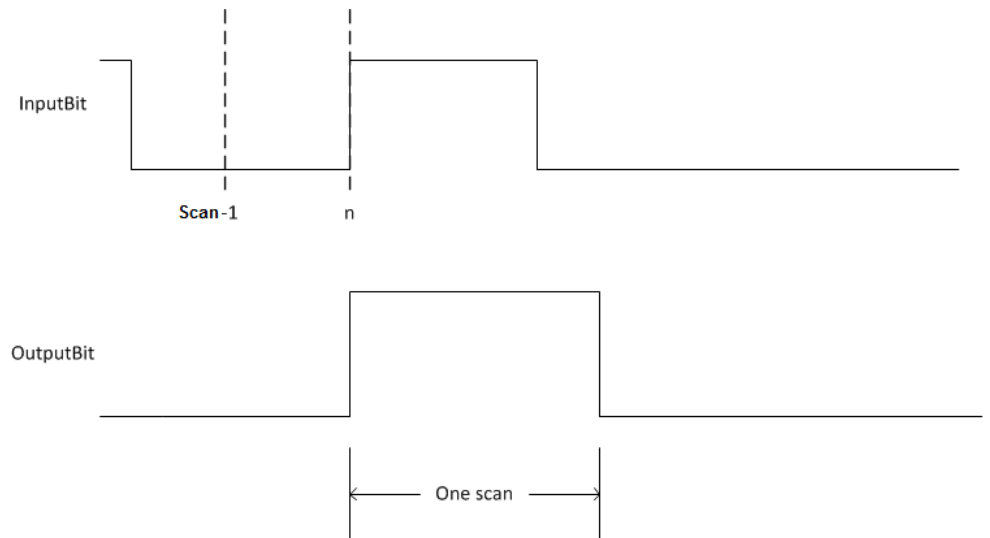
**Structure FBD\_ONESHOT**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
InputBit	BOOL	Bit d'entrée. La valeur par défaut est mise à zéro.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
OutputBit	BOOL	Bit de sortie

**Description**

Si InputBit est vrai et avait faux la dernière fois que l'instruction a été scrutée, OutputBit est défini, sinon OutputBit est mis à zéro.



**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Bloc fonctionnel**

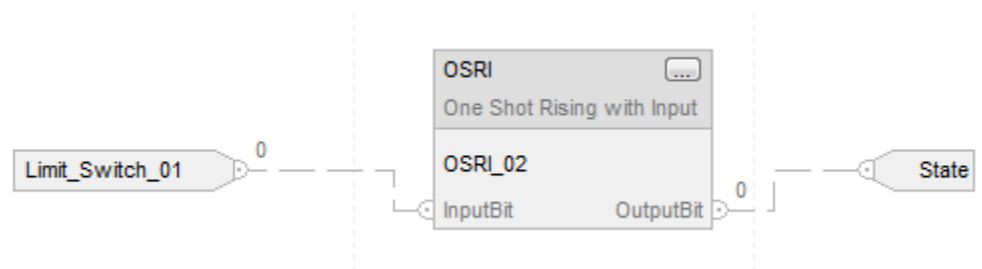
Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.Enable-in est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.Enable-in est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	L'historique d'InputBit précédent est défini pour demander une transition de l'état faux à vrai d'InputBit.
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel

**Exemples**

**Bloc fonctionnel**



Lorsque limit\_switch1 passe de zéro à défini, l'instruction OSRI définit OutputBit pour une scrutation.

#### Texte structuré

```
OSRI_01.InputBit := limit_switch1;
```

```
OSRI(OSRI_01);
```

```
State := OSRI_O1.OutputBit;
```

#### Voir aussi

[Instructions de bit](#) sur la [page 73](#)

[Impulsion sur front descendant \(OSF\)](#) sur la [page 80](#)

[Impulsion \(ONS\)](#) sur la [page 78](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)


## Activation de sortie (OTE)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction OTE définit ou met le bit de données à zéro en fonction de la condition d'échelon.

#### Langages disponibles

#### Diagramme à relais

data\_bit  


#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

### Diagramme à relais

Opérande	Type de données	Format	Description
Bit de données	BOOL	étiquette	Bit à modifier. Il existe divers mode d'adressage d'opérande possibles pour le bit de données, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

### Affecte les indicateurs d'état mathématique

Non

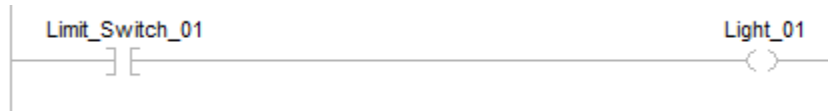
### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Le bit de données est défini sur faux
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Le bit de données est défini sur faux
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Le bit de données est défini sur vrai.
Post-scrutation	Le bit de données est défini sur faux.

**Exemple****Diagramme à relais**

Lorsque l'interrupteur est vrai, l'instruction OTE définit Light\_01 sur vrai.  
Lorsque l'interrupteur est faux, l'instruction OTE définit Light\_01 sur faux.

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

**Verrouillage de sortie (OTL)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction OTL définit (verrouille) le bit de données.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

## Diagramme à relais

Opérande	Type de donnée	Format	Description (Description)
Bit de données	BOOL	étiquette	Bit à modifier. Il existe divers mode d'adressage d'opérande possibles pour le bit de données, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

## Description (Description)

Lorsque la condition d'échelon est vraie, l'instruction OTL définit le bit de données sur vrai. Le bit de données reste à l'état vrai jusqu'à ce qu'il soit mis à zéro, généralement par l'instruction OTU. Lorsque la condition d'échelon passe à l'état faux, l'instruction OTL ne modifie pas l'état du bit de données.

## Affecter les indicateurs d'état mathématique

Non

## Défauts majeurs/mineurs

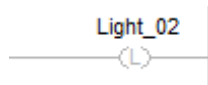
Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

Pour Controllers Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580, si l'opérande se trouve dans une référence indirecte de tableau et que l'indice est hors plage, l'automate ne génère pas de défaut majeur lorsque l'instruction OTL est fautive.



**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Le bit de données est défini sur vrai.
Post-scrutation	N/A

**Exemple****Diagramme à relais**

Lorsqu'elle est activée, l'instruction OTL allume la lumière.

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)

**Déverrouillage de sortie (OTU)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction OTU met (verrouille) le bit de données à zéro.

**Langages disponibles****Diagramme à relais**

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

### Diagramme à relais

Opérande	Type de données	Format	Description
Bit de données	BOOL	étiquette	Bit à modifier. Il existe divers mode d'adressage d'opérande possibles pour le bit de données, reportez-vous à la section <i>Adressage de bits</i> pour en voir des exemples.

### Description

Lorsque la condition d'échelon est vraie, l'instruction OTU définit le bit de données sur faux. Lorsque la condition d'échelon passe à l'état faux, l'instruction OTU ne modifie pas l'état du bit de données.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Le bit de données est défini sur faux.
Post-scrutation	N/A

## Exemple

### Diagramme à relais



Lorsqu'elle est activée, l'instruction OTU met Light\_02 à zéro.

### Voir aussi

[Instructions de bit](#) sur la [page 73](#)

[Adressage de bits](#) sur la [page 900](#)

[Indexer via des tableaux](#) sur la [page 899](#)



# Instructions de temporisateur et de compteur

## Instructions de temporisateur et de compteur

Les temporisateurs et les compteurs contrôlent les opérations en fonction du temps ou du nombre d'événements.

### Instructions disponibles

#### Diagramme à relais

<a href="#">TON</a>	<a href="#">TOF</a>	<a href="#">RTO</a>	<a href="#">CTU</a>	<a href="#">CTD</a>	<a href="#">RES</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

#### Bloc fonctionnel et Texte structuré

<a href="#">TONR</a>	<a href="#">TOFR</a>	<a href="#">RTOR</a>	<a href="#">CTUD</a>
----------------------	----------------------	----------------------	----------------------

Si vous voulez	Utiliser cette instruction
chronométrer combien de temps un temporisateur est activé	TON
chronométrer combien de temps un temporisateur est désactivé	TOF
accumuler le temps	RTO
chronométrer combien de temps un temporisateur est activé avec la réinitialisation intégrée dans le bloc fonctionnel	TONR
chronométrer combien de temps un temporisateur est désactivé avec la réinitialisation intégrée dans le bloc fonctionnel	TOFR
accumuler le temps avec la réinitialisation intégrée dans le bloc fonctionnel	RTOR
comptage progressif	CTU
comptage dégressif	CTD

effectuer un comptage progressif et un comptage dégressif dans le bloc fonctionnel	CTUD
réinitialiser un temporisateur ou un compteur	RES

La base de temps est de 1 ms pour tous les temporisateurs. Par exemple, la valeur .PRE d'un temporisateur de 2 secondes doit être de 2 000.

**Voir aussi**

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

**Comptage dégressif (CTD)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction CTD effectue un comptage vers le bas chaque fois que la condition d'entrée d'échelon passe de l'état faux à vrai.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

## Diagramme à relais

Opérande	Type de données	Format	Description
Counter	COUNTER	étiquette	Structure du compteur
Preset	DINT	immédiate	Valeur de Counter.PRE.
Accum	DINT	immédiate	Valeur de Counter.ACC.

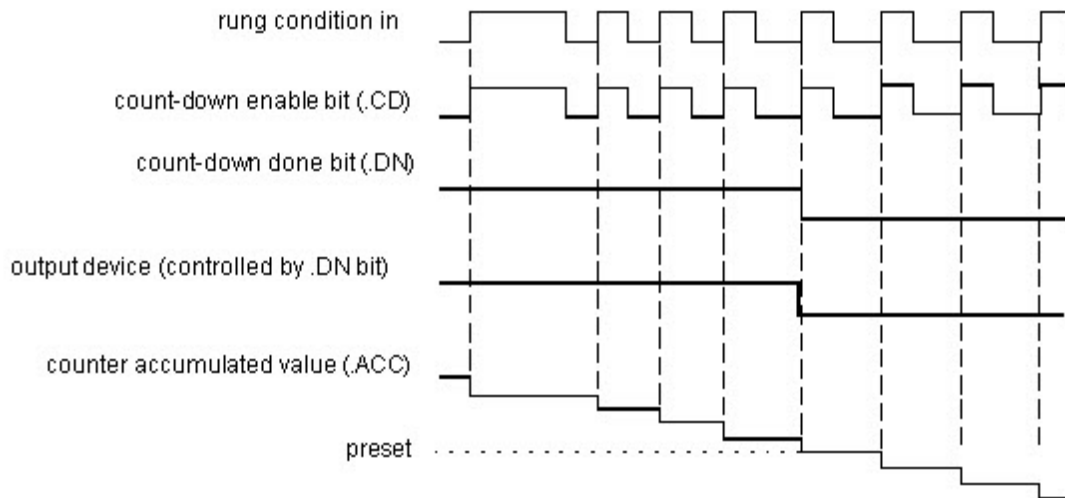
## Structure COUNTER

Mnémonique	Type de données	Description
.CD	BOOL	Le bit d'activation du comptage dégressif contient la condition d'entrée d'échelon de la dernière exécution de l'instruction.
.DN	BOOL	Le bit de fin, lorsqu'il est mis à zéro, indique que l'opération de comptage est terminée.
.OV	BOOL	Le bit de débordement, lorsqu'il est défini, indique que le compteur a incrémenté au-delà de la limite supérieure de 2 147 483 647.
.UN	BOOL	Le dépassement inférieur, lorsqu'il est défini, indique que le compteur a décrémenté au-delà de la limite inférieure de -2 147 483 648.
.PRE	DINT	La valeur prédéfinie spécifie la valeur accumulée qui doit être atteinte avant que l'instruction indique qu'elle est terminée.
.ACC	DINT	La valeur accumulée spécifie le nombre de transitions que l'instruction a compté.

## Description

L'instruction CTD est généralement utilisée avec une instruction CTU qui fait référence à la même structure du compteur.

Lorsque la condition d'entrée d'échelon est définie sur vrai et que .CD est faux, .ACC sera décrémenté d'une unité. Lorsque la condition d'entrée d'échelon est fausse, .CD sera défini sur faux.



**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

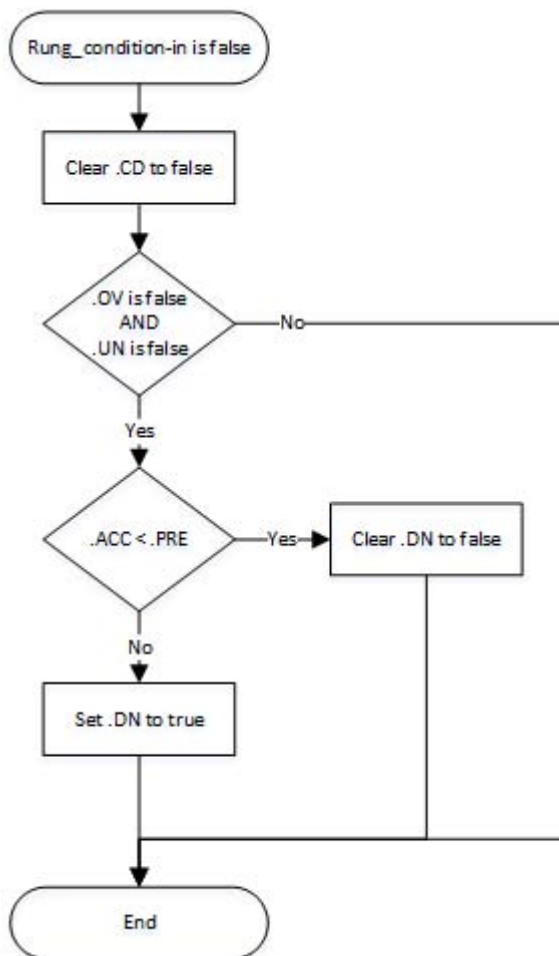
**Exécution**

**Diagramme à relais**

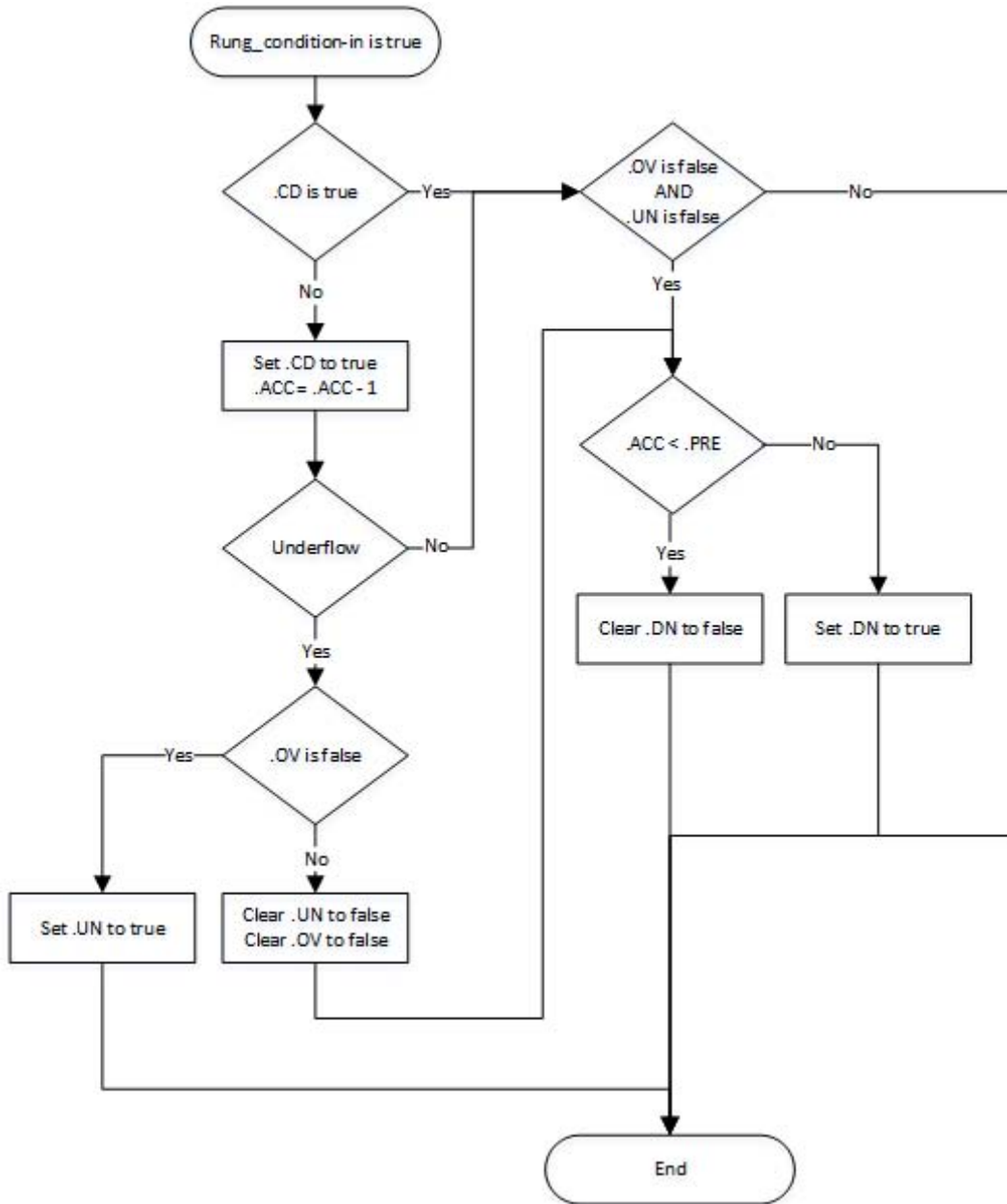
Condition/État	Action entreprise
Pré-scrutation	Le bit .CD est défini sur vrai pour empêcher des décréments non valides pendant la première scrutation du programme.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme CTD (faux)
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme CTD (vrai)
Post-scrutation	N/A



Organigramme CTD (faux)

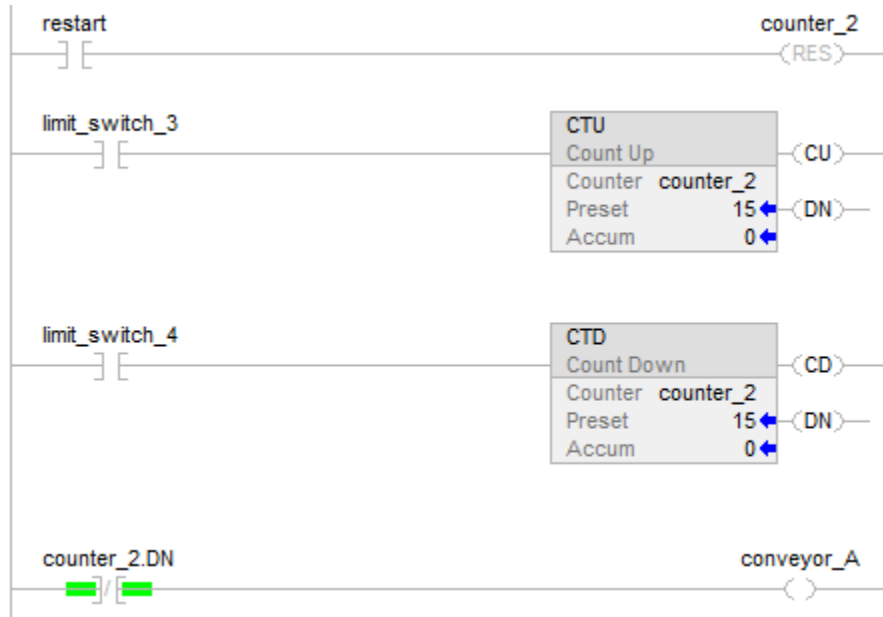


Organigramme CTD (vrai)



## Exemple

## Diagramme à relais



Un convoyeur amène des parties dans une zone tampon. Chaque fois qu'une partie entre, limit\_switch\_3 est activé et counter\_2 est incrémenté de 1. Chaque fois qu'une partie quitte, limit\_switch\_4 est activé et counter\_2 est décrémenté de 1. S'il y a 100 parties dans la zone tampon (counter\_2.dn est vrai), conveyor\_A s'active et indique au convoyeur d'arrêter d'apporter d'autres pièces tant que le tampon ne dispose plus de place pour plus de parties.

## Voir aussi

[Indexer via des tableaux](#) sur la [page 899](#)

[Instructions de compteur](#) sur la [page 101](#)

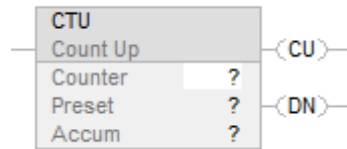
## Comptage progressif (CTU)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction CTU effectue un comptage vers le haut chaque fois que la condition d'entrée d'échelon passe de l'état faux à vrai.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

- 
- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.
- 

#### Diagramme à relais

Opérande	Type de données	Format	Description
Counter	COUNTER	étiquette	Structure du compteur
Preset	DINT	immédiate	Valeur de Counter.PRE.
Accum	DINT	immédiate	Valeur de Counter.ACC.

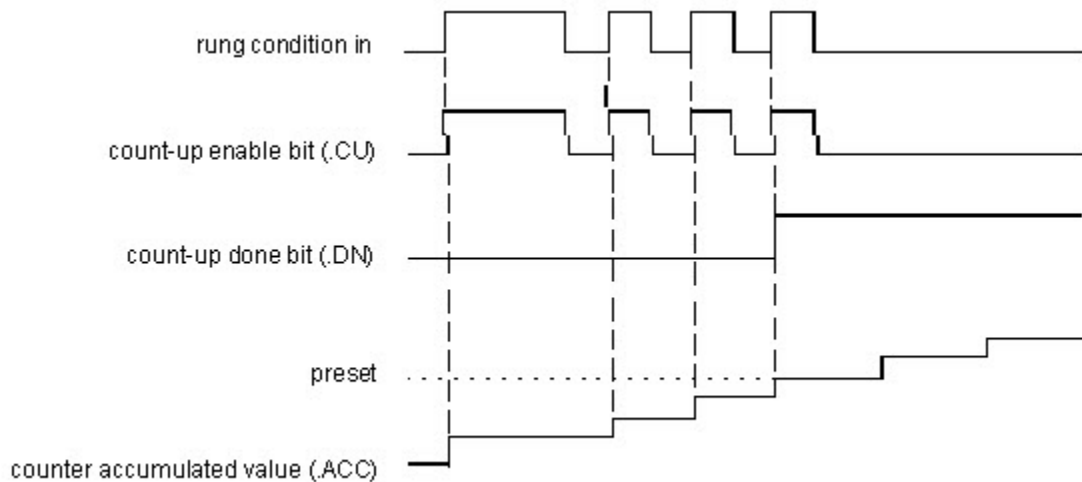
#### Structure COUNTER

Mnémonique	Type de données	Description
.CU	BOOL	L'activation du comptage progressif contient la condition d'entrée d'échelon de la dernière exécution de l'instruction.
.DN	BOOL	Le bit de fin, lorsqu'il est défini, indique que l'opération de comptage est terminée.

.OV	BOOL	Le bit de débordement, lorsqu'il est défini, indique que le compteur a incrémenté au-delà de la limite supérieure de 2 147 483 647.
.UN	BOOL	Le dépassement inférieur, lorsqu'il est défini, indique que le compteur a décrémenté au-delà de la limite inférieure de -2 147 483 648.
.PRE	DINT	La valeur prédéfinie spécifie la valeur accumulée qui doit être atteinte avant que l'instruction indique qu'elle est terminée.
.ACC	DINT	La valeur accumulée spécifie le nombre de transitions que l'instruction a compté.

### Description

Lorsque la condition d'entrée d'échelon est définie sur vrai et que .CU est faux, .ACC sera décrémenté d'une unité. Lorsque la condition d'entrée d'échelon est fausse, .CU sera défini sur faux.



### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

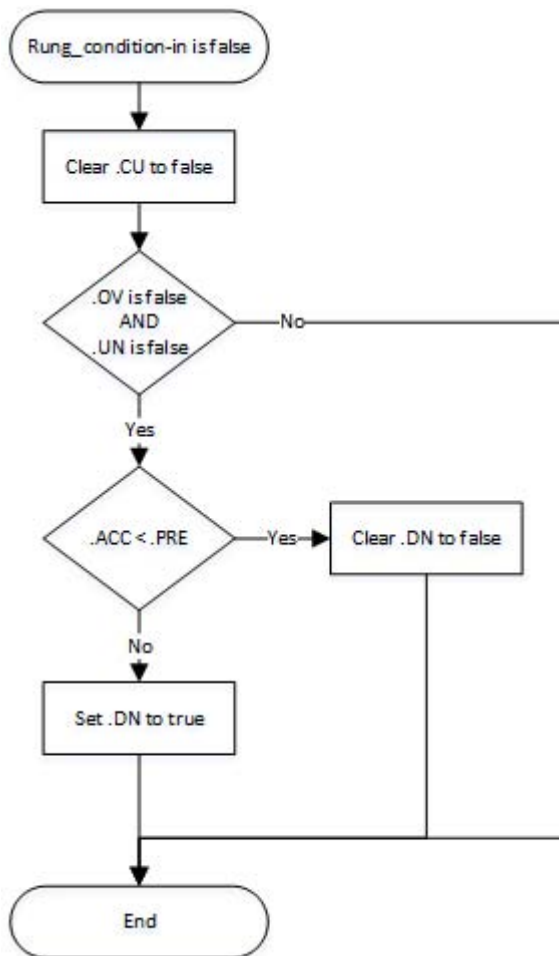
Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution**

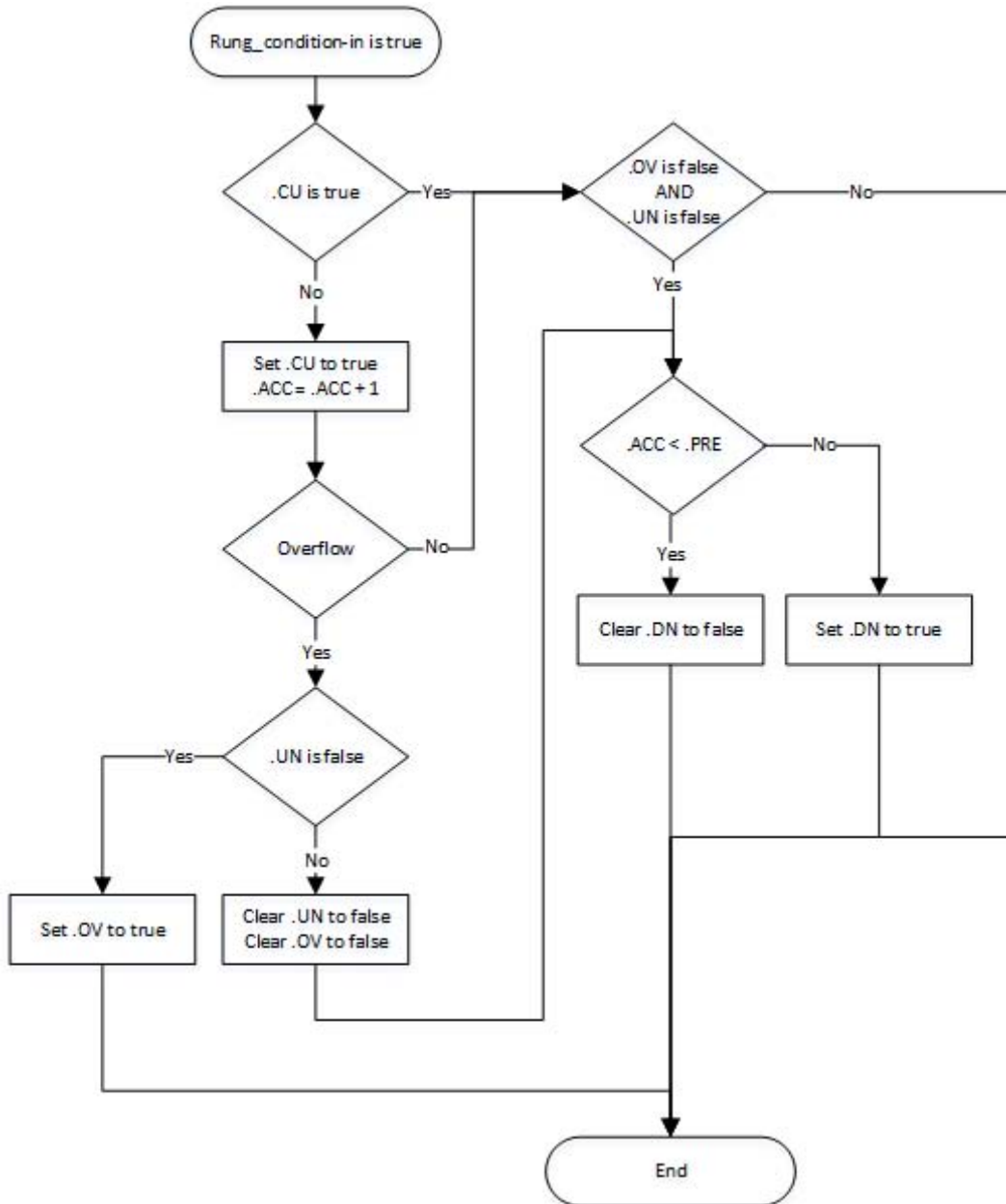
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit .CU est défini sur vrai pour empêcher des incréments non valides pendant la première scrutation du programme.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme CTU (faux)
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme CTU (vrai)
Post-scrutation	N/A

**Organigramme CTU (faux)**

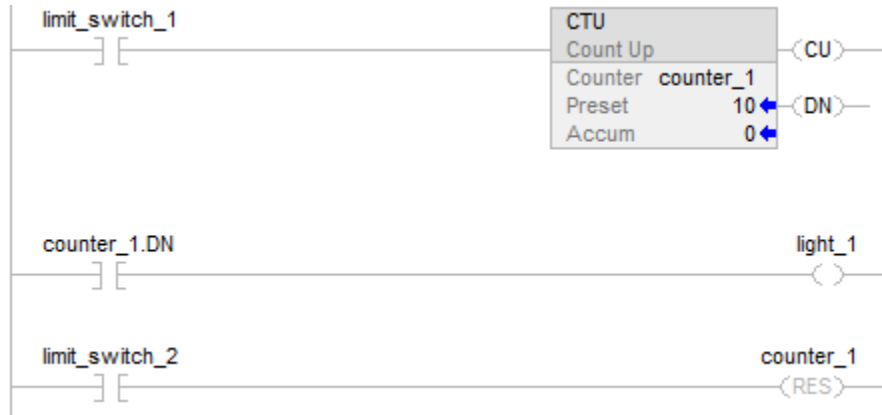


Organigramme CTU (vrai)



**Exemple**

**Diagramme à relais**



Lorsque limit\_switch\_1 passe 10 fois de l'état désactivé à l'état actif, le bit .DN est défini sur vrai light\_1 est activé. Si limit\_switch\_1 continue à passer l'état désactivé à l'état actif, counter\_1 continue à incrémenter son comptage et le bit .DN reste défini. Lorsque limit\_switch\_2 est activé, l'instruction RES réinitialise counter\_1 (met les bits d'état et la valeur .ACC à zéro ) et light\_1 est désactivé.

**Voir aussi**

[Indexer via des tableaux](#) sur la [page 899](#)

[Instructions de compteur](#) sur la [page 101](#)

**Comptage progressif/dégressif (CTUD)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction CTUD effectue un comptage progressif par incrément d'un lorsque CUEnable passe de zéro à défini. L'instruction effectue un comptage dégressif par incrément d'un lorsque CDEnable passe de zéro à défini.

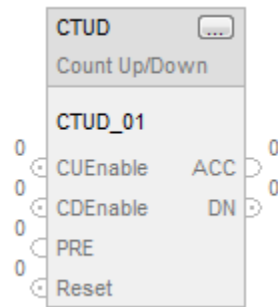
**Langages disponibles**

**Diagramme à relais**

Cette instruction n'est pas disponible dans un diagramme à relais.



**Bloc fonctionnel**



**Texte structuré**

CTUD(CTUD\_tag)

**Opérandes**

**Texte structuré**

Variable	Type	Format	Description
CTUD tag	FBD_COUNTER	Structure	Structure CTUD

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
CTUD tag	FBD_COUNTER	Structure	Structure CTUD

**Structure FBD\_COUNTER**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
CUEnable	BOOL	Permet d'activer le comptage progressif. Lorsque l'entrée passe de zéro à défini, le totalisateur effectue un comptage progressif par incrément d'un. La valeur par défaut est mise à zéro.

CDEnable	BOOL	Permet d'activer le comptage dégressif. Lorsque l'entrée passe de zéro à défini, le totalisateur effectue un comptage dégressif par incrément d'un. La valeur par défaut est mise à zéro.
PRE	DINT	Valeur prédéfinie du compteur. Correspond à la valeur que la valeur accumulée doit atteindre avant que DN soit défini. Valide = tout entier La valeur par défaut est 0.
Reset	BOOL	Demande de réinitialisation du temporisateur. Si définir, le compteur se réinitialise. La valeur par défaut est mise à zéro.

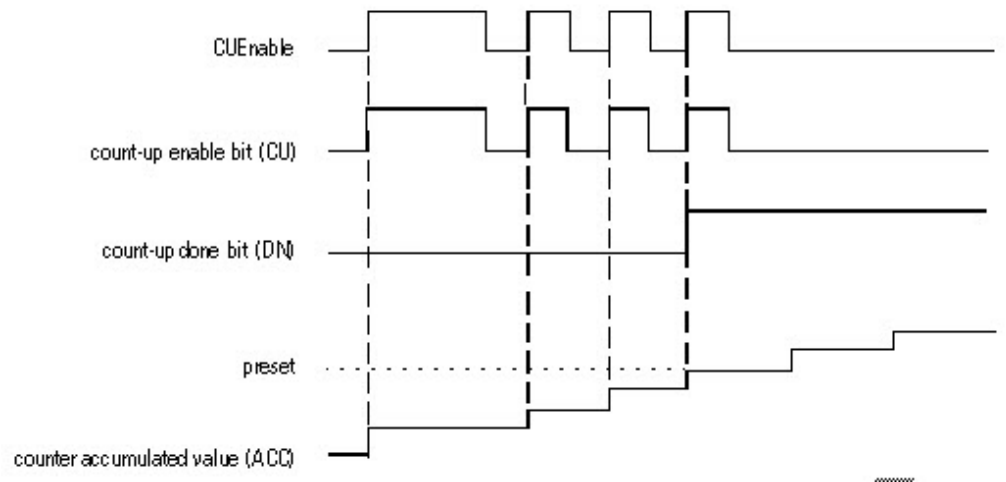
Paramètre de sortie	Type de données	Description
EnableOut	BOOL	L'instruction a produit un résultat valide.
ACC	DINT	Valeur accumulée.
CU	BOOL	Comptage progressif activé.
CD	BOOL	Comptage dégressif activé.
DN	BOOL	Comptage terminé. Définir lorsque la valeur accumulée est supérieure ou égale à la valeur prédéfinie.
OV	BOOL	Débordement du compteur. Indique que le compteur a dépassé la limite supérieure de 2 147 483 647. Le compteur repasse alors à -2 147 483 648 et recommence à effectuer un comptage dégressif.
UN	BOOL	Dépassement inférieur du compteur. Indique que le compteur a dépassé la limite inférieure de -2 147 483 648. Le compteur repasse alors à 2 147 483 647 et recommence à effectuer un comptage dégressif.

### Description

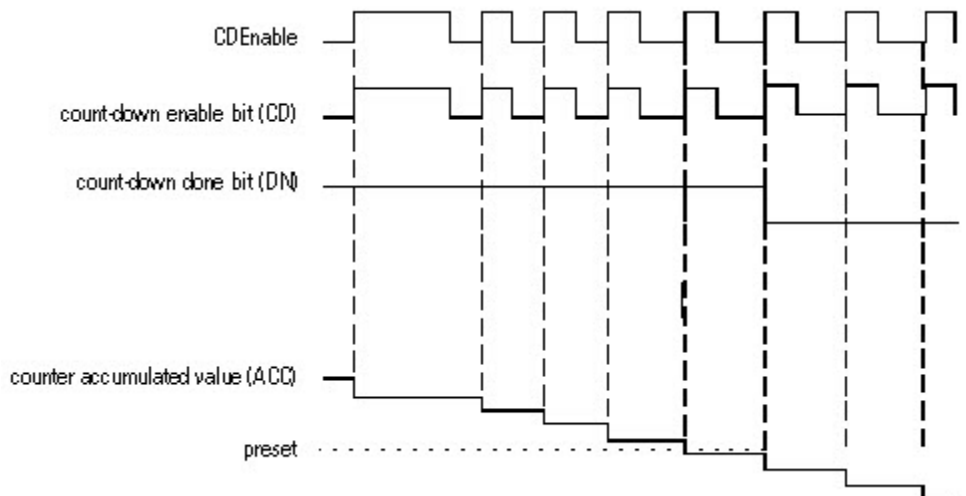
Si définir sur vrai et que CUEnable est vrai, l'instruction CTUD incrémente le compteur d'un. Si définir sur vrai et que CDEnable est vrai, l'instruction CTUD décrémente le compteur d'un.

Les paramètres d'entrée de CUEnable et de CDEnable peuvent être basculés pendant la même scrutation. L'instruction exécute le comptage progressif avant le comptage dégressif.

### Comptage progressif



### Comptage dégressif



Si désactivé, l'instruction CTUD conserve sa valeur accumulée. Définir le paramètre d'entrée Reset pour réinitialiser l'instruction.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Bloc fonctionnel**

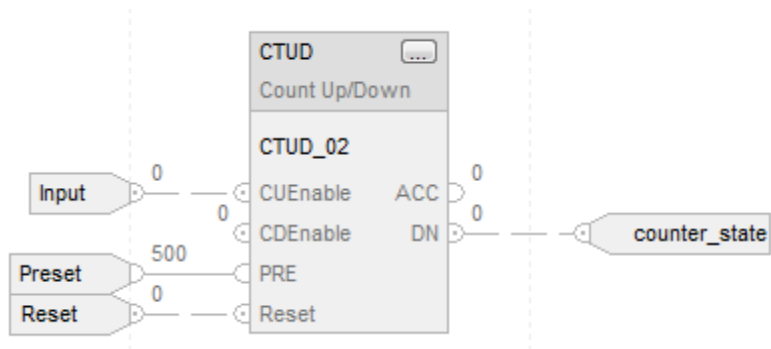
Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux. Initialisez les données pour demander une transition « de zéro à un » de CuEnable ou CdEnable pour réaliser ACC.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute.
Première exécution de l'instruction	Initialisez les données pour demander une transition « de zéro à un » de CuEnable ou CdEnable pour réaliser ACC.
Première scrutation de l'instruction	Initialisez les données pour demander une transition « de zéro à un » de CuEnable ou CdEnable pour réaliser ACC.
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemple**

**Bloc fonctionnel**



**Texte structuré**

```
CTUD_01.PRE := 500;

CTUD_01.Reset := Reset;

CTUD_01.CUEnable := Input;

CTUD(CTUD_01);

counter_state := CTUD_01.DN;
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Comptage progressif \(CTU\)](#) sur la [page 107](#)

[Comptage dégressif \(CTD\)](#) sur la [page 102](#)

[Réinitialisation \(RES\)](#) sur la [page 117](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

**Réinitialisation (RES)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction RES réinitialise une structure TIMER, COUNTER ou CONTROL.

**Langages disponibles****Diagramme à relais**

-(RES)-

**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

### Diagramme à relais

Opérande	Type de données	Format	Description
Structure	TIMER CONTROL COUNTER	Étiquette	Structure à réinitialiser

### Description

Si définir sur vrai, l'instruction RES met trois éléments à zéro :

Lors de l'utilisation d'une instruction RES pour un :	L'instruction met valeurs suivantes à zéro :
TIMER	Valeur .ACC sur 0 bit d'état du contrôle sur faux
COUNTER	Valeur .ACC sur 0 bit d'état du contrôle sur faux
CONTROL	Valeur .POS sur 0 bit d'état du contrôle sur faux

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

### Exécution

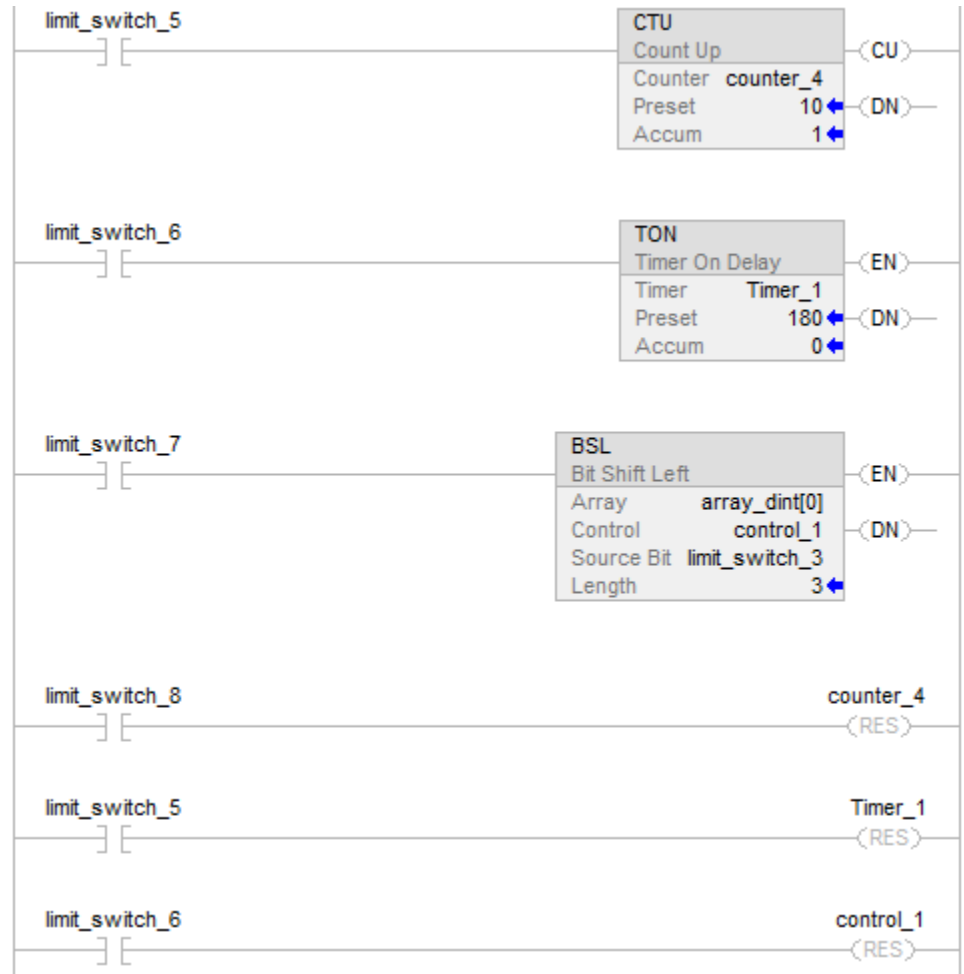
### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.

Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Réinitialiser la structure spécifiée.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Exemple de réinitialisation**

Dans l'exemple précédent :

lorsque `limit_switch_8` est activé, réinitialiser `counter_4`

lorsque `limit_switch_5` est activé, réinitialiser `Timer_1`

lorsque `limit_switch_6` est activé, réinitialiser `control_1`

**Voir aussi**

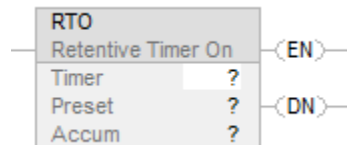
[Instruction de compteur](#) sur la [page 101](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Temporisateur de rémanence activé (RTO)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction RTO est un temporisateur rémanent qui accumule le temps lorsque l'instruction est activée.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**


---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---



**Diagramme à relais**

Opérande	Type de données	Format	Description
Timer	TIMER	étiquette	Structure de temporisateur
Preset	DINT	immédiate	Valeur de Timer.PRE.
Accum	DINT	immédiate	Valeur de Timer.ACC.

**Structure TIMER**

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation contient la condition d'entrée d'échelon de la dernière exécution de l'instruction.
.TT	BOOL	Le bit de temporisation, lorsqu'il est défini, indique que l'opération de temporisation est en cours.
.DN	BOOL	Le bit de fin, lorsqu'il est défini, indique que l'opération de temporisation est terminée (ou mise en pause).
.PRE	DINT	La valeur prédéfinie spécifie la valeur accumulée (à raison d'une milliseconde par incrément) qui doit être atteinte avant que l'instruction indique qu'elle est terminée.
.ACC	DINT	La valeur accumulée spécifie le nombre de millisecondes qui se sont écoulées depuis l'activation de l'instruction RTO.

**Description**

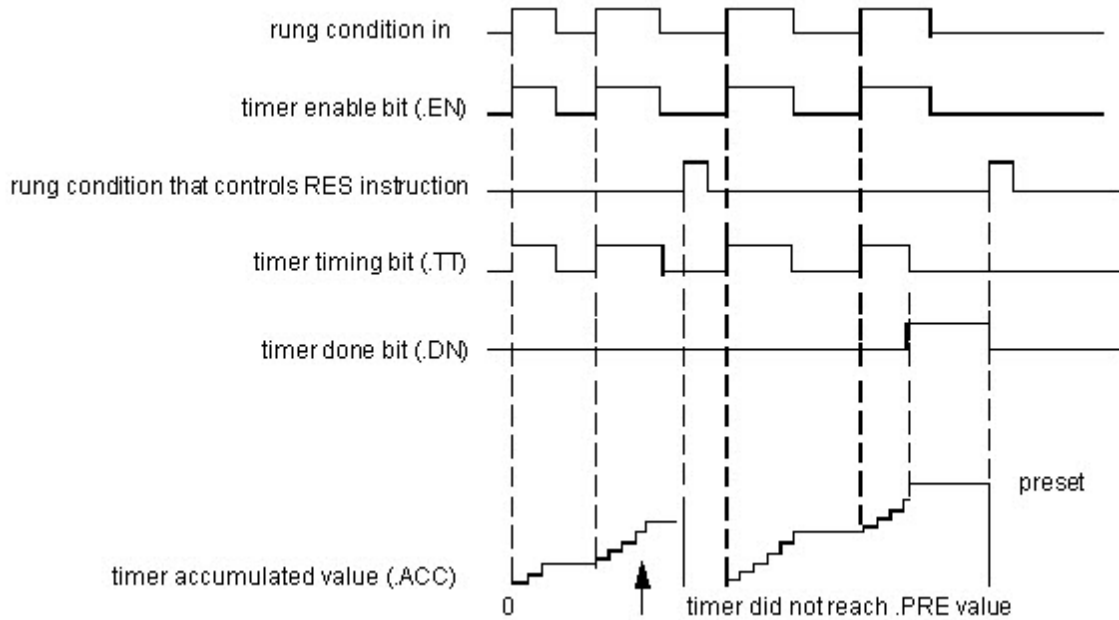
L'instruction RTO accumule le temps jusqu'à la situation suivante :

- Le temporisateur est désactivé.
- Le temporisateur se termine.

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur .PRE.

Le temporisateur définira le bit .DN sur vrai lorsque le temporisateur se termine.

Lorsqu'elle est activée, la temporisation peut être mise en pause en définissant le bit .DN sur vrai, puis reprise en définissant le bit .DN sur faux.



**Fonctionnement d'un temporisateur**

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

$$ACC = ACC + (current\_time - last\_time\_scanned)$$

Après avoir mis à jour ACC, le temporisateur définit last\_time\_scanned = current\_time. Cela rend le temporisateur disponible pour la scrutation suivante.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

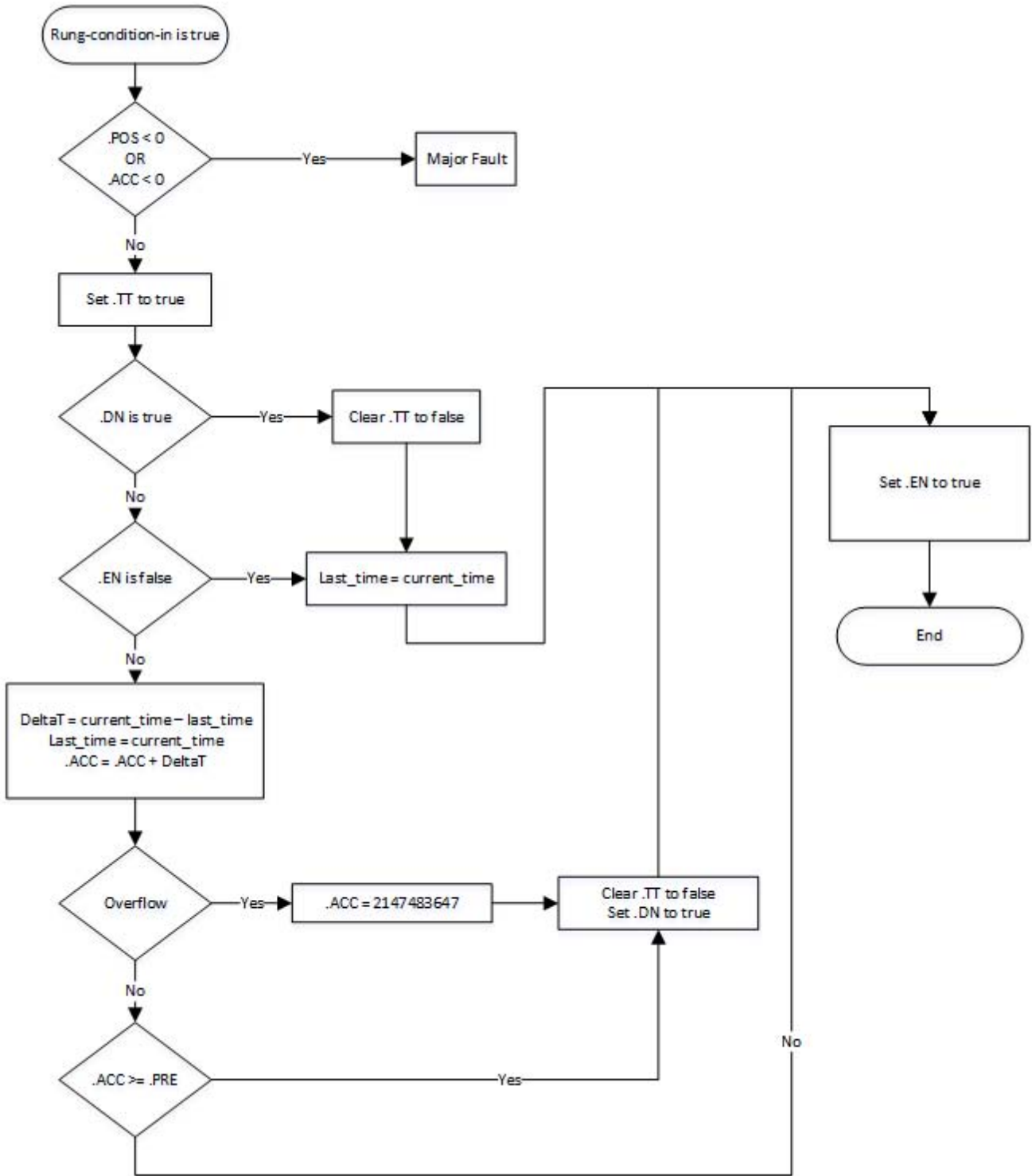
Un défaut majeur se produira si :	Type de défaut	Code de défaut
.PRE < 0	4	34
.ACC < 0	4	34

Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

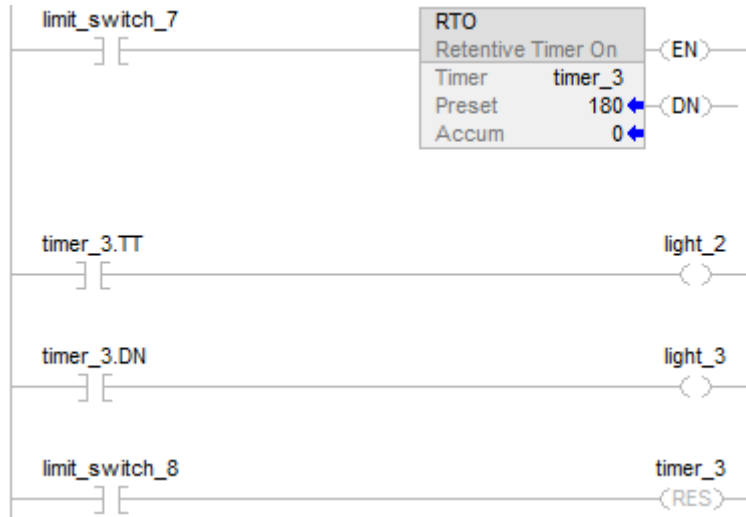
<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Le bit .EN est défini sur faux. Le bit .TT est défini sur faux.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Le bit .EN est défini sur faux. Le bit .TT est défini sur faux.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme RTO (vrai).
Post-scrutation	N/A

Organigramme RTO (vrai)



## Exemple

## Diagramme à relais



Lorsque `limit_switch_7` est défini, `light_2` est activé pendant 180 millisecondes (`timer_3` fonctionne). Lorsque `timer_3.acc` atteint 180, `light_2` est désactivé et `light_3` est activé. `Light_3` reste activé jusqu'à la réinitialisation de `timer_3`. Si `limit_switch_7` est mis à zéro pendant le fonctionnement de `timer_3`, `light_2` est désactivé. Lorsque `limit_switch_7` est défini, l'instruction RES réinitialise `timer_3` (met les bits d'état et la valeur .ACC à zéro).

## Voir aussi

[Indexer via des tableaux](#) sur la [page 899](#)

[Réinitialisation \(RES\)](#) sur la [page 117](#)

[Temporisateur au déclenchement \(TOF\)](#) sur la [page 130](#)

[Temporisateur à l'enclenchement \(TON\)](#) sur la [page 140](#)

[Instructions de temporisateur et de compteur](#) sur la [page 101](#)

## Temporisateur de rémanence activé avec réinitialisation (RTOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

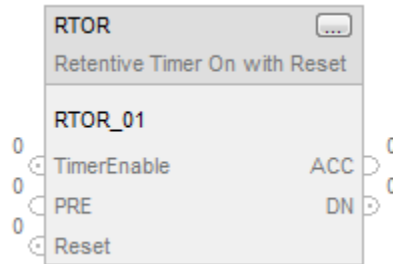
L'instruction RTOR est un temporisateur rémanent qui accumule le temps lorsque `TimerEnable` est défini.

### Langages disponibles

#### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

#### Bloc fonctionnel



#### Texte structuré

RTOR(RTOR\_tag)

#### Opérandes

#### Texte structuré

Variable	Type	Format	Description
RTOR tag	FBD_TIMER	Structure	Structure RTOR

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

#### Bloc fonctionnel

Opérande	Type	Format	Description
RTOR tag	FBD_TIMER	Structure	Structure RTOR

#### Structure FBD\_TIMER

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
TimerEnable	BOOL	Si définir, cela active le fonctionnement du temporisateur et accumule le temps. La valeur par défaut est mise à zéro.

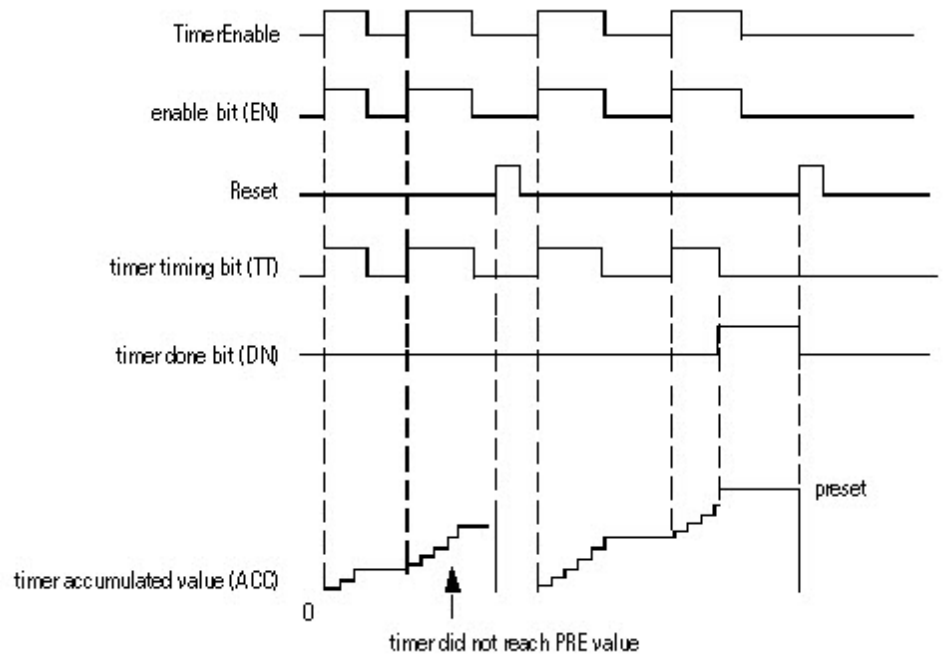
PRE	DINT	Valeur prédéfinie du temporisateur. Correspond à la valeur par incrément de 1 ms qu'ACC doit atteindre avant la fin de la temporisation. Si non valide, l'instruction définit le bit approprié dans Status et le temporisateur ne s'exécute pas. Valide = 0 à l'entier positif maximal
Reset	BOOL	Demande de réinitialisation du temporisateur. Si défini, le temporisateur se réinitialise. Lorsque le paramètre d'entrée Reset est défini, l'instruction met EN, TT et DN à zéro, et définit ACC = 0.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	L'instruction a produit un résultat valide.
ACC	DINT	Temps accumulé en millisecondes. Cette valeur est conservée même si l'entrée TimerEnable est mise à zéro.
EN	BOOL	Sortie d'activation du temporisateur. Indique que l'instruction du temporisateur est activée.
TT	BOOL	Sortie de temporisation du temporisateur. Si défini, une opération de temporisation est en cours.
DN	BOOL	Sortie de temporisation terminée. Indique à quel moment le temps accumulé est supérieur ou égal à la valeur prédéfinie.
Status	DINT	État du bloc fonctionnel.
InstructFault (Status.0)	BOOL	L'instruction a détecté l'une des erreurs d'exécution suivantes. Il ne s'agit pas d'une erreur d'automate mineure ou majeure. Contrôlez les bits d'état restants pour déterminer la cause du problème.
PresetInv (Status.1)	BOOL	La valeur prédéfinie n'est pas valide.

### Description

L'instruction RTOR accumule le temps jusqu'à ce qu'elle est fausse. Lorsque l'instruction RTOR est fausse, elle conserve sa valeur ACC. Vous devez mettre la valeur .ACC à zéro en utilisant l'entrée Reset.

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur PRE.



Définir le paramètre d'entrée Reset pour réinitialiser l'instruction. Si TimerEnable est défini lorsque Reset est vrai, l'instruction RTOR recommence la temporisation lorsque Reset est mis à zéro.

### Fonctionnement d'un temporisateur

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

- $ACC = ACC + (current\_time - last\_time\_scanned)$
- Après avoir mis à jour ACC, le temporisateur définit  $last\_time\_scanned = current\_time$ . Cela rend le temporisateur disponible pour la scrutation suivante.

---

**Important :** Veillez à scruter le temporisateur au moins toutes les 69 minutes lorsqu'il fonctionne. Sinon, la valeur ACC ne sera pas correcte.

---

La valeur  $last\_time\_scanned$  dispose d'une plage allant jusqu'à 69 minutes. Le calcul du temporisateur recommence si vous ne le scrutez pas dans les 69 minutes. La valeur ACC ne sera pas correcte si cela se produit.

Pendant qu'un temporisateur fonctionne, scrutez-le pendant la période de 69 minutes si vous l'avez placé dans l'un des éléments suivants :

- Sous-routine



- Section de code comprise entre les instructions JMP et LBL
- Graphe de fonctionnement séquentiel (SFC)
- Événement ou tâche périodique
- Routine d'état d'une phase

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

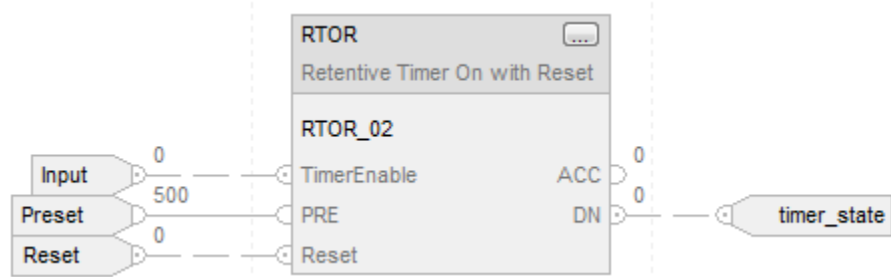
### Exécution

#### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute. Lorsque le paramètre d'entrée Reset est défini, l'instruction met EN, TT et DN à zéro, et définit ACC = 0.
Première exécution de l'instruction	EN, TT et DN sont définis sur faux. L'instruction s'exécute.
Première scrutation de l'instruction	N/A
Post-scrutation	EnableIn et EnableOut sont définis sur faux.

#### Texte structuré

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemple****Bloc fonctionnel****Texte structuré**

```
RTOR_01.PRE := 500;
```

```
RTOR_01.Reset := Reset;
```

```
RTOR_01.TimerEnable := Input;
```

```
RTOR(RTOR_01);
```

```
timer_state := RTOR_01.DN;
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Temporisateur de rémanence activé \(RTO\)](#) sur la [page 120](#)

[Réinitialisation \(RES\)](#) sur la [page 117](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

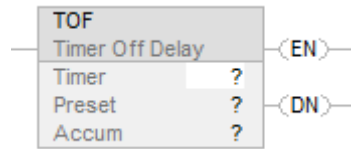
**Temporisateur au déclenchement (TOF)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction TOF est un temporisateur non rémanent qui accumule le temps lorsque l'instruction est activée (condition d'entrée d'échelon est fausse).

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

### Diagramme à relais

Opérande	Type de données	Format	Description
Timer	TIMER	étiquette	Structure de temporisateur
Preset	DINT	immédiate	Valeur de Timer.PRE.
Accum	DINT	immédiate	Valeur de Timer.ACC.

### Structure TIMER

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation contient la condition d'entrée d'échelon de la dernière exécution de l'instruction.
.TT	BOOL	Le bit de temporisation, lorsqu'il est défini, indique que l'opération de temporisation est en cours.

.DN	BOOL	Le bit de fin, lorsqu'il est mis à zéro, indique que l'opération de temporisation est terminée (ou mise en pause).
.PRE	DINT	La valeur prédéfinie spécifie la valeur accumulée (à raison d'une milliseconde par incrément) qui doit être atteinte avant que l'instruction indique qu'elle est terminée.
.ACC	DINT	La valeur accumulée spécifie le nombre de millisecondes qui se sont écoulées depuis l'activation de l'instruction TOF.

**Description**

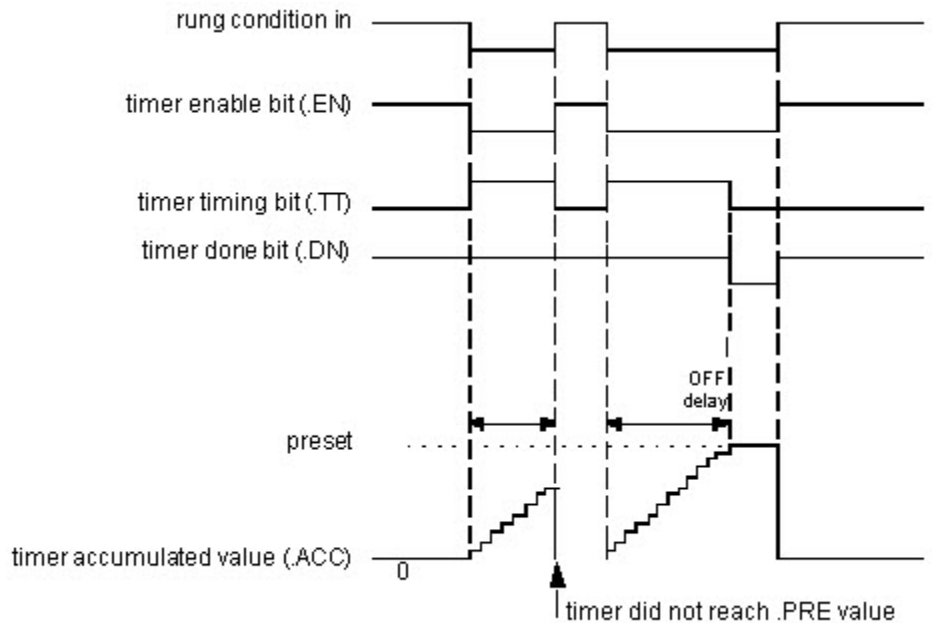
L'instruction TOF accumule le temps jusqu'à la situation suivante :

- Le temporisateur est désactivé
- Le temporisateur se termine

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur .PRE.

Le temporisateur définit le bit .DN sur faux lorsque le temporisateur se termine.

Lorsqu'elle est activée, la temporisation peut être mise en pause en définissant le bit .DN sur faux, puis reprise en définissant le bit .DN sur vrai.



**Fonctionnement d'un temporisateur**

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

$$ACC = ACC + (current\_time - last\_time\_scanned)$$

Après avoir mis à jour ACC, le temporisateur définit `last_time_scanned = current_time`. Cela rend le temporisateur disponible pour la scrutation suivante.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
.PRE < 0	4	34
.ACC < 0	4	34

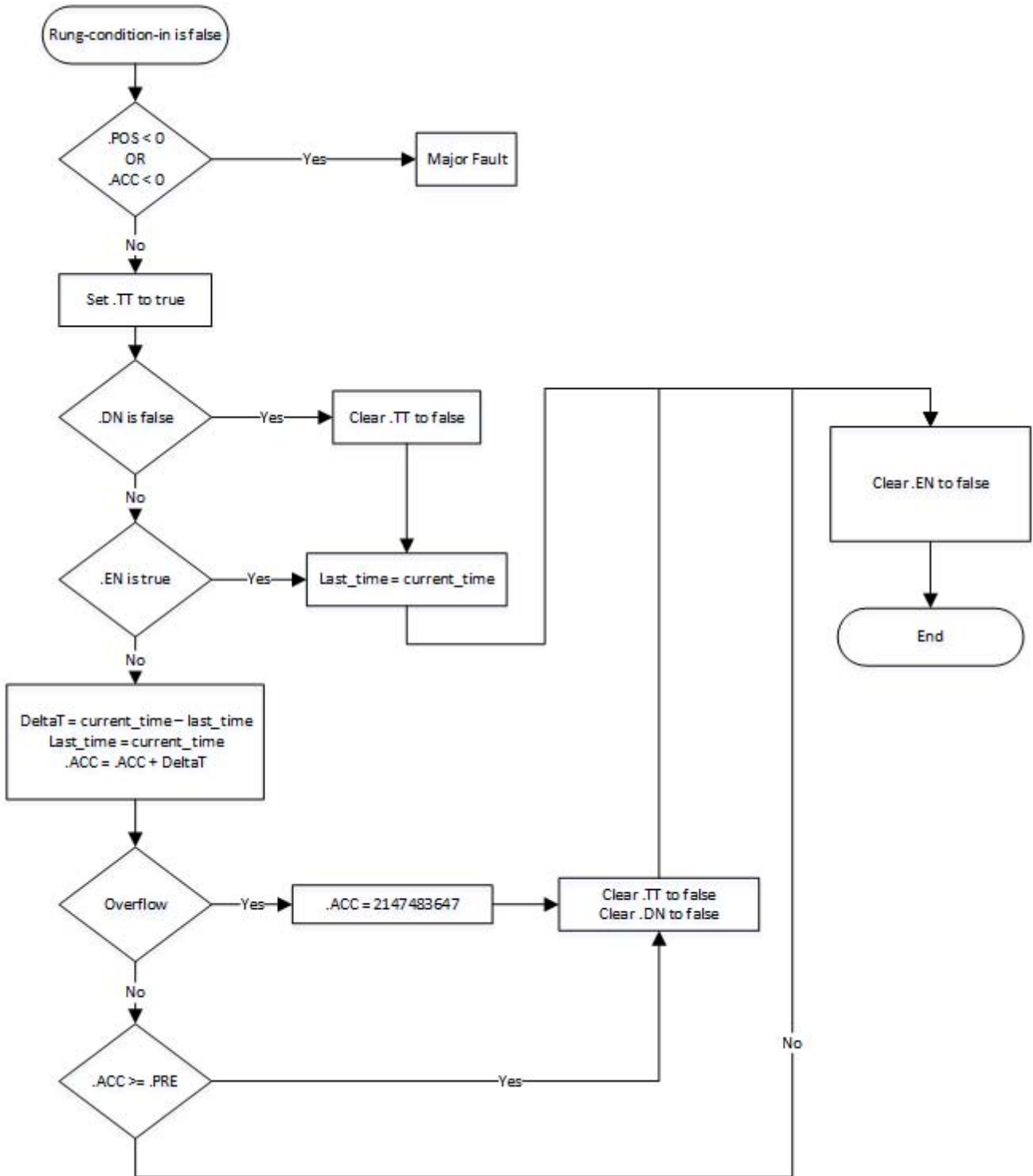
Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

### Diagramme à relais

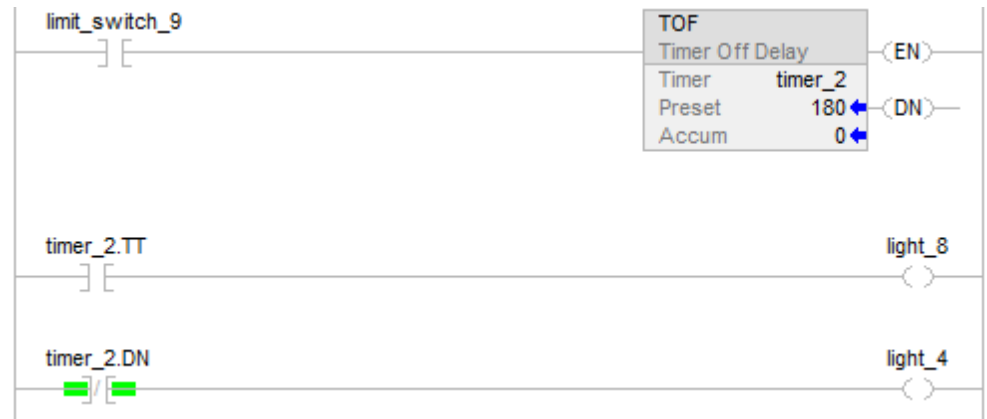
Condition/État	Action entreprise
Pré-scrutation	Le bit .EN est défini sur faux. Le bit .TT est défini sur faux. Le bit .DN est défini sur faux. La valeur .ACC est définie égale à la valeur .PRE.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme TOF (faux).
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Le bit de .EN est défini sur vrai. Le bit .TT est défini sur faux. Le bit de .DN est défini sur vrai. La valeur .ACC est définie sur zéro.
Post-scrutation	Le bit .EN est défini sur faux. Le bit .TT est défini sur faux. Le bit .DN est défini sur faux. La valeur .ACC est définie égale à la valeur .PRE.

Organigramme TOF (faux)



## Exemple

### Diagramme à relais



Lorsque `limit_switch_9` est défini, `light_8` est activé pendant 180 millisecondes (`timer_2` fonctionne). Lorsque `timer_2.acc` atteint 180, `light_8` est désactivé et `light_4` est activé. `light_4` reste activé jusqu'à ce que l'instruction TOF soit activée. Si `limit_switch_9` est vrai pendant le fonctionnement de `timer_2`, `light_8` est désactivé.

### Voir aussi

[Instructions de temporisateur et de compteur](#) sur la [page 101](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Temporisateur au déclenchement avec réinitialisation (TOFR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

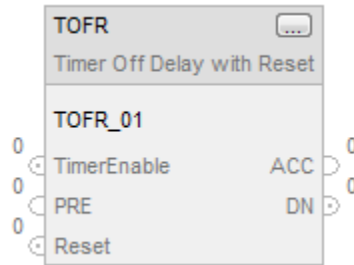
L'instruction TOFR est un temporisateur non rémanent qui accumule le temps lorsque `TimerEnable` est mis à zéro.

### Langages disponibles

### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

**Bloc fonctionnel**



**Texte structuré**

TOFR(TOFR\_tag)

**Opérandes**

**Texte structuré**

Variable	Type	Format	Description
TOFR tag	FBD_TIMER	Structure	Structure TOFR

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
TOFR tag	FBD_TIMER	Structure	Structure TOFR

**Structure FBD\_TIMER**

Paramètres d'entrée	Type de données	Description
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
TimerEnable	BOOL	Si mis à zéro, cela active le fonctionnement du temporisateur et accumule le temps. La valeur par défaut est mise à zéro.
PRE	DINT	Valeur prédéfinie du temporisateur. Correspond à la valeur par incrément de 1 ms qu'ACC doit atteindre avant la fin de la temporisation. Si non valide, l'instruction définit le bit approprié dans Status et le temporisateur ne s'exécute pas. Valide = 0 à l'entier positif maximal



Reset	BOOL	<p>Demande de réinitialisation du temporisateur. Si définir, le temporisateur se réinitialise. La valeur par défaut est mise à zéro.</p> <p>Lorsque le paramètre d'entrée Reset est défini, l'instruction met EN, TT et DN à zéro, et définit ACC = PRE. Notez que cela est différent de l'utilisation d'une instruction RES sur une instruction TOF.</p>
-------	------	---

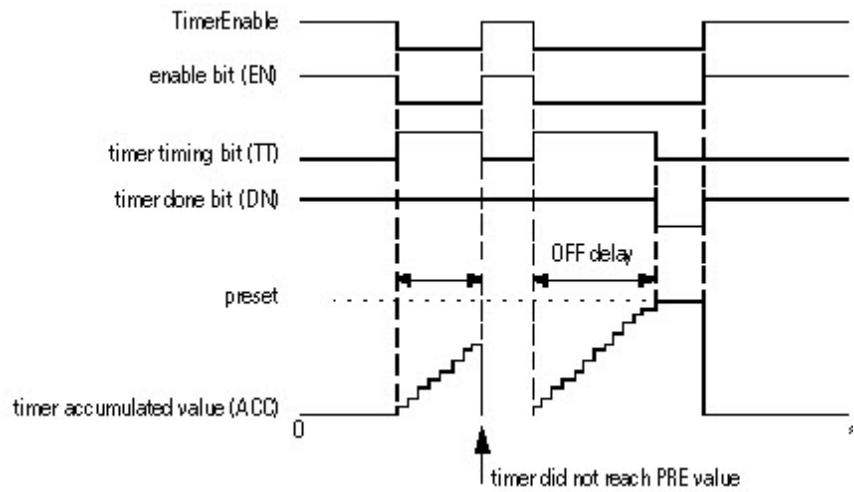
Paramètres de sortie	Type de données	Description
EnableOut	BOOL	L'instruction a produit un résultat valide.
ACC	BOOL	Temps accumulé en millisecondes.
EN	BOOL	Sortie d'activation du temporisateur. Indique que l'instruction du temporisateur est activée.
TT	BOOL	Sortie de temporisation du temporisateur. Si définir, une opération de temporisation est en cours.
DN	BOOL	Sortie de temporisation terminée. Indique à quel moment le temps accumulé est supérieur ou égal à la valeur prédéfinie.
Status	DINT	État du bloc fonctionnel.
InstructFault (Status.0)	BOOL	L'instruction a détecté l'une des erreurs d'exécution suivantes. Il ne s'agit pas d'une erreur d'automate mineure ou majeure. Contrôlez les bits d'état restants pour déterminer la cause du problème.
PresetInv (Status.1)	BOOL	La valeur prédéfinie n'est pas valide.

### Description

Lorsque définir sur vrai, l'instruction TOFR accumule le temps jusqu'à la situation suivante :

- L'instruction TOFR est désactivée
- $ACC \geq PRE$

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur PRE.



Définir le paramètre d'entrée Reset pour réinitialiser l'instruction. Si TimerEnable est faux lorsque Reset est vrai, l'instruction TOFR n'effectue pas de nouveau la temporisation lorsque Reset est faux.

### Fonctionnement d'un temporisateur

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

$$ACC = ACC + (current\_time - last\_time\_scanned)$$

Après avoir mis à jour ACC, le temporisateur définit `last_time_scanned = current_time`. Cela rend le temporisateur disponible pour la scrutation suivante.

---

**Important :** Veillez à scruter le temporisateur au moins toutes les 69 minutes lorsqu'il fonctionne. Sinon, la valeur ACC ne sera pas correcte.

---

La valeur `last_time_scanned` dispose d'une plage allant jusqu'à 69 minutes. Le calcul du temporisateur recommence si vous ne le scrutez pas dans les 69 minutes. La valeur ACC ne sera pas correcte si cela se produit.

Pendant qu'un temporisateur fonctionne, scrutez-le pendant la période de 69 minutes si vous l'avez placé dans l'un des éléments suivants :

- Sous-routine
- Section de code comprise entre les instructions JMP et LBL
- Graphe de fonctionnement séquentiel (SFC)
- Événement ou tâche périodique

- Routine d'état d'une phase

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag. EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag. EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'algorithme principal de l'instruction sera exécuté et les sorties seront mises à jour.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	EN, TT et DN sont mis à zéro et la valeur ACC n'est pas modifiée.
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

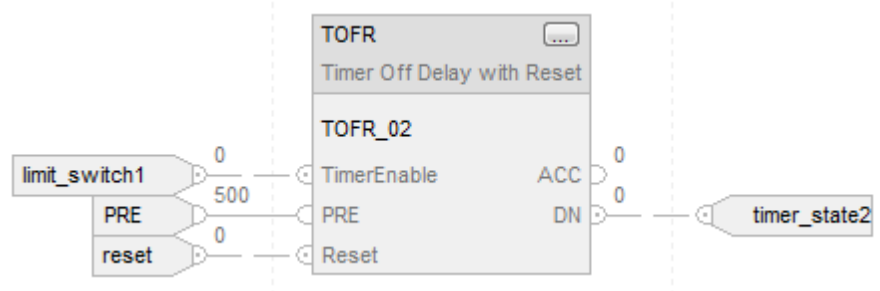
#### Texte structuré

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

#### Exemple

Chaque scrutation après limit\_switch1 est mise à zéro, l'instruction TOFR incrémente la valeur ACC du temps écoulé jusqu'à ce que la valeur ACC atteigne la valeur PRE. Lorsque  $ACC \geq PRE$ , le paramètre DN est mis à zéro et timer\_state2 est défini.

**Bloc fonctionnel**



**Texte structuré**

```

TOFR_01.PRE := 500;

TOFR_01.Reset := Reset;

TOFR_01.TimerEnable := Input;

TOFR(TOFR_01);

timer_state := TOFR_01.DN;
    
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

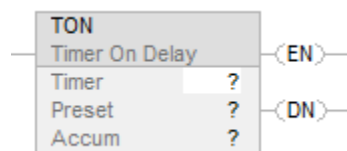
**Temporisateur à l'enclenchement (TON)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction TON est un temporisateur non rémanent qui accumule le temps lorsque l'instruction est activée.

**Langages disponibles**

**Diagramme à relais**



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

---

<b>Important :</b>	Une opération inattendue peut se produire dans les cas suivants :
	<ul style="list-style-type: none"> <li>• Les opérandes d'étiquette de sortie sont remplacés.</li> <li>• Les membres d'un opérande de structure sont remplacés.</li> <li>• Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.</li> </ul>

---

### Diagramme à relais

Opérande	Type de données	Format	Description
Timer	TIMER	étiquette	Structure de temporisateur
Preset	DINT	immédiate	Valeur de Timer.PRE.
Accum	DINT	immédiate	Valeur de Timer.ACC.

### Structure TIMER

Mnémone	Type de données	Description
.EN	BOOL	Le bit d'activation contient la condition d'entrée d'échelon de la dernière exécution de l'instruction.
.TT	BOOL	Le bit de temporisation, lorsqu'il est défini, indique que l'opération de temporisation est en cours.
.DN	BOOL	Le bit de fin, lorsqu'il est défini, indique que l'opération de temporisation est terminée (ou mise en pause).
.PRE	DINT	La valeur prédéfinie spécifie la valeur accumulée (à raison d'une milliseconde par incrément) qui doit être atteinte avant que l'instruction indique qu'elle est terminée.
.ACC	DINT	La valeur accumulée spécifie le nombre de millisecondes qui se sont écoulées depuis l'activation de l'instruction TON.

### Description

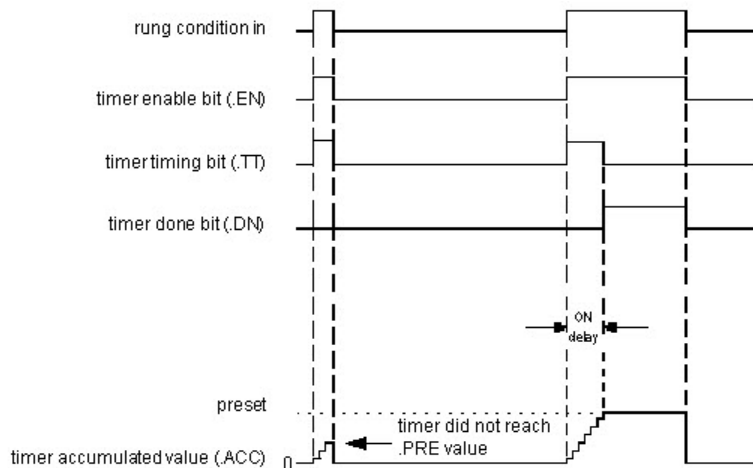
L'instruction TON accumule le temps à partir du temps auquel elle a été activée jusqu'à la situation suivante :

- Le temporisateur est désactivé
- Le temporisateur se termine

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur .PRE.

Le temporisateur définira le bit .DN sur vrai lorsque le temporisateur se termine.

Lorsqu'elle est activée, la temporisation peut être mise en pause en définissant le bit .DN sur vrai, puis reprise en définissant le bit .DN sur faux.



### Fonctionnement d'un temporisateur

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

$$ACC = ACC + (current\_time - last\_time\_scanned)$$

Après avoir mis à jour ACC, le temporisateur définit last\_time\_scanned = current\_time. Cela rend le temporisateur disponible pour la scrutation suivante.

### Affecte les indicateurs d'état mathématique

Non

**Défauts majeurs/mineurs**

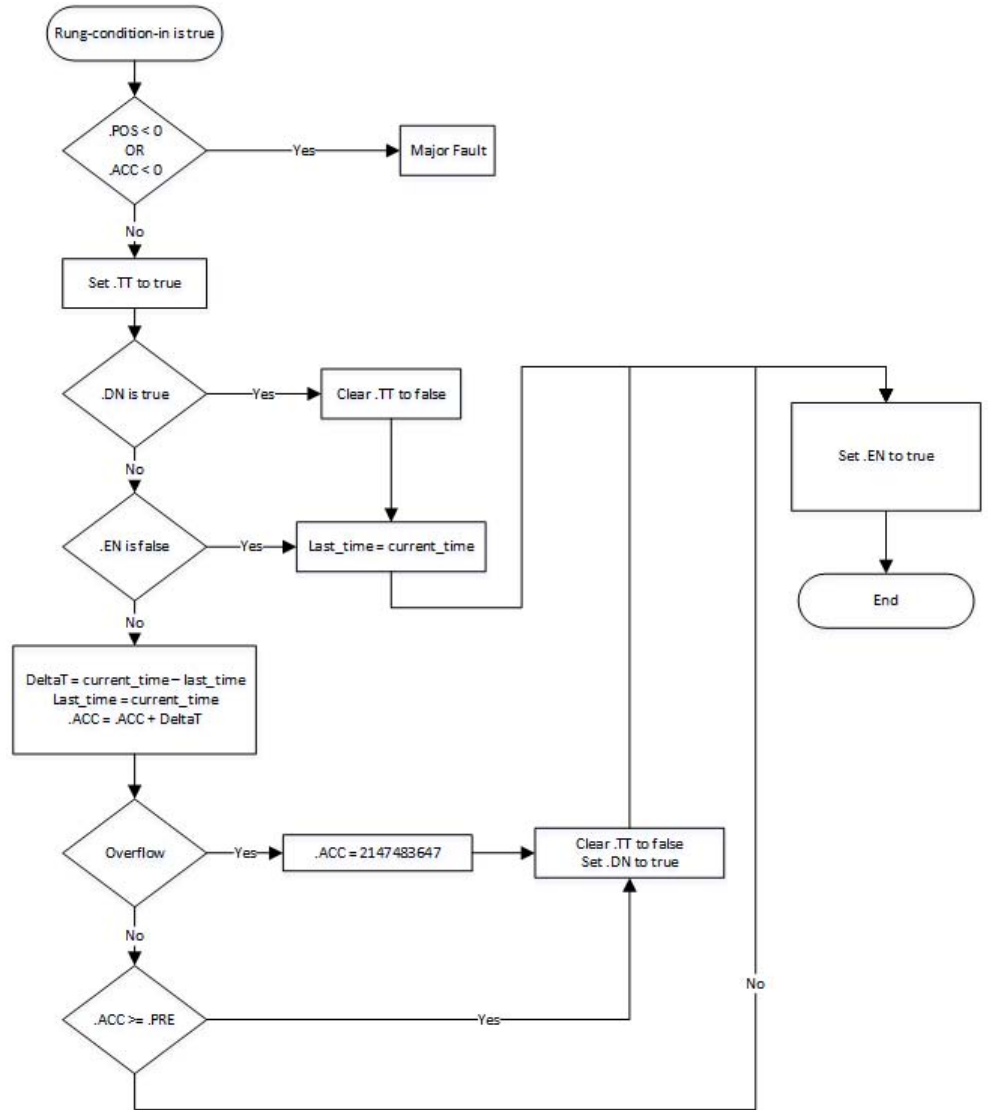
Un défaut majeur se produira si :	Type de défaut	Code de défaut
.PRE < 0	4	34
.ACC < 0	4	34

Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit .EN est défini sur faux. Le bit .TT est défini sur faux. Le bit .DN est défini sur faux. La valeur .ACC est définie sur zéro.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Le bit .EN est défini sur faux. Le bit .TT est défini sur faux. Le bit .DN est défini sur faux. La valeur .ACC est définie sur zéro.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Reportez-vous à l'organigramme TON (vrai)
Post-scrutation	Le bit .EN est défini sur faux. Le bit .TT est défini sur faux. Le bit .DN est défini sur faux. La valeur .ACC est définie sur zéro.

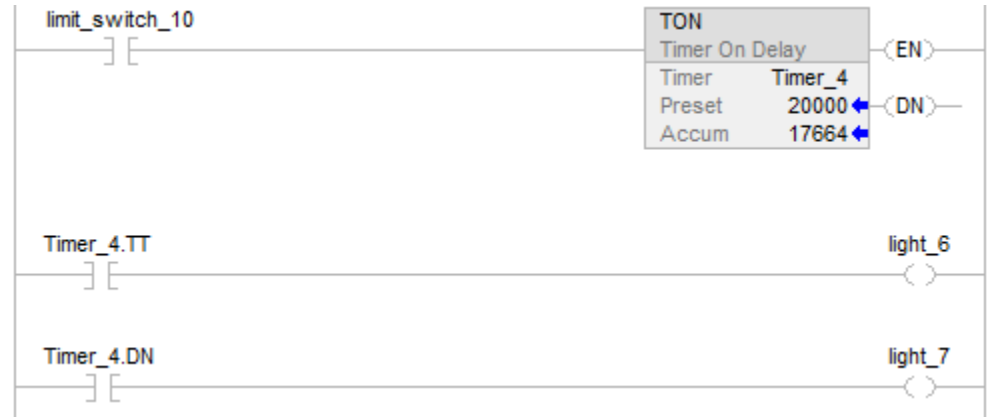
Organigramme TON (vrai)





## Exemple

### Diagramme à relais



Lorsque `limit_switch_10` est défini sur vrai, `light_6` est activé pendant 20 000 millisecondes (`Timer_4` fonctionne). Lorsque `Timer_4.acc` atteint 20 000, `light_6` est désactivé et `light_7` est activé. Si `limit_switch_10` est défini sur faux pendant le fonctionnement de `Timer_4`, `light_6` est désactivé. Lorsque `limit_switch_10` est défini sur faux, les bits d'état de `Timer_4` et la valeur `.ACC` sont réinitialisés.

### Voir aussi

[Instructions de compteur](#) sur la [page 101](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Temporisateur à l'enclenchement avec réinitialisation (TONR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

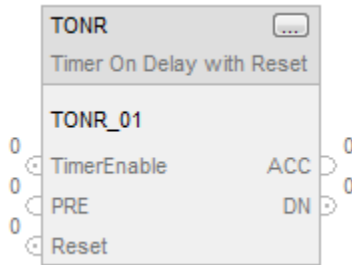
L'instruction TONR est un temporisateur non rémanent qui accumule le temps lorsque `TimerEnable` est défini.

### Langages disponibles

### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

**Bloc fonctionnel**



**Texte structuré**

TONR(TONR\_tag);

**Opérandes**

**Texte structuré**

Opérande	Type	Format	Description
TONR tag	FBD_TIMER	Structure	Structure TONR

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
TONR tag	FBD_TIMER	Structure	Structure TONR

**Structure FBD\_TIMER**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
TimerEnable	BOOL	Si définir, cela active le fonctionnement du temporisateur et accumule le temps. La valeur par défaut est mise à zéro.
PRE	DINT	Valeur prédéfinie du temporisateur. Correspond à la valeur par incrément de 1 ms qu'ACC doit atteindre avant la fin de la temporisation. Si non valide, l'instruction définit le bit approprié dans Status et le temporisateur ne s'exécute pas. Valide = 0 à l'entier positif maximal

Reset	BOOL	<p>Demande de réinitialisation du temporisateur. Si définir, le temporisateur se réinitialise.</p> <p>La valeur par défaut est mise à zéro.</p> <p>Lorsque le paramètre d'entrée Reset est défini, l'instruction met EN, TT et DN à zéro, et définit ACC = 0.</p>
-------	------	---

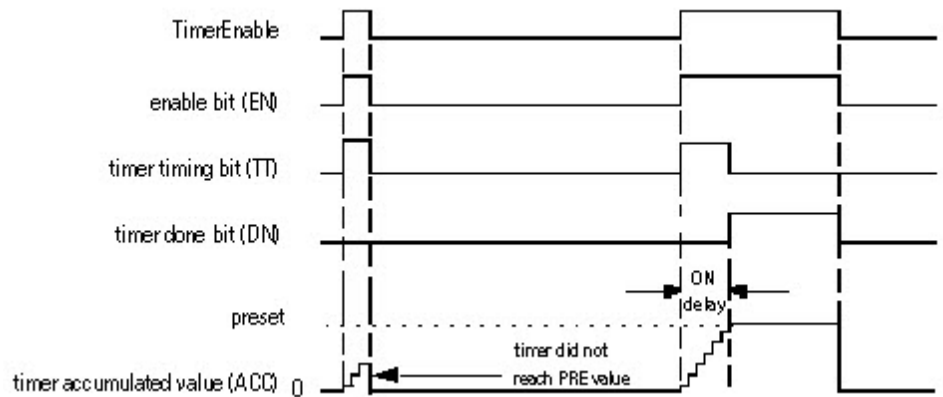
Paramètre de sortie	Type de données	Description
EnableOut	BOOL	L'instruction a produit un résultat valide.
ACC	BOOL	Temps accumulé en millisecondes.
ENF	BOOL	Sortie d'activation du temporisateur. Indique que l'instruction du temporisateur est activée.
TT	BOOL	Sortie de temporisation du temporisateur. Si définir, une opération de temporisation est en cours.
DN	BOOL	Sortie de temporisation terminée. Indique à quel moment le temps accumulé est supérieur ou égal à la valeur prédéfinie.
Status	DINT	État du bloc fonctionnel.
InstructFault (Status.0)	BOOL	L'instruction a détecté l'une des erreurs d'exécution suivantes. Il ne s'agit pas d'une erreur d'automate mineure ou majeure. Contrôlez les bits d'état restants pour déterminer la cause du problème.
PresetInv (Status.1)	BOOL	La valeur prédéfinie n'est pas valide.

### Description

Lorsque définir sur vrai, l'instruction TONR accumule le temps jusqu'à la situation suivante :

- L'instruction TONR est désactivée
- $ACC \geq PRE$

La base de temps est toujours de 1 milliseconde. Par exemple, pour un temporisateur de 2 secondes, entrez 2 000 comme valeur PRE.



Définir le paramètre d'entrée Reset pour réinitialiser l'instruction. Si TimerEnable est défini lorsque Reset est vrai, l'instruction TONR recommence la temporisation lorsque Reset est faux.

### Fonctionnement d'un temporisateur

Un temporisateur fonctionne en soustrayant le temps de sa dernière scrutation du temps actuel :

- $ACC = ACC + (current\_time - last\_time\_scanned)$

Après avoir mis à jour ACC, le temporisateur définit  $last\_time\_scanned = current\_time$ . Cela rend le temporisateur disponible pour la scrutation suivante.

---

**Important :** Veillez à scruter le temporisateur au moins toutes les 69 minutes lorsqu'il fonctionne. Sinon, la valeur ACC ne sera pas correcte.

---

La valeur  $last\_time\_scanned$  dispose d'une plage allant jusqu'à 69 minutes. Le calcul du temporisateur recommence si vous ne le scrutez pas dans les 69 minutes. La valeur ACC ne sera pas correcte si cela se produit.

Pendant qu'un temporisateur fonctionne, scrutez-le pendant la période de 69 minutes si vous l'avez placé dans l'un des éléments suivants :

- Sous-routine
- Section de code comprise entre les instructions JMP et LBL
- Graphe de fonctionnement séquentiel (SFC)
- Événement ou tâche périodique
- Routine d'état d'une phase

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Bloc fonctionnel**

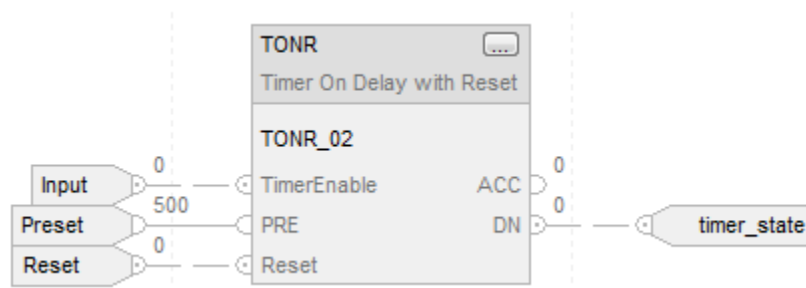
Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'algorithme principal de l'instruction est exécuté et les sorties sont mises à jour.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	EN, TT et DN sont mis à zéro et la valeur ACC est définie sur 0.
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemple**

**Bloc fonctionnel**



**Texte structuré**

TONR\_01.PRE := 500;

TONR\_01.Reset := Reset;

TONR\_01.TimerEnable := Input;

TONR(TONR\_01);

timer\_state := TONR\_01.DN;

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Temporisateur à l'enclenchement \(TON\)](#) sur la [page 140](#)

[Réinitialisation \(RES\)](#) sur la [page 117](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Entrée/Sortie

### Instructions d'entrée/sortie

Les instruction d'entrée/sortie lisent ou écrivent des données vers l'automate ou à partir de celui-ci, ou un bloc de données vers un autre module sur autre réseau ou à partir de celui-ci.

#### Instructions disponibles

##### Un diagramme à relais et texte structuré

<a href="#">MSG</a>	<a href="#">GSV</a>	<a href="#">SSV</a>	<a href="#">IOT</a>
---------------------	---------------------	---------------------	---------------------

#### Bloc fonctionnel

Indisponible

Si vous voulez :	Utilisez cette instruction :
Envoyer donner vers ou à partir d'un autre module	MSG
Obtenir les informations d'état de l'automate	GSV
Définir les informations d'état de l'automate	SSV
Envoyer les valeurs de sortie vers un module d'E/S ou utiliser l'automate à un point spécifique dans votre logique Déclencher une tâche d'événement dans un autre automate	IOT

#### Voir aussi

[Spécifier les détails de communication](#) sur la [page 179](#)

[Spécification de messages CIP](#) sur la [page 288](#)

[Sélection du type de message](#) sur la [page 269](#)

[Exemples de configuration MSG](#) sur la [page 162](#)

[Déterminer les informations relatives à la mémoire de l'automate](#) sur la [page 198](#)

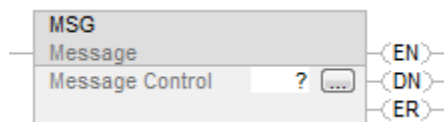
## Message (MSG)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction MSG lit ou écrit de manière asynchrone un bloc de données sur un autre module sur un réseau.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

MSG(MessageControl);

#### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description
Message	MSG	étiquette	Structure message

#### Texte structuré

Opérande	Type	Format	Description
Message	MSG	étiquette	Structure message

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.



**Structure MESSAGE**

---

**Important :** Si vous vérifiez les bits d'état plus d'une fois  
 Utilisez une copie des bits si vous les vérifiez à plus d'un  
 emplacement dans votre logique. Sinon, les bits peuvent  
 changer pendant la scrutation et votre logique ne  
 fonctionnera pas comme vous le prévoyez.  
 L'une des façons de faire une copie consiste à utiliser le  
 mot FLAGS. Copiez le mot FLAGS sur une autre  
 étiquette et vérifiez les bits dans la copie.

---



---

**Important :** Ne modifiez pas les bits suivants d'une instruction MSG :

- DN
- EN
- ER
- EW
- ST

---

Ne modifiez pas ces bits eux-mêmes ou en tant que partie du mot FLAGS. Si vous  
 le faites, l'automate peut rencontrer un défaut non récupérable. L'automate efface  
 le projet de sa mémoire lorsqu'il rencontre un défaut non récupérable.

Mnémonique	Type de données	Description	
.FLAGS	INT	Le membre .FLAGS fournit un accès aux membres d'état (bits) dans un mot unique de 16 bits.	
		<b>Le bit suivant :</b> 2	<b>Correspond au membre suivant :</b> .EW
		4	.ER
		5	.DN
		6	.ST
		7	.EN
		8	.TO
		9	.EN_CC
		<b>Important :</b> Ne modifiez pas les bits EW, ER, DN ou ST du membre FLAGS. Par exemple, n'effacez pas la totalité du mot FLAGS. L'automate ignore les modifications et utilise les valeurs stockées en interne des bits.	

.ERR	INT	Si le bit .ER est défini, le mot de code d'erreur identifie les codes d'erreur de l'instruction MSG.
.EXERR	INT	Le mot de code d'erreur étendu spécifie des informations de code d'erreur supplémentaires pour certains codes d'erreur.
.REQ_LEN	INT	La longueur requise spécifie le nombre de mots que l'instruction de message va tenter de transférer.
.DN_LEN	INT	La longueur de fin identifie le nombre de mots réellement transféré.
.EW	BOOL	Le bit d'activation est défini lorsque l'automate détecte qu'une demande de message est entrée dans la files d'attente. L'automate réinitialise le bit .EW lorsque le bit .ST est défini. <b>Important</b> : Ne modifiez pas le bit EW. L'automate ignore les modifications et utilise les valeurs stockées en interne du bit.
.ER	BOOL	Le bit d'erreur est défini lorsque l'automate détecte l'échec d'un transfert. Le bit .ER est réinitialisé au prochain passage d'EnableIn de l'état faux à vrai. <b>Important</b> : Ne modifiez pas le bit ER. L'automate ignore les modifications et utilise les valeurs stockées en interne du bit.
.DN	BOOL	Le bit de fin est défini lorsque le dernier paquet du message est transféré correctement. Le bit .DN est réinitialisé au prochain passage d'EnableIn de l'état faux à vrai. <b>Important</b> : Ne modifiez pas le bit DN. L'automate ignore les modifications et utilise les valeurs stockées en interne du bit.
.ST	BOOL	Le bit de départ est défini lorsque l'automate commence à exécuter l'instruction MSG. Le bit .ST bit est réinitialisé lorsque le bit .DN ou le bit .ER est défini. <b>Important</b> : Ne modifiez pas le bit ST. L'automate ignore les modifications et utilise les valeurs stockées en interne du bit.

.EN	BOOL	Le bit d'activation est défini lorsqu'EnableIn passe à l'état vrai et reste défini jusqu'à ce que le bit .DN ou le bit .ER soit défini et qu'EnableIn est faux. Si EnableIn passe à l'état faux, mais que le bit .DN bit et le bit .ER sont mis à zéro, le bit .EN reste défini. <b>Important</b> : Ne modifiez pas le bit EN. L'automate ignore les modifications et utilise les valeurs stockées en interne du bit.
.TO	BOOL	Si vous définissez manuellement le bit .TO, l'automate arrête de traiter le message et définit le bit .ER.
.EN_CC	BOOL	Le bit d'activation de cache détermine comment gérer la connexion MSG. Si vous souhaitez que l'automate maintienne la connexion (par exemple lorsque vous répétez la même instruction MSG à plusieurs reprises), définissez le bit .EN_CC. Si vous exécutez rarement l'instruction MSG et que vous avez d'autres besoins pour une connexion à l'automate, mettez le bit .EN_CC à zéro. Les connexions pour les instructions MSG qui sortent par le port série ne sont pas mises en cache, même si le bit .EN_CC est défini.
.ERR_SRC	SINT	Affiche le chemin de l'erreur dans la boîte de dialogue Configuration du message (Message Configuration).
.DestinationLink	INT	Pour modifier liaison destinataire d'un DH+ ou CIP avec le message d'ID de la source, définissez ce membre sur la valeur requise.
.DestinationNode	INT	Pour modifier la station destinataire d'un DH+ ou CIP avec le message d'ID de la source, définissez ce membre sur la valeur requise.
.SourceLink	INT	Pour modifier la liaison source d'un DH+ ou CIP avec le message d'ID de la source, définissez ce membre sur la valeur requise.
.Class	INT	Pour modifier le paramètre Class d'un message CIP générique, définissez ce membre sur la valeur requise.
.Attribute	INT	Pour modifier le paramètre Attribute d'un message CIP générique, définissez ce membre sur la valeur requise.
.Instance	DINT	Pour modifier le paramètre Instance d'un message CIP générique, définissez ce membre sur la valeur requise.

.LocalIndex	DINT	Si vous utilisez un astérisque [*] pour désigner le numéro d'élément du tableau local, LocalIndex fournit le numéro d'élément. Pour modifier le numéro d'élément, définissez ce membre sur la valeur requise.	
		<b>Si le message :</b>	<b>Le tableau local se trouve alors dans :</b>
		Lit les données	Élément destinataire
		Écrit les données	Élément source
.Channel	SINT	Pour envoyer le message par un canal différent du module 1756-DHRIO, définissez ce membre sur la valeur requise. Utiliser le caractère ASCII A ou B.	
.Rack	SINT	Pour modifier le numéro de rack d'un message de bloc-transfert, définissez ce membre sur le numéro du rack requis (octal).	
.Group	SINT	Pour modifier le numéro de groupe d'un message de bloc-transfert, définissez ce membre sur le numéro du groupe requis (octal).	
.Slot	SINT	Pour modifier le numéro de logement d'un message de bloc-transfert, définissez ce membre sur le numéro du logement requis.	
		<b>Si le message passe par ce réseau :</b>	<b>Spécifiez le numéro de logement dans :</b>
		I/O à distance universel	octal
		ControlNet	décimal (0 à 15)
.Path	STRING	<p>Pour envoyer le message vers un autre automate, définissez ce membre sur le nouveau chemin.</p> <p>Entrez le chemin sous forme de valeurs hexadécimales.</p> <p>Supprimez les virgules [,]</p> <p>Par exemple, pour un chemin de 1, 0, 2, 42, 1, 3, entrez \$01\$00\$02\$2A\$01\$03.</p> <p>Pour parcourir un dispositif et créer automatiquement une partie ou l'intégralité de la nouvelle chaîne, cliquez avec le bouton droit sur une étiquette de chaîne et choisissez Atteindre l'Éditeur de chemin de message.</p>	

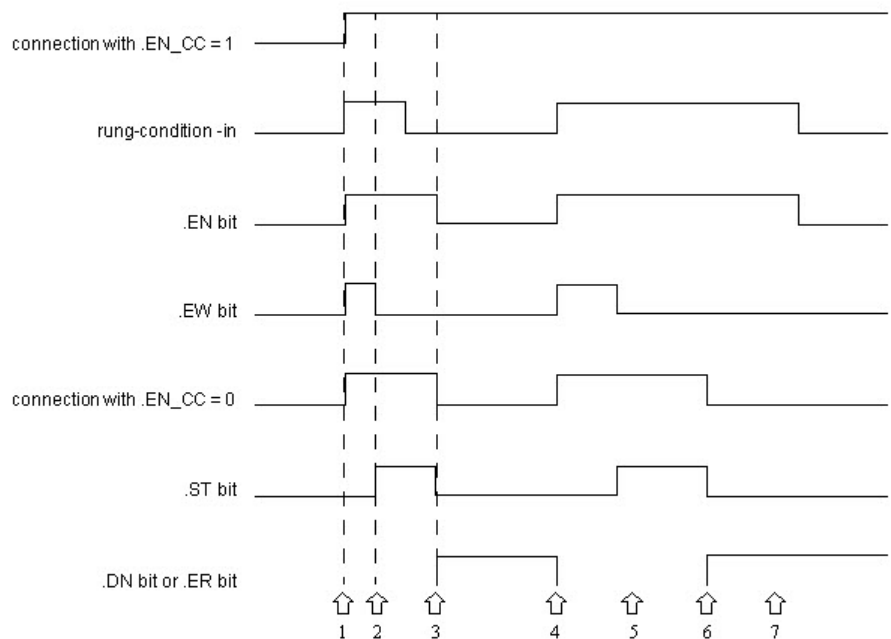
.RemoteIndex	DINT	Si vous utilisez un astérisque [*] pour désigner le numéro d'élément du tableau à distance, RemoteIndex fournit le numéro d'élément. Pour modifier le numéro d'élément, définissez ce membre sur la valeur requise.	
		<b>Si le message</b>	<b>Le tableau à distance se trouve alors dans</b>
		Lit les données	Élément source
		Écrit les données	Élément destinataire
.RemoteElement	STRING	Pour spécifier une étiquette ou une adresse différente dans l'automate vers lequel le message est envoyé, définissez ce membre sur la valeur requise. Entrez l'étiquette ou l'adresse sous forme de caractères ASCII.	
		<b>Si le message</b>	<b>Le tableau à distance se trouve alors dans</b>
		Lit les données	Élément source
		Écrit les données	Élément destinataire
.UnconnectedTimeout	DINT	<p>Dépassement de délai pour un message non connecté ou pour établir une connexion. La valeur par défaut est 30 secondes.</p> <p>Si le message n'est pas connecté, le bit ER s'active si l'automate n'obtient pas de réponse dans le temps UnconnectedTimeout.</p> <p>Si le message est connecté, le bit ER s'active si l'automate n'obtient pas de réponse pour établir la connexion dans le temps UnconnectedTimeout.</p>	
.ConnectionRate	DINT	Dépassement de délai pour un message	

.TimeoutMultiplier	SINT	<p>connecté une fois qu'une connexion est établie. Ce dépassement de délai correspond à la réponse de l'autre dispositif.</p> <p>Le dépassement de délai s'applique uniquement une fois que la connexion est établie.</p> <p>Dépassement de délai = ConnectionRate x TimeoutMultiplier</p> <p>La valeur par défaut de ConnectionRate est de 7,5 secondes.</p> <p>La valeur par défaut de TimeoutMultiplier est de 0 (ce qui équivaut à un facteur multiplicatif de 4).</p> <p>Le dépassement de délai par défaut pour les messages connectés est de 30 secondes (7,5 secondes x 4 = 30 secondes).</p> <p>Pour modifier la valeur du dépassement de délai, modifiez ConnectionRate et conservez la valeur par défaut de TimeoutMultiplier.</p>
--------------------	------	---

**Description**

L'instruction MSG transfère des éléments de données. Il s'agit d'une instruction de transition :

- Dans le diagramme à relais, basculer EnableIn de zéro à défini à chaque exécution de l'instruction.
- La taille de chaque élément dépend du type de données que vous spécifiez et du type de commande de message que vous utilisez.



Où	Description
1	EnableIn est vrai .EN est défini .EW est défini La connexion est ouverte
2	Le message est envoyé .ST est défini .EW est mis à zéro
3	Le message est terminé ou erroné EnableIn est faux. .DN ou .ER est défini .ST est mis à zéro La connexion est fermée (si .EN_CC = 0) .EN est mis à zéro (car EnableIn est faux)
4	EnableIn est vrai et .DN ou .ER a été défini précédemment .EN est défini .EW est défini La connexion est ouverte .DN ou .ER est mis à zéro
5	Le message est envoyé .ST est défini .EW est mis à zéro
6	Le message est terminé ou erroné EnableIn est toujours vrai .DN ou .ER est défini .ST est mis à zéro La connexion est fermée (si .EN_CC = 0)
7	EnableIn passe à l'état faux et .DN ou .ER est défini .EN est mis à zéro

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérands.

### Exécution

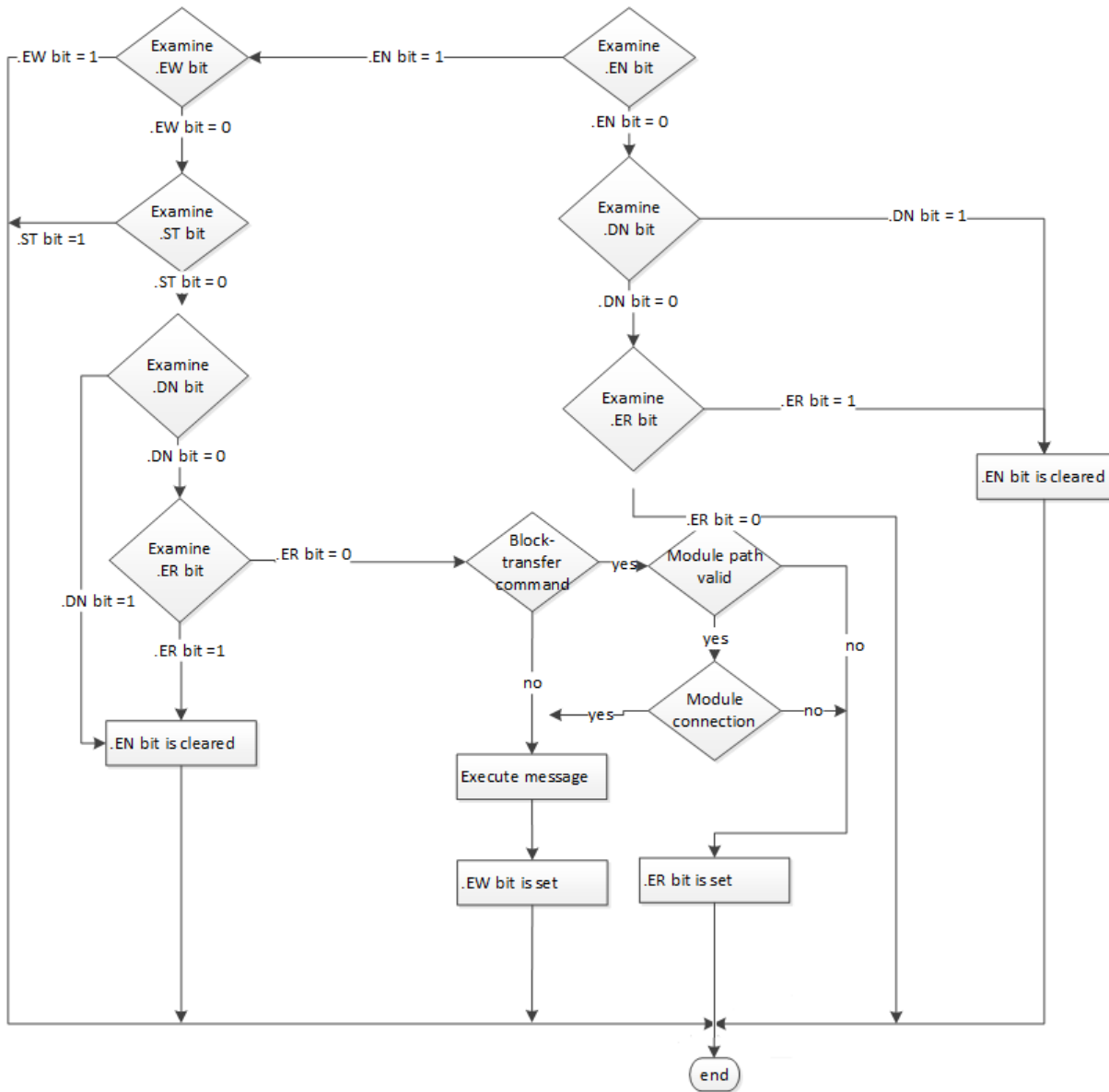
#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Les bits .EWS, .ST, .DN et .ER sont mis à zéro.
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme MSG (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme MSG (vrai)
Post-scrutation	N/A

Texte structuré

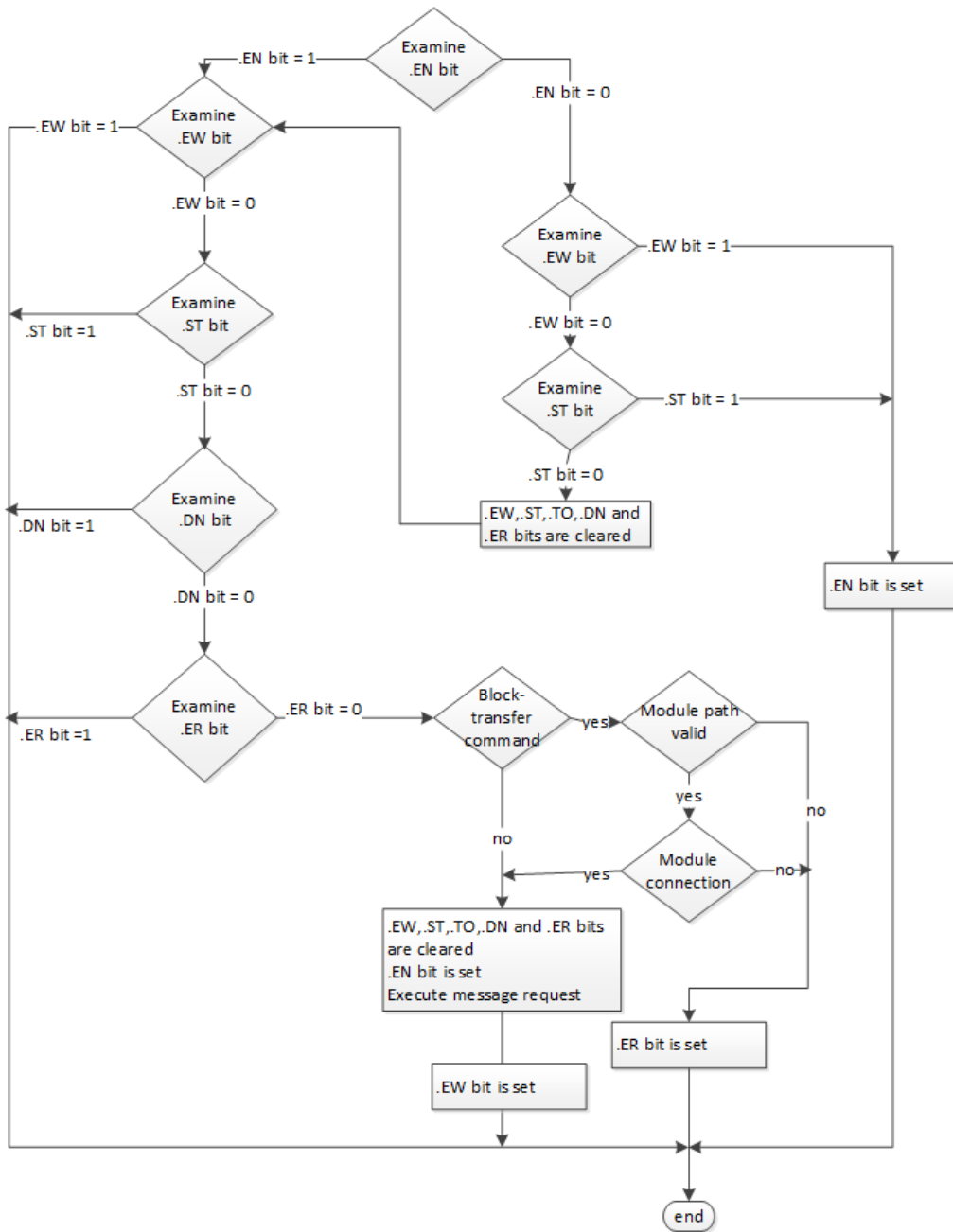
Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à l'organigramme MSG (vrai)
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais

Organigramme MSG (faux)



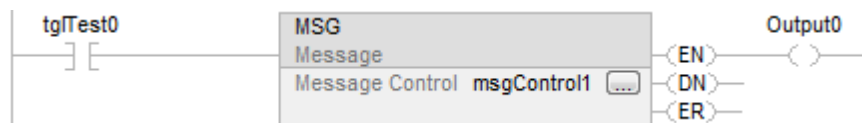


### Organigramme MSG (vrai)



### Exemple

### Diagramme à relais



**Texte structuré**

MSG (MessageControl);

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Codes d'erreur des messages](#) sur la [page 172](#)

[Sélection du type de message](#) sur la [page 269](#)

[Spécifier les détails de communication](#) sur la [page 179](#)

[Exemples de configuration MSG](#) sur la [page 162](#)

[Attributs communs](#) sur la [page 885](#)

## Exemples de configuration MSG

Les exemples suivants montrent les étiquettes de source et de destination, ainsi que les éléments pour contrôler différentes combinaisons d'automates.

Le tableau explique le chemin pour les instructions MSG provenant d'un automate LOGIX 5000 et écrites vers un autre automate.

Chemin du message	Exemple de source et de destination	
LOGIX 5000 -> LOGIX 5000	Étiquette de source	array_1[0]
	Étiquette de destination	array_2[0]
	Vous pouvez utiliser une étiquette alias pour l'étiquette de source dans l'automate LOGIX 5000 d'origine. Vous ne pouvez pas utiliser un alias pour l'étiquette de destination. La destination doit être une étiquette de base.	
LOGIX 5000 -> PLC-5 LOGIX 5000 -> SLC	Étiquette de source	array_1[0]
	Élément destinataire	N7:10
	Vous pouvez utiliser une étiquette alias pour l'étiquette de source dans l'automate LOGIX 5000 d'origine.	
LOGIX 5000 -> PLC-2	Étiquette de source	array_1[0]
	Élément destinataire	010

Le tableau explique le chemin pour les instructions MSG provenant d'un automate LOGIX 5000 et lues à partir d'un autre automate.

Chemin du message	Exemple de source et de destination	
LOGIX 5000 -> LOGIX 5000	Étiquette de source	array_1[0]
	Étiquette de destination	array_2[0]
	Vous ne pouvez pas utiliser un alias pour l'étiquette de source. La source doit être une étiquette de base. Vous pouvez utiliser une étiquette alias pour l'étiquette de destination dans l'automate LOGIX 5000 d'origine.	

LOGIX 5000 -> PLC-5 LOGIX 5000 -> SLC	Élément source	N7:10
	Étiquette de destination	array_1[0]
	Vous pouvez utiliser une étiquette alias pour l'étiquette de destination dans l'automate LOGIX 5000 d'origine.	
LOGIX 5000 -> PLC-2	Élément source	010
	Étiquette de destination	array_1[0]

Voir aussi

[Message \(MSG\)](#) sur la [page 152](#)

## Types et codes des défauts majeurs

La liste des défauts majeurs comprend les éléments suivants :

Type	Code	Cause	Méthode de récupération
1	1	L'automate sous tension est en mode Exécution.	Exécutez le gestionnaire de mise sous tension.
1	16	Détection d'un défaut de configuration de communication des E/S. (CompactLogix automates 1768-L4x uniquement.)	Reconfigurez le nombre de modules de communication du côté du bus 1768 de l'automate : <ul style="list-style-type: none"> <li>1768-L43 possède deux modules au maximum.</li> <li>1768-L45 possède quatre modules au maximum. <ul style="list-style-type: none"> <li>Jusqu'à quatre modules Sercos</li> <li>Jusqu'à deux modules de communication NetLinx</li> </ul> </li> </ul>
1	40	Si l'automate utilise une batterie, celle-ci ne contient alors pas une charge suffisante pour enregistrer le programme utilisateur lors de la mise hors tension.  Si l'automate utilise un ESM (module de stockage d'énergie), celui-ci ne contient alors pas une charge suffisante pour enregistrer le programme utilisateur lors de la mise hors tension.	<ul style="list-style-type: none"> <li>Pour les automates qui utilisent une batterie, remplacez la batterie.</li> <li>Pour les automates qui utilisent un ESM (module de stockage d'énergie) : <ul style="list-style-type: none"> <li>Laissez l'ESM se charger complètement avant de mettre l'automate hors tension.</li> <li>Remplacez l'ESM si celui-ci est amovible ou remplacez l'automate s'il ne l'est pas.</li> </ul> </li> <li>Si le problème persiste, contactez l'assistance Rockwell Automation.</li> </ul>
1	60	Pour un automate n'ayant pas de carte mémoire installée, l'automate : <ul style="list-style-type: none"> <li>A détecté un défaut non récupérable.</li> <li>A effacé le projet de la mémoire.</li> </ul>	<ol style="list-style-type: none"> <li>Effacez le défaut.</li> <li>Téléchargez le projet.</li> <li>Passer sur le mode Exécution distante ou Exécution.</li> </ol> <p>Si le défaut persiste :</p> <ol style="list-style-type: none"> <li>Avant d'effectuer des cycles d'alimentation de l'automate, enregistrez l'état des indicateurs d'état OK et RS232.</li> <li>Contactez l'assistance Rockwell Automation.</li> </ol>
1	61	Pour un automate avec une carte mémoire installée, l'automate : <ul style="list-style-type: none"> <li>A détecté un défaut non récupérable.</li> <li>A écrit des informations de diagnostic sur la carte mémoire.</li> <li>A effacé le projet de la mémoire.</li> </ul>	<ol style="list-style-type: none"> <li>Effacez le défaut.</li> <li>Téléchargez le projet.</li> <li>Passer sur le mode Exécution distante/Exécution.</li> </ol> <p>Si le défaut persiste, contactez l'assistance Rockwell Automation.</p>

1	62	<p>Pour un automate avec une carte Secure Digital (SD) installée, l'automate :</p> <ul style="list-style-type: none"> <li>• A détecté un défaut non récupérable.</li> <li>• A écrit des informations de diagnostic sur la carte mémoire.</li> </ul> <p>Lorsqu'il est dans cet état, l'automate n'ouvre aucune connexion et ne permet pas de passer en mode Exécution.</p>	<p>1. Effacez le défaut. 2. Téléchargez le projet. 3. Passez sur le mode Exécution distante ou Exécution. Si le défaut persiste, contactez l'assistance Rockwell Automation.</p>
3	16	Échec d'une connexion de module d'E/S requise.	<p>Vérifiez les éléments suivants :</p> <ul style="list-style-type: none"> <li>• Le module d'E/S est dans le châssis.</li> <li>• Les conditions requises pour le détrompage électronique.</li> <li>• L'onglet Défaut majeur dans Propriété de l'automate et l'onglet Connexion dans Propriété du module pour obtenir plus d'informations sur le défaut.</li> </ul>
3	20 21	Problèmes possibles avec le châssis.	Non récupérable : remplacez le châssis.
3	23	Au moins une connexion requise n'a pas été établie avant de passer en mode Exécution.	Attendez que le témoin d'E/S de l'automate passe au vert avant de passer en mode Exécution.
4	16	Rencontre d'une instruction inconnue.	Supprimez l'instruction inconnue. Cela s'est probablement produit en raison d'un processus de conversion du programme.
4	20	Indice de tableau trop grand, la structure de contrôle .POS ou .LEN n'est pas valide.	Ajustez la valeur pour qu'elle se situe dans la plage définie. Ne dépassez pas la taille du tableau et n'allez pas au-delà des dimensions définies.
4	21	Structure de contrôle .LEN ou .POS < 0.	Modifiez la valeur pour qu'elle soit > 0.
4	31	Les paramètres de l'instruction JSR ne correspondent pas avec ceux de l'instruction SBR ou RET associée.	Transmettez le nombre approprié de paramètres. Si vous transmettez un trop grand nombre de paramètres, les paramètres excédentaires seront ignorés sans aucune erreur.
4	34	Une instruction de temporisateur a une valeur prédéfinie négative ou une valeur accumulée.	Corrigez le programme pour ne pas charger de valeur négative dans une valeur prédéfinie ou accumulée du temporisateur.
4	42	Instruction JMP vers une étiquette qui n'existait pas ou a été supprimée.	Corrigez la cible de l'instruction JMP ou ajoutez l'étiquette manquante.
4	82	Un graphe de fonctionnement séquentiel (SFC) a appelé une sous-routine et celle-ci a essayé de revenir à l'appel SFC. Se produit lorsque SFC utilise une instruction JSR ou FOR pour appeler la sous-routine.	Supprimez le retour vers l'appel SFC.
4	83	Les données testées ne se trouvaient pas à l'intérieur des limites requises. Cela se produit avec des indices de tableaux utilisés avec des tableaux booléens et un adressage au niveau des bits.	Ajustez la valeur pour qu'elle se situe dans la plage valide. Ne dépassez pas la taille du tableau et n'allez pas au-delà des dimensions définies.
4	84	Débordement de pile.	Réduisez les niveaux d'imbrication de la sous-routine ou le nombre de paramètres transmis.
4	89	Dans une instruction SFR, la routine cible ne contient pas l'étape cible.	Corrigez la cible de l'instruction SFR ou ajoutez l'étape manquante.
4	90	Utilisation d'une instruction de sécurité en dehors d'une tâche de sécurité.	Placez l'instruction de sécurité à l'intérieur de la tâche de sécurité.
4	91	L'instruction Phase d'équipement est appelée depuis l'extérieur d'un programme de Phase d'équipement.	Utilisez uniquement l'instruction dans un programme de Phase d'équipement.

4	94	Limites d'imbrication dépassées.	Restructurez le projet afin de réduire les niveaux d'imbrication des sous-routines.
4	990 - 999	Défaut majeur défini par l'utilisateur.	
6	1	Le chien de garde de la tâche a expiré. La tâche utilisateur ne s'est pas terminée pendant la période spécifiée. Une erreur du programme a provoqué une boucle infinie ou le programme est trop complexe pour s'exécuter aussi rapidement que spécifié, ou une tâche de priorité supérieure empêche de terminer cette tâche (tentative d'effectuer un trop grand nombre d'opérations avec un seul automate).	Augmentez le chien de garde de la tâche, raccourcissez le temps d'exécution, donnez une priorité plus élevée à cette tâche, simplifiez les tâches de priorité plus élevée ou déplacez une partie du code vers un autre automate.
7	40	Échec du stockage dans la mémoire non volatile.	<ul style="list-style-type: none"> <li>• Réessayez de stocker le projet dans la mémoire non volatile.</li> <li>• S'il n'est pas possible de stocker le projet dans la mémoire non volatile, remplacez la carte mémoire.</li> <li>• Si vous utilisez un automate 1756-L7x, vérifiez que la carte SD est déverrouillée.</li> </ul>
7	41	Échec du chargement à partir de la mémoire non volatile en raison d'une incompatibilité de type d'automate.	Utilisez un automate du type correct ou téléchargez le projet est stockez-le sur la carte mémoire.
7	42	Échec du chargement de la mémoire non volatile, car la révision du firmware du projet dans la mémoire non volatile ne correspond pas à la révision du firmware de l'automate.	Mettez à jour le firmware de l'automate vers le même niveau de révision que celui du projet se trouvant dans la mémoire non volatile.
7	43	Échec du chargement à partir de la mémoire non volatile en raison d'une somme de contrôle erronée.	Contactez l'assistance Rockwell Automation.
7	44	Échec de la restauration de la mémoire du processeur.	Contactez l'assistance Rockwell Automation.
7	50	Impossible de vérifier le certificat du fichier journal. Lorsque l'automate démarre, il essaye de vérifier la combinaison clé/certificat du fichier journal. Selon la vérification, l'automate entreprend l'une des actions suivantes : <ul style="list-style-type: none"> <li>• Si l'automate vérifie le certificat du fichier journal existant, il poursuit avec le répertoire de journal existant.</li> <li>• S'il n'est pas possible de vérifier le certificat existant, l'automate consigne un défaut majeur et tente de créer un certificat. <ul style="list-style-type: none"> <li>• Si l'automate crée correctement un certificat, il crée un sous-répertoire de journal de sauvegarde, déplace les fichiers existants vers ce répertoire et poursuit la journalisation et la signature avec la nouvelle clé de vérification et le certificat de fichier journal.</li> <li>• Si l'automate ne peut pas créer de certificat, il écrit les entrées de journal dans le répertoire de journal existant, mais ne met pas à jour les fichiers de signatures dans ce répertoire.</li> </ul> </li> </ul>	Effacez le défaut et soumettez l'automate à un cycle d'alimentation. Si le problème persiste, contactez l'assistance Rockwell Automation.

8	1	Tentative de placement de l'automate en mode Exécution avec un interrupteur à clé pendant le téléchargement.	Attendez que le téléchargement se termine et effacez le défaut.
11	1	La position réelle a dépassé la limite de surcourse positive.	Déplacez l'axe dans le sens négatif jusqu'à ce que sa position se situe dans la limite de surcourse, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	2	La position réelle a dépassé la limite de surcourse négative.	Déplacez l'axe dans le sens positif jusqu'à ce que sa position se situe dans la limite de surcourse, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	3	La position réelle a dépassé la tolérance d'erreur de position.	Déplacez la position au sein de la tolérance, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	4	La connexion avec le canal A, B ou Z du codeur est interrompue.	Reconnectez le canal du codeur, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	5	Détection d'un événement de bruit du codeur ou les signaux de codeur ne sont pas un quadrature.	Rectifiez le câblage du codeur, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	6	L'entrée Défaut de variateur a été activée.	Effacez le défaut de variateur, puis exécutez Réinitialisation de défaut d'axe de mouvement.
11	7	La connexion synchrone a occasionné une défaillance.	Exécutez d'abord Réinitialisation de défaut d'axe de mouvement. Si cela ne fonctionne pas, retirez le servo-module, puis rebranchez-le. En cas d'échec, remplacez le servo-module.
11	8	Le servo-module a détecté un défaut matériel grave.	Remplacez le module.
11	9	La connexion asynchrone a occasionné une défaillance.	Exécutez d'abord Réinitialisation de défaut d'axe de mouvement. Si cela ne fonctionne pas, retirez le servo-module, puis rebranchez-le. En cas d'échec, remplacez le servo-module.
11	10	Un défaut du moteur s'est produit.	Reportez-vous à l'étiquette d'axe DriveFaults pour obtenir plus d'informations.
11	11	Un défaut thermique du moteur s'est produit.	Reportez-vous à l'étiquette d'axe DriveFaults pour obtenir plus d'informations.
11	12	Un défaut thermique du moteur s'est produit.	Reportez-vous à l'étiquette d'axe DriveFaults pour obtenir plus d'informations.
11	13	Un défaut de boucle SERCOS s'est produit.	Vérifiez l'intégrité du réseau de boucle de fibre optique SERCOS et les dispositifs sur celui-ci.
11	14	Un défaut d'entrée d'activation de variateur s'est produit.	Réactivez l'entrée d'activation de variateur et effacez le défaut.
11	15	Un défaut de perte de phase de variateur s'est produit.	Restaurez la connexion pleine puissance vers le variateur et effacez le défaut.
11	16	Un défaut de protection de variateur s'est produit.	Reportez-vous à étiquette d'axe GuardFaults pour plus obtenir plus d'informations.
11	32	La tâche de mouvement a rencontré un chevauchement.	Le découpage intermédiaire du groupe est trop élevé pour conserver un fonctionnement correct. Effacez l'étiquette de défaut du groupe, augmentez le taux de mise à jour du groupe, puis effacez le défaut majeur.

12	32	L'alimentation a été effectuée vers un automate secondaire disqualifié et aucun châssis partenaire ou automate n'a été détecté lors de la mise sous tension.	Vérifiez que : <ul style="list-style-type: none"> <li>• Un châssis partenaire est connecté.</li> <li>• L'alimentation est appliquée aux deux châssis redondants.</li> <li>• Les automates partenaires possèdent les mêmes éléments suivants : <ul style="list-style-type: none"> <li>• référence.</li> <li>• numéro de logement.</li> <li>• révision du firmware.</li> </ul> </li> </ul>
12	33	Un automate non-partenaire a été identifié dans le nouveau châssis principal après une commutation.	Soit : <ul style="list-style-type: none"> <li>• Supprimez l'automate non partenaire et résolvez la cause de la commutation.</li> <li>• Ajoutez un automate partenaire au châssis secondaire.</li> <li>• Résolvez la cause de la commutation et synchronisez le système.</li> </ul>
12	34	Juste après une commutation, les positions des interrupteurs des automates principal et secondaire ne correspondent pas. L'ancien automate principal est en mode Programme et le nouvel automate principal est en mode Exécution.	Soit : <ul style="list-style-type: none"> <li>• Modifiez les interrupteurs en les faisant passer du mode Exécution à Programme puis Exécution à deux reprises pour effacer le défaut.</li> <li>• Utilisez l'application Logix Designer pour passer en ligne avec les automates. Effacez ensuite les défauts et faites passer les modes des deux automates sur Exécution.</li> </ul>
14	1	Le chien de garde de la tâche de sécurité a expiré. La tâche utilisateur ne s'est pas terminée pendant une période spécifiée. Une erreur du programme a provoqué une boucle infinie, le programme est trop complexe pour s'exécuter aussi rapidement que spécifié, une tâche de priorité supérieure empêche de terminer cette tâche ou le partenaire de sécurité a été supprimé.	Effacez le défaut. S'il existe une signature de tâche de sécurité, la mémoire de sécurité est réinitialisée et la tâche de sécurité commence à s'exécuter. S'il n'existe pas de signature de tâche de sécurité, vous devez télécharger de nouveau le programme pour permettre l'exécution de la tâche de sécurité. Réinsérez le partenaire de sécurité, s'il a été supprimé.
14	2	Il existe une erreur dans une routine de la tâche de sécurité.	Corrigez l'erreur dans la routine dans la logique du programme utilisateur.
14	3	Le partenaire de sécurité est manquant.	Installez un partenaire de sécurité compatible.
14	4	Le partenaire de sécurité est introuvable.	Installez un partenaire de sécurité compatible.
14	5	Le matériel du partenaire de sécurité n'est pas compatible.	Installez un partenaire de sécurité compatible.
14	6	Le firmware du partenaire de sécurité n'est pas compatible.	Installez un partenaire de sécurité compatible.
14	7	La tâche de sécurité est inexploitable. Ce défaut se produit lorsque la logique de sécurité n'est pas valide, par exemple lorsqu'il existe des éléments qui ne correspondent pas dans la logique entre l'automate principal et le partenaire de sécurité, en cas d'expiration de délai du chien de garde ou lorsque la mémoire est endommagée.	Effacez le défaut. S'il existe une signature de tâche de sécurité, la mémoire de sécurité est réinitialisée à l'aide de la signature de tâche de sécurité et la tâche de sécurité commence à s'exécuter. S'il n'existe pas de signature de tâche de sécurité, vous devez télécharger de nouveau le programme pour permettre l'exécution de la tâche de sécurité.
14	8	Temps système coordonné maître (CST) introuvable.	Effacez le défaut. Configurez un dispositif pour qu'il soit le maître du temps CST.
14	9	Défaut non récupérable de l'automate du partenaire de sécurité.	Effacez le défaut et téléchargez le programme. Si le défaut persiste, remplacez le partenaire de sécurité.

17	34	La température interne de l'automate a dépassé la limite de fonctionnement.	Il convient de prendre des mesures pour réduire la température ambiante du module. Respectez les limites recommandées pour la température ambiante (entrée) et appliquez les dégagements requis autour du châssis.
17	37	L'automate a récupéré après un défaut de température interne.	Généré lorsque l'automate récupère après un arrêt automatique. L'arrêt se produit lorsque la température du module dépasse le seuil de température du défaut de préservation. Lorsque la température redescend à un niveau acceptable, cela réactive les tensions de l'automate et génère le défaut Type 17, Code 37.
18	1	Le variateur CIP Motion n'est pas initialisé correctement.	Pour déterminer une action corrective, reportez-vous à Attributs de défauts d'initialisation pour obtenir plus d'informations sur le type de défaut qui s'est produit.
18	2	Le variateur CIP Motion n'est pas initialisé correctement. Ce défaut est indiqué lorsqu'il se produit un défaut d'initialisation spécifique au fabricant.	Pour déterminer l'action corrective, reportez-vous à Défauts d'initialisation CIP - attributs Mfg pour obtenir plus d'informations sur le défaut qui s'est produit.
18	3	Le bit Défaut d'axe physique est défini, ce qui indique un défaut sur l'axe physique.	Pour déterminer une action corrective, reportez-vous à Attributs de défaut d'axe CIP pour obtenir plus d'informations sur le défaut qui s'est produit.
18	4	Le bit Défaut d'axe physique est défini, ce qui indique un défaut sur l'axe physique. Ce défaut est indiqué lorsqu'il se produit un défaut d'axe spécifique au fabricant.	Pour déterminer une action corrective, reportez-vous à Défaut d'initialisation CIP - attributs Mfg pour obtenir plus d'informations sur le défaut qui s'est produit.
18	5	Un défaut de mouvement s'est produit.	Pour déterminer une action corrective, reportez-vous à l'attribut et aux bits du défaut de mouvement pour obtenir plus d'informations sur le défaut qui s'est produit.
18	6	Un défaut de variateur CIP Motion s'est produit. Généralement, le défaut affecte tous les axes associés au module et tous les axes associés sont arrêtés.	Reconfigurez le module de mouvements en défaut pour corriger le défaut.
18	7	Un défaut de groupe de mouvements s'est produit. Généralement, le défaut affecte tous les axes associés à un groupe de mouvements.	Reconfigurez l'intégralité du sous-système de mouvement pour corriger le défaut.
18	8	Un défaut s'est produit pendant la configuration d'un variateur CIP Motion. Généralement, ce défaut se produit après une tentative infructueuse de mise à jour d'un attribut de configuration d'axe d'un variateur CIP Motion.	Pour déterminer l'action corrective, reportez-vous à Défaut de configuration dans les attributs Code d'erreur d'attribut et ID d'erreur d'attribut associés au mouvement ou au module 1756-ENxT.
18	9	Un défaut de récupération de la position absolue (APR) s'est produit et il est impossible de récupérer la position absolue de l'axe.	Pour déterminer l'action corrective, reportez-vous à Défaut APR afin de déterminer la cause du défaut.
18	10	Un défaut de récupération de la position absolue (APR) s'est produit et il est impossible de récupérer la position absolue de l'axe. Ce défaut est indiqué lorsqu'il se produit un défaut APR spécifique au fabricant.	Pour déterminer l'action corrective, reportez-vous à Défaut APR - attributs Mfg afin de déterminer la cause du défaut.
18	128	Un défaut spécifique à la fonction de sécurité Mouvement de protection s'est produit. Ce défaut n'est applicable que lors de l'utilisation d'un variateur avec la fonctionnalité Mouvement de protection.	Pour déterminer l'action corrective, reportez-vous aux attributs de Mouvement de protection et aux bits d'État de protection afin de déterminer la cause du défaut.



20	1	Une licence requise est introuvable ou a expiré pendant la transition en mode exécution ou en mode test.	Insérez une CmCard contenant toutes les licences requises par le projet dans l'automate.
----	---	--	--

Mots-clés : faults:4, fault code:1, fault codes:1

## Types et codes des défauts mineurs

Vous trouverez ci-dessous la liste des codes et types de défaut mineur.

La liste des défauts mineurs comprend les éléments suivants :

Type	Code	Cause	Méthode de récupération
1	15	<ul style="list-style-type: none"> <li>• Une alimentation 1769 est connectée directement au CompactBus 1768 de l'automate, avec une configuration non valide.</li> <li>• L'alimentation 1768 de l'automate est défaillante.</li> </ul>	<ul style="list-style-type: none"> <li>• Mettez hors tension le CompactBus 1768 et remettez sous tension le système.</li> <li>• Remplacez l'alimentation.</li> </ul>
3	1	Condition de bus désactivé Les connexions entre l'automate et les modules d'E/S sont interrompues.	<p>Exécutez ces étapes pour identifier la source du défaut BUS DÉACTIVÉ :</p> <ol style="list-style-type: none"> <li>1. Le nombre de modules locaux d'extension du projet correspond au nombre de modules installés physiquement dans le système.</li> <li>2. Tous les supports de fixation sont verrouillés et les modules d'E/S sont installés de manière sécurisée sur les supports de fixation.</li> <li>3. Tous les modules d'E/S POINT 1734 sont configurés pour utiliser la vitesse de transmission automatique.</li> </ol> <p>Si ces étapes ne résolvent pas la condition de défaut, contactez l'assistance Rockwell Automation.</p>
3	94	La mise à jour du RPI actuel d'un module d'E/S se superpose à sa mise à jour RPI précédente.	<p>Définissez le taux RPI des modules d'E/S sur une valeur numérique plus élevée.</p> <p>Rockwell Automation recommande de ne pas exécuter les systèmes de contrôle CompactLogix 5370 L2 et CompactLogix 5370 L3 avec des défauts de superposition RPI des modules.</p>
3	100	Il existe un risque de perte d'intégrité des données avec le module, car l'une ou les deux tailles d'entrée/sortie est supérieure à 16 octets et le module ne prend pas en charge le démarrage et la perte d'intégrité.	<p>Méthodes de récupération :</p> <ul style="list-style-type: none"> <li>• Diminuez les tailles d'entrée/sortie en leur donnant une valeur inférieure ou égale à 16 octets, ce qui évite les problèmes de perte d'intégrité des données.</li> <li>• Contactez le fournisseur du module pour demander une version qui prenne en charge la fonction d'intégrité au démarrage et à l'arrêt.</li> <li>• Pour de plus amples informations, consultez Rockwell Automation ID de réponse de la base de connaissances <a href="https://support.rockwellautomation.com/s/article/1028837">1028837</a>.</li> </ul>
4	4	Un débordement arithmétique s'est produit dans une instruction.	Corrigez le programme en examinant les opérations arithmétiques (ordre) ou en ajustant les valeurs.
4	5	Dans une instruction GSV/SSV, l'instance spécifiée est introuvable.	Vérifiez le nom de l'instance.

4	6	Dans une instruction GSV/SSV : <ul style="list-style-type: none"> <li>Le nom de la classe spécifié n'est pas pris en charge</li> <li>Le nom de l'attribut spécifié n'est pas valide</li> </ul>	Vérifiez les noms de la classe et de l'attribut.
4	7	L'étiquette de destination GSV/SSV était trop petite pour contenir toutes les données.	Modifiez la destination ou la source pour que l'espace disponible soit suffisant.
4	30	Paramètres incorrects transmis au port ASCII.	Vérifiez les paramètres de configuration ASCII.
4	35	Le temps delta PID $\leq 0$ .	Modifiez le temps delta PID pour qu'il soit $> 0$ .
4	36	Point de consigne PID hors limites.	Modifiez le point de consigne pour qu'il soit compris dans la plage autorisée.
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	<ul style="list-style-type: none"> <li>Vérifiez qu'aucune instruction n'est écrite dans le membre LEN de l'étiquette de la chaîne.</li> <li>Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.</li> </ul>
4	52	La chaîne de sortie est plus grande que la destination.	Créez un nouveau type de données de chaîne suffisamment grand pour la chaîne de sortie. Utilisez le nouveau type de données de chaîne comme type de données pour la destination.
4	53	Le numéro de sortie se situe au-delà des limites du type de données de destination.	Soit : <ul style="list-style-type: none"> <li>Réduisez la taille de la valeur ASCII.</li> <li>Utilisez un type de données de destination plus grand.</li> </ul>
4	56	La valeur Démarrage ou Quantité n'est pas valide.	<ul style="list-style-type: none"> <li>Vérifiez que la valeur Démarrage se situe entre 1 et la taille DATA de la Source.</li> <li>Vérifiez que la valeur Démarrage plus la valeur Quantité est inférieure ou égale à la taille DATA de la Source.</li> </ul>
4	57	Échec d'exécution de l'instruction AHL, car le port série est défini sur Pas d'échange.	Soit : <ul style="list-style-type: none"> <li>Changez le paramètre Ligne de commande du port série.</li> <li>Supprimez l'instruction AHL.</li> </ul>
6	2	Superposition de tâches périodiques. La tâche périodique ne s'est pas terminée avant que ce soit le moment de la réexécuter.	Effectuez les modifications telles que la simplification des programmes, l'allongement de la période ou l'augmentation de la priorité relative.
6	3	Superposition de tâches d'événement. La tâche d'événement ne s'est pas terminée avant que ce soit le moment de la réexécuter.	Effectuez les modifications telles que la simplification des programmes, l'allongement de la période, l'augmentation de la priorité relative ou le ralentissement de l'événement de déclenchement.
7	49	Lorsque l'automate charge un projet à partir d'une mémoire non volatile, il enregistre ce défaut mineur et définit l'objet FaultLog, l'attribut MinorFaultBits et le bit 7.	Effacez le défaut.
9	0	Erreur inconnue lors du traitement du port série.	Contactez le support technique de Rockwell Automation si le problème persiste.
9	1	La ligne CTS n'est pas correcte pour la configuration actuelle.	Débranchez et reconnectez le câble du port série à l'automate. Vérifiez que le câblage est correct.

9	2	Erreur de la liste d'appel. Un défaut a été détecté dans la liste d'appel du maître DF1 (par exemple, plus de stations que la taille de fichier spécifiées, plus de 255 stations spécifiées, tentative d'indexation au-delà de la fin de la liste, ou appel de l'adresse de diffusion (STN N°255)).	Vérifiez les erreurs suivantes : <ul style="list-style-type: none"> <li>• Le nombre total de stations est supérieur à l'espace disponible dans l'étiquette de la liste d'appel.</li> <li>• Le nombre total de stations est supérieur à 255.</li> <li>• Le pointeur de la station en cours est supérieur à la fin de l'étiquette de la liste d'appel.</li> <li>• Un nombre de stations supérieur à 254 a été trouvé.</li> </ul>
9	3	L'étiquette de la station active maître RS-232 DF1 n'est pas spécifiée.	Spécifiez l'étiquette à utiliser pour l'étiquette de station active dans l'onglet Protocole du port série (Serial Port Protocol) de la boîte de dialogue Propriétés de l'automate (Controller Properties).
9	5	Délai d'expiration d'appel de l'esclave DF1. Le chien de garde d'appel a expiré pour l'esclave. Ce maître n'a pas appelé cet automate dans le délai spécifié.	Déterminez et corrigez le retard d'appel.
9	9	Perte de contact avec le modem. Les lignes de commande DCD ou DSR ne sont pas reçues dans la séquence et/ou l'état approprié.	Connexion modem correcte à l'automate.
9	10	Les données ont été supprimées ou perdues du port série.	Ralentissez le taux auquel l'initiateur envoie les données.
10	10	La batterie n'a pas été détectée ou doit être remplacée.	Installez une nouvelle batterie.
10	11	La batterie du partenaire de sécurité n'est pas détectée ou doit être remplacée.	Installez une nouvelle batterie.
10	12	Le Module de stockage d'énergie (ESM) n'est pas installé. Si l'automate est mis hors tension, la maintenance de l'attribut WallClockTime et du programme n'est pas assurée.	Installez un ESM dans l'automate.
10	13	L'ESM installé n'est pas compatible avec l'automate.	Remplacez l'ESM installé par un ESM compatible avec l'automate.
10	14	L'ESM doit être remplacé en raison d'un défaut matériel. Il n'est pas en mesure d'assurer la maintenance de l'attribut WallClockTime ou du programme de l'automate à la mise hors tension.	Remplacez l'ESM.
10	15	L'ESM ne peut pas stocker suffisamment d'énergie à l'intérieur pour assurer la maintenance de l'attribut WallClockTime ou du programme de l'automate à la mise hors tension.	Remplacez l'ESM.
10	16	La source d'alimentation ininterrompue (UPS) est introuvable ou n'est pas prête.	Soit : <ul style="list-style-type: none"> <li>• Installez l'UPS.</li> <li>• Vérifiez l'UPS pour vous assurer qu'elle est correctement chargée afin de servir de source d'alimentation de secours en cas de coupure de courant.</li> </ul>
10	17	La batterie de l'UPS est défectueuse et doit être remplacée.	Remplacez la batterie de l'UPS.
13	21	Heure locale hors plage.	Assurez-vous que l'heure locale est définie sur la date et l'heure correspondantes.
14	12	Le projet de sécurité est configuré en tant que SIL2/PLd et un partenaire de sécurité est présent.	Assurez-vous qu'aucun partenaire de sécurité n'est installé à droite de l'automate principal.

17	1...n	Échec du diagnostic de l'automate interne.	Contactez le support technique de Rockwell Automation avec le type et le code de défaut.
17	35	La température interne de l'automate approche de la limite de fonctionnement.	Il convient de prendre des mesures pour réduire la température ambiante du module. Respectez les limites recommandées pour la température ambiante (entrée) et appliquez les dégagements requis autour du châssis.
17	36	Absence de ventilateur ou celui-ci ne conserve pas la vitesse souhaitée.	Remplacez le ventilateur.
19	4	Défaut du port Ethernet	Détection d'une tempête de données EtherNet/IP. Examinez le trafic réseau sur le port Ethernet et effacez le défaut. Si le problème persiste, contactez le support technique de Rockwell Automation pour obtenir de l'aide.
20	1	Une licence requise est introuvable ou a expiré, alors que l'automate est en mode exécution ou en mode test.	Insérez une CmCard contenant toutes les licences requises par le projet dans l'automate.

Mots-clés : fault code:2, fault codes:2, faults:2Mots-clés : défauts:5

## Codes d'erreur des messages

Les codes d'erreur dépendent du type d'instruction MSG.

Voir aussi

[Codes d'erreur](#) sur la [page 172](#)

[Codes d'erreur étendus](#) sur la [page 174](#)

[Codes d'erreur PLC et SLC \(.ERR\)](#) sur la [page 176](#)

[Codes d'erreur des blocs-transferts](#) sur la [page 178](#)

## Codes d'erreur

L'application Logix Designer n'affiche pas toujours la description complète.

Code d'erreur (hexa)	Description	Affichage dans le logiciel	
0001	Echec de la connexion (codes d'erreur étendus)	Identique à la description	
0002	Ressource insuffisante		
0003	Valeur non valide		
0004	Erreur de syntaxe IOI (voir les codes d'erreur étendus)		
0005	Destination inconnue, classe non prise en charge, instance non définie ou élément de structure non défini (voir les codes d'erreur étendus)		
0006	Espace de trame insuffisant		
0007	Connexion perdue		
0008	Service non pris en charge		
0009	Erreur dans un segment de données ou valeur d'attribut incorrecte		
000A	Erreur dans la liste d'attributs		
000B	Cet état existe déjà		
000C	Conflit entre les modèles objet		
000D	L'objet existe déjà		
000E	Impossible de définir l'attribut		
000F	Autorisation refusée		
0010	Conflit entre les états du dispositif		
0011	La réponse ne convient pas		
0012	Fragmenter primitive		
0013	Données de commande insuffisantes		
0014	Attribut non pris en charge		
0015	Trop de données		
001A	Demande de pont trop longue		
001B	Réponse de pont trop longue		
001C	Liste d'attributs manquante		
001D	Liste d'attributs incorrecte		Identique à la description
001E	Erreur de service imbriqué		
001F	Pannes liée à la connexion (codes d'erreur étendus)		
0022	Réponse reçue non valide		
0025	Erreur de segment clé		
0026	Erreur d'IOI incorrect		
0027	Attribut inattendu dans la liste		
0028	Erreur DeviceNet - ID de membre incorrect		
0029	Erreur DeviceNet - membre impossible à définir		
00D1	Module pas en état d'exécution	Erreur inconnue	
00FB	Port de message non pris en charge		
00FC	Type de données de message non pris en charge		
00FD	Message non initialisé		
00FE	Délai d'expiration du message		
00FF	Erreur générale (voir les codes d'erreur étendus)		

## Codes d'erreur étendus

L'application Logix Designer n'affiche pas de texte pour les codes d'erreur étendus.

Ci-après les codes d'erreur étendus du code d'erreur 0001.

Code d'erreur étendu (hexa)	Description
0100	Connexion en cours d'utilisation
0103	Transport non pris en charge
0106	Conflit de propriété
0107	Connexion introuvable
0108	Type de connexion non valide
0109	Taille de connexion non valide
0110	Module non configuré
0111	EPR non pris en charge
0113	Echec d'écriture de MSG
0114	Module incorrect
0115	Type de dispositif incorrect
0116	Révision incorrecte
0118	Format de configuration non valide
011A	Plus de connexions disponibles pour l'application
0203	Délai d'expiration de la connexion
0204	Non connecté - Délai d'expiration du message
0205	Non connecté - Erreur d'envoi de paramètres
0206	Message trop long
0301	Pas de mémoire tampon
0302	Largeur de bande non disponible
0303	Aucun écran disponible
0305	Signature non correspondante
0311	Port non disponible
0312	Adresse de liaison non disponible
0315	Type de segment incorrect
0317	Connexion non scrutée

Ci-après les codes d'erreur étendus du code d'erreur 001F.

Code d'erreur étendu (hexa)	Description
0203	Délai d'expiration de la connexion

Ci-après les codes d'erreur étendus des codes d'erreur 0004 et 0005.

Code d'erreur étendu (hexa)	Description
0000	statut étendu Mémoire insuffisante
0001	statut étendu Instances insuffisantes

Ci-après les codes d'erreur étendus du code d'erreur 00FF.

Code d'erreur étendu (hexa)	Description
2001	IOI excessif
2002	Valeur de paramètre incorrecte
2018	Rejet du sémaphore
201B	Taille trop petite
201C	Taille non valide
2100	Echec de privilège
2101	Position incorrecte de l'interrupteur à clé
2102	Mot de passe incorrect
2103	Aucun mot de passe émis
2104	Adresse hors limites
2105	Adresse et nombre de hors limite
2106	Données en cours d'utilisation
2107	Le type est incorrect ou non pris en charge
2108	Automate en mode de transfert ou de téléchargement
2109	Tentative de modification du nombre de dimensions du tableau
210A	Nom de symbole incorrect
210B	Ce symbole n'existe pas
210E	Echec de la recherche
210F	La tâche ne peut pas démarrer
2110	Impossible d'écrire
2111	Impossible de lire
2112	Routine partagée non modifiable
2113	Automate en mode Défaut
2114	Mode Exécution interdit

## Codes d'erreur PLC et SLC (.ERR)

Les révisions du firmware Logix 10.x et ultérieures fournissent de nouveaux codes d'erreur pour les erreurs associées aux types de message PLC et SLC™ (messages PCCC).

Ces modifications permettent au logiciel RSLogix 5000 d'afficher une description plus pertinente pour de nombreuses erreurs. Auparavant, le logiciel ne donnait pas de description pour les erreurs associées au code d'erreur 00F0.

Par ailleurs, ces modifications rendent les codes d'erreur plus cohérents avec les erreurs renvoyées par les autres Controllers, tels que les Controllers PLC-5®.

Le tableau suivant affiche les modifications apportées aux codes d'erreur entre les versions R9.x et antérieures et les versions R10.x et ultérieures. Suite à ces modifications, le membre .ERR renvoie une valeur unique pour chaque erreur PCCC. Le .EXERR n'est plus requis pour ces erreurs.

### Codes d'erreur PLC et SLC (hexa)

Versions R9.x et antérieures		Versions R10.x et ultérieures		Description
.ERR	.EXERR	.ERR	.EXERR	
0010		1000		Commande ou format illégal(e) par le processeur local
0020		2000		Le module de communication ne fonctionne pas
0030		3000		La station distante n'existe pas, est déconnectée ou arrêtée
0040		4000		Processeur connecté mais en défaut (matériel)
0050		5000		Numéro de station erroné
0060		6000		La fonction requise n'est pas disponible
0070		7000		Le processeur est en mode Programme
0080		8000		Le fichier de compatibilité du processeur n'existe pas
0090		9000		La station distante ne peut pas mémoriser la commande
00B0		B000		Le processeur est en transfert donc il n'est pas accessible
00F0	0001	F001		Le processeur a converti l'adresse incorrectement
00F0	0002	F002		Adresse incomplète
00F0	0003	F003		Adresse incorrecte
00F0	0004	F004		Format d'adresse illégal - symbole introuvable
00F0	0005	F005		Format d'adresse illégal - le symbole présente 0 caractère ou un nombre de caractères supérieur au nombre maximal pris en charge par le dispositif
00F0	0006	F006		Le fichier d'adresses n'existe pas dans le processeur cible
00F0	0007	F007		Le fichier de destination est trop petit pour le nombre de mots demandé
00F0	0008	F008		Impossible de terminer la demande Situation modifiée lors de l'opération à paquets multiples



00F0	0009	F009		Les données ou le fichier sont trop grands Mémoire non disponible
00F0	000A	F00A		Le processeur cible ne peut pas mettre les informations requises en paquets
00F0	000B	F00B		Erreur de privilège ; accès refusé
00F0	000C	F00C		La fonction requise n'est pas disponible
00F0	000D	F00D		La demande est redondante
00F0	000E	F00E		La commande ne peut pas être exécutée
00F0	000F	F00F		Dépassement ; dépassement d'histogramme
00F0	0010	F010		Pas d'accès
00F0	0011	F011		Le type de données demandé ne correspond pas aux données disponibles
00F0	0012	F012		Paramètres de commande incorrects
00F0	0013	F013		La référence de l'adresse existe dans une zone effacée
00F0	0014	F014		Echec de l'exécution de la commande pour une raison inconnue Dépassement d'histogramme PLC-3®
00F0	0015	F015		Erreur de conversion de donnée
00F0	0016	F016		Le scrutateur n'est pas disponible pour communiquer avec un adaptateur de châssis 1771
00F0	0017	F017		L'adaptateur n'est pas disponible pour communiquer avec le module
00F0	0018	F018		La réponse du module 1771 est incorrecte
00F0	0019	F019		Etiquette dupliquée
00F0	001A	F01A		Propriétaire du fichier actif - le fichier est en cours d'utilisation
00F0	001B	F01B		Propriétaire du Program actif - quelqu'un effectue un chargement ou une modification en ligne
00F0	001C	F01C		Le fichier disque est protégé en écriture ou autrement inaccessible (hors ligne seule)
00F0	001D	F01D		Le fichier disque est en cours d'utilisation par une autre application Mise à jour non effectuée (hors ligne uniquement)

## Codes d'erreur des blocs-transferts

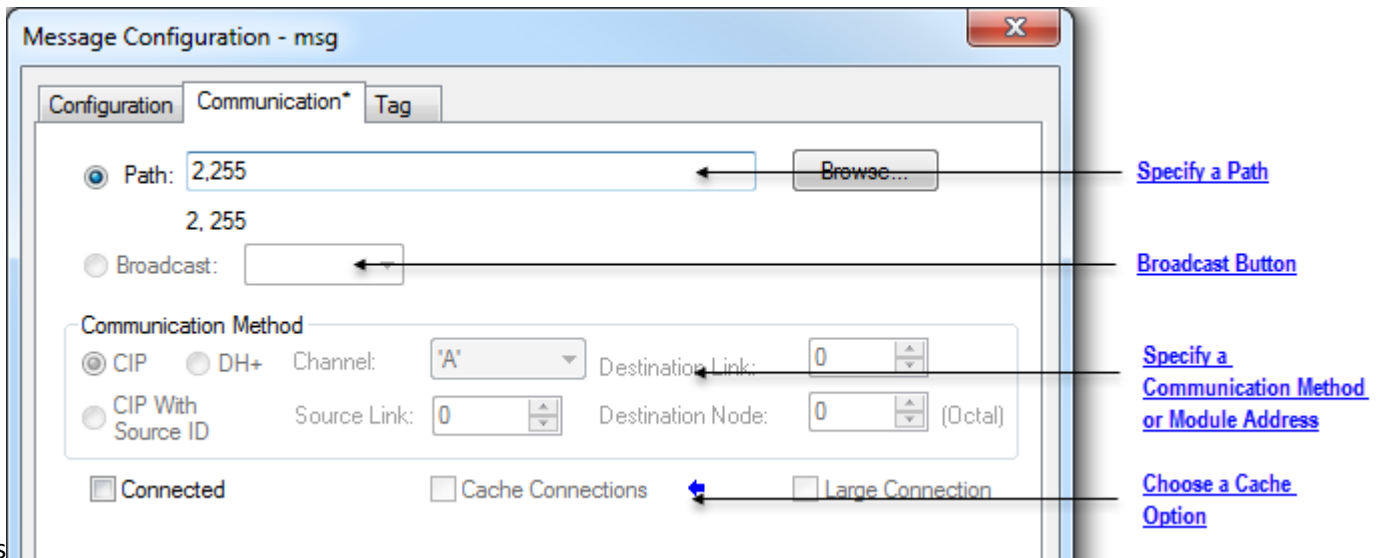
Il s'agit des codes d'erreur spécifiques des blocs-transferts de LOGIX 5000.

Code d'erreur (hexa)	Description	Affichage dans le logiciel
00D0	Le scrutateur n'a pas reçu de réponse de bloc-transfert du module de bloc-transfert dans les 3,5 secondes suivant la demande.	Erreur inconnue
00D1	Le contrôle global de la réponse de lecture ne correspondait pas au contrôle global du flux de données.	
00D2	Le scrutateur a demandé une lecture ou une écriture, mais le module de bloc-transfert a répondu par le contraire.	
00D3	Le scrutateur a demandé une certaine longueur et le module de bloc-transfert a répondu avec une autre longueur.	
00D6	Le scrutateur a reçu une réponse du module de bloc-transfert indiquant que la demande d'écriture a échoué.	
00EA	Le scrutateur n'a pas été configuré pour communiquer avec le rack contenant ce module de bloc- transfert.	
00EB	L'emplacement logique spécifié n'est pas disponible pour la taille de rack donnée.	
00EC	Une demande de bloc-transfert est actuellement en cours et une réponse est requise avant qu'une autre demande ne puisse démarrer.	
00ED	La taille de la demande de bloc-transfert ne correspond pas à des tailles de demandes de bloc-transfert correctes.	
00EE	Le type de demande de bloc-transfert ne correspond pas à la valeur attendue BT_READ ou BT_WRITE.	
00EF	Le scrutateur n'a pas été en mesure de trouver un emplacement disponible dans le tableau de bloc-transfert pour accepter la demande de bloc-transfert.	
00F0	Le scrutateur a reçu une demande de réinitialisation des voies E/S à distance alors que des blocs-transferts étaient en attente.	
00F3	Les files d'attente pour les blocs-transferts à distance sont pleines.	
00F5	Aucune voie de communication n'est configurée pour le rack ou l'emplacement demandé.	
00F6	Aucune voie de communication n'est configurée pour les E/S à distance.	
00F7	Le délai d'expiration du bloc-transfert, défini dans l'instruction, est arrivé à terme avant la fin de l'opération	
00F8	Erreur dans le protocole de bloc-transfert - bloc-transfert non sollicité.	
00F9	Des données de bloc-transfert ont été perdues à cause d'une mauvaise voie de communication.	
00FA	Le module de bloc-transfert a demandé une longueur différente de celle associée à l'instruction de bloc-transfert.	
00FB	Le contrôle global des données de lecture du bloc-transfert était incorrect.	
00FC	Un transfert de données d'écriture de bloc-transfert était incorrect entre l'adaptateur et le module de bloc-transfert.	
00FD	La somme de la taille du bloc-transfert et de la taille de l'index dans la table de données du bloc-transfert était supérieure à la taille du fichier de table de données du bloc-transfert.	

## Spécifier les détails de communication

Configurer une diffusion dans des programmes en logique à relais ou de texte structuré. Dans les diagrammes logiques à relais, ajouter un échelon et cliquer sur la propriété de **MSG** pour accéder à la boîte de dialogue **Configuration du message** (Message Configuration) et définir un nouveau message. Dans un texte structuré, entrer **MSG** (aMsg), puis cliquer avec le bouton droit sur aMsg pour accéder à la boîte de dialogue **Configuration du message** (Message Configuration) et configurer le message.

Pour configurer une instruction MSG, spécifier les éléments suivants dans l'onglet **Communication** (Communication) :



### Spécifier un chemin

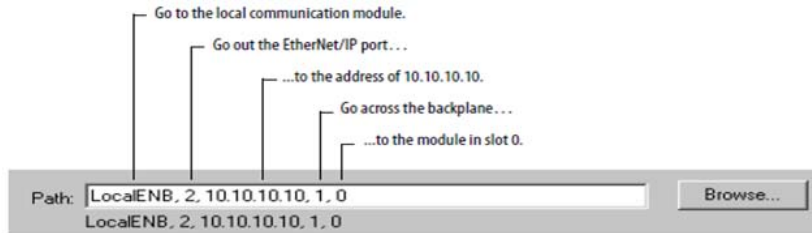
Le chemin montre le parcours du message pour arriver à destination. Il utilise des noms de configuration des E/S de l'automate, des nombres que vous entrez, ou les deux. Vous pouvez définir ce chemin comme chemin par défaut avec le bouton Diffusion (broadcast), qui peut être activé avec le protocole de système et le type de message.

Si	Alors
La configuration des E/S de l'automate dispose du module qui récupère le message.	Rechercher pour sélectionner le module.
La configuration des E/S de l'automate dispose uniquement du module de communication local.	Rechercher pour sélectionner le module de communication local et saisir le reste du chemin.
La configuration des E/S de l'automate ne dispose d'aucun des modules nécessaires pour le message.	Saisir le chemin.

**Astuce :** THIS est également admis, ce qui indique le chemin réflexif. THIS est utilisé pour envoyer un message non connecté à l'automate.

**Exemples**

La configuration des E/S de l'automate dispose uniquement du module de communication local :



Pour saisir un chemin, utiliser le format :

port, adresse suivante, port, adresse suivante,

Où	Est	
	Pour ce réseau	Type (Type)
Port	Fond de panier	1
	DF1 (série, canal série 0)	2
	ControlNet	
	EtherNet/IP	
	Canal DH+ A	3
	Canal DH+ B	
	DF1 Canal 1 (canal série 1)	
Adresse suivant	Fond de panier	Numéro de logement du module
	DF1 (série)	Adresse de station (0-254)
	ControlNet	Numéro de station (1-99 décimal)
	DH+	8# suivi par le numéro de station (1-77 octal) Par exemple, pour spécifier l'adresse de station octal de 37, saisir 8#37.
	EtherNet/IP	Spécifier un module sur un réseau EtherNet/IP en utilisant l'un de ces formats au choix : <ul style="list-style-type: none"> <li>• Adresse IP. Par exemple, 10.10.10.10</li> <li>• Adresse IP:Port. Par exemple, 10.10.10.10:24</li> <li>• Nom DNS. Par exemple, tanks</li> <li>• Nom DNS:Port. Par exemple, tanks:24</li> </ul>

### Bouton Diffusion (Broadcast)

Le bouton **Diffusion** (Broadcast) est utilisé avec le port série.

- Cette fonctionnalité du logiciel RSLogix 5000, à partir de la version 18, améliore la capacité à définir une route et un type de message, nécessaires pour envoyer un message à sa destination.

Le bouton **Diffusion** (Broadcast) lorsqu'il est activé, vous permet de définir le chemin comme étant par défaut en sélectionnant un canal disponible dans la zone de liste déroulante. Le nombre de canaux de la zone de liste déroulante dépend de l'automate en cours.

Par défaut, le bouton **Chemin** (Path) de l'onglet **Communication** (Communication) est actif.

Procéder aux étapes suivantes pour activer le bouton **Diffusion** (Broadcast) et sélectionner un canal pour définir un chemin par défaut pour le message.

1. Dans l'**Organisateur de l'automate** (Controller Organizer), cliquer avec le bouton droit de la souris sur **Automate** (Controller) et choisir **Propriétés** (Properties). La boîte de dialogue **Propriétés de l'automate** (Controller Properties) s'affiche.
2. Cliquer sur l'onglet **Protocole système** (System Protocol).
3. Sélectionner **DF1 Master** (DF1 Master) dans la boîte de dialogue **Protocoles** (Protocol). Le Mode d'appel par défaut est « Basé sur les messages » (un esclave peut initier des messages).
4. Cliquer sur **OK**.
5. Dans le diagramme logique à relais, cliquer sur la case à l'intérieur de l'étiquette MSG. La boîte de dialogue **Configuration du message** (Message Configuration) s'ouvre et affiche l'onglet **Configuration** (Configuration).
6. Dans la boîte de dialogue **Type de message** (Message Type), sélectionner **Écriture de la table de données CIP** (CIP Data Table Write).
7. Cliquer sur **OK**. Vous avez activé le bouton **Diffusion** (Broadcast) dans l'onglet **Communication** (Communication).
8. Cliquer sur l'onglet **Communication** (Communication).
9. À côté du bouton **Diffusion** (Broadcast), sélectionner un canal dans la zone de liste déroulante. Le nombre de canaux de la zone de liste déroulante dépend de l'automate.

Lorsque vous sélectionnez le canal 0 ou 1, le chemin du message correspondant dans la boîte de dialogue **Configuration de message** (Message Configuration) prend par défaut la valeur 2 255 (canal 0) ou la valeur 3 255 (canal 1). Le Chemin (Path) devient grisé afin de ne pas vous permettre de saisir un chemin manuellement.

10. Cliquer sur **OK**.

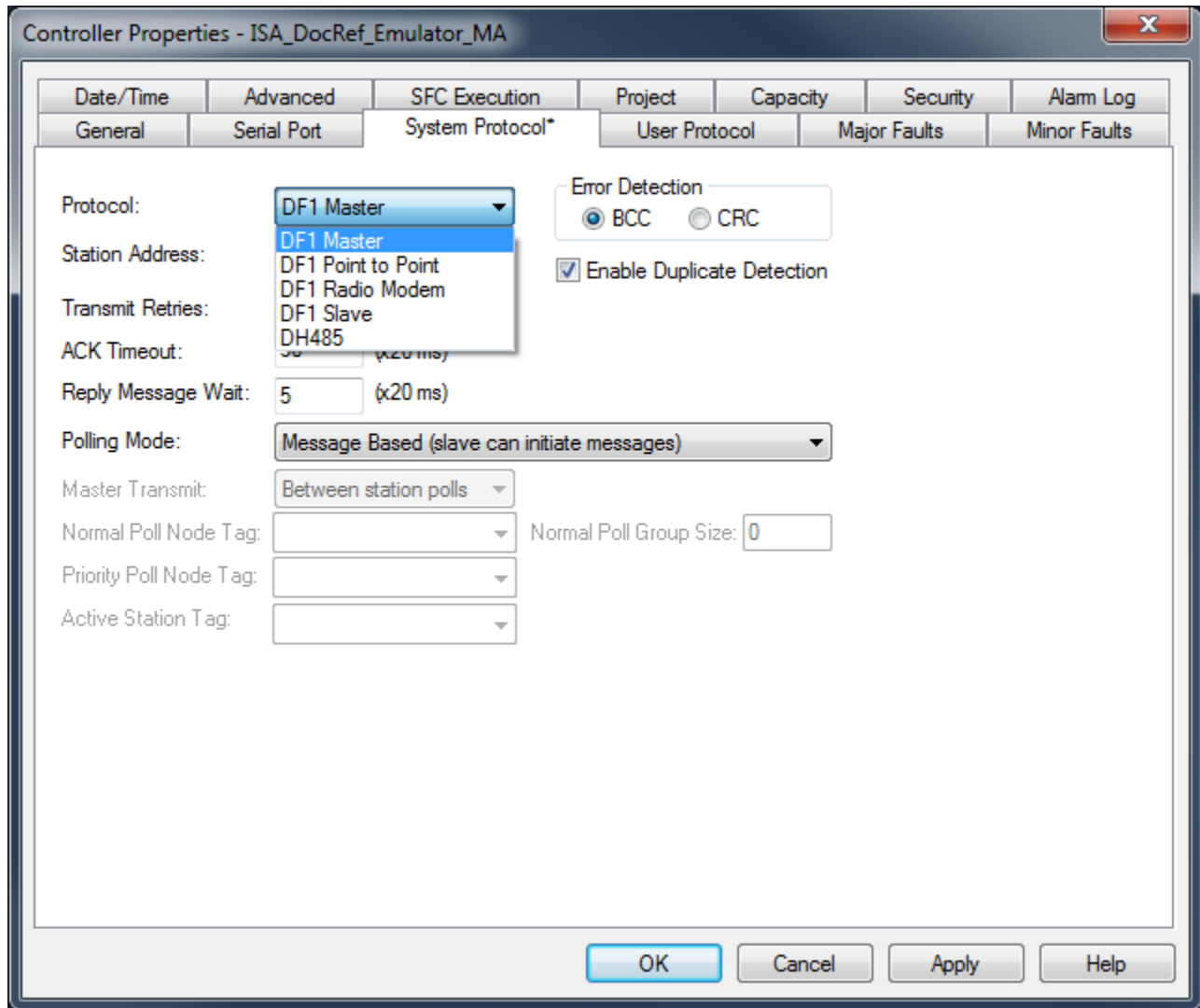
### **Configuration de l'onglet Protocole système (System Protocol)**

Pour lancer la diffusion dans les automates ControlLogix de l'application Logix Designer, vous devez configurer l'onglet **Protocole Système** (System Protocol) dans la boîte de dialogue **Propriétés de l'automate** (Controller Properties). Le protocole doit être compatible avec le type de message d'écriture dans la boîte de dialogue **Configuration de message** (Message Configuration).

Suivez ces étapes pour mettre en place un protocole système qui soit compatible avec la fonctionnalité de diffusion.

1. Créer ou ouvrir un automate existant dans l'application.
2. Dans l'**Organisateur de l'automate** (Controller Organizer), cliquer avec le bouton droit de la souris sur le nom de l'automate et sélectionner **Propriétés** (Properties). La boîte de dialogue **Propriétés de l'automate** (Controller Properties) s'affiche.

3. Si votre automate a un port série, cliquer sur l'onglet **Protocole système** (System Protocol).



4. Dans la boîte Protocole (Protocol), sélectionner un protocole.

---

**IMPORTANT :** La case **Type de message** (Message Type) dans la boîte de dialogue **Onglet de Configuration du Message** (Message Configuration Tab) doit être écrite par pour être compatible avec le protocole système. Autrement, le bouton **Diffusion** (Broadcast) est désactivé.

---

5. Saisir l'information dans l'onglet **Protocole système** (System Protocol) pour chaque protocole décrit dans les tableaux suivants.

Rubrique	Description (Description)
Protocol	DF-1 Maître

Adresse de station (Station Address)	Saisir le numéro d'adresse de station de l'automate
Nb de réessais de transmission (Transmit Retries)	3
Délai d'expiration ACK (ACK Timeout)	50
Attente message réponse (Reply Message Wait)	5
Mode d'appel (Polling Mode)	Sélectionner l'un des modes suivants : <ul style="list-style-type: none"> <li>• <b>Basé sur message</b> (Message Based) Fait appel à l'esclave en utilisant l'instruction de message</li> <li>• <b>L'esclave peut initier les messages</b> (Slave can initiate message) pour la diffusion d'esclave à esclave</li> <li>• <b>Standard.</b> pour que la planification interroge l'esclave</li> </ul>
Détection d'erreur (Error Detection)	BCC
Détection de doublons (Duplicate Detection)	Activé (coché)
<b>Rubrique</b>	<b>Description (Description)</b>
Protocol	DF-1 Esclave
Adresse de station (Station Address)	Saisir le numéro d'adresse de station de l'automate
Nb de réessais de transmission (Transmit Retries)	3
Délai d'expiration d'appel esclave (Slave Poll Timeout)	3000
Suppression EOT (EOT Suppression)	Désactivé (non coché)
Détection d'erreur (Error Detection)	BCC
Détection de doublons (Duplicate Detection)	Activé (coché)
<b>Rubrique</b>	<b>Description (Description)</b>
Protocol	DF-1 Esclave
Adresse de station (Station Address)	Saisir le numéro d'adresse de station de l'automate
Activer le stockage et la transmission (Enable Store and Forward)	Activer la case à cocher (coché) pour utiliser l'étiquette stocker et transmettre
Détection d'erreur (Error Detection)	BCC

6. Cliquer sur **OK**.



### Pour transfert par bloc

Pour les messages de bloc-transfert, ajouter les modules suivants aux E/S de configuration de l'automate :

Pour les messages de transfert par bloc dans ce réseau :	Ajouter ces modules à la configuration des E/S :
ControlNet	Module de communication local (par exemple, module 1756-CNB) Module d'adaptateur à distance (par exemple, module 1771-ACN)
I/O à distance universel	Module de communication local (par exemple, module 1756-DHRIO) Un module d'adaptateur à distance (par exemple, module 1771-ASB) pour chaque rack ou portion de rack dans le châssis Module de transfert par bloc (facultatif)

### Spécifier une Méthode de communication ou une Adresse de module

Utiliser les tableaux suivants pour sélectionner une méthode de communication ou une adresse de module pour le message :

Si le dispositif destinataire est	Choisir	Et spécifier	
Automate LOGIX 5000	CIP	Pas d'autres spécifications requises.	
Automate PLC-5 sur un réseau EtherNet/IP			
Automate PLC-5 sur un réseau ControlNet			
Automate SLC 5/05			
Automate PLC-5 sur un réseau DH+	DH+	Canal	Canal A ou B du module 1756-DHRIO connecté au réseau DH+.
Automate SLC sur un réseau DH+		Liaison source	L'ID de la liaison est affecté au fond de panier de l'automate dans la table de routage du module 1756-DHRIO. La station source de la table de routage est automatiquement le numéro de logement de l'automate.
Processeur PLC-3		Liaison destinataire	L'ID de la liaison DH+ à distance dans laquelle réside le dispositif cible.
Processeur PLC-2		Station destinataire	Adresse de station du dispositif cible, en octal.

		S'il n'existe qu'une seule liaison DH+ et que vous n'avez pas utilisé le logiciel RSLinx Classic pour configurer le module DH/RIO pour des liaison destinataire, spécifier 0 comme liaisons source et destinataire.	
Une application sur une station de travail qui reçoit un message non sollicité routé vers un réseau EtherNet/IP ou ControlNet à travers le logiciel RSLinx Classic ou FactoryTalk Linx	CIP avec ID de la source  Ceci permet à l'application de recevoir des données de l'automate.	Liaison source	L'ID à distance de la rubrique dans le logiciel RSLinx Classic ou le raccourci dans FactoryTalk Linx.
		Liaison destinataire	L'ID de liaison virtuelle configuré dans le logiciel RSLinx Classic ou FactoryTalk Linx (0...65535).
		Station destinataire	L'ID de la destination (0...77 octal) fourni par l'application au logiciel RSLinx Classic ou FactoryTalk Linx. Pour la rubrique DDE du logiciel RSLinx Classic, utilisez 77.
		Le numéro de logement de l'automate ControlLogix est utilisé comme Station source.	
Module de transfert par bloc dans un réseau d'I/O à distance universel	RIO	Canal	Canal A ou B du module 1756-DHRIO connecté au réseau RIO.
		Rack	Numéro du rack (octal) du module.
		Groupe (Group)	Numéro de groupe du module.
		Logement	Numéro de logement du module.
Module de transfert par bloc dans un réseau ControlNet	ControlNet	Logement	Numéro de logement du module.

**Choisir une option de cache**

En fonction de la configuration de l'instruction MSG, il peut utiliser une connexion pour envoyer ou recevoir des données.

Type de message :	Méthode de communication :	Utiliser une connexion :
Lecture ou écriture de la table de données CIP		Au choix (1)
PLC-2, PLC-3, PLC-5, ou SLC (ou tous les types)	CIP CIP avec ID de la source	
	DH+	X
CIP Générique		Au choix (2)
Lecture ou écriture de transfert par bloc		X

1. Les messages de lecture ou d'écriture de la table de données CIP peuvent être connectés ou déconnectés. Pour la plupart des applications, Rockwell Automation recommande de laisser les messages de lecture ou d'écriture de la table de données CIP en connectés.
2. Les messagerie générique CIP peuvent être connectés ou déconnectés. Mais, pour la plupart des applications, nous vous recommandons de laisser les messages génériques CIP déconnectés.

Si une instruction MSG utilise une connexion, vous avez le choix de laisser la connexion ouverte (cache) ou de fermer la connexion une fois le message transmis.

Si vous :	Alors :
Mettre la connexion en cache	La connexion reste ouverte après que l'instruction MSG ait été exécutée. Ceci optimise le temps d'exécution. Le fait d'ouvrir la connexion à chaque fois qu'un message s'exécute augmente le temps d'exécution.
Ne pas mettre la connexion en cache	La connexion se ferme après que l'instruction MSG ait été exécutée. Cela libère cette connexion pour d'autres usages.

L'automate a un certain nombre de limitations pour le nombre de connexions que vous pouvez mettre en cache.

Si vous avez cet automate :	Alors vous pouvez mettre en cache :
CompactLogix 5370 ou ControlLogix 5570	Jusqu'à 32 connexions.
ControlLogix 5580	Jusqu'à 256 connexions.

Si plusieurs messages sont envoyés au même dispositif, les messages peuvent partager une même connexion.

Si les instructions MSG sont à destination de :	Et qu'ils sont :	Alors :
Différents dispositifs		Chaque instruction MSG utilise 1 connexion.
Un même dispositif	Activé simultanément	Chaque instruction MSG utilise 1 connexion.
	PAS activés simultanément	L'instruction MSG utilise 1 connexion et 1 tampon de cache. Ils partagent la connexion et le tampon.

**Astuce :** Pour partager une connexion, si l'automate alterne entre l'envoi d'un message de lecture de bloc-transfert et d'écriture de bloc-transfert au même module, les deux messages comptent comme une seule connexion. La mise en cache des deux messages compte comme un dans la liste de cache.

## Recommandations

Lorsque vous planifiez et que vous programmez vos instructions MSG, suivez ces lignes directrices :

Recommandation	Détails
Pour chaque instruction MSG, créer une étiquette de contrôle.	Chaque instruction MSG requiert sa propre étiquette de contrôle.
	Type de données = MESSAGE
	Niveau d'accès = automate
	L'étiquette ne peut pas faire partie d'un tableau ou d'un type de données défini par l'utilisateur.
Garder les données de source et/ou de destination au niveau d'accès de l'automate.	Une instruction MSG peut uniquement accéder aux étiquettes qui sont dans le dossier Étiquettes de l'automate (Controller Tags) (niveau d'accès de l'automate).
Si votre MSG est pour un dispositif qui utilise des entiers de 16 bits, utiliser un tampon de INT dans le MSG et de DINT tout au long du projet.	Si votre message est pour un dispositif qui utilise des entiers de 16 bits, comme automate PLC-5 ou SLC 500, et qu'il transfère des entiers (pas REAL), utiliser un tampon de INT dans le message et de DINT tout au long du projet.
	Cela augmente l'efficacité de votre projet car les automates Logix exécutent plus efficacement en utilisant moins de mémoire lorsqu'ils travaillent avec des entiers de 32 bits (DINT).
	Pour effectuer la conversion entre INT et DINT, reportez-vous au document <a href="#">Logix 5000 Controllers Common Procedures Programming Manual</a> , publication <a href="#">1756-PM001</a> .
Mettre en cache les MSG connectés et qui sont lancés le plus fréquemment.	Mettre en cache la connexion des instructions MSG qui sont lancés le plus fréquemment, jusqu'au nombre maximal autorisé pour votre révision d'automate. Ceci optimise le temps d'exécution car l'automate n'a donc pas à ouvrir une connexion à chaque fois qu'un message s'exécute.
Pour les automates CompactLogix 5370 ou ControlLogix 5570, si vous souhaitez activer plus de 16 MSG à la fois, utilisez un type de stratégie de gestion. Pour les automates ControlLogix 5580, si vous souhaitez activer plus de 256 MSG à la fois, utilisez un type de stratégie de gestion.	Pour les automates CompactLogix 5370 ou ControlLogix 5570, si vous activez plus de 16 MSG à la fois, certaines instructions MSG peuvent tarder à intégrer la file d'attente. Pour les automates ControlLogix 5580, si vous activez plus de 256 MSG à la fois, certaines instructions MSG peuvent tarder à intégrer la file d'attente. Pour vous aider à vous assurer que chaque message s'exécute, utilisez l'une des options suivantes :
	Activer chaque message de façon séquence.
	Activer les messages par groupe.
	Programmer un message pour communiquer avec des dispositifs multiples. Pour plus d'information, reportez-vous au document <a href="#">LOGIX 5000 Controllers Common Procedures Programming Manual</a> , publication <a href="#">1756-PM001</a> .
	Logique de programme pour coordonner l'exécution des messages. Pour plus d'information, reportez-vous au document <a href="#">LOGIX 5000 Controllers Common Procedures Programming Manual</a> , publication <a href="#">1756-PM001</a> .
(Uniquement pour les automates CompactLogix 5370 ou ControlLogix 5570) Veillez à ce que le	L'automate peut avoir 10...40 tampons non connectés. Le nombre par défaut est 10 pour les automates CompactLogix 5370 ou ControlLogix 5570.

nombre de MSG non connectés et non mis en cache soit inférieur au nombre de tampons non connectés.	Si tous les tampons non connectés sont utilisés lorsqu'une instruction sort de la file d'attente des messages, l'instruction renvoie une erreur et ne transfère pas les données.
	Vous pouvez augmenter le nombre de tampons non connectés (40 max), mais suivez toujours la ligne directrice 5.
	Pour augmenter le nombre de tampons non connectés, reportez-vous au document <a href="#">LOGIX 5000 Controllers Common Procedures Programming Manual</a> , publication <a href="#">1756-PM001</a> .

## Spécification de messages SLC

Utilisez les types de message SLC pour communiquer avec les automates SLC et MicroLogix. Le tableau suivant indique à quels types de données l'instruction vous permet d'accéder. Il précise également le type de données LOGIX 5000 correspondant.

Pour ce type de données SLC ou MicroLogix :	Utilisez ce type de données LOGIX 5000 :
F	REAL
L (automates MicroLogix 1200 et 1500)	DINT
N	INT

## Spécification de messages de bloc-transfert

Les types de messages de bloc-transfert sont utilisés pour communiquer avec les modules de bloc-transfert sur un réseau I/O à distance universel.

Pour :	Sélectionner cette commande :
Lire les données d'un module de bloc-transfert Ce type de message remplace l'instruction BTR	Lecture de transfert par bloc
Écrire les données vers un module de transfert par bloc Ce type de message remplace l'instruction BTW	Écriture de transfert par bloc

Pour configurer un message de transfert par bloc, suivez ces lignes directrices :

- Les étiquettes de source (pour BTW) et de destination (pour BTR) doivent avoir une taille suffisante pour accepter les données demandées, sauf pour les structures MESSAGE, AXIS et MODULE.
- Spécifiez le nombre d'entiers (INT) de 16 bits à envoyer ou recevoir. Vous pouvez spécifier de 0 à 64 entiers.

**Astuce :** Pour que le module de transfert par bloc détermine le nombre d'entiers de 16 bits à envoyer (BTR), ou pour que l'automate envoie 64 entiers (BTW), tapez **0** comme nombre d'éléments.

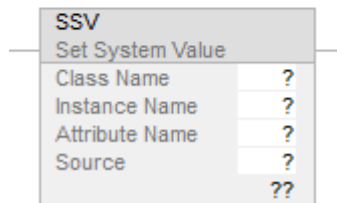
## Lire la valeur système (GSV) et Définir la valeur système (SSV)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

Les instructions GSV/SSV lisent et définissent les données système de l'automate stockées dans des objets.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Ces instructions ne sont pas disponibles dans le bloc fonctionnel.

#### Texte structuré

GSV(ClassName,InstanceName,AttributeName,Dest)

SSV(ClassName,InstanceName,AttributeName,Source)

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à *Conversion de données*.

#### Un diagramme à relais et texte structuré

Opérande	Type (Type)	Format	Description (Description)
Nom de la classe		nom	Le nom de la classe d'objet
Instance name		nom	Le nom de l'objet spécifique, lorsqu'un nom est requis pour l'objet

Nom de l'attribut		nom	L'attribut de l'objet Le type de données dépend de l'attribut que vous sélectionnez
Destination (GSV)	SINT INT DINT REAL structure	étiquette	La destination des données d'attribut
Source (SSV)	SINT INT DINT REAL structure	étiquette	L'étiquette qui contient les données que vous souhaitez copier vers l'attribut

### Description (Description)

Les instructions GSV/SSV lisent et définissent les données d'état de l'automate stockées dans des objets. L'automate stocke les données d'état dans des objets. Il n'y a pas de fichier d'état comme dans le processeur PLC-5.

Lorsqu'elle est vraie, l'instruction GSV extrait les informations spécifiées et les place dans la destination. Lorsqu'elle est vraie, l'instruction SSV définit l'attribut spécifié avec les données de la source.

Lorsque vous entrez une instruction GSV ou SSV, le logiciel de programmation affiche les classes d'objet valides, les noms d'objet et les noms d'attribut de chaque instruction. Pour l'instruction GSV, vous pouvez obtenir les valeurs de tous les attributs. Pour l'instruction SSV, le logiciel affiche uniquement les attributs que vous pouvez définir (SSV).

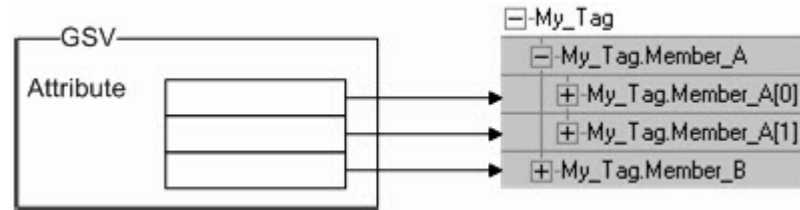


**ATTENTION** : utilisez les instructions SSV avec prudence. Les changements apportés aux objets peuvent déclencher un fonctionnement inattendu de l'automate ou blesser les personnes.

Vous devez tester et vérifier que les instructions ne modifient pas les données que vous ne voulez pas modifier.

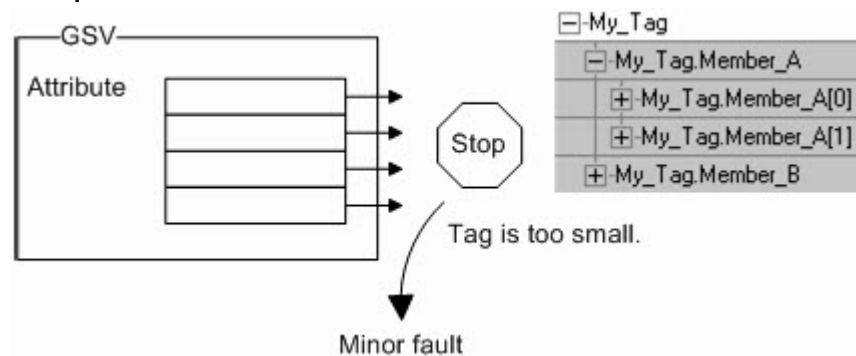
Les instructions SSV et les instructions GSV peuvent écrire et lire les autres membres d'une étiquette en passant par un membre. Si l'étiquette est trop petite, les instructions n'écrivent pas et ne lisent pas les données. À la place, elles enregistrent un défaut mineur dans le journal.

**Exemple 1**



Member\_A est trop petit pour l'attribut. L'instruction GSV écrit donc la dernière valeur dans Member\_B.

**Exemple 2**



My\_Tag est trop petit pour l'attribut. L'instruction GSV s'arrête alors et enregistre un défaut mineur dans le journal. L'étiquette Destination reste inchangée.

Les objets GSV/SSV définissent les attributs de chaque objet et leurs types de données associés. Par exemple, l'attribut MajorFaultRecord de l'objet Program requiert un type de données DINT[11].

**Affecter les indicateurs d'état mathématique**

Nbre.

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
Il y a une adresse d'objet non valide	4	5
L'objet spécifié qui n'est pas compatible avec GSV/SSV	4	6
Il y a un attribut incorrect	4	6
L'instruction SSV ne dispose pas de suffisamment d'informations	4	6



La destination de GSV n'était pas assez grande pour contenir toutes les données demandées	4	7
---	---	---

Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

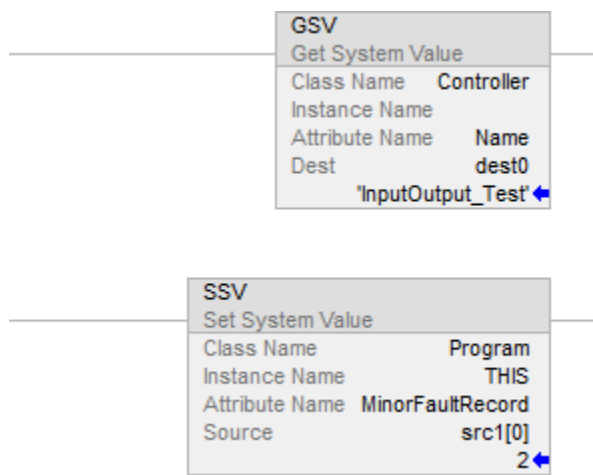
Condition	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau du diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vrai dans le tableau du Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau du diagramme à relais.

**Exemple**

**Diagrammes à relais**



**Texte structuré**

GSV (Program,THIS,LASTSCANTIME,dest1);

```
SSV (Program, THIS, MinorFaultRecord, src[0]);
```

**Voir aussi**

[Conversions de données](#) sur la [page 889](#)

[Attributs communs](#) sur la [page 885](#)

[Objets GSV/SSV](#) sur la [page 210](#)

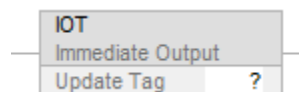
[Objets de Sécurité GSV/SSV](#) sur la [page 263](#)

[Exemple de programmation GSV/SSV](#) sur la [page 206](#)

## Sortie immédiate (IOT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction IOT met à jour immédiatement les données de sortie spécifiées (étiquette de sortie d'un module d'E/S ou d'une étiquette produite). La connexion au module doit être ouverte afin de permettre à l'instruction IOT de s'exécuter.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

IOT (output\_tag)

## Opérandes

### Diagramme à relais

Opérande	TYPE	FORMAT	DESCRIPTION
Update Tag		Étiquette	L'étiquette contient des données que vous souhaitez copier dans l'étiquette d'attribut que vous voulez mettre à jour ; ou bien : Étiquette de sortie d'un module d'E/S ou étiquette produite

### Texte structuré

Les opérandes sont les mêmes que ceux de l'instruction IOT du diagramme à relais.

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

### Description

L'instruction IOT outrepassa l'intervalle entre paquets demandé (RPI) d'une connexion de sortie et envoie les nouvelles données sur la connexion.

Une connexion de sortie est une connexion qui est associée à l'étiquette produite d'un module d'E/S ou à l'étiquette produite. Si la connexion est pour une étiquette produite, l'instruction IOT envoie également le déclencheur d'événements à l'automate de consommation. Cela permet à l'instruction IOT de déclencher une tâche d'événement dans l'automate de consommation.

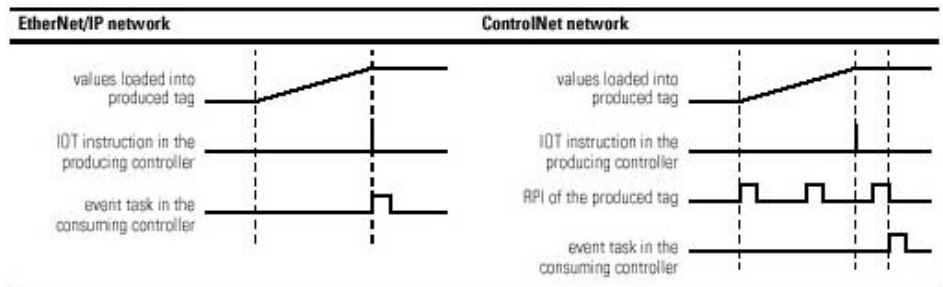
Pour utiliser une instruction IOT et une étiquette produite pour déclencher une tâche d'événement dans un automate de consommation, cocher la case Envoyer déclencheur d'événement au consommateur par programme (instruction IOT) (Programmatically (IOT Instruction) Send Event Trigger to Consumer) dans l'onglet Connexion (Connection) de la boîte de dialogue **Propriétés d'étiquette** (Tag Properties).

**Astuce :** Pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580, au moment du contrôle de 5069 E/S sur un réseau à distance, il est procédé à une optimisation pour grouper les connexions de module configurées avec le même taux RPI dans un seul paquet avant de les envoyer sur le réseau. Si l'instruction IOT est utilisée sur l'une des étiquettes, l'instruction IOT peut provoquer une mise à jour immédiate de certaines étiquettes de données pour d'autres modules configurés avec le même RPI, se trouvent dans le même fond de panier et sont regroupées avec cette étiquette. Si cela n'est pas souhaité, cela peut être évité en faisant un RPI pas tout à fait identique au RPI d'autres connexions de module.

Le type de réseau entre automates définit à quel moment l'automate de consommation reçoit les nouvelles données et le déclencheur d'événements par le biais de l'instruction IOT.

Dans ce réseau	Le dispositif de consommation reçoit les données et le déclencheur d'événements
Fond de panier	Immédiatement
EtherNet/IP	Immédiatement
ControlNet	À l'intérieur de l'intervalle de paquets courant (API) de l'étiquette consommé (connexion)

Les diagrammes suivants comparent la réception des données par le biais d'une instruction IOT sur les réseaux EtherNet/IP et ControlNet.



**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction se met à jour la connexion de l'étiquette spécifiée et réinitialise le temporisateur RPI de la connexion.
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le Diagramme à relais
Post-scrutation	N/A

**Exemple**

Lorsque l'instruction IOT s'exécute, elle envoie immédiatement les valeurs de l'étiquette Local:5:0 au module de la sortie.

**Diagramme à relais****Texte structuré**

IOT (Local:5:0);

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Accéder aux valeurs système

Cette procédure vous aidera à obtenir ou à utiliser l'information d'état de votre automate LOGIX 5000.

Si vous voulez :	Reportez-vous à cette rubrique d'aide :
utilisez des mots-clés spécifiques dans votre logique pour contrôler des événements spécifiques	<a href="#">Contrôle des indicateurs d'état</a> sur la <a href="#">page 269</a>
obtenir ou définir une valeur système	<a href="#">Obtenir et définir une valeur système</a> sur la <a href="#">page 204</a>
Obtenir les informations sur la mémoire de l'automate	<a href="#">Déterminer les informations relatives à la mémoire de l'automate</a> sur la <a href="#">page 198</a>

## Déterminer les informations relatives à la mémoire de l'automate

La mémoire de l'automate est divisée en deux types de mémoire : une mémoire E/S et une mémoire d'extension. Le tableau suivant indique la manière dont l'automate utilise chacun de ces deux types de mémoire :

Cet élément	Utilise ce type de mémoire
Étiquettes E/S	Mémoire E/S
étiquettes produites	
étiquettes consommées	
Communication via des MSG	
Communication avec des stations de travail	
étiquettes autres que les étiquettes E/S, produites ou consommées	mémoire d'extension
routines logiques	
communication avec des étiquettes appelées (OPC/DDE) qui utilisent le logiciel RSLinx Classic.	Mémoire E/S et mémoire d'extension

Notez que l'automate renvoie des valeurs dans le nombre de mots de 32 bits. Pour connaître une valeur en octets, il vous suffit de multiplier par 4.

Utilisez cette procédure pour obtenir les informations suivantes concernant la mémoire de l'automate :

- mémoire d'extension et E/S disponible (libre)
- mémoire d'extension et E/S totale
- bloc continu de mémoire d'extension et E/S maximal

### Obtention des informations relatives à la mémoire depuis l'automate

Pour obtenir des informations relatives à la mémoire depuis l'automate, exécutez une instruction Message (MSG) configurée ainsi :

Dans la boîte de dialogue Propriétés du message, onglet Configuration :

For this item	Type or select	Which means:
Message Type	CIP Generic	Execute a Control and Information Protocol command.
Service Type	Custom	Create a CIP Generic message that is not available in the drop-down list.
Service Code	3	Use the GetAttributeList service. This lets you read specific information about the controller.
Class	72	Get information from the user memory object.
Instance	1	This object contains only 1 instance.
Attribute	0	Null value
Source Element	<i>source_array</i> of type SINT[12]	
	<b>In this element</b>	<b>Enter: Which means:</b>
	<i>source_array</i> [0]	5 Get 5 attributes
	<i>source_array</i> [1]	0 Null value
	<i>source_array</i> [2]	1 Get free memory
	<i>source_array</i> [3]	0 Null value
	<i>source_array</i> [4]	2 Get total memory
	<i>source_array</i> [5]	0 Null value
	<i>source_array</i> [6]	5 Get largest contiguous block of additional free expansion memory
	<i>source_array</i> [7]	0 Null value
	<i>source_array</i> [8]	6 Get largest contiguous block of free I/O memory
	<i>source_array</i> [9]	0 Null value
	<i>source_array</i> [10]	7 Get largest contiguous block of free expansion memory
	<i>source_array</i> [11]	0 Null value
Source Length	12	Write 12 bytes (12 SINTs).
Destination	<i>INT_array</i> of type INT[29]	

Dans la boîte de dialogue Propriétés du message, onglet Communication :

For this item	Type:
Path	1, <i>slot_number_of_controller</i>

### Sélection de l'information de mémoire dont vous avez besoin

L'instruction MSG renvoie les informations suivantes dans *INT\_array* (l'étiquette de destination de l'instruction MSG).

**Important :** Dans le cas de l'automate 1756-L55M16, l'instruction MSG renvoie deux valeurs pour chaque catégorie de mémoire d'extension. Pour déterminer la mémoire d'extension totale ou libre d'un automate 1756-L55M16, ajoutez les deux valeurs pour la catégorie.

If you want the:	Then copy these array elements:	Description:
amount of free I/O memory (32-bit words)	<i>INT_array</i> [3]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [4]	upper 16 bits of the 32 bit value
amount of free expansion memory (32-bit words)	<i>INT_array</i> [5]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [6]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—amount of additional free expansion memory (32-bit words)	<i>INT_array</i> [7]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [8]	upper 16 bits of the 32 bit value
total size of I/O memory (32-bit words)	<i>INT_array</i> [11]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [12]	upper 16 bits of the 32 bit value
total size of expansion memory (32-bit words)	<i>INT_array</i> [13]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [14]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—additional expansion memory (32-bit words)	<i>INT_array</i> [15]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [16]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—largest contiguous block of additional free expansion memory (32-bit words)	<i>INT_array</i> [19]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [20]	upper 16 bits of the 32 bit value
largest contiguous block of free I/O memory (32-bit words)	<i>INT_array</i> [23]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [24]	upper 16 bits of the 32 bit value
largest contiguous block of free expansion memory (32-bit words)	<i>INT_array</i> [27]	lower 16 bits of the 32 bit value
	<i>INT_array</i> [28]	upper 16 bits of the 32 bit value

### Conversion d'INT en DINT

L'instruction MSG renvoie chaque valeur de mémoire sous forme de deux INT distincts.

- Le premier INT représente les 16 bits inférieurs de la valeur.
- Le deuxième INT représente les 16 bits supérieurs de la valeur.

Pour convertir ces deux INT distincts en une seule valeur exploitable, utilisez une instruction Copy (COP) où :

Dans cette opérande :	Spécifier :	Qui signifie :
Source (Source)	premier INT de la paire de 2 éléments (16 bits inférieurs)	Commencer avec les 16 bits inférieurs
Destination	Étiquette DINT dans laquelle stocker la valeur de 32 bits	Copier la valeur dans l'étiquette DINT



Longueur (Length)	1	Copier 1 fois le nombre d'octets dans le type de données de destination. Dans ce cas, l'instruction copie 4 octets (32 bits), qui combinent les 16 bits supérieurs et inférieurs dans une seule valeur de 32 bits.
-------------------	---	--

## Codes d'état de DeviceNet

Vous trouverez ci-dessous les codes d'état de DeviceNet.

Code d'état	Description de l'état	Action recommandée
0-63	Adresse de station DeviceNet pour le scrutateur ou le dispositif esclave.	Aucun.
65	L'option AutoScan est activée et le scrutateur est en mode Attente.	Aucun.
67	Le scrutateur est un scrutateur secondaire.	Aucun.
68	Le scrutateur principal n'a pas détecté de scrutateur secondaire.	Configurez un autre scrutateur comme scrutateur secondaire.
69	Les configurations du scrutateur principal et du scrutateur secondaire ne correspondent pas.	Vérifiez la configuration du scrutateur secondaire.
70	L'adresse du scrutateur est déjà utilisée par un autre dispositif du réseau.	Remplacez l'adresse du scrutateur par une adresse inutilisée.
71	Données non valides dans la liste de scrutation.	Utilisez le logiciel RSNetWorx pour reconfigurer la liste de scrutation.
72	Le dispositif esclave a cessé de communiquer. Si la communication n'est pas rétablie avec le dispositif esclave lors de la tentative suivante, le code d'état devient 78.	<ul style="list-style-type: none"> <li>• Vérifiez les connexions réseau et l'alimentation du dispositif esclave.</li> <li>• Si le dispositif esclave est appelé, vérifiez que le retard de l'interscrutation est suffisant pour que le dispositif puisse renvoyer des données.</li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
73	Les informations d'identité du dispositif esclave ne correspondent pas au détrompage électronique dans le scrutateur.	<ul style="list-style-type: none"> <li>• Assurez-vous que le dispositif esclave correct est connecté à cette adresse.</li> <li>• Assurez-vous que le dispositif esclave correspond au détrompage électronique spécifié (fournisseur, code produit, type de produit).</li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
74	Le scrutateur a détecté un dépassement des données sur le port de communication DeviceNet.	<ul style="list-style-type: none"> <li>• Vérifiez le trafic de communication du réseau.</li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
75	Au moins l'un des deux problèmes suivants est présent. <ul style="list-style-type: none"> <li>• Le scrutateur ne possède pas de liste de scrutation.</li> <li>• Le scrutateur n'a pas reçu de communication en provenance d'un autre dispositif.</li> </ul>	<p>Vérifiez les éléments suivants.</p> <ul style="list-style-type: none"> <li>• Une liste de scrutation a été configurée pour le scrutateur.</li> <li>• Le scrutateur est correctement connecté au réseau.</li> </ul>

76	Pas de trafic réseau direct pour le scrutateur. Le scrutateur entend une autre communication réseau, mais pas une communication qui lui est destinée.	Aucun.
77	Pendant l'initialisation, la taille des données attendue par le dispositif esclave ne correspond pas à celle indiquée dans l'entrée correspondante de la liste de scrutation.	<ul style="list-style-type: none"> <li>• Utilisez le logiciel RSNetWorx pour vérifier que le dispositif esclave et la liste de scrutation contiennent les tailles d'entrée et de sortie correctes pour le dispositif esclave.</li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
78	Le dispositif esclave est configuré dans une liste de scrutation, mais ne communique pas.	<ul style="list-style-type: none"> <li>• Vérifiez les connexions réseau et l'alimentation du dispositif esclave.</li> <li>• Si le dispositif esclave est appelé, assurez-vous que le retard de l'interscrutation est suffisamment long pour qu'il puisse renvoyer ses données.</li> <li>• Au besoin, utilisez le logiciel RSNetWorx pour exécuter les actions suivantes :             <ul style="list-style-type: none"> <li>• Ajoutez le dispositif esclave au réseau DeviceNet</li> <li>• Supprimez le dispositif esclave de la liste de scrutation du scrutateur.</li> <li>• Inhibez le dispositif esclave dans la liste de scrutation du scrutateur.</li> </ul> </li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
79	Le scrutateur n'est pas parvenu à transmettre un message.	<ul style="list-style-type: none"> <li>• Assurez-vous que le scrutateur est connecté à un réseau valide.</li> <li>• Vérifiez que tous les câbles sont correctement connectés.</li> <li>• Vérifiez la vitesse de transmission du réseau.</li> </ul>
80	Le scrutateur est en mode Attente.	<p>Si nécessaire, passez le scrutateur en mode Exécution en procédant ainsi :</p> <ul style="list-style-type: none"> <li>• Placez l'automate en mode Exécution/Exécution à distance à l'aide de l'interrupteur à clé sur l'automate ou via l'application Logix Designer ET</li> <li>• Activez le bit O.CommandRegister.Run pour le scrutateur.</li> </ul>
81	L'automate a placé le scrutateur en mode de défaut.	Le bit O.CommandRegister.Fault pour le scrutateur est activé. Corrigez la condition qui a provoqué l'activation de ce bit par l'automate, puis désactivez ce bit.

82	Erreur détectée dans la séquence de messages E/S fragmentés depuis le dispositif esclave.	<ul style="list-style-type: none"> <li>• Utilisez le logiciel RSNetWorx pour exécuter les actions suivantes. <ul style="list-style-type: none"> <li>• Vérifiez l'entrée de la liste de scrutation pour le dispositif esclave, afin de vous assurer que les tailles de données d'entrée et de sortie sont correctes</li> <li>• Vérifiez la configuration du dispositif esclave</li> </ul> </li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
83	Le dispositif esclave renvoie des réponses d'erreur lorsque le scrutateur tente de communiquer avec lui.	<ul style="list-style-type: none"> <li>• Utilisez le logiciel RSNetWorx pour exécuter les actions suivantes. <ul style="list-style-type: none"> <li>• Vérifiez l'exactitude de la liste de scrutation.</li> <li>• Vérifiez la configuration du dispositif esclave Le dispositif esclave se trouve peut-être dans la liste de scrutation d'un autre scrutateur.</li> </ul> </li> <li>• Éteignez le dispositif esclave, puis rallumez-le.</li> <li>• Vérifiez que le dispositif esclave fonctionne correctement.</li> </ul>
84	Le scrutateur initialise le réseau DeviceNet.	Aucun. Ce code s'efface lorsque le scrutateur tente d'initialiser tous les dispositifs esclaves du réseau.
85	La taille des données envoyée pendant l'exécution par le dispositif esclave ne correspond pas à la taille indiquée dans l'entrée correspondante de la liste de scrutation.	Les données d'appel de longueur variable n'étant pas prises en charge, vérifiez que le dispositif esclave fonctionne correctement.
86	Le dispositif esclave est en mode Attente ou ne génère pas de données pendant que le scrutateur est en mode Exécution.	<ul style="list-style-type: none"> <li>• Vérifiez la configuration et l'état du dispositif esclave.</li> <li>• Si vous configurez une relation maître/esclave entre 2 scrutateurs, assurez-vous que ceux-ci sont en mode Exécution.</li> </ul>
87	Le scrutateur ne peut pas écouter les entrées partagées depuis le dispositif esclave, car le scrutateur propriétaire n'a pas établi de communication avec ce dispositif esclave.	<ul style="list-style-type: none"> <li>• Vérifiez la connexion et la configuration du scrutateur propriétaire.</li> <li>• Le dispositif esclave ne génère peut-être pas de données.</li> </ul>
88	Le scrutateur ne peut pas écouter les entrées partagées depuis le dispositif esclave, car les paramètres E/S (par exemple, appelé ou échantillonné, détrompage électronique, taille des données) pour ce dispositif esclave sont configurés différemment entre ce scrutateur et le scrutateur propriétaire.	Dans ce scrutateur, reconfigurez les paramètres E/S pour l'entrée de la liste de scrutation des entrées partagées, afin que ces paramètres correspondent à ceux du scrutateur propriétaire.
89	Le scrutateur n'est pas parvenu à configurer un dispositif esclave avec les paramètres Récupération automatique de dispositif (ADR).	Vérifiez que vous avez installé un dispositif esclave compatible.

90	L'automate a placé le scrutateur en mode désactivé.	Si nécessaire, activez le scrutateur en désactivant le bit O.CommandRegister.DisableNetwork pour le scrutateur.
91	Condition de bus désactivé probablement due à des erreurs de signal ou de câble.	<ul style="list-style-type: none"> <li>• Éteignez, puis rallumez le scrutateur, les dispositifs esclaves et/ou le réseau.</li> <li>• Vérifiez que tous les dispositifs sont configurés avec la même vitesse de transmission.</li> <li>• Vérifiez le câblage de DeviceNet pour vous assurer qu'il n'y a aucun court-circuit entre les fils CAN (bleu et blanc) et les fils d'alimentation ou le blindage (noir, rouge et blindage).</li> <li>• Vérifiez les sources de bruits suivantes au niveau du système du support : <ul style="list-style-type: none"> <li>• Dispositif situé à proximité d'un câble d'alimentation à haute tension.</li> <li>• Résistance de terminaison utilisée incorrecte, ou aucune résistance de terminaison utilisée.</li> <li>• Mauvaise mise à la terre.</li> </ul> </li> <li>• Dispositif sur un réseau produisant du bruit ou données incorrectes sur le réseau.</li> </ul>
92	Le câble DeviceNet n'alimente pas le port de communication du scrutateur.	<ul style="list-style-type: none"> <li>• Vérifiez que l'alimentation 24 Vcc du réseau fonctionne correctement.</li> <li>• Vérifiez que le câble est en bon état.</li> <li>• Vérifiez les connexions des câbles au scrutateur.</li> </ul>
95	Le firmware du scrutateur est en train d'être mis à jour, ou une configuration est en cours de téléchargement.	Aucun. Ne déconnectez pas le scrutateur pendant la mise à jour, sinon les données présentes dans la mémoire du scrutateur seront perdues.
97	L'automate a placé le scrutateur en mode Interruption.	Le bit O.CommandRegister.HaltScanner pour le scrutateur est activé. Désactivez ce bit, puis éteignez et rallumez le scrutateur.
98	Erreur générale du firmware.	Remplacez le dispositif.
99	Défaillance du système.	Remplacez le dispositif.

## Obtenir et définir une valeur système

L'automate stocke les données système dans des objets. Il n'y a pas de fichiers d'états comme dans l'automate PLC-5. Utilisez l'instruction GSV/SSV pour lire et définir les données système de l'automate stockées dans des objets :

- L'instruction GSV extrait les informations spécifiées et les place dans la destination.
- L'instruction SSV configure l'attribut spécifié avec les données de la source.

Attention : Utilisez l'instruction SSV avec prudence. Les changements apportés aux objets peuvent déclencher un fonctionnement inattendu de l'automate ou blesser les personnes.

Pour lire ou définir une valeur système :

1. Ouvrez le projet d'application Logix Designer.
2. Dans le menu **Aide** (Help), cliquez sur **Contenu** (Contents).
3. Cliquez sur **Index** (Index).
4. Saisissez **gsv/ssv objects**, puis cliquez sur **Affichage** (Display).
5. Cliquez sur l'objet qui vous intéresse.

Pour lire ou définir	Cliquez sur
axe d'un servo-module	AXIS
tranche de temps de surcharge système	CONTROLLER
matériel physique d'un automate	CONTROLLERDEVICE
temps système coordonné pour les dispositifs d'un même châssis	CST
driver de communication DF1 pour le port série (uniquement pour les automates avec ports série)	DF1
historique des défauts d'un automate	FAULTLOG
attributs d'une instruction Message	MESSAGE
état, défauts, chemin de communication et mode d'un module	MODULE
groupe d'axes	MOTIONGROUP
informations de défaut ou temps de scrutation d'un programme	PROGRAM
numéro d'instance d'une routine	ROUTINE
configuration du port série (uniquement pour les automates avec ports série)	SERIALPORT
propriétés ou temps écoulé pour une tâche	TASK
heure locale d'un automate	WALLCLOCKTIME
état de synchronisation temporelle d'un automate	TIMESYNCHRONIZE

6. Dans la liste des attributs pour l'objet, identifiez l'attribut auquel vous voulez accéder.

## 7. Créer une étiquette pour la valeur de l'attribut.

Si le type de données de l'attribut est	Alors
un élément (p. ex., DINT)	Créez une étiquette pour l'attribut.
plusieurs éléments (p. ex., DINT[7])	Créez un type de données personnalisé qui correspond à l'organisation des données utilisée par l'attribut. Ensuite, créez une étiquette pour l'attribut, puis utilisez le type de données que vous avez créé.

## 8. Dans votre routine logique à relais, saisissez l'instruction appropriée.

Pour	Saisissez cette instruction
lire la valeur d'un attribut	GSV
définir la valeur d'un attribut	SSV

## 9. Assigner les opérandes requises à l'instruction.

Reportez-vous à l'instruction GSV/SSV pour en savoir plus sur ces opérandes.

**Voir aussi**

[Lire la valeur système \(GSV\) et Définir la valeur système](#) sur la [page 190](#)

## Exemple de programmation GSV/SSV

Les exemple suivant utilisent l'instruction GSV pour obtenir les informations de défaut.

**Exemple 1 : Obtention d'une information de défaut E/S**

Cet exemple obtient des informations de défaut du module d'E/S disc\_in\_2 et place les données dans la structure disc\_in\_2\_info définie par l'utilisateur.

**Diagramme à relais**



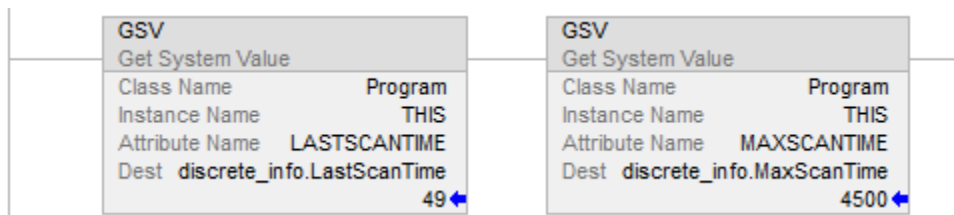
**Texte structuré**

```
GSV(MODULE,disc_in_2,FaultCode,disc_in_2_info.FaultCode);
GSV(MODULE,disc_in_2,FaultInfo,disc_in_2_info.FaultInfo);
GSV(MODULE,disc_in_2,Mode,disc_in_2_info.Mode);
```

**Exemple 2 : Obtention de l'information d'état d'un programme**

Cet exemple obtient des informations d'état sur le programme discret et place les données dans la structure discrete\_info définie par l'utilisateur.

**Diagramme à relais**



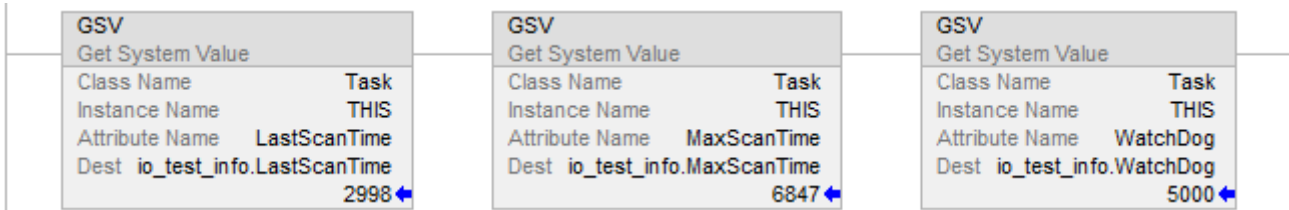
**Texte structuré**

```
GSV(PROGRAM,DISCRETE,LASTSCANTIME,discrete_info.LastScanTime);
GSV(PROGRAM,DISCRETE,MAXSCANTIME,discrete_info.MaxScanTime);
```

### Exemple 3 : Obtention de l'information sur l'état d'une tâche

Cet exemple obtient des informations d'état sur la tâche IO\_test et place les données dans la structure io\_test\_info définie par l'utilisateur.

#### Diagramme à relais



#### Texte structuré

```
GSV(TASK, IO_TEST, LASTSCANTIME, io_test_info.LastScanTime);
GSV(TASK, IO_TEST, MAXSCANTIME, io_test_info.MaxScanTime);
GSV(TASK, IO_TEST, WATCHDOG, io_test_info.Watchdog);
```

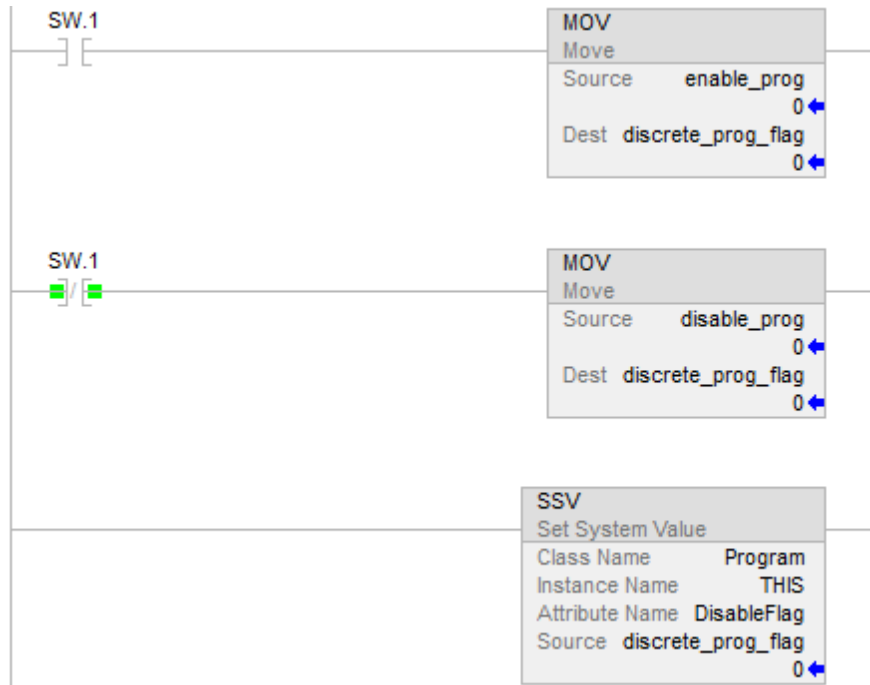
#### Définition d'indicateurs Activé et Désactivé

Les exemples suivants utilisent l'instruction SSV pour activer ou désactiver un programme. Vous pouvez également utiliser cette méthode pour activer ou désactiver un module d'E/S, ce qui est une solution de programme similaire à l'utilisation de bit d'inhibition avec un processeur PLC-5.

En se basant sur l'état de SW.1, placer la valeur appropriée dans l'attribut d'indicateur désactivé du programme discret.



## Diagramme à relais



## Texte structuré

```

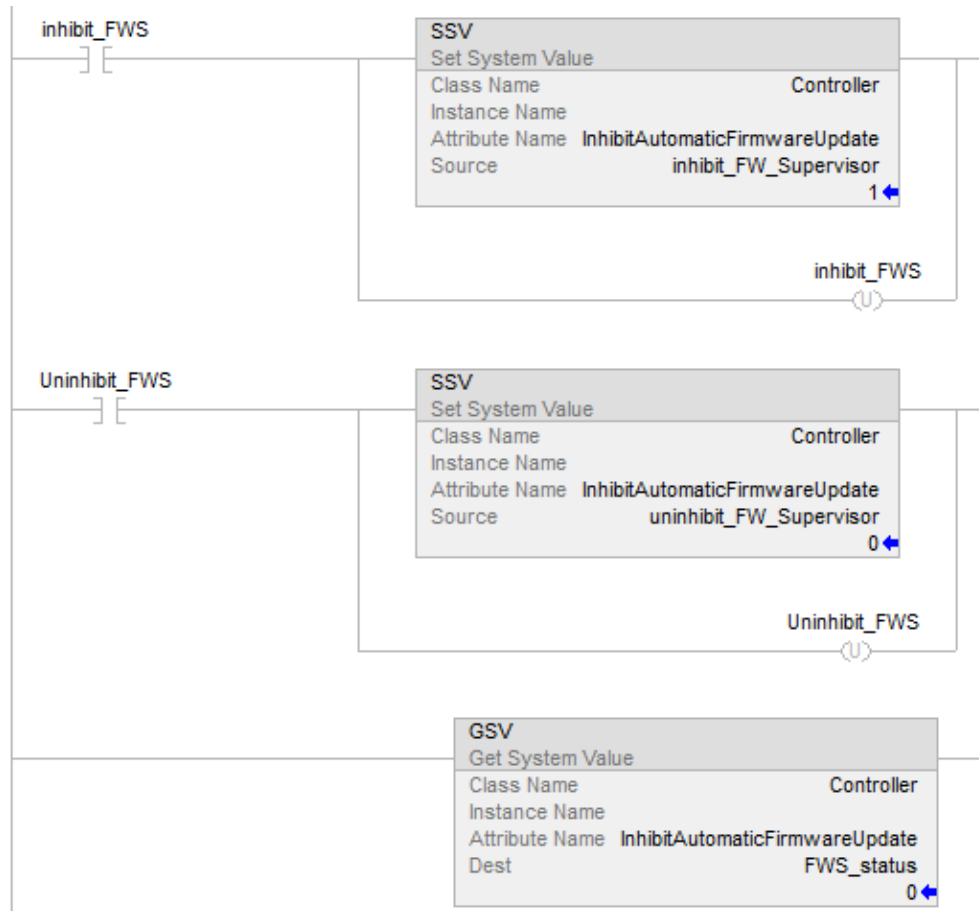
IF SW.1 THEN
discrete_prog_flag := enable_prog;
ELSE
discrete_prog_flag := disable_prog;
END_IF;
SSV (PROGRAM, DISCRETE, DISABLEFLAG, discrete_prog_flag);

```

### Inhiber et Annuler l'inhibition de la mise à jour automatique du firmware de FirmwareSupervisor.

L'exemple suivant utilise l'instruction GSV/SSV pour inhiber ou annuler l'inhibition de l'attribut Mise à jour automatique du firmware de l'automate. Si vous écrivez une valeur de 1, cela inhibe la fonctionnalité. Si vous écrivez une valeur de 0, cela annule l'inhibition de la fonctionnalité. L'état de l'attribut peut également être lu avec une instruction GSV.

## Diagramme à relais



## Objets GSV/SSV

Lorsque vous saisissez une instruction GSV/SSV, spécifiez l'objet et son attribut auxquels accéder. Dans certains cas, il peut y avoir plusieurs instances du même type d'objet. Veillez bien à indiquer le nom de l'objet. Par exemple, chaque tâche possède son propre objet TASK, qui nécessite de spécifier le nom de la tâche pour que l'accès soit accordé.

---

**Important :** Pour l'instruction GSV, seule la taille spécifiée des données est copiée dans la destination. Par exemple, si l'attribut est spécifié comme étant de type SINT et que la destination est de type DINT, seuls les 8 bits inférieurs de la destination DINT sont mis à jour, laissant les 24 bits restants inchangés.

---

**Important :** La mémoire tampon d'alarme a été supprimée des fonctions disponibles par abonnement pour créer des alarmes dans le firmware v21 et elle n'est plus disponible. Les instructions GSV qui faisaient précédemment référence à l'attribut de mémoire tampon d'alarme seront invalidées lors de la vérification du projet. Il incombe au programmeur de modifier comme il convient les codes d'application qui reposaient sur cet attribut.

---

Voici les objets GSV/SSV. Les objets accessibles varient en fonction de l'automate.

- [AddOnInstructionDefinition](#) sur la [page 211](#)
- [Axis](#) sur la [page 215](#)
- [Controller](#) sur la [page 226](#)
- [ControllerDevice](#) sur la [page 228](#)
- [CoordinateSystem](#) sur la [page 230](#)
- [CST](#) sur la [page 235](#)
- [DF1](#) sur la [page 238](#)
- [FaultLog](#) sur la [page 241](#)
- [HardwareStatus](#) sur la [page 242](#)
- [Message](#) sur la [page 234](#)
- [Module](#) sur la [page 244](#)
- [MotionGroup](#) sur la [page 232](#)
- [Programme](#) sur la [page 252](#)
- [Redondance](#) sur la [page 247](#)
- [Routine](#) sur la [page 247](#)
- [Safety](#) sur la [page 252](#)
- [SerialPort](#) sur la [page 254](#)
- [Tâche](#) sur la [page 255](#)
- [TimeSynchronize](#) sur la [page 257](#)
- [WallClockTime](#) sur la [page 262](#)

#### Voir aussi

[Lire la valeur système \(GSV\) et Définir la valeur système \(SSV\)](#) sur la [page 190](#)

[Instructions d'entrée/sortie](#) sur la [page 151](#)

## Accès à l'objet AddOnInstruction Definition

L'objet **AddOnInstructionDefinition** vous permet de personnaliser les instructions d'ensembles de logiques courantes, offre une interface commune à cette logique et document l'instruction.

Pour plus d'informations, reportez-vous à LOGIX 5000 Controllers Add-On Instructions Programming Manual, publication 1756-PM010.

Attribut	Type de données	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description
LastEditDate	LINT	GSV	Aucun	Date et horodatage de la dernière édition d'une définition d'instruction complémentaire.
MajorRevision	DINT	GSV	Aucun	Numéro de révision majeure de l'instruction complémentaire.
MinorRevision	DINT	GSV	Aucun	Numéro de révision mineure de l'instruction complémentaire.
Name	String	GSV	GSV	Nom de l'instruction complémentaire.
RevisionExtendedText	String	GSV	Aucun	Texte décrivant la révision de l'instruction complémentaire.
SafetySignature ID	DINT	GSV	Aucun	Dans un projet de sécurité, le numéro d'ID, la date et l'horodatage de la définition d'une instruction complémentaire.
SignatureID	DINT	GSV	Aucun	Numéro d'identification 32 bits d'une définition d'instruction complémentaire.
Vendor	String	GSV	Aucun	Fournisseur ayant créé l'instruction complémentaire.

#### Voir aussi

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet ALARMBUFFER

L'objet ALARMBUFFER fait partie de l'infrastructure Éditeur/Abonné. L'infrastructure Éditeur/Abonné fait partie du sous-système de communication de l'automate Logix. Le sous-système de communication de l'automate Logix met en place les modèles de messagerie Éditeur/Abonné pour CIP, ce qui permet aux autres dispositifs de recevoir les messages envoyés par le sous-système de l'automate. Actuellement les sous-systèmes Alarmes Numériques et Analogiques et Phase d'équipement par lots utilisent l'infrastructure Éditeur/Abonné pour fournir les messages via CIP aux applications abonnées.

Utiliser l'objet ALARMBUFFER pour vous aider à déterminer l'existence de connexions au sous-système Éditeur/Abonné et à l'état de ces connexions. Une instance d'objet AlarmBuffer existe pour chaque application abonnée. Cela signifie qu'un objet AlarmBuffer peut exister à un moment donné, mais peut ne pas exister à un autre moment. C'est pourquoi une instruction Lire la valeur système (GSV) renvoie un état dans une partie de l'étiquette de destination (INT[0].0). Quand le bit d'état est à zéro, cela veut dire le plus probablement que l'objet AlarmBuffer n'existe plus.

Attribut	Type de données	Instruction	Description	
AlarmBufferInstance	DINT[n]	GSV	Renvoie les ID de l'objet AlarmBuffer.	
			DINT[0]	Nombre d'objets AlarmBuffer.
			DINT[1...(n-1)]	Les ID de l'objet AlarmBuffer.
			Si le nombre d'objets AlarmBuffer est supérieur à n-1, seuls les ID des premiers (n-1) objets sont renvoyés. Vous n'avez pas besoin de spécifier un ID d'instance d'AlarmBuffer pour cet attribut.	
AlarmBufferStatus	INT[2]	GSV	Renvoie l'état de l'objet AlarmBuffer spécifié. Vous devez spécifier l'ID de l'instance d'AlarmBuffer pour obtenir l'état de cette instance en particulier.	
			INT[0].0	1-L'attribut AlarmBufferStatus est valide. 0-L'attribut AlarmBufferStatus est non valide.
			INT[1]	Valeur de l'attribut AlarmBuffer Status.
			L'attribut Status contient les éléments suivants :	
			INT[1].0	1-Paquet à plusieurs messages activés. 0-Paquet à plusieurs messages désactivés.
			INT[1].1	1-Tampon activé. 0-Tampon désactivé.
			INT[1].2	1-Données stockées dans le tampon. 0-Tampon vide.
			INT[1].3	1-Tampon plein. 0-Tampon n'est pas plein.
			INT[1].4	1-Les messages État d'initialisation NE SERONT PAS envoyés (au moment de l'abonnement et d'une commutation de Redondance). 0-Les messages État d'initialisation SERONT envoyés.
			Tous les autres bits sont réservés et définis sur 0.	
BufferSize	INT[2]	GSV	Renvoie la taille du tampon (en kB) de l'objet AlarmBuffer spécifié. Vous devez spécifier l'ID de l'instance de tampon d'alarme pour obtenir la taille du tampon de cette instance en particulier.	
			INT[0].0	1-L'attribut BufferSize est valide. 0-L'attribut BufferSize est non valide.
			INT[1]	Valeur de l'attribut Buffer Size.
BufferUsage	INT[2]	GSV	Renvoie le pourcentage d'espace tampon utilisé par l'objet AlarmBuffer spécifié. Vous devez spécifier l'ID de l'instance d'AlarmBuffer pour obtenir la valeur d'utilisation du tampon de cette instance en particulier.	
			INT[0].1	1-L'attribut BufferUsage est valide. 0-L'attribut BufferUsage est non valide.
			INT[1]	Valeur d'attribut BufferUsage.

SubscriberName	STRING	GSV	<p>Renvoie le nom d'abonné de l'objet AlarmBuffer spécifié. Vous devez spécifier l'ID de l'instance d'AlarmBuffer pour obtenir le nom d'abonné de cette instance en particulier.</p> <p>Tout type de chaîne peut être référencée comme étiquette de destination.</p> <p>Si le Nom de l'abonné ne peut pas rentrer dans la chaîne de l'étiquette de destination, alors seule la partie du nom qui peut rentrer dans l'étiquette de destination est fournie par l'instruction.</p> <p>Si l'instance d'objet AlarmBuffer spécifiée par l'ID d'instance n'existe pas lors de l'appel de l'instruction, alors la longueur de la chaîne (membre .LEN) est définie sur zéro.</p> <p>Noter que si aucun nom d'abonné n'est fourni lorsque l'objet AlarmBuffer est créé par un abonné, alors l'attribut du nom de l'abonné est défini comme étant le numéro de série d'un dispositif associé à la connexion par laquelle le service Création a été appelé sur l'objet AlarmBuffer.</p>
----------------	--------	-----	---

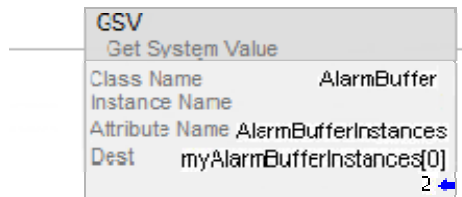
### Exemple d'instruction GSV

Votre programme peut contenir une instruction GSV pour obtenir une liste des AlarmBufferInstances actuelles dans l'automate. L'instruction renverra le nombre total d'objets de tampon d'alarme présents dans l'automate (DINT[0]) ainsi que l'ID de l'instance d'objet AlarmBuffer associé (DINT[1] – DINT[n-1]) pour chaque objet AlarmBuffer qui est présent dans l'automate. L'instruction GSV affiche la valeur du nombre d'objets AlarmBuffer (DINT[0]) sous le nom d'étiquette Dest (destination).

Votre programme peut utiliser l'ID de l'instance d'objet AlarmBuffer pour obtenir les informations liées à une instance spécifique de l'objet AlarmBuffer présent dans l'automate. Un mot d'état (INT[0]) indiquant des données valides ou non valides, est renvoyé à l'étiquette de destination pour les attributs AlarmBufferStatus, BufferSize, et BufferUsage, car les objets de tampon d'alarme peuvent être créés et supprimés à tout moment. La valeur renvoyée est en (INT[1]) lorsque Attribute Name est égal à AlarmBufferStatue, BufferSize, ou BufferUsage. La valeur renvoyée est le nom de l'abonné lorsque Attribute Name est SubscriberName. Aucun état n'est renvoyé pour l'attribut SubscriberName.

### Diagramme à relais

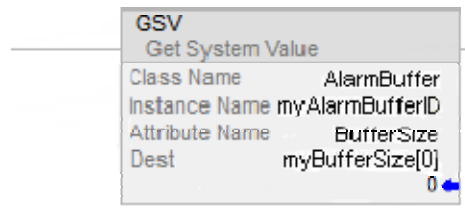
L'exemple suivant montre une instruction GSV récupérant les ID d'objet AlarmBuffer.



Bien que le GSV des AlarmBufferInstances renvoie des valeurs dans un tableau, vous ne pouvez pas utiliser l'adresse du tableau pour obtenir les valeurs d'attribut

de cette instance. Vous devez copier ou déplacer la valeur située dans `myAlarmBufferInstances[x]`, (où  $x = 1, 2, 3, \dots$ ) vers une étiquette directe (non indexée) comme `myAlarmBufferID`, montrée dans l'illustration suivante.

L'exemple suivant montre une instruction GSV récupérant la taille de tampon de l'objet `AlarmBuffer`.



Le nombre qui s'affiche sous le nom d'étiquette `Dest` (destination) est la valeur de bit valide ou non valide lorsque `Attribute Name` est `AlarmBufferStatus`, `BufferSize` ou `BufferUsage`.

### Texte structuré

L'exemple suivant montre une instruction GSV récupérant les ID d'objet `AlarmBuffer`.

- `GSV(AlarmBuffer, AlarmBufferInstances, myAlarmBufferInstances[0]);`

L'exemple suivant montre une instruction GSV récupérant l'objet `AlarmBuffer`.

- `GSV(AlarmBuffer, myAlarmBufferID, BufferSize, myBufferSize[0]);`

## Accès à l'objet Axis

L'objet `AXIS` fournit des informations d'état sur un axe. Indiquer le nom de l'étiquette de l'axe pour déterminer l'objet `AXIS` qui vous intéresse.

Pour plus d'information sur l'objet `AXIS`, reportez-vous à *SERCOS and Analog Motion Configuration and Startup User Manual*, publication MOTION-UM001.

Lorsqu'un attribut est marqué par une astérisque (\*), cela signifie qu'il est situé à la fois dans l'automate `ControlLogix` et dans le module de mouvement. Lorsque vous utilisez une instruction `SSV` pour écrire l'une de ces valeurs, l'automate met à jour automatiquement la copie dans le module. Cependant, ce processus n'est pas immédiat. L'étiquette d'état de l'axe, `ConfigUpdateInProgress` est fourni pour indiquer quand le processus sera achevé.

Par exemple, si vous exécutez une instruction `SSV` sur l'attribut `PositionLockTolerance`, l'attribut `ConfigUpdateInProgress` de l'étiquette `Axis` est défini jusqu'à ce que le module soit mis à jour correctement. De ce fait, la logique suivant l'instruction `SSV` pourrait attendre la réinitialisation de ce bit avant de se poursuivre dans le programme.

Attribut	Type de donnée	Instruction	Description (Description)		
* AccelerationFeedForwardGain	REAL	GSV SSV	Le pourcentage de sortie de commande du couple requis pour produire l'accélération commandée.		
ACStopMode	SINT	GSV SSV	Le type d'arrêt à réaliser sur votre axe.		
			Valeur (Value)	Signification	
			0	arrêt rapide	
			1	fermeture rapide	
2	fermeture matériel				
ActualPosition	REAL	GSV	La position réelle en unités de position de votre axe.		
ActualVelocity	REAL	GSV	La vitesse réelle de votre axe en unités de position/seconde.		
AnalogInput1	REAL	GSV SSV	Cet attribut s'applique uniquement à une Entrée analogique 2 associée à l'axe, un variateur Kinetix7000. Cet attribut, dont la valeur est un entier dans la plage +/-16384, représente la valeur analogique d'un dispositif analogique connecté aux entrées analogiques du variateur Kinetix7000. Ces entrées sont utiles pour les applications de conversion/toile dotées d'un capteur de force (mesurant la force de tension de toile d'un rouleau) ou d'un tenseur (mesurant directement la force de tension de toile/position), qui peut être connecté directement au variateur qui contrôle la toile.		
AverageVelocity	REAL	GSV	La vitesse moyenne de votre axe en unités de position/seconde.		
AverageVelocityTimebase	REAL	GSV SSV	La base de temps, en secondes, de la vitesse moyenne de votre axe.		
AxisConfigurationState	SINT	GSV	L'état de la configuration de l'axe.		
			Valeur (Value)	Signification	
			0 – 126	pas encore configuré	
			127	données d'axe consommées non valides (en raison de versions incompatibles entre le producteur et le consommateur)	
			128	configuré	
			3	en attente de réponse	
4	configuré				
AxisEventBits	DINT	GSV	Les bits d'événement d'asservissement de votre boucle d'asservissement. (Dans la structure AXIS, il s'agit du membre AxisEvent.)		
			Bit	Nom du bit	Signification
			0	WatchEventArmed Status	armement de l'événement de surveillance
			1	WatchEventStatus	événement de surveillance
			2	RegEvent1Armed Status	armement de l'événement d'enregistrement



			3	RegEvent1Status	événement d'enregistrement
			4	HomeEventArmed Status	armement de l'événement de prise d'origine
			5	HomeEventStatus	événement de prise d'origine
AxisState	SINT	GSV	L'état de fonctionnement de l'axe.		
			Valeur (Value)	Signification	
			0	axe prêt	
			1	contrôle du variateur direct	
			2	Commande d'asservissement	
			3	défaut de l'axe	
			4	fermeture de l'axe	
Bandwidth	REAL	GSV SSV	La bande passante du gain unitaire (Hz) que l'automate utilise pour calculer les gains d'une instruction Appliquer les réglages d'axe (MAAT).		
C2CConnectionInstance	DINT	GSV	L'instance de connexion de l'automate produisant les données de l'axe.		
C2CMapTableInstance	DINT	GSV	L'instance de mappage de l'automate produisant les données de l'axe.		
CommandPosition	REAL	GSV	La position de commande de votre axe, en unités de position.		
CommandVelocity	REAL	GSV	La vitesse de commande de votre axe, en unités de position.		
ConversionConstant	REAL	GSV SSV	Le facteur de conversion utilisé pour convertir vos unités en comptages de retour (en comptages/unité de position).		
DampingFactor	REAL	GSV SSV	La valeur utilisée pour calculer la bande passante de l'asservissement de position maximale pendant l'exécution de l'instruction Exécution du mouvement de réglage d'axe (MRAT).		
*DriveFaultAction	SINT	GSV SSV	L'opération exécutée lorsqu'un défaut variateur se produit.		
			Valeur (Value)	Signification	
			0	fermeture de l'axe	
			1	désactivation du variateur	
			2	arrêt du mouvement commandé	
3	changement du bit d'état uniquement				
DynamicsConfigurationBits	DINT	GSV SSV	Dans la version 16, la gestion par l'automate du changement pour un profil de courbe en S a été améliorée. Voulez-vous revenir au comportement de la version 15 ou antérieure pour les courbes en S ? NON — Laisser ces bits sur ON (par défaut). OUI — Passer à l'état OFF un ou plusieurs de ces bits :		
			<b>Pour désactiver cette modification</b>		<b>Désactiver ce bit</b>

			<p><b>Retard à l'arrêt de la courbe en S réduit</b>                  Cette modification s'applique à l'instruction Arrêt d'axe de mouvement (MAS). Elle vous permet d'utiliser un jerk de décélération supérieure pour arrêter plus vite un axe qui <b>accélère</b>.                  L'automate utilise la variation de décélération de l'instruction d'arrêt si elle est supérieure au jerk de décélération actuel.</p>	0
			<p><b>Inversions de la vitesse de la courbe en S réduites</b>                  Dans les versions antérieures à la version 16, vous pouviez provoquer l'inversion momentanée du sens de l'axe lorsque vous diminuez le jerk de décélération pendant que l'axe décélérait. Ce phénomène se produisait généralement lorsque vous tentiez de relancer une marche par à-coups ou d'effectuer un déplacement avec un taux de décélération plus faible pendant que l'axe s'arrêtait. Cette modification empêche l'inversion de l'axe dans ces cas-là.</p>	1
			<p><b>Dépassements de la vitesse de la courbe en S réduits</b>                  Vous pouvez provoquer le dépassement de la vitesse programmée de l'axe si vous diminuez la variation d'accélération pendant que l'axe accélère. Cette modification autorise un dépassement maximal de 50 % de la vitesse programmée.</p>	2
FaultConfigurationBits	DINT		Type d'axe	Configuration des défauts
*FeedbackFaultAction	SINT	GSV SSV	L'opération exécutée en cas de défaut de perte de codeur.	
			Valeur (Value)	Signification
			0	fermeture de l'axe
			1	désactivation du variateur
			2	arrêt du mouvement commandé
			3	changement du bit d'état uniquement
*FeedbackNoiseFaultAction	SINT	GSV SSV	L'opération exécutée en cas de défaut de bruit de codeur.	
			Valeur (Value)	Signification
			0	fermeture de l'axe

			1	désactivation du variateur
			2	arrêt du mouvement commandé
			3	changement du bit d'état uniquement
*FrictionCompensation	REAL	GSV SSV	Le niveau de sortie fixe, en volts, utilisé pour compenser la friction statique.	
GroupInstance	DINT	GSV	Le numéro de l'instance du groupe de mouvements qui contient votre axe.	
HardOvertravelFaultAction	SINT	GSV SSV	<b>Valeur (Value)</b>	<b>Signification</b>
			0	fermeture
			1	désactivation du variateur
			2	arrêt du mouvement
			3	état seulement
HomeConfigurationBits	DINT	GSV SSV	Les bits de configuration du mouvement pour votre axe.	
			Bit	Signification
			0	direction de prise d'origine
			1	interrupteur de prise d'origine normalement fermé
			2	front négatif du zéro codeur de la prise d'origine
HomeMode	SINT	GSV SSV	Le mode de prise d'origine de votre axe.	
			Valeur (Value)	Signification
			0	prise d'origine passive
			1	prise d'origine active (valeur par défaut)
			2	absolute
HomePosition	REAL	GSV SSV	La position de prise d'origine de votre axe, en unités de position.	
HomeReturnSpeed	REAL	GSV SSV	La vitesse de retour de prise d'origine de votre axe, en unités de position/seconde.	
HomeSequence	SINT	GSV SSV	Le type de séquence de prise d'origine de votre axe.	
			Valeur (Value)	Signification
			0	prise d'origine immédiate
			1	prise d'origine de type interrupteur
			2	prise d'origine de type zéro codeur
3	prise d'origine de type interrupteur-zéro codeur (valeur par défaut)			
HomeSpeed	REAL	GSV SSV	La vitesse de prise d'origine de votre axe, en unités de position/seconde.	
Instance	DINT	GSV	Le numéro d'instance de l'axe.	
InterpolatedActualPosition	REAL	GSV	Cet attribut fournit la position d'axe réelle interpolée, pour les captures de position basées sur le temps. La position est donnée en unités de position, et est basée sur la valeur de l'attribut InterpolationTime. Pour interpoler une position d'axe réelle, utilisez une instruction SSV pour définir l'attribut InterpolationTime.	

InterpolatedCommandPosition	REAL	GSV	<p>Cet attribut fournit la position de l'axe de commande interpolée, pour les captures de position basées sur le temps.</p> <p>La position est donnée en unités de position, et est basée sur la valeur de l'attribut InterpolationTime.</p> <p>Pour interpoler une position de l'axe de commande, utilisez une instruction SSV pour définir l'attribut InterpolationTime.</p>																														
InterpolationTime	DINT	GSV SSV	<p>Utilisez cet attribut pour fournir une référence pour les captures de position basées sur le temps.</p> <p>Pour interpoler une position, utilisez une instruction SSV pour définir l'attribut InterpolationTime. L'automate met ensuite à jour les attributs suivants :</p> <ul style="list-style-type: none"> <li>• InterpolatedActualPosition</li> <li>• InterpolatedCommandPosition</li> </ul> <p>Pour fournir une valeur pour InterpolationTime, vous pouvez utiliser tout événement qui produit un horodatage CST, tel que :</p> <ul style="list-style-type: none"> <li>• Attribut RegistrationTime</li> <li>• horodatage d'une sortie numérique</li> </ul> <p>L'attribut InterpolationTime utilise les valeurs 32 bits inférieures d'un horodatage CST.</p>																														
MapTableInstance	DINT	GSV	L'instance de mappage E/S d'un servo-module.																														
MasterOffset	REAL	GSV	Le décalage de position actuellement appliqué au maître d'une came de position. Donné en unités de position de l'axe maître.																														
MaximumAcceleration	REAL	GSV SSV	L'accélération maximale de votre axe, en unités de position/seconde <sup>2</sup> .																														
MaximumDeceleration	REAL	GSV SSV	La décélération maximale de votre axe, en unités de position/seconde <sup>2</sup>																														
*MaximumNegativeTravel	REAL	GSV SSV	La limite de course négative maximale, en unités de position.																														
*MaximumPositiveTravel	REAL	GSV SSV	La limite de course positive maximale, en unités de position.																														
MaximumSpeed	REAL	GSV SSV	La vitesse maximale de votre axe, en unités de position/seconde.																														
ModuleChannel	SINT	GSV	Le canal de votre servo-module.																														
MotionStatusBits	DINT	GSV	<p>Les bits d'état de mouvement pour votre axe. (Dans la structure AXIS, il s'agit du membre MotionStatus.)</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Nom du bit</th> <th>Signification</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>AccelStatus</td> <td>accélération</td> </tr> <tr> <td>1</td> <td>DecelStatus</td> <td>décélération</td> </tr> <tr> <td>2</td> <td>MoveStatus</td> <td>move</td> </tr> <tr> <td>3</td> <td>JogStatus</td> <td>jog</td> </tr> <tr> <td>4</td> <td>GearingStatus</td> <td>gear</td> </tr> <tr> <td>5</td> <td>HomingStatus</td> <td>home</td> </tr> <tr> <td>6</td> <td>StoppingStatus</td> <td>arrêt</td> </tr> <tr> <td>7</td> <td>AxisHomedStatus</td> <td>état de la prise d'origine</td> </tr> <tr> <td>8</td> <td>PositionCamStatus</td> <td>came de position</td> </tr> </tbody> </table>	Bit	Nom du bit	Signification	0	AccelStatus	accélération	1	DecelStatus	décélération	2	MoveStatus	move	3	JogStatus	jog	4	GearingStatus	gear	5	HomingStatus	home	6	StoppingStatus	arrêt	7	AxisHomedStatus	état de la prise d'origine	8	PositionCamStatus	came de position
Bit	Nom du bit	Signification																															
0	AccelStatus	accélération																															
1	DecelStatus	décélération																															
2	MoveStatus	move																															
3	JogStatus	jog																															
4	GearingStatus	gear																															
5	HomingStatus	home																															
6	StoppingStatus	arrêt																															
7	AxisHomedStatus	état de la prise d'origine																															
8	PositionCamStatus	came de position																															

			9	TimeCamStatus	comes de temps
			10	PositionCamPendingStatus	came de position en attente
			11	TimeCamPendingStatus	came de temps en attente
			12	GearingLockStatus	verrouillage de synchronisation
			13	PositionCamLockStatus	verrouillage de la came de position
			14	MasterOffsetMoveStatus	mouvement décalage maître
			15	CoordinatedMotionStatus	mouvement coordonné
			16	TransformStateStatus	état de la transformation
			17	ControlledByTransformStatus	contrôle par transformation
*OutputLPFilterBandwidth	REAL	GSV SSV	La bande passante (Hz) du filtre passe-bas de sortie numérique de l'asservissement.		
*OutputLimit	REAL	GSV SSV	La valeur, en volts, de la tension maximale de sortie de l'asservissement de votre axe.		
*OutputOffset	REAL	GSV SSV	La valeur, en volts, utilisée pour compenser les effets des décalages cumulés de la sortie DAC du servo-module et de l'entrée du servovariateur.		
PositionError	REAL	GSV	La différence entre la position de commande et la position réelle d'un axe.		
*PositionErrorFaultAction	SINT	GSV SSV	L'opération exécutée en cas de défaut d'erreur de position.		
			Valeur (Value)	Signification	
			0	fermeture de l'axe	
			1	désactivation du variateur	
			2	arrêt du mouvement commandé	
			3	changement du bit d'état uniquement	
*PositionErrorTolerance	REAL	GSV SSV	Le degré d'erreur de position (en unités de position) que l'asservissement tolère avant de générer un défaut d'erreur de position.		
PositionIntegratorError	REAL	GSV	La somme de l'erreur de position pour en axe, en unités de position.		
*PositionIntegralGain	REAL	GSV SSV	La valeur (1/ms <sup>2</sup> ) utilisée pour obtenir un positionnement précis de l'axe malgré les perturbations telles que la friction statique et la gravité.		
PositionLockTolerance	REAL	GSV SSV	Le degré d'erreur de position (en unités de position) que le servo-module tolère lors de la fourniture de l'indication d'état verrouillé de la position réelle.		
*PositionProportionalGain	REAL	GSV SSV	La valeur (1/ms) par laquelle l'automate multiplie l'erreur de position pour compenser celle-ci.		
PositionServoBandwidth	REAL	GSV SSV	La bande passante du gain unitaire que l'automate utilise pour calculer les gains d'une instruction Appliquer les réglages d'axe (MAAT).		

*PositionUnwind	DINT	GSV SSV	La valeur utilisée pour réaliser le rebouclage automatique de l'axe rotatif, en comptages/révolution.		
ProcessStatus	INT	GSV	L'état de la dernière instruction Exécution des diagnostics de raccordement d'axe (MRHD).		
			Valeur (Value)	Signification	
			0	test réussi	
			1	test en cours	
			2	test abandonné par l'utilisateur	
			3	la durée du test a dépassé le délai d'expiration de 2 secondes	
			4	le test a échoué en raison d'un défaut de l'asservissement	
ProgrammedStopMode	SINT	GSV SSV	Le type d'arrêt à réaliser sur votre axe.		
			Valeur (Value)	Signification	
			0	arrêt rapide	
			1	fermeture rapide	
			2	fermeture matériel	
Registration1Position	REAL	GSV	La position d'enregistrement de votre axe, en unités de position.		
RegistrationTime	DINT	GSV	<p>Vous pouvez utiliser cet attribut pour fournir un horodatage pour les captures de position basées sur le temps :</p> <ul style="list-style-type: none"> <li>• L'attribut RegistrationTime contient les 32 bits inférieurs de l'horodatage CST d'un événement d'enregistrement d'axe.</li> <li>• L'horodatage CST est mesuré en microsecondes.</li> <li>• Pour interpoler une position basée sur un événement d'enregistrement d'axe, procédez ainsi : <ul style="list-style-type: none"> <li>• Utilisez une instruction GSV pour obtenir la valeur de l'attribut RegistrationTime.</li> <li>• Utilisez une instruction SSV pour régler l'attribut InterpolationTime à la valeur de l'attribut RegistrationTime.</li> </ul> </li> </ul>		
RotaryAxis	SINT	GSV Étiquette (Tag)	<p>0 = linéaire 1 = rotatif</p> <p>Lorsque l'attribut Axe rotatif est défini sur vrai (1), le rebouclage de l'axe est autorisé. Il est ainsi possible de disposer d'un nombre infini de positions en rebouclant la position de l'axe chaque fois que l'axe effectue une révolution physique complète. Le nombre de comptages de codeur par révolution physique de l'axe est donné par l'attribut Rebouclage de position. Pour le fonctionnement linéaire, les comptages ne rebouclent pas. Ils sont limités à +/- 2 milliards.</p>		
ServoFaultBits	DINT	GSV	Les bits de défaut d'asservissement de votre boucle d'asservissement. (Dans la structure AXIS, il s'agit du membre AxisEvent.)		
			Bit	Nom du bit	Signification

			0	PosSoftOvertravel Fault	défaut de surcourse positive
			1	NegSoftOvertravel Fault	défaut de surcourse négative
			2	PositionErrorFault	défaut d'erreur de position
			3	FeedbackFault	défaut de perte du canal A du codeur
			4	FeedbackFault	défaut de perte du canal B du codeur
			5	FeedbackFault	défaut de perte du canal Z du codeur
			6	FeedbackNoiseFault	défaut de bruit du codeur
			7	DriveFault	défaut de variateur
			8	ModuleSyncFault	défaut de la connexion synchrone
			9	ModuleHardwareFault	défaut matériel de l'asservissement
ServoOutputLevel	REAL	GSV	Le niveau de tension de sortie, en volts, pour la boucle d'asservissement de l'axe.		
ServoStatusBits	DINT	GSV	Les bits d'état de votre boucle d'asservissement. (Dans la structure AXIS, il s'agit du membre ServoStatus.)		
			Bit	Nom du bit	Signification
			0	ServoActionStatus	action d'asservissement
			1	DriveEnableStatus	variateur activé
			2	OutputLimitStatus	limite de sortie
			3	PositionLockStatus	verrouillage de position
			13	TuneStatus	procédure de réglage
			14	ProcessStatus	Test de diagnostic
15	ShutdownStatus	fermeture de l'axe			
*SoftOvertravelFaultAction	SINT	GSV SSV	L'opération exécutée en cas de défaut de surcourse logicielle.		
			Valeur (Value)	Signification	
			0	fermeture de l'axe	
			1	désactivation du variateur	
			2	arrêt du mouvement commandé	
3	changement du bit d'état uniquement				
StartActualPosition	REAL	GSV	La position réelle de votre axe, en unités de position, lorsqu'un nouveau mouvement commandé commence pour l'axe.		
StartCommandPosition	REAL	GSV	La position de commande de votre axe, en unités de position, lorsqu'un nouveau mouvement commandé commence pour l'axe.		

StartMasterOffset	REAL	GSV	Le décalage maître lorsque l'instruction Mouvement de positionnement d'axe (MAM) a exécuté l'un des types de déplacement suivants : <ul style="list-style-type: none"> <li>• AbsoluteMasterOffset</li> <li>• IncrementalMasterOffset</li> </ul> Donné en unités de position de l'axe maître.	
StrobeActualPosition	REAL	GSV	La position réelle, en unités de position, d'un axe lorsque l'instruction Echantillonnage des positions d'un groupe d'axes (MGSP) s'exécute.	
StrobeCommandPosition	REAL	GSV	La position de commande, en unités de position, d'un axe lorsque l'instruction Echantillonnage des positions d'un groupe d'axes (MGSP) s'exécute.	
StrobeMasterOffset	REAL	GSV	Le décalage maître lorsque l'instruction Echantillonnage des positions d'un groupe d'axes (MGSP) s'exécute. Donné en unités de position de l'axe maître.	
TestDirectionForward	SINT	GSV	La direction de la course de l'axe pendant l'instruction Exécution des diagnostics de raccordement d'axe (MRHD), telle que vue par le servo-module.	
			Valeur (Value)	Signification
			0	direction négative (vers l'arrière)
			1	direction positive (vers l'avant)
TestIncrement	REAL	GSV SSV	Le mouvement nécessaire pour lancer le test Exécution des diagnostics de raccordement d'axe (MRHD).	
*TorqueScaling	REAL	GSV SSV	La valeur utilisée pour convertir la sortie de la boucle d'asservissement en tension équivalente pour le variateur.	
TuneAcceleration	REAL	GSV	La valeur de l'accélération, en unités de position/seconde <sup>2</sup> , mesurée pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneAccelerationTime	REAL	GSV	La durée, en secondes, de l'accélération mesurée pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneDeceleration	REAL	GSV	La valeur de la décélération, en unités de position/seconde, mesurée pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneDecelerationTime	REAL	GSV	La durée, en secondes, de la décélération mesurée pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneInertia	REAL	GSV	La valeur de l'inertie de l'axe, en mV/comptages K/seconde, telle que calculée à partir des mesures réalisées par l'automate pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneRiseTime	REAL	GSV	Le temps de montée de l'axe, en secondes, mesuré pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	



TuneSpeedScaling	REAL	GSV	Le facteur de mise à l'échelle du variateur de l'axe, en mV/comptages K/seconde, mesuré pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuneStatus	INT	GSV	L'état de la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).	
			Valeur (Value)	Signification
			0	réglage réussi
			1	réglage en cours
			2	réglage abandonné par l'utilisateur
			3	la durée du réglage a dépassé le délai d'expiration de 2 secondes
			4	le réglage a échoué en raison d'un défaut de l'asservissement
			5	l'axe atteint la fin de course de réglage
			7	vitesse de réglage trop faible pour réaliser des mesures
TuningConfigurationBits	DINT	GSV SSV	Les bits de configuration de réglage pour votre axe.	
			Bit	Signification
			0	direction du réglage (0=vers l'avant, 1=vers l'arrière)
			1	position de réglage de l'intégrateur d'erreur
			2	vitesse de réglage de l'intégrateur d'erreur
			3	vitesse de réglage du bit d'anticipation
			5	vitesse de réglage du filtre passe-bas
TuningSpeed	REAL	GSV SSV	La vitesse maximale, en unités de position/seconde, lancée par l'instruction Exécution du mouvement de réglage d'axe (MRAT).	
TuningTravelLimit	REAL	GSV SSV	La limite de course utilisée par l'instruction Exécution du mouvement de réglage d'axe (MRAT) pour limiter l'action pendant le réglage.	
VelocityCommand	REAL	GSV	La référence de vitesse actuelle, en unités de position/seconde, pour la boucle d'asservissement de vitesse d'un axe.	
VelocityError	REAL	GSV	La différence, en unités de position/seconde, entre la vitesse commandée et la vitesse réelle de l'axe asservi.	
VelocityFeedback	REAL	GSV	La vitesse réelle de votre axe, en unités de position/seconde, telle qu'estimée par le servo-module.	
*VelocityFeedforwardGain	REAL	GSV SSV	Le pourcentage de sortie de commande de vitesse requis pour générer la vitesse commandée.	
*VelocityIntegralGain	REAL	GSV SSV	La valeur (1/ms) par laquelle l'automate multiplie la valeur de VelocityError pour compenser l'erreur de vitesse.	

VelocityIntegratorError	REAL	GSV	La somme des erreurs de vitesse d'un axe donné.
*VelocityProportionalGain	REAL	GSV SSV	La valeur (1/ms) par laquelle l'automate multiplie la valeur de VelocityError pour compenser l'erreur de vitesse.
*VelocityScaling	REAL	GSV SSV	La valeur utilisée pour convertir la sortie de la boucle d'asservissement en tension équivalente pour le variateur.
VelocityServoBandwidth	REAL	GSV SSV	La bande passante (Hz) du variateur, telle que calculée à partir des mesures réalisées pendant la dernière instruction Exécution du mouvement de réglage d'axe (MRAT).
WatchPosition	REAL	GSV	La position de surveillance en unités de position de votre axe.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet Controller

L'objet **Controller** fournit des informations d'état relatives à l'exécution de l'automate.

Attribut	Type de donnée	Instruction	Description (Description)
Audit Value	DINT[2], LINT	GSV	La Valeur d'audit est une valeur unique générée lorsqu'un projet est téléchargé vers l'automate ou chargé depuis un dispositif de stockage amovible. Lorsqu'une modification est détectée, cette valeur est mise à jour. Pour spécifier les modifications à surveiller, utilisez l'attribut ChangesToDetect. <b>Astuce</b> : nous vous conseillons d'utiliser le type de données DINT[2], pour éviter les limitations lorsque vous travaillez avec des types de données LINT dans les automates Rockwell Automation.
ChangesToDetect	DINT[2], LINT	GSV, SSV	Utilisé pour indiquer les modifications à surveiller. Lorsqu'une modification surveillée est détectée, l'attribut Audit Value est mis à jour. <b>Astuce</b> : nous vous conseillons d'utiliser le type de données DINT[2], pour éviter les limitations lorsque vous travaillez avec des types de données LINT dans les automates Rockwell Automation.
CanUseRPIFrom Producer	DINT	GSV	Identifie si le RPI spécifié par le producteur doit être utilisé. Signification de la valeur 0 Ne pas utiliser le RPI spécifié par le producteur 1 Utiliser le RPI spécifié par le producteur
ControllerLog Execution Modification Count	DINT	GSV SSV	Nombre d'entrées du journal de l'automate provenant d'une modification des propriétés des tâches/du programme, d'une modification en ligne ou d'une modification de la tranche de temps de l'automate. Peut également être configuré pour inclure les entrées de journal tirées des forces. Ce nombre est réinitialisé si l'état de la RAM est en état mauvais. Le nombre n'est pas plafonné à la taille DINT maximale, et une réinitialisation est possible.
ControllerLog TotalEntryCount	DINT	GSV SSV	Nombre d'entrées de journal de l'automate depuis la dernière mise à niveau du firmware. Ce nombre est réinitialisé si l'état de la RAM est en état mauvais. Le nombre est plafonné à la taille DINT maximale.

DataTablePad Percentage	INT	GSV	Pourcentage (de 0 à 100) de mémoire libre de la table de données réservée.
IgnoreArrayFaults DuringPostScan	SINT	GSV SSV	<p>Utilisé pour configurer la suppression des défauts sélectionnés rencontrés lors de la post-scrutation d'une action SFC. Valide uniquement lorsque les SFC sont configurées pour une réinitialisation automatique.</p> <ul style="list-style-type: none"> <li>0. Cette valeur ne supprime pas les défauts pendant l'exécution de la post-scrutation. Il s'agit du comportement par défaut recommandé.</li> <li>1. Cette valeur supprime automatiquement les défauts majeurs 4/20 (indice de tableau trop important) et 4/83 (valeur hors limites) pendant la post-scrutation des actions SFC.</li> </ul> <p>Lorsqu'un défaut est supprimé, l'automate utilise un gestionnaire de défauts interne pour effacer automatiquement le défaut. L'instruction défectueuse est alors ignorée, et l'exécution reprend à l'instruction suivante.</p> <p>Le gestionnaire de défauts étant un système interne, vous n'avez pas à en configurer un pour bénéficier de ce comportement. En fait, même si un gestionnaire de défauts est configuré, un défaut supprimé ne le déclenchera pas.</p>
InhibitAutomatic FirmwareUpdate	BOOL	GSV SSV	<p>Indique si l'utilitaire de supervision de firmware doit être activé.</p> <ul style="list-style-type: none"> <li>0. L'utilitaire de supervision de firmware s'exécute.</li> <li>1. L'utilitaire de supervision de firmware ne s'exécute pas.</li> </ul>
KeepTestEditsOn Switch over	SINT	GSV	<p>Indique si les modifications de test lors de la commutation de l'automate doivent être conservées.</p> <ul style="list-style-type: none"> <li>0. Les tests des modifications lors de la commutation sont automatiquement désactivés.</li> <li>1. Les tests des modifications lors de la commutation sont exécutés.</li> </ul>
Nom (Name)	Chaîne	GSV	Nom de l'automate.
Redundancy Activé	SINT	GSV	<p>Indique si l'automate est configuré pour la redondance.</p> <ul style="list-style-type: none"> <li>0. L'automate n'est pas configuré pour la redondance.</li> <li>1. L'automate est configuré pour la redondance.</li> </ul>
ShareUnused TimeSlice	INT	GSV SSV	<p>Indique comment la tâche continue et les tâches en arrière-plan partagent la tranche de temps inutilisée.</p> <ul style="list-style-type: none"> <li>0. Le système d'exploitation n'octroie pas le contrôle à la tâche continue, même si les tâches en arrière-plan sont terminées.</li> <li>1. Une tâche continue s'exécute, même si les tâches en arrière-plan sont terminées. Il s'agit de la valeur par défaut.</li> <li>2. Cette valeur, ou une valeur supérieure, consigne un défaut mineur et ne modifie pas le paramètre.</li> </ul>
TimeSlice	INT	GSV SSV	Pourcentage de CPU disponible (de 10 à 90) affecté aux communications. Cette valeur ne peut pas être modifiée lorsque l'interrupteur à clé est sur la position Exécution.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet ControllerDevice

L'objet **ControllerDevice** identifie le matériel physique de l'automate.

Attribut	Type de données	Instruction	Description
DeviceName	SINT[33]	GSV	Chaîne ASCII qui identifie la référence de l'automate et de la carte mémoire. Le premier octet contient le nombre de caractères ASCII renvoyés dans la chaîne de tableau.
ProductCode	INT	GSV	Chaque valeur correspond à un type d'automate particulier : 15 SoftLogix5800 49 PowerFlex® avec DriveLogix5725 52 PowerFlex avec DriveLogix5730 53 Émulateur 54 1756-L61 ControlLogix 55 1756-L62 ControlLogix 56 1756-L63 ControlLogix 57 1756-L64 ControlLogix 64 1769-L31 CompactLogix 65 1769-L35E CompactLogix 67 1756-L61S GuardLogix 68 1756-L62S GuardLogix 69 1756-LSP GuardLogix 72 1768-L43 CompactLogix 74 1768-L45 CompactLogix 76 1769-L32C CompactLogix 77 1769-L32E CompactLogix 80 1769-L35CR CompactLogix 85 1756-L65 ControlLogix 86 1756-L63S GuardLogix 87 1769-L23E-QB1 CompactLogix 88 1769-L23-QBFC1 CompactLogix 89 1769-L23E-QBFC1 CompactLogix 92 1756-L71 93 1756-L72 94 1756-L73 95 1756-L74 96 1756-L75 106 1769-L30ER 107 1769-L33ER 108 1769-L36ERM 109 1769-L30ER-NSE 110 1769-L33ERM 146 1756-L7SP 147 1756-L72S 148 1756-L73S 149 1769-L24ER-QB1B 150 1769-L24ER-QBFC1B 151 1769-L27ERM-QBFC1B 153 1769-L16ER-BB1B 154 1769-L18ER-BB1B 155 1769-L18ERM-BB1B 156 1769-L30ERM

			158 1756-L71S
ProductRev	INT	GSV	Identifie la version de produit actuelle. L'affichage doit être hexadécimal. L'octet inférieur contient la version majeure, l'octet supérieur la version mineure.
SerialNumber	DINT	GSV	Numéro de série du dispositif. Le numéro de série est attribué lors de la fabrication du dispositif.
Status	INT	GSV	<p>Bits d'état du dispositif</p> <p>7...4 Signification</p> <p>0000 Réservé</p> <p>0001 Mise à jour flash en cours</p> <p>0010 Réservé</p> <p>0011 Réservé</p> <p>0100 Erreur de flash</p> <p>0101 Modes de défaut</p> <p>0110 Exécution</p> <p>0111 Programme</p> <p>Bits d'état de défaut</p> <p>11...8 Signification</p> <p>0001 Défaut mineur récupérable</p> <p>0010 Défaut mineur irrécupérable</p> <p>0100 Défaut majeur récupérable</p> <p>1000 Défaut majeur irrécupérable</p> <p>Bits d'état de l'automate</p> <p>13...12 Signification</p> <p>01 Interrupteur à clé en mode Exécution</p> <p>10 Interrupteur à clé en mode Programme</p> <p>11 Interrupteur à clé en mode Exécution à distance</p> <p>15...14 Signification</p> <p>01 L'automate est en train de changer de modes</p> <p>10 Mode de débogage si l'automate est en mode Exécution</p>
Type	INT	GSV	Identifie le dispositif comme étant un automate. Automate = 14.
Vendor	INT	GSV	Identifie le fournisseur du dispositif. Allen-Bradley = 0001.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet CoordinateSystem

L'objet COORDINATESYSTEM fournit des informations d'état sur l'exécution du système de coordonnées du mouvement.

Attribut	Type de données	Instruction	Signification
CoordinateMotionStatus	DINT	GSV SSV	Définir lorsqu'un verrouillage d'axe est demandé pour une instruction MCLM ou MCCM et que l'axe a franchi la position de verrouillage. Mettre à zéro lorsqu'une instruction MCLM ou MCCM est initiée.
AccelStatus	BOOL	GSV SSV	Définir à quel moment le vecteur accélère. Mettre à zéro lorsqu'une combinaison est en cours ou lorsque le déplacement du vecteur atteint la vitesse ou décélère.
DecelStatus	BOOL	GSV SSV	Définir à quel moment le vecteur décélère. Mettre à zéro lorsqu'une combinaison est en cours ou lorsque le déplacement du vecteur accélère ou lorsque le mouvement se termine.
ActualPosToleranceStatus	BOOL	GSV SSV	Définir uniquement le type de terminaison Tolérance réelle. Le bit est défini une fois que les deux conditions suivantes sont satisfaites. 1) L'interpolation est terminée. 2) La distance réelle vers le point final programmé est inférieure à la valeur Tolérance réelle du système de coordonnées configuré. Il reste défini lorsque l'instruction se termine. Il est réinitialisé au démarrage d'une nouvelle instruction.
CommandPosToleranceStatus	BOOL	GSV SSV	Définir tous les types de terminaison chaque fois que la distance vers le point de terminaison programmé est inférieure à la valeur Tolérance commandée du système de coordonnées configuré et reste défini lorsque l'instruction se termine. Il est réinitialisé au démarrage d'une nouvelle instruction.
StoppingStatus	BOOL	GSV SSV	Le bit d'état d'arrêt est mis à zéro lorsque l'instruction MCCM s'exécute.
MoveStatus	BOOL	GSV SSV	Définir à quel moment MCCM commence le mouvement de l'axe. Mettre à zéro sur le bit .PC de la dernière instruction de mouvement ou d'instruction de mouvement qui s'exécute et provoque un arrêt.
MoveTransitionStatus	BOOL	GSV SSV	Définir à quel moment le type de terminaison Aucune décélération ou Tolérance commandée est satisfait. Lorsque la combinaison colinéaire se déplace, le bit n'est pas défini, car la machine est toujours sur le trajet. Il est mis à zéro lorsqu'une combinaison se termine, que le mouvement d'une instruction en attente démarre ou lorsqu'une instruction de mouvement s'exécute et provoque un arrêt. Indique qu'elle n'est pas sur le chemin.

MovePendingStatus	BOOL	GSV SSV	<p>Le bit de déplacement en attente est défini une fois que l'instruction de mouvement coordonné est mise en file d'attente. Une fois que l'instruction a commencé à s'exécuter, le bit est mis à zéro, sous réserve qu'aucune instruction de mouvement coordonné suivante n'ait été mise en file d'attente entre-temps. Dans le cas d'une instruction de mouvement coordonné unique, le bit d'état peut ne pas être détecté par l'utilisateur dans l'application Logix Designer, car la transition de la mise en file d'attente à l'exécution est plus rapide que la période intermédiaire de mise à jour.</p> <p>La valeur réelle du bit se produit dans le cas de plusieurs instructions. Tant qu'une instruction se trouve dans la file d'attente des instructions, le bit d'attente est défini. Cela fournit au programmeur de l'application Logix Designer un moyen d'uniformiser l'exécution de plusieurs instructions de mouvement coordonné. Le diagramme logique à relais contenant des instructions de mouvement coordonné peut être réalisé pour s'exécuter plus rapidement lorsque le programmeur autorise la mise en file d'attente des instructions pendant qu'une instruction précédente est en cours d'exécution. Lorsque le bit MovePendingStatus est mis à zéro, l'instruction de mouvement coordonné suivante peut être exécutée (c'est-à-dire configurée dans la file d'attente).</p>
MovePendingQueueFullStatus	BOOL	GSV SSV	Définir à quel moment la file d'attente des instructions est pleine. Il est mis à zéro lorsque la file d'attente dispose de place pour prendre en charge une nouvelle instruction de mouvement coordonné.
TransformSourceStatus	BOOL	GSV SSV	Le système de coordonnées est la source d'une transformation active.
TransformTargetStatus	BOOL	GSV SSV	Le système de coordonnées est la cible d'une transformation active.
CoordMotionLockStatus	BOOL	GSV SSV	<p>Définir lorsqu'un verrouillage d'axe est demandé pour une instruction MCLM ou MCCM et que l'axe a franchi la position de verrouillage. Mettre à zéro lorsqu'une instruction MCLM ou MCCM est initiée. Pour l'énumération Avant immédiat uniquement et Inverse immédiat uniquement, le bit est défini immédiatement lorsque MCLM or MCCM est initié.</p> <p>Lorsque l'énumération est Position avant uniquement ou Position inverse uniquement, les bit est défini lorsque l'Axe maître franchit la Position de verrouillage dans le sens spécifié. Le bit n'est jamais défini si l'énumération est AUCUN.</p> <p>Le bit CoordMotionLockStatus est mis à zéro lorsque l'axe maître inverse le sens et que l'axe esclave s'arrête en suivant l'axe maître. Le bit CoordMotionLockStatus est de nouveau défini lorsque le système de coordonnées esclave reprend en suivant l'axe maître. Le bit CoordMotionLockStatus est également mis à zéro lors de l'initialisation d'un MCS.</p>

coordinateDefinition	UDINT	GSV	Définition des coordonnées des coordonnées de la géométrie
zeroAngleOffset4	REAL	GSV/SSV	Orientation angle zéro pour le quatrième axe des géométries non cartésiennes.
zeroAngleOffset5	REAL	GSV/SSV	Orientation angle zéro pour le cinquième axe des géométries non cartésiennes.
zeroAngleOffset6	REAL	GSV/SSV	Orientation angle zéro pour le sixième axe des géométries non cartésiennes.
linkLength3	REAL	GSV/SSV	Longueur linéaire de la liaison du poignet d'un robot.
ballScrewPitch	REAL	GSV/SSV	Pas d'une vis couplée indépendante SCARA.
ActiveToolFrameID	DINT	GSV/étiquette	Identificateur d'outil actif spécifié par l'utilisateur dans l'instruction MCTO.
MaxOrientationSpeed	REAL	GSV/SSV	Vitesse maximale des axes d'orientation du système de coordonnées.
MaxOrientationAccel	REAL	GSV/SSV	Accélération maximale des axes d'orientation du système de coordonnées.
MaxOrientationDecel	REAL	GSV/SSV	Décélération maximale des axes d'orientation du système de coordonnées.
ActiveWorkFrameID	REAL	GSV/étiquette	Cadre de travail actif
SwingArmOffsetA3	REAL	GSV/SSV	Décalage le long de l'axe X à partir du centre de la plaque de base inférieure jusqu'au cadre de la jointure 4 pour la géométrie Delta à 5 axes.
SwingArmOffsetD3	REAL	GSV/SSV	Décalage le long de l'axe Z à partir du centre de la plaque de base inférieure jusqu'au cadre de la jointure 4 pour la géométrie Delta à 5 axes.
SwingArmOffsetA4	REAL	GSV/SSV	Décalage le long du cadre J4 sur l'axe X jusqu'au cadre de la jointure 5 pour la géométrie Delta à 5 axes.
SwingArmOffsetD4	REAL	GSV/SSV	Décalage le long du cadre J4 sur l'axe Z jusqu'au cadre de la jointure 5 pour la géométrie Delta à 5 axes.
SwingArmOffsetD5	REAL	GSV/SSV	Décalage le long du cadre J5 sur l'axe Z jusqu'au cadre EOA pour la géométrie Delta à 5 axes.
SwingArmCouplingRatioNum	UINT16	GSV/SSV	Le rapport entre l'axe de rotation et l'axe d'inclinaison.
SwingArmCouplingRatioDen	UINT16	GSV/SSV	Le rapport entre l'axe de rotation et l'axe d'inclinaison.
SwingArmCouplingDirection	UINT	GSV/SSV	Sens relatif de l'axe de rotation J4 couplé par rapport à l'axe d'inclinaison J5 pour la géométrie de robot Delta J1J2J3J4J5.

## Accès à l'objet MotionGroup

L'objet MOTIONGROUP fournit des informations d'état sur un groupe d'axes du servo-module. Indiquez le nom de l'étiquette du groupe de mouvements pour spécifier l'objet MOTIONGROUP qui vous intéresse.



Attribut	Type de donnée	Instruction	Description (Description)
Alternate1UpdateMultiplier	USINT	GSV	La période de mise à jour pour les axes associés à la planification de mise à jour de l'alternative 1.
Alternate1UpdatePeriod	UDINT	GSV	La période de mise à jour pour les axes associés à la planification de mise à jour de l'alternative 1. La valeur est le produit de la planification de mise à jour de l'alternative 1 et de la période intermédiaire de mise à jour.
Alternate2UpdateMultiplier	USINT	GSV	La période de mise à jour pour les axes associés à la planification de mise à jour de l'alternative 2.
Alternate2UpdatePeriod	UDINT	GSV	La période de mise à jour pour les axes associés à la planification de mise à jour de l'alternative 2. La valeur est le produit de la planification de mise à jour de l'alternative 1 et de la période intermédiaire de mise à jour.
AutoTagUpdate	USINT	GSV SSV	Contrôle la conversion automatique et la mise à jour des attributs d'état de mouvement.
CoarseUpdatePeriod	UDINT	GSV	La période intermédiaire de mise à jour, couramment appelée « période de mise à jour de base ».
Cycle Start Time	LTIME	GSV	Cette valeur de 64 bits (ms) correspond à l'événement Temporisateur qui déclenche le cycle de mise à jour.
INSTANCE	DINT	GSV	Le numéro d'instance de cet objet MOTION_GROUP
MaximumInterval	LTIME	GSV SSV	L'intervalle maximal entre deux exécutions successives de cette tâche.
MinimumInterval	LTIME	GSV	L'intervalle minimal entre deux exécutions successives de cette tâche.
StartTime	LTIME	GSV	La valeur de l'heure locale lors du démarrage de la dernière exécution de la tâche
TaskAverageIOTime	UDINT	GSV SSV	La durée moyenne qui s'écoule entre l'entrée et la sortie de la tâche de mouvement, c'est-à-dire le temps écoulé entre le démarrage de la tâche de mouvement et l'envoi des données de connexion. (constante de temps = 250 CUP)
TaskAverageScanTime	UDINT	GSV SSV	La durée de scrutation moyenne de la tâche de mouvement. (constante de temps = 250 CUP)
TaskLastIOTime	UDINT	GSV	La dernière durée écoulée entre l'entrée et la sortie de la tâche de mouvement, c'est-à-dire le temps écoulé entre le démarrage de la tâche de mouvement et l'envoi des données de connexion.
TaskLastScanTime	UDINT	GSV	La dernière durée de scrutation de la tâche de mouvement. (temps écoulé)
TaskMaximumIOTime	UDINT	GSV SSV	La durée maximale qui s'écoule entre l'entrée et la sortie de la tâche de mouvement, c'est-à-dire le temps écoulé entre le démarrage de la tâche de mouvement et l'envoi des données de connexion.
TaskMaximumScanTime	UDINT	GSV SSV	La durée de scrutation maximale de la tâche de mouvement. (temps écoulé)
Time Offset	LTIME	GSV	La valeur du décalage de temps entre l'heure locale et la valeur du temporisateur local, pour l'automate associé à la valeur actuelle de l'heure de démarrage du cycle.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

**Accès à l'objet Message**

Accédez à l'objet Message via les instructions GSV/SSV. Indiquez le nom de l'étiquette du message pour déterminer l'objet Message qui vous intéresse. L'objet Message offre une interface pour configurer et déclencher des communications d'égal à égal. Cet objet remplace le type de données MG du processeur PLC-5.

Attribut	Type de données	Instruction	Description
ConnectionPath	SINT[130]	GSV SSV	Données pour configurer le chemin de connexion. Les deux premiers octets (octet de poids faible et octet de poids fort) correspondent à la longueur en octets du chemin de connexion.
ConnectionRate	DINT	GSV SSV	Débit des paquets demandé de la connexion.
MessageType	SINT	GSV SSV	Indique le type de message. Signification de la valeur : <ul style="list-style-type: none"> <li>• 0. Non initialisé</li> </ul>
Port	SINT	GSV SSV	Indique le port sur lequel le message doit être envoyé. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 1. Fond de panier.</li> <li>• 2. Port série.</li> </ul>
Timeout Multiplicateur	SINT	GSV SSV	Détermine le délai d'expiration et de fermeture d'une connexion. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Le délai d'expiration de la connexion est atteint au bout de quatre fois la vitesse de mise à jour. Il s'agit du mode par défaut.</li> <li>• 1. Le délai d'expiration de la connexion est atteint au bout de huit fois la vitesse de mise à jour.</li> <li>• 2. Le délai d'expiration de la connexion est atteint au bout de 16 fois la vitesse de mise à jour.</li> </ul>
Unconnected Timeout	DINT	GSV SSV	Délai d'expiration en microsecondes pour tous les messages non connectés. La valeur par défaut est de 30 000 000 microsecondes (30 s).

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet CST

L'objet Temps système coordonné (CST) fournit un temps système de coordonnées pour les dispositifs d'un même châssis.

Attribut	Type de données	Instruction	Description
CurrentStatus	INT	GSV	<p>État actuel du temps système coordonné. Chaque bit a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• 0. Défaut matériel du temporisateur. Le temporisateur matériel interne du dispositif présente un défaut.</li> <li>• 1. Rampe activée. La valeur actuelle des 16 bits inférieurs du temporisateur augmente jusqu'à la valeur demandée, au lieu de passer à la valeur inférieure.</li> <li>• 2. Temps système maître. L'objet CST est une source de temps maître dans le système ControlLogix.</li> <li>• 3. Synchronisé. La valeur CurrentValue 64 bits de l'objet CST est synchronisée par l'objet CST maître via une mise à jour du temps système.</li> <li>• 4. Maître du réseau local. L'objet CST est la source de temps maître du réseau local.</li> <li>• 5. Mode Relais électromécanique. L'objet CST fonctionne en mode de relais électromécanique temporel.</li> <li>• 6. Maître en double détecté. Un temps maître en double a été détecté pour le réseau local. Ce bit est toujours à 0 pour les stations qui dépendent du temps.</li> <li>• 7. Inutilisé.</li> <li>• 8-9. 00. Station dépendant du temps.</li> <li>• 01. Station maître de temps.</li> <li>• 10. Station relais électromécanique de temps.</li> <li>• 11. Inutilisé.</li> <li>• 10-15. Inutilisé.</li> </ul>
CurrentValue	DINT[2]	GSV	<p>Valeur actuelle du temporisateur. DINT[0] contient les 32 bits inférieurs ; DINT[1] contient les 32 bits supérieurs. La source du temporisateur est ajustée pour correspondre à la valeur fournie dans les services de mise à jour et depuis la synchronisation du réseau local de communication. L'ajustement se fait soit par augmentation progressive jusqu'à la valeur demandée, soit par réglage immédiat à la valeur demandé, en fonction de la valeur de l'attribut CurrentStatus.</p>

### Voir aussi

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

**Accès à l'objet Datalog** L'objet DATALOG fournit des informations d'état sur un journal de données spécifique. Indiquez le nom du journal de données pour spécifier l'objet DATALOG qui vous intéresse.

Attribut	Type de donnée	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description (Description)
CaptureFull	BOOL	GSV	None	Un état qui indique : <ul style="list-style-type: none"> <li>• Soit que la saisie de données la plus récente arrête de collecter des échantillons ;</li> <li>• soit que les plus anciens échantillons dans la saisie de données la plus récente sont remplacés suite à un dépassement de la taille de la saisie.</li> </ul>
CollectionCapacity	DINT	GSV	None	Affiche la fréquence fournie par l'automate concernant le nombre d'octets qui peuvent être collectés par seconde pour chaque type d'automate. Le pourcentage de CPU pour le journal des données peut être calculé à partir de cette fréquence et du nombre d'octets que l'automate doit collecter pour tous les journaux de données configurés.
CollectionState	INT	GSV	None	Affiche l'état actuel de collecte des données du journal des données. Il peut être : <ul style="list-style-type: none"> <li>• <b>Hors ligne</b> – non connecté à l'automate.</li> <li>• <b>Désactivé</b> – le journal de données ne consigne pas les données tant qu'il n'est pas activé.</li> <li>• <b>Prêt pour le déclencheur</b> – prêt pour le déclencheur de lancement ou pour le déclencheur d'instantané. L'attente pour le déclencheur d'arrêt est intégrée à la Collecte d'échantillons. Cet état peut coexister avec une Saisie complète.</li> <li>• <b>Collecte d'échantillons</b> – la logique collecte les échantillons, aucun pré ou post-échantillon. L'état de collecte de pré-échantillons est intégré à l'état Prêt pour le déclencheur. Cet état peut coexister avec une Saisie complète.</li> <li>• <b>Collecte de post-échantillons</b> – le déclencheur d'arrêt a été activé et la collecte de post-échantillons a commencé. Cet état peut coexister avec une Saisie complète.</li> <li>• <b>Saisie complète</b> – soit la saisie de données la plus récente arrête de collecter des échantillons, soit les plus anciens échantillons dans la saisie de données la plus récente sont remplacés suite à un dépassement de la taille de la saisie. Cet état peut co-exister avec les états Prêt pour le déclencheur, Collecte</li> </ul>

				<p>d'échantillons, Collecte de post-échantillons ou Journal des données plein.</p> <ul style="list-style-type: none"> <li>• <b>Journal des données plein</b> – la consignation des données s'arrête suite à un dépassement des saisies de données. Cet état peut coexister avec une Saisie complète. La collecte des données peut être réactivée en émettant une commande de réinitialisation ou une commande d'effacement suivie d'une commande d'activation.</li> <li>• <b>Erroné</b> – un défaut est survenue et la collecte des données a été arrêtée. Plus aucune autre donnée ne sera collectée jusqu'à ce que le défaut soit supprimée et qu'un service de réinitialisation ou d'activation soit émis. Cet état peut coexister avec une Saisie complète.</li> </ul>
CurrentCaptureNumber	INT	GSV	None	Indique le numéro de la saisie en cours. Par exemple, si la configuration indique 10 saisies de données à conserver, le numéro de la saisie en cours est compris entre 1 et 10.
DataCapturesToKeep	SINT	GSV	None	Indique le nombre configuré de saisies de données à conserver pour le journal de données spécifié.
Activé	SINT	GSV	None	Indique si le journal de données spécifié est activé ou non.
FaultReason	INT	GSV	None	Indique la raison du défaut actuel.
PreviousCaptureUsedStorage	DINT	GSV	None	Indique la quantité d'espace de stockage utilisée par la saisie de données précédente.
ReservedStorage	DINT	GSV	None	Indique le pourcentage de l'espace de stockage total réservé pour stocker le journal de données actuel.
UsedStorage	DINT	GSV	None	Indique le pourcentage de l'espace de stockage total contenant actuellement les échantillons de données collectées pour le journal des données actuel.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet DF1

L'objet DF1 offre une interface avec le driver de communication DF1 que vous pouvez configurer pour le port série.

Attribut	Type de données	Instruction	Description
ACKTimeout	DINT	GSV	Durée d'attente pour la confirmation de la transmission d'un message (point à point et maître uniquement). Les valeurs autorisées sont comprises entre 0 et 32 767. Le retard en nombre de périodes de 20 ms. La valeur par défaut est 50 (1 seconde).
Diagnostic Compteurs	INT[19]	GSV	Tableau des compteurs de diagnostic pour le driver de communication DF1.

Décalage de mot		Point à point DF1	Maître-esclave DF1
0	Signature (0x0043)	Signature (0x0042)	Signature (0x0044)
1	Bits du modem	Bits du modem	Bits du modem
2	Paquets transmis	Paquets transmis	Paquets transmis
3	Paquets reçus	Paquets reçus	Paquets reçus
4	Paquets non remis	Paquets non remis	Paquets non remis
5	Inutilisé	Messages dont l'envoi a été retenté	Messages dont l'envoi a été retenté
6	NAK reçus	NAK reçus	Inutilisé
7	ENQ reçues	Paquets d'appel reçus	Inutilisé
8	Mauvais paquets avec NAK	Mauvais paquets sans ACK	Mauvais paquets sans ACK
9	Aucune mémoire n'a envoyé de NAK	Aucune mémoire sans ACK	Inutilisé
10	Paquets en double reçus	Paquets en double reçus	Paquets en double reçus
11	Mauvais caractères reçus	Inutilisé	Inutilisé
12	Nombre de récupérations DCD	Nombre de récupérations DCD	Nombre de récupérations DCD
13	Nombre de modems perdus	Nombre de modems perdus	Nombre de modems perdus
14	Inutilisé	Inutilisé	Temps de scrutation prioritaire maximal
15	Inutilisé	Inutilisé	Dernier temps de scrutation prioritaire
16	Inutilisé	Inutilisé	Temps de scrutation normale maximal
17	Inutilisé	Inutilisé	Dernier temps de scrutation normale
18	ENQ envoyées	Inutilisé	Inutilisé

Doublon Détection	SINT	GSV	Active la détection de messages en double. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Détection de messages en double désactivée.</li> <li>• Différent de zéro. Détection de messages en double activée.</li> </ul>
Embedded ResponseEnable	SINT	GSV	Active la fonctionnalité de réponse imbriquée (point à point uniquement). Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Fonctionnalité activée uniquement après la réception d'une réponse imbriquée. Il s'agit du mode par défaut.</li> <li>• 1. Fonctionnalité activée sans restrictions.</li> </ul>
EnableStoreFwd	SINT	GSV	Active le comportement « stockage et transmission » lors de la réception d'un message. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Ne pas transmettre le message.</li> <li>• Différent de zéro. Voir la table « stockage et transmission » lors de la réception d'un message. Il s'agit du mode par défaut.</li> </ul>
ENQTransmit Limite	SINT	GSV	Nombre de requêtes (ENQ) devant être envoyées lorsque le délai défini pour ACK (point à point uniquement). Les valeurs autorisées sont comprises entre 0 et 127. Le paramètre par défaut est 3.
EOTSuppression	SINT	GSV	Active la suppression des transmissions EOT en réponse aux paquets d'appel (esclave uniquement). Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Suppression EOT désactivée (désactivé).</li> <li>• Différent de zéro. Suppression EOT activée.</li> </ul>
ErrorDetection	SINT	GSV	Spécifie la méthode de détection d'erreur. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. BCC. Il s'agit du mode par défaut.</li> <li>• 1. CRC.</li> </ul>
MasterMessageTransmit	SINT	GSV	Valeur actuelle de la transmission de message maître (maître uniquement). Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Entre appels de la station. Il s'agit du mode par défaut.</li> <li>• 1. En séquence d'appel. Remplace le numéro de station du maître.</li> </ul>
MaxStation Adresse	SINT	GSV	Valeur actuelle (de 0 à 31) de l'adresse de station maximale sur un réseau DH-485. La valeur par défaut est 31.
NAKReceiveLimit	SINT	GSV	Le nombre de NAK reçus en réponse à un message avant d'arrêter la transmission (communication point à point uniquement). Valeurs valides entre 0 et 127. La valeur par défaut est 3.
NormalPollGroupSize	INT	GSV	Nombre de stations à appeler dans le tableau de station d'appel normal après avoir appelé toutes les stations du tableau de station d'appel prioritaire (maître uniquement). Valeurs valides entre 0 et 255. La valeur par défaut est 0.

PollingMode	SINT	GSV	Le mode d'appel actuel (maître uniquement). Le paramètre par défaut est 1. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Basé sur les messages ; les esclaves ne sont pas autorisés à initier des messages.</li> <li>• 1. Basé sur les messages ; les esclaves sont autorisés à initier des messages. Il s'agit du mode par défaut.</li> <li>• 2. Standard ; transfert d'un seul message par scrutation de station.</li> <li>• 3. Standard ; transfert de plusieurs messages par scrutation de station.</li> </ul>
ReplyMessage Wait	DINT	GSV	Temps d'attente (en tant que maître) après avoir reçu un ACK avant d'appeler l'esclave pour obtenir une réponse (maître uniquement). Valeurs valides entre 0 et 65 535. Le retard en nombre de périodes de 20 ms. La valeur par défaut est de 5 périodes (100 ms).
SlavePollTimeout	DINT	GSV	Durée, en ms, pendant laquelle l'esclave doit attendre l'appel du maître avant qu'il se déclare incapable de transmettre des messages car le maître est inactif (esclave uniquement). Valeurs valides entre 0 et 32 767. Le retard en nombre de périodes de 20 ms. La valeur par défaut est de 3 000 périodes (1 min.).
StationAddress	INT	GSV	Adresse de station actuelle du port série. Valeurs valides entre 0 et 254. La valeur par défaut est 0.
TokenHoldFactor	SINT	GSV	Valeur actuelle (de 1 à 4) de nombre maximal de messages envoyés par cette station avant de transmettre le jeton sur un réseau DH-485. La valeur par défaut est 1.
TransmitRetries	SINT	GSV	Nombre de tentatives de renvoi d'un message sans obtention de confirmation (maître et esclave uniquement). Valeurs valides entre 0 et 127. La valeur par défaut est 3.
PendingACK Timeout	DINT	SSV	Valeur en attente pour l'attribut ACKTimeout.
Pending Duplicate Detection	SINT	SSV	Valeur en attente pour l'attribut DuplicateDetection.
Pending Embedded ResponseEnable	SINT	SSV	Valeur en attente pour l'attribut EmbeddedResponse.
PendingEnable StoreFwd	SINT	SSV	Valeur en attente pour l'attribut EnableStoreFwd.
PendingENQ TransmitLimit	SINT	SSV	Valeur en attente pour l'attribut ENQTransmitLimit.
PendingEOT Suppression	SINT	SSV	Valeur en attente pour l'attribut EOTSuppression.
PendingError Détection	SINT	SSV	Valeur en attente pour l'attribut ErrorDetection.
PendingMaster Message Transmit	SINT	SSV	Valeur en attente pour l'attribut MasterMessageTransmit.
PendingMax StationAddress	SINT	SSV	Valeur en attente pour l'attribut MaxStationAddress.
PendingNAK ReceiveLimit	SINT	SSV	Valeur en attente pour l'attribut NAKReceiveLimit.



PendingNormalPollGroupSize	INT	SSV	Valeur en attente pour l'attribut NormalPollGroupSize.
PendingPollingMode	SINT	SSV	Valeur en attente pour l'attribut PollingMode.
PendingReplyMessageWait	DINT	SSV	Valeur en attente pour l'attribut ReplyMessageWait.
PendingSlavePollTimeout	DINT	SSV	Valeur en attente pour l'attribut SlavePollTimeout.
PendingStationAdresse	INT	SSV	Valeur en attente pour l'attribut StationAddress.
PendingTokenHoldFactory	SINT	SSV	Valeur en attente pour l'attribut TokenHoldFactor.
PendingTransmitRetries	SINT	SSV	Valeur en attente pour l'attribut TransmitRetries.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet FaultLog

L'objet FaultLog fournit des informations de défaut sur l'automate.

Attribut	Type de donnée	Instruction	Description (Description)
MajorEvents	INT	GSV SSV	Le nombre de défauts majeurs survenus depuis la dernière réinitialisation de ce compteur.
MajorFaultBits	DINT	GSV SSV	Les bits indiquent la raison du défaut majeur actuel. Chaque bit a une signification spécifique : 1 Perte d'alimentation 3 E/S 4 Exécution d'instruction (programme) 5 Gestionnaire de défauts 6 Chien de garde 7 Pile 8 Changement de mode 11 Mouvement
MinorEvents	INT	GSV SSV	Le nombre de défauts mineurs survenus depuis la dernière réinitialisation de ce compteur.
MinorFaultBits	DINT	GSV SSV	Les bits indiquent la raison du défaut mineur actuel. Chaque bit a une signification spécifique : 4 - Exécution d'instruction (programme) 6 - Chien de garde 9 - Port série 10 - Module de stockage d'énergie (ESM) ou alimentation sans interruption (UPS) 20 - La licence/une licence CodeMeter requise est introuvable ou a expiré.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

**Accès à l'objet  
HardwareStatus**

L'objet **HardwareStatus** permet d'obtenir des informations d'état pour l'alimentation sans interruption (UPS), les ventilateurs et les températures avec les instructions GSV pour les CompactLogix projets avec l'automate 5480. Cet objet est pris en charge lors des routines Diagramme à relais et Texte structuré et dans les instructions Complémentaire.

Attribut	Type de donnée		Instruction	Description (Description)
FanSpeeds	Structure de :		GSV	Vitesse des ventilateurs.
	Nombre de ventilateurs	USINT		Si le produit ne supporte aucun ventilateur (zéro), dans ce cas, l'appareil ne prend pas en compte les ventilateurs.
	Vitesse du ventilateur	SINT[9] pour 2 ventilateurs : SINT[0] = Nombre de ventilateurs SINT[1-4] = Vitesse du ventilateur n°1 SINT[5-8] = Vitesse du ventilateur n°2		RPM
FanStatus	Structure de :		GSV	Indique si le ventilateur présente un défaut.
	Nombre d'indicateurs d'état du ventilateur	USINT		Si le produit ne supporte aucun ventilateur (zéro), dans ce cas, l'appareil ne prend pas en compte l'état du ventilateur.
	État du ventilateur	SINT[3] pour 2 ventilateurs : SINT[0] = Nombre de ventilateurs SINT[1] = État du ventilateur n°1 SINT[2] = État du ventilateur n°2		<ul style="list-style-type: none"> <li>• 0 Le ventilateur ne présente aucun défaut</li> <li>• 1 Le ventilateur présente un défaut</li> </ul>
TemperatureFault Levels	Structure de :		GSV	Le niveau du défaut en degrés Celsius
	Nombre de niveau de défaut de température	USINT		Si le nombre de niveau de défaut de température est de zéro, dans ce cas, l'appareil ne prend pas en compte les niveaux de défaut de la température.
	Niveau de défaut de température	SINT[3] pour 1 capteur de température : SINT[0] = Nombre de niveaux de défaut de température SINT[1-2] = Niveau du défaut de température n°1		Température en degrés Celsius

Attribut	Type de donnée		Instruction	Description (Description)
Températures	Structure de :		GSV	Valeurs de la température en degrés Celsius
	Nombre de températures	USINT		Si le produit ne supporte aucune température (zéro), dans ce cas, l'appareil ne prend pas en compte les températures.
	Température	SINT[3] pour 1 capteur de température : SINT[0] = Nombre de températures SINT[1-2] = Température n°1		Température en degrés Celsius
UPSBatteryFailure	SINT		GSV	Indique si la batterie UPS présente un défaut. <ul style="list-style-type: none"> <li>• 0 La batterie UPS connectée n'a détecté aucun défaut.</li> <li>• 1 La batterie UPS connectée a détecté un problème avec la batterie connectée.</li> </ul>
UPSBuffering	SINT		GSV	Indique si l'UPS fournit une alimentation depuis la batterie. <ul style="list-style-type: none"> <li>• 0. L'UPS ne fournit aucune alimentation depuis la batterie.</li> <li>• 1. L'UPS fournit une alimentation depuis la batterie.</li> </ul>
UPSInhibited	SINT		GSV	Demande à l'UPS de couper l'alimentation. <ul style="list-style-type: none"> <li>• 0. L'automate ne souhaite pas que l'alimentation soit coupée à ce moment.</li> <li>• 1. L'UPS va couper l'alimentation.</li> </ul>
UPSReady	SINT		GSV	Indique si l'UPS est prête en se basant sur : un chargement >= à 85 %, l'absence de défaut dans le câblage, une tension d'entrée suffisante et un signal d'inhibition inactif. <ul style="list-style-type: none"> <li>• 0. UPS non prête</li> <li>• 1. UPS prête</li> </ul>
UPSSupported	SINT		GSV	Indique si l'UPS est prise en charge. <ul style="list-style-type: none"> <li>• 0. Non compatible</li> <li>• 1. Prise en charge</li> </ul>

## Accès à l'objet Message

Accédez à l'objet Message via les instructions GSV/SSV. Indiquez le nom de l'étiquette du message pour déterminer l'objet Message qui vous intéresse. L'objet Message offre une interface pour configurer et déclencher des communications d'égal à égal. Cet objet remplace le type de données MG du processeur PLC-5.

Attribut	Type de données	Instruction	Description
ConnectionPath	SINT[130]	GSV SSV	Données pour configurer le chemin de connexion. Les deux premiers octets (octet de poids faible et octet de poids fort) correspondent à la longueur en octets du chemin de connexion.
ConnectionRate	DINT	GSV SSV	Débit des paquets demandé de la connexion.
MessageType	SINT	GSV SSV	Indique le type de message. Signification de la valeur : <ul style="list-style-type: none"> <li>• 0. Non initialisé</li> </ul>
Port	SINT	GSV SSV	Indique le port sur lequel le message doit être envoyé. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 1. Fond de panier.</li> <li>• 2. Port série.</li> </ul>
Timeout Multiplicateur	SINT	GSV SSV	Détermine le délai d'expiration et de fermeture d'une connexion. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Le délai d'expiration de la connexion est atteint au bout de quatre fois la vitesse de mise à jour. Il s'agit du mode par défaut.</li> <li>• 1. Le délai d'expiration de la connexion est atteint au bout de huit fois la vitesse de mise à jour.</li> <li>• 2. Le délai d'expiration de la connexion est atteint au bout de 16 fois la vitesse de mise à jour.</li> </ul>
Unconnected Timeout	DINT	GSV SSV	Délai d'expiration en microsecondes pour tous les messages non connectés. La valeur par défaut est de 30 000 000 microsecondes (30 s).

### Voir aussi

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet Module

L'objet Module fournit des informations d'état sur un module. Pour sélectionner un objet Module particulier, définissez l'opérande de nom d'objet de l'instruction GSV/SSV sur le nom du module. Le module spécifié doit être présent dans la section Configuration des E/S de l'organisateur de l'automate, et doit avoir un nom.

Attribut	Type de données	Instruction	Description
EntryStatus	INT	GSV	<p>Indique l'état actuel de l'entrée d'adressage spécifiée. Les 12 bits inférieurs doivent être masqués lors des opérations de comparaison. Seuls les bits 12 à 15 sont valides. Chaque valeur a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• <b>16#0000. En attente.</b> L'automate est mis sous tension.</li> <li>• <b>16#1000. Anormal.</b> Une des connexions de l'objet Module du module associé a rencontré un problème. Cette valeur ne doit pas être utilisée pour déterminer si la défaillance du module est due au fait que l'objet Module quitte cet état régulièrement lorsqu'il tente de se reconnecter au module. Il est préférable pour cela de réaliser un test de l'état Exécution en cours (16#4000). Recherchez les attributs FaultCode dont la valeur n'est pas égale à 0 pour déterminer si un module présente un défaut. Si un défaut est présent, les attributs FaultCode et FaultInfo sont valides jusqu'à ce que la condition de défaut soit corrigée.</li> <li>• <b>16#2000. Validation en cours.</b> L'objet Module vérifie l'intégrité de l'objet Module avant d'établir les connexions au module.</li> <li>• <b>16#3000. Connexion en cours.</b> L'objet Module initie les connexions au module.</li> <li>• <b>16#4000. Exécution.</b> Toutes les connexions au module sont établies et les données sont en cours de transfert.</li> <li>• <b>16#5000. Fermeture.</b> L'objet Module est en train de fermer toutes les connexions au module.</li> <li>• <b>16#6000. Inactivé.</b> L'objet Module est inhibé (le bit d'inhibition de l'attribut Mode est défini).</li> <li>• <b>16#7000. Attente.</b> L'objet parent dont dépend cet objet Module ne s'exécute pas.</li> <li>• <b>16#9000. Mise à jour du firmware.</b> L'utilitaire de supervision de firmware tente de flasher le module.</li> <li>• <b>16#A000. Configuration.</b> L'automate est en train de charger la configuration dans le module.</li> </ul>
FaultCode	INT	GSV	Nombre qui identifie le défaut du module, le cas échéant.
FaultInfo	DINT	GSV	Fournit des informations spécifiques sur le code de défaut de l'objet Module.
Firmware SupervisorStatus	INT	GSV	<p>Identifie l'état de fonctionnement actuel de l'utilitaire de supervision de firmware. Signification des valeurs :</p> <ul style="list-style-type: none"> <li>• 0. Les mises à jour du module ne sont pas exécutées.</li> <li>• 1. Les mises à jour du module sont exécutées.</li> </ul>
ForceStatus	INT	GSV	<p>Indique l'état des forces. Signification des bits :</p> <ul style="list-style-type: none"> <li>• 0. Forces installées (1=où, 0=non).</li> <li>• 1. Forces activées (1=où, 0=non).</li> </ul>
Instance	DINT	GSV	Fournit le numéro d'instance de cet objet module.

LEDStatus	INT	GSV	<p>Spécifie l'état actuel de l'indicateur d'état des E/S à l'avant de l'automate.(1) Signification des valeurs :</p> <ul style="list-style-type: none"> <li>• 0. Indicateur d'état désactivé : aucun objet Module n'est configuré pour l'automate. (Il n'y a aucun module dans la section Configuration des E/S de l'organisateur de l'automate.)</li> <li>• 1. Voyant clignotant en rouge : aucun des objets Module ne s'exécute.</li> <li>• 2. Voyant clignotant en vert : au moins un objet Module ne s'exécute pas.</li> <li>• 3. Voyant vert continu : tous les objets Module s'exécutent.</li> </ul> <p>Vous ne saisissez pas de nom d'objet avec cet attribut, car cet attribut s'applique à l'ensemble des modules.</p>
Mode	INT	GSV SSV	<p>Indique le mode actuel de l'objet Module. Chaque bit a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• 0. Si défini, génère un défaut majeur si l'une des connexions de l'objet Module présente un défaut pendant que l'automate est en mode Exécution.</li> <li>• 2. Si défini, provoque le passage de l'objet Module dans l'état Inactivé après fermeture de toutes les connexions au module.</li> </ul>
Path	Tableau SINT	GSV	<p>Indique le chemin du module référencé. Il s'agit d'un nouvel attribut présent depuis la version 24. Chaque octet a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• 0-1. Longueur du chemin en octets. Si 0, la longueur du tableau SINT est insuffisante pour contenir le chemin du module renvoyé.</li> </ul> <p>Si la longueur du tableau SINT est insuffisante pour contenir ce chemin, le tableau est mis à zéro et un défaut mineur est consigné.</p>

(1) Bien que les automates 1756-L7x ne disposent pas en face avant d'un écran d'indication d'état, ils utilisent cette fonctionnalité.

**Voir aussi**

[Défauts du Module : 16#0000 - 16#00ff](#) sur la [page 271](#)

[Défauts du Module : 16#0100 - 16#01ff](#) sur la [page 273](#)

[Défauts du Module : 16#0200 - 16#02ff](#) sur la [page 279](#)

[Défauts du Module : 16#0300 - 16#03ff](#) sur la [page 280](#)

[Défauts du Module : 16#0800 - 16#08ff](#) sur la [page 283](#)

[Défauts du Module : 16#fd00 - 16#fdff](#) sur la [page 283](#)

[Défauts du Module : 16#fe00 - 16#feff](#) sur la [page 284](#)

[Défauts du Module : 16#ff00 - 16#ffff](#) sur la [page 287](#)

**Accès à l'objet Routine** L'objet Routine fournit des informations d'état sur une routine. Indiquez le nom de la routine pour spécifier l'objet Routine qui vous intéresse.

Attribut	Type de données	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description
Instance	DINT	GSV	GSV	Fournit le numéro d'instance de cet objet routine. Les valeurs correctes sont comprises entre 0 et 65 535.
Name	String	GSV	GSV	Nom de la routine.
SFCPaused	INT	GSV	Aucun	Dans une routine SFC, indique si la SFC est en pause. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. La SFC n'est pas en pause.</li> <li>• 1. La SFC est en pause.</li> </ul>
SFCResuming	INT	GSV SSV	Aucun	Dans une routine SFC, indique si la SFC reprend son exécution. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. La SFC ne s'exécute pas. Cet attribut est automatiquement défini sur 0 à la fin d'une scrutation au cours de laquelle le graphe a été exécuté.</li> <li>• 1. La SFC s'exécute. Les temporisateurs d'action et d'étape conservent leur valeur précédente s'ils sont configurés ainsi. Cet attribut est automatiquement défini sur 1 lors de la première scrutation après la sortie de pause d'un graphe.</li> </ul>

Voir aussi

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet Redondance

L'objet REDUNDANCY fournit des informations d'état sur le système de redondance.

Pour cette information	Obtenir cet attribut	Type de donnée	GSV/SSV	Description (Description)	
État de redondance du châssis complet	ChassisRedundancy State	INT	GSV	<b>Si</b>	<b>Alors</b>
				16#2	Principal avec secondaire synchronisé
				16#3	Principal avec secondaire disqualifié
				16#4	Principal sans secondaire
				16#10	Principal verrouillé pour mise à jour
État de redondance du châssis partenaire	PartnerChassis RedundancyState	INT	GSV	<b>Si</b>	<b>Alors</b>
				16#8	Secondaire synchronisé

				16#9	Secondaire disqualifié avec principal
				16#E	Sans partenaire
				16#12	Secondaire verrouillé pour mise à jour
État de redondance de l'automate	ModuleRedundancy State	INT	GSV	<b>Si</b>	<b>Alors</b>
				16#2	Principal avec secondaire synchronisé
				16#3	Principal avec secondaire disqualifié
				16#4	Principal sans secondaire
				16#6	Principal avec synchronisation du secondaire
				16#F	Principal se verrouillant pour mise à jour
				16#10	Principal verrouillé pour mise à jour
État de redondance du partenaire	PartnerModule RedundancyState	INT	GSV	<b>Si</b>	<b>Alors</b>
				16#7	Synchronisation du secondaire
				16#8	Secondaire synchronisé
				16#9	Secondaire disqualifié avec principal
				16#E	Sans partenaire
				16#11	Secondaire se verrouillant pour mise à jour
				16#12	Secondaire verrouillé pour mise à jour
Résultats des vérifications de la compatibilité avec l'automate partenaire	CompatibilityResults	INT	GSV	<b>Si</b>	<b>Alors</b>
				0	Indéterminé
				1	Pas de partenaire compatible
				2	Partenaire entièrement compatible
État du processus de synchronisation (qualification)	Qualification InProgress	INT	GSV	<b>Si</b>	<b>Alors</b>
				-1	La synchronisation (qualification) n'est pas en cours.
				0	Non pris en charge
				De 1 à 999	Pour les modules qui peuvent mesurer leur pourcentage d'exécution, le pourcentage de synchronisation (qualification) achevé.
				50	Pour les modules qui ne peuvent pas mesurer leur pourcentage d'exécution, la synchronisation (qualification) est en cours.
100	La synchronisation (qualification) est terminée.				



Correspondance ou non entre les paramètres de l'interrupteur à clé de l'automate et de son partenaire	KeyswitchAlarm	DINT	GSV	<b>Si</b>	<b>Alors</b>
				0	L'une des conditions suivantes est vraie : les interrupteurs à clé correspondent aucun partenaire présent
				1	Les interrupteurs à clé ne correspondent pas
Position de l'interrupteur à clé du partenaire	PartnerKeyswitch	DINT	GSV	<b>Si</b>	<b>Alors</b>
				0	Inconnu
				1	RUN
				2	PROG
				3	REM
État des défauts mineurs du partenaire (si l'attribut ModuleRedundancyState indique qu'un partenaire est présent)	PartnerMinorFaults	DINT	GSV	<b>Le bit suivant :</b>	<b>Correspond à ce défaut mineur</b>
				1	Défaut de mise sous tension
				3	Défaut d'E/S
				4	Problème lié à une instruction (programme)
				6	Superposition de tâche périodique (chien de garde)
				9	Problème de port série (non disponible pour les projets 1756-L7x)
				10	Batterie faible ou problème de module de stockage d'énergie
Mode du partenaire	PartnerMode	DINT	GSV	<b>Si</b>	<b>Alors</b>
				16#0	Mise sous tension
				16#1	Program
				16#2	Exécution
				16#3	Test
				16#4	Anormal
				16#5	Exécution à programme
				16#6	Test à programme
				16#7	Programme à exécution
				16#8	Test à exécution
				16#9	Exécution à test
				16#A	Programme à test
				16#B	Défaut
				16#C	Défaut à programme
Dans une paire de châssis redondants, identification d'un châssis particulier indifféremment de l'état du châssis	PhysicalChassisID	INT	GSV	<b>Si</b>	<b>Alors</b>
				0	Inconnu
				1	Châssis A
				2	Châssis B

Numéro de logement du module Redondance (par exemple, 1756-RM, 1756-RM2) dans le châssis	SRMSlotNumber	INT	GSV		
Taille du dernier chargement croisé Taille du dernier chargement croisé si vous aviez un châssis secondaire	LastDataTransferSize	DINT	GSV	Cet attribut indique la taille des données qui ont été ou qui auraient été chargées de façon croisée lors de la dernière scrutation. Taille en DINT (mots de 4 octets). Vous devez configurer l'automate pour la redondance. Vous n'avez pas besoin d'un châssis secondaire. Présence d'un châssis secondaire synchronisé	
				OUI	Indique le nombre de DINT qui ont été chargés de façon croisée lors de la dernière scrutation.
				NON	Indique le nombre de DINT qui auraient été chargés de façon croisée lors de la dernière scrutation.
Taille du chargement croisé maximal Taille du chargement croisé maximal si vous aviez un châssis secondaire	MaxDataTransferSize	DINT	GSV SSV	Taille en DINT (mots de 4 octets). Vous devez configurer l'automate pour la redondance. Vous n'avez pas besoin d'un châssis secondaire. Pour réinitialiser cette valeur, utilisez une instruction SSV avec une valeur source de 0. Présence d'un châssis secondaire synchronisé ?	
				OUI	Indique le plus grand nombre de DINT qui ont été chargés de façon croisée.
				NON	Indique le nombre de DINT qui auraient été chargés de façon croisée.
Mode du partenaire	PartnerMode	DINT	GSV	<b>Si</b>	<b>Alors</b>
				16#0	Mise sous tension
				16#1	Program
				16#2	Exécution
				16#3	Test
				16#4	Anormal
				16#5	Exécution à programme
				16#6	Test à programme
				16#7	Programme à exécution
				16#8	Test à exécution
				16#9	Exécution à test
16#A	Programme à test				

				16#B	Défaut
				16#C	Défaut à programme
Dans une paire de châssis redondants, identification d'un châssis particulier indifféremment de l'état du châssis	PhysicalChassisID	INT	GSV	<b>Si</b>	<b>Alors</b>
				0	Inconnu
				1	Châssis A
				2	Châssis B
Numéro de logement du module 1757-SRM dans le châssis	SRMSlotNumber	INT	GSV		
<ul style="list-style-type: none"> <li>Taille du dernier chargement croisé</li> <li>Taille du dernier chargement croisé si vous aviez un châssis secondaire</li> </ul>	LastDataTransferSize	DINT	GSV	Cet attribut indique la taille des données qui ont été ou qui auraient été chargées de façon croisée lors de la dernière scrutation.	
				<ul style="list-style-type: none"> <li>Taille en DINT (mots de 4 octets).</li> <li>Vous devez configurer l'automate pour la redondance.</li> <li>Vous n'avez pas besoin de châssis secondaire.</li> </ul>	
				Présence d'un châssis secondaire synchronisé ?	
				OUI	Indique le nombre de DINT qui ont été chargés de façon croisée lors de la dernière scrutation.
				NON	Indique le nombre de DINT qui auraient été chargés de façon croisée lors de la dernière scrutation.
<ul style="list-style-type: none"> <li>Taille du chargement croisé maximal</li> <li>Taille du chargement croisé maximal si vous aviez un châssis secondaire</li> </ul>	MaxDataTransferSize	DINT	GSV SSV	Cet attribut donne la taille maximale de l'attribut LastDataTransfer Size.	
				<ul style="list-style-type: none"> <li>Taille en DINT (mots de 4 octets).</li> <li>Vous devez configurer l'automate pour la redondance.</li> <li>Vous n'avez pas besoin de châssis secondaire.</li> <li>Pour réinitialiser cette valeur, utilisez une instruction SSV avec une valeur source de 0.</li> </ul>	
				Présence d'un châssis secondaire synchronisé ?	
				OUI	Indique le plus grand nombre de DINT qui ont été chargés de façon croisée.
				NON	Indique le nombre de DINT qui auraient été chargés de façon croisée.

## Accès à l'objet Programme

L'objet Programme fournit des informations d'état sur un programme. Indiquez le nom du programme pour spécifier l'objet Programme qui vous intéresse.

Attribut	Type de donnée	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description (Description)
DisableFlag	SINT	GSV SSV	None	Contrôle l'exécution de ce programme. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Exécution activée.</li> <li>• Différent de zéro. Exécution désactivée.</li> </ul>
	DINT	GSV	GSV	Une valeur non égale à zéro désactive la fonctionnalité.
LastScanTime	DINT	GSV SSV	None	Durée de la dernière exécution du programme. Temps en microsecondes.
MajorFault Record	DINT[11]	GSV SSV	GSV SSV	Enregistre les défauts majeurs pour ce programme

**Astuce :** Rockwell Automation recommande de créer une structure définie par l'utilisateur pour simplifier l'accès à l'attribut MajorFaultRecord :

Nom (Name)	Type de donnée	Style	Description (Description)	
TimeLow	DINT	Décimal	Les 32 bits inférieurs de la valeur d'horodatage de défaut	
TimeHigh	DINT	Décimal	Les 32 bits supérieurs de la valeur d'horodatage de défaut	
Type (Type)	INT	Décimal	Type de défaut (programme, E/S, etc.)	
Code	INT	Décimal	Code unique du défaut (dépend du type de défaut)	
Info	DINT[8]	Hexadécimal	Informations sur le défaut (dépend du code et du type de défaut)	
MaxScanTime	DINT	GSV SSV	None	Durée d'exécution maximale enregistrée pour ce programme. Temps en microsecondes.
Nom (Name)	Chaîne	GSV	GSV	Nom du programme.

### Voir aussi

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accéder à l'objet Safety

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'objet Safety Controller fournit des informations sur l'état de sécurité et la signature de sécurité. Les attributs SafetyTask et SafetyFaultRecord peuvent récupérer les informations sur les défauts irrécupérables.

Reportez-vous à [Manuel utilisateur des automates GuardLogix](#), publication [1756-UM020](#).

Attribut	Type de données	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description
SafetyLockedState	SINT	GSV	Aucun	Indique si l'automate est verrouillé ou déverrouillé de sécurité.
SafetySILConfiguration	SINT	GSV	Aucun	Indique la configuration de sécurité SIL. <ul style="list-style-type: none"> <li>• 2 -- SIL2/PLd</li> <li>• 3 -- SIL3/PLe</li> </ul>
SafetyStatus	INT	GSV	Aucun	Indique l'état de sécurité. Chaque valeur a une signification spécifique : : <ul style="list-style-type: none"> <li>• 1000000000000000 -- Tâche de sécurité OK.</li> <li>• 10000000000000001 -- Tâche de sécurité inexploitable.</li> <li>• 0000000000000000 -- Partenaire manquant.</li> <li>• 00000000000000001 -- Partenaire non disponible.</li> <li>• 00000000000000010 -- Matériel incompatible.</li> <li>• 00000000000000011 -- Firmware incompatible.</li> </ul>
SafetySignature Exists	SINT	GSV	GSV	Indique si la signature de la tâche de sécurité est présente.
SafetySignature ID (uniquement applicable à Automates Compact GuardLogix 5370 et GuardLogix 5570)	SINT	GSV	Aucun	Numéro d'identification 32 bits.
SafetySignature (uniquement applicable à Automates Compact GuardLogix 5370 et GuardLogix 5570)	String	GSV	Aucun	Le numéro d'identification 32 bits inclut le numéro d'ID, la date et l'horodatage.
SafetyTaskFault Record	DINT[11]	GSV	Aucun	Enregistre les défauts de la tâche de sécurité.
SafetySignatureIDLong (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580)	SINT[33]	GSV	Aucun	ID de la signature de sécurité 32 octets dans un tableau en octets. Le 1er octet est la taille de l'ID de la signature de sécurité en octets et les 31 octets restants sont ID de signature.
SafetySignatureIDHex (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580).	String	GSV	Aucun	Représentation de chaîne hexadécimale 64 caractères de l'ID de signature.
SafetySignatureDateTime (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580).	String	GSV	Aucun	Jour et heure de la signature de sécurité sur 27 caractères au format mm/dd/yyyy, hh:mm:ss.iii<AM ou PM>>

## Accès à l'objet SerialPort

L'objet SerialPort offre une interface pour le port de communication série.

Attribut	Type de données	Instruction	Description
BaudRate	DINT	GSV	Indique la vitesse de transmission. Les valeurs possibles sont les suivantes : 110, 300, 600, 1 200, 2 400, 4 800, 9 600 et 19 200 (valeur par défaut).
ComDriverID	SINT	GSV	Indique le pilote. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0xA2. DF1. Il s'agit du mode par défaut.</li> <li>• 0xA3. ASCII.</li> </ul>
DataBits	SINT	GSV	Indique le nombre de bits de données par caractère. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 7. Sept bits de données. ASCII uniquement.</li> <li>• 8. Huit bits de données. Il s'agit du mode par défaut.</li> </ul>
DCDDelay	INT	GSV	Indique la durée à attendre pour que la détection de porteuse de données (DCD) atteigne un niveau bas avant de déclarer une erreur de paquet. Le retard est donné en nombre de paquets d'1 s. La valeur par défaut est 0 compteur.
Parity	SINT	GSV	Indique la parité. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Aucune parité. Il s'agit du mode par défaut.</li> <li>• 1. Parité impaire. ASCII uniquement.</li> <li>• 2. Parité paire.</li> </ul>
RTSOffDelay	INT	GSV	Durée qui doit s'écouler avant la désactivation de la ligne RTS une fois le dernier caractère transmis. Valeurs autorisées : De 0 à 32 767 Le retard en nombre de périodes de 20 ms. La valeur par défaut est de 0 milliseconde.
RTSSendDelay	INT	GSV	Durée qui doit s'écouler avant la transmission du premier caractère d'un message une fois la ligne RTS activée. Valeurs autorisées : De 0 à 32 767 Le retard en nombre de périodes de 20 ms. La valeur par défaut est de 0 milliseconde.
StopBits	SINT	GSV	Indique le nombre de bits d'arrêt. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 1. Un bit d'arrêt. Il s'agit du mode par défaut.</li> <li>• 2. Deux bits d'arrêt. ASCII uniquement.</li> </ul>
PendingBaudRate	DINT	SSV	Valeur en attente pour l'attribut BaudRate.
PendingCOM DriverID	SINT	SSV	Valeur en attente pour l'attribut COMDriverID.
PendingDataBits	SINT	SSV	Valeur en attente pour l'attribut DataBits.
PendingDCD Delay	INT	SSV	Valeur en attente pour l'attribut DCDDelay.
PendingParity	SINT	SSV	Valeur en attente pour l'attribut Parity.
PendingRTSOff Delay	INT	SSV	Valeur en attente pour l'attribut RTSOffDelay.
PendingRTSSendDelay	INT	SSV	Valeur en attente pour l'attribut RTSSendDelay.
PendingStopBits	SINT	SSV	Valeur en attente pour l'attribut StopBits.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

**Accès à l'objet Tâche**

L'objet TASK fournit des informations d'état sur une tâche. Indiquez le nom de la tâche pour spécifier l'objet TASK qui vous intéresse.

Attribut	Type de données	Instruction au sein d'une Tâche Standard	Instruction au sein d'une Tâche de Sécurité	Description
DisableUpdateOutputs	DINT	GSV SSV	Aucun	Active ou désactive le traitement des sorties à la fin d'une tâche. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour activer le traitement des sorties à la fin d'une tâche.</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour désactiver le traitement des sorties à la fin d'une tâche.</li> </ul>
EnableTimeOut	DINT	GSV SSV	Aucun	Active ou désactive la fonction de délai d'expiration d'une tâche d'événement. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour désactiver la fonction de délai d'expiration.</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour activer la fonction de délai d'expiration.</li> </ul>
InhibitTask	DINT	GSV SSV	Aucun	Empêche la tâche de s'accomplir. Si la tâche est inhibée, l'automate procède quand même à la pré-scrutation de la tâche lorsqu'il est en train de passer du mode Programme au mode Exécution ou Test. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour activer la tâche</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour inhiber (désactiver) la tâche</li> </ul>
Instance	DINT	GSV	GSV	Fournit le numéro d'instance de cet objet TASK. Les valeurs valides sont comprises entre 0 et 31.
LastScanTime	DINT	GSV SSV	Aucun	Durée de la dernière exécution du programme lors de son dernier lancement. Temps en microsecondes.

MaximumInterval	DINT[2]	GSV SSV	Aucun	L'intervalle de temps maximal entre deux exécutions successives de la tâche. DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur. Une valeur de 0 signifie 1 exécution ou moins de la tâche.
MaximumScanTime	DINT	GSV SSV	Aucun	Durée d'exécution maximale enregistrée pour ce programme. Temps en microsecondes.
MinimumInterval	DINT[2]	GSV SSV	Aucun	L'intervalle de temps minimal entre deux exécutions successives de la tâche. DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur. Une valeur de 0 signifie 1 exécution ou moins de la tâche.
Name	String	GSV	GSV	Nom de la tâche.
OverlapCount	DINT	GSV SSV	GSV SSV	Nombre de fois que la tâche a été déclenchée alors qu'elle était déjà en cours d'exécution. Valide pour un événement ou une tâche périodique. Pour effacer le compteur, définir l'attribut sur 0.
Priority	INT	GSV SSV	GSV	Priorité relative de cette tâche par rapport aux autres tâches. Les valeurs autorisées sont comprises entre 0 et 15.
Rate	DINT	GSV SSV	GSV	L'intervalle de temps entre deux exécutions de la tâche. Temps en microsecondes.
StartTime	DINT[2]	GSV SSV	Aucun	Valeur de WALLCLOCKTIME lors du démarrage de la dernière exécution de la tâche. DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur.
Status	DINT	GSV SSV	Aucun	Fournit les informations d'état relatives à la tâche. Une fois que l'automate définit l'un de ces bits, il faut les mettre à zéro manuellement. Pour déterminer si : <ul style="list-style-type: none"> <li>• une instruction EVENT a déclenché la tâche (uniquement la tâche d'événement), examiner le bit 0</li> <li>• un délai d'expiration a déclenché la tâche (uniquement la tâche d'événement), examiner le bit 1</li> <li>• un chevauchement s'est produit pour cette tâche, examiner le bit 2</li> </ul>



Watchdog	DINT	GSV SSV	GSV	<p>Limite de temps pour l'exécution de tous les programmes associés à cette tâche. Temps en microsecondes.</p> <p>Si vous saisissez 0, les valeurs suivantes sont attribuées :</p> <p><b>Durée :</b> 0,5 s 5,0 s</p> <p><b>Type de tâche :</b> périodique continue</p>
----------	------	------------	-----	--

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)

## Accès à l'objet TimeSynchronize

L'objet TIMESYNCHRONIZE fournit une interface CIP (Common Industrial Protocol) à la norme IEEE 1588 (IEC 61588) pour un protocole de synchronisation d'horloges de précision pour les systèmes de contrôle et de mesure en réseau. Vous pouvez accéder à l'objet TIMESYNCHRONIZE via les instructions GSV/SSV.

Pour plus d'informations sur cet objet, reportez-vous au document Integrated Architecture® and CIP Sync Configuration Application Techniques, publication IA-AT003.

Attribut	Type de données	Instruction	Description	
ClockType	INT	GSV	Type de l'horloge.	
			<b>Bit</b>	<b>Type d'horloge</b>
			0	Horloge normale
			1	Horloge limite
			2	Horloge transparente égal à égal
			3	Horloge transparente de bout en bout
			4	Station de gestion
			Tous les autres bits sont réservés.	
CurrentTimeMicroseconds	LINT	GSV	Valeur actuelle du temps système, en microsecondes.	
CurrentTimeNanoseconds	LINT	GSV	Valeur actuelle du temps système, en nanosecondes.	
DomainNumber	SINT	GSV	Le domaine de l'horloge PTP. La valeur est comprise entre 0 et 255. Le paramètre par défaut est de 0.	
CurrentTimeMicroseconds	LINT	GSV	Valeur actuelle du temps système, en microsecondes.	

CurrentTimeNanoseconds	LINT	GSV	Valeur actuelle du temps système, en nanosecondes.						
DomainNumber	SINT	GSV	Le domaine de l'horloge PTP. La valeur est comprise entre 0 et 255. Le paramètre par défaut est de 0.						
GrandMasterClockInfo	Structure	GSV	Informations de propriété pour l'horloge maître. Nécessite 24 octets de stockage.						
<b>Structure des informations de l'horloge maître :</b>									
ClockIdentity	SINT[8]								
ClockClass	INT								
TimeAccuracy	INT								
OffsetScaledLogVariance	INT								
CurrentUtcOffset	INT								
TimePropertyFlags	INT								
TimeSource	INT								
Priority1	INT								
Priority2	INT								
IsSynchronized	DINT	GSV	L'horloge locale est synchronisée avec un maître. <table border="1"> <thead> <tr> <th>Valeur</th> <th>Signification</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Non synchronisée</td> </tr> <tr> <td>1</td> <td>Synchronisée</td> </tr> </tbody> </table>	Valeur	Signification	0	Non synchronisée	1	Synchronisée
Valeur	Signification								
0	Non synchronisée								
1	Synchronisée								
LocalClockInfo	Structure	GSV	Informations de propriété pour l'horloge locale. Nécessite 20 octets de stockage.						
<b>Structure des informations de l'horloge locale :</b>									
ClockIdentity	SINT[8]								
ClockClass	INT								
TimeAccuracy	INT								
OffsetScaledLogVariance	INT								
CurrentUtcOffset	INT								
TimePropertyFlags	INT								
TimeSource	INT								
ManufacturerIdentity	DINT			GSV	L'IEEE OUI (Identité unique de l'organisation) du fabricant.				
MaxOffsetFromMaster	LINT	GSV / SSV	Décalage maximal par rapport au maître, en nanosecondes.						
MeanPathDelayToMaster	LINT	GSV	Retard moyen du chemin entre les horloges maître et locale, en nanosecondes.						
NumberOfPorts	INT	GSV	Le nombre de ports de cette horloge.						
OffsetFromMaster	LINT	GSV	La différence calculée entre l'horloge locale et l'horloge maître, basée sur le message de synchronisation le plus récent, en nanosecondes.						

PTPEnable	DINT	GSV / SSV	Le statut d'activation pour la synchronisation de temps/PTP/CIP sur le dispositif.	
			<b>Valeur</b>	<b>Signification</b>
			0	Désactiver
			1	Enabled
ParentClockInfo	Structure	GSV	Informations de propriété pour l'horloge parent. Nécessite 16 octets de stockage.	
<b>Structure des informations de l'horloge parent :</b>				
ClockIdentity	SINT[8]			
PortNumber	INT			
ObservedOffsetScaledLogVariance	INT			
ObservedPhaseChangeRate	DINT			
PortEnableInfo	Structure	GSV	La configuration d'activation de port pour chaque port du dispositif. Taille = 2 + (nombre de ports activés x 4) Taille max. = 42 octets	
<b>Structure de l'état d'activation de port :</b>				
NumberOfPorts	INT		Le nombre maximal de ports est de 10.	
<i>Structure répétée pour le nombre de ports :</i>				
PortNumber	INT			
PortEnable	INT			
PortLogAnnounceIntervallInfo	Structure	GSV	L'intervalle entre deux messages « Annonce » (Announce) successifs envoyés par une horloge maître sur chaque port PTP du dispositif. Taille = 2 + (nombre de ports activés x 4) Taille max. = 42 octets	
<b>Structure de l'intervalle d'annonce de consignment de port :</b>				
NumberOfPorts	INT		Le nombre maximal de ports est de 10.	
<i>Structure répétée pour le nombre de ports :</i>				
PortNumber	INT			
PortLogAnnounceInterval	INT			
PortLogSyncIntervallInfo	Structure	GSV	L'intervalle entre deux messages « Synchronisation » (Sync) successifs envoyés par un maître sur chaque port PTP du dispositif. Taille = 2 + (nombre de ports activés x 4) Taille max. = 42 octets	

<b>Structure de l'intervalle de synchronisation de consignation de port :</b>			
NumberOfPorts	INT		Le nombre maximal de ports est de 10.
<i>Structure répétée pour le nombre de ports :</i>			
PortNumber	INT		
PortLogAnnounceInterval	INT		
PortPhysicalAddressInfo	Structure	GSV	L'adresse physique et de protocole de chaque port du dispositif. Taille = 2 + (nombre de ports activés x 36) Taille max. = 362 octets
<b>Structure de l'adresse physique du port :</b>			
NumberOfPorts	INT		Le nombre maximal de ports est de 10.
<i>Structure répétée pour le nombre de ports :</i>			
PortNumber	INT		
Protocol	SINT[16]		
SizeOfAddress	INT		
Port Address	SINT[16]		
PortProfileIdentityInfo	Structure	GSV	Profil de chaque port du dispositif. Taille = 2 + (nombre de ports activés x 10) Taille max. = 102
<b>Structure de l'identité du profil de port :</b>			
NumberOfPorts	INT		Le nombre maximal de ports est de 10.
<i>Structure répétée pour le nombre de ports :</i>			
PortNumber	INT		
ClockIdentity	SINT[8]		
PortProtocolAddressInfo	Structure	GSV	L'adresse réseau et de protocole de chaque port du dispositif. Taille = 2 + (nombre de ports activés x 22) Taille max. = 222
<b>Structure de l'adresse de protocole du port :</b>			
NumberOfPorts	INT		Le nombre maximal de ports est de 10.
<i>Structure répétée pour le nombre de ports :</i>			
PortNumber	INT		
NetworkProtocol	INT		
SizeOfAddress	INT		
PortAddress	SINT[16]		
PortStateInfo	Structure	GSV	L'état actuel de chaque port PTP du dispositif. Taille = 2 + (nombre de ports activés x 4) Taille max. = 42 octets

<b>Structure de l'état du port :</b>			
NumberOfPorts	INT		Le nombre maximal de ports est de 10.
<i>Structure répétée pour le nombre de ports :</i>			
PortNumber	INT		
PortState	INT		
Priority1	SINT	GSV / SSV	Valeur Priority1 (outrépassement du maître) pour l'horloge locale. <b>Astuce</b> : La valeur est non signée.
Priority2	SINT	GSV / SSV	Valeur Priority2 (disjoncteur) pour l'horloge locale. <b>Astuce</b> : La valeur est non signée.
ProductDescription	Structure	GSV	Description du dispositif qui contient l'horloge. Nécessite 68 octets de stockage.
<b>Structure de la description du produit :</b>			
Size	DINT		
Description	SINT[64]		
RevisionData	Structure	GSV	Données de version du dispositif qui contient l'horloge. Nécessite 36 octets de stockage.
<b>Structure des données de version :</b>			
Size	DINT		
Revision	SINT[32]		
StepsRemoved	INT	GSV	Le nombre de régions de synchronisation CIP entre l'horloge locale et l'horloge maître (c'est-à-dire le nombre d'horloges limites + 1).
SystemTimeAndOffset	Structure	GSV	Le temps système, en microsecondes, et le décalage par rapport à la valeur de l'horloge locale.
<b>Structure du temps système et du décalage :</b>			
SystemTime	LINT		
SystemOffset	LINT		
UserDescription	Structure	GSV	Description utilisateur du dispositif qui contient l'horloge. Nécessite 132 octets de stockage.
<b>Structure de la description utilisateur :</b>			
Size	DINT		
Description	SINT[128]		

## Accès à l'objet WallClockTime

L'objet WallClockTime fournit un horodatage que l'automate peut utiliser pour la planification.

**Astuce :** Le réglage de l'objet WALLCLOCKTIME est limité à pas plus d'une mise à jour toutes les 15 secondes.

---

**Important :** Pour vous assurer que l'heure correcte est lue en utilisant l'instruction GSV, incluez l'instruction GSV WALLCLOCKTIME dans une seule tâche utilisateur.

---



---

**Important :** Pour vous assurer que l'heure correcte est lue en utilisant l'instruction GSV, placez la paire d'instructions UID/UIE autour des instances de l'instruction GSV WALLCLOCKTIME dans les tâches utilisateur qui peuvent être interrompues par les instances de l'instruction GSV WALLCLOCKTIME dans d'autres tâches. Aucune paire UID/UIE n'est requise lorsque l'instruction GSV WALLCLOCKTIME existe dans une seule tâche utilisateur.

---

Attribut	Type de donnée	Instruction	Description (Description)
ApplyDST	SINT	GSV SSV	Permet d'identifier s'il faut activer l'heure d'été. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• 0. Ne pas ajuster à l'heure d'été.</li> <li>• Différent de zéro. Ajuster à l'heure d'été.</li> </ul>
CSTOffset	DINT[2]	GSV SSV	Décalage positif à partir de CurrentValue de l'objet CST (temps système coordonné). DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur. Valeur en microns. Le paramètre par défaut est de 0.
CurrentValue	DINT[2]	GSV SSV	Valeur actuelle de l'heure locale. DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur. Nombre de microsecondes qui se sont écoulées depuis 0000 heure le 1er janvier 1970. Les objets CST et WALLCLOCKTIME sont liés mathématiquement dans l'automate. Par exemple, si vous ajoutez CST CurrentValue et WALLCLOCKTIME CSTOffset, le résultat est WALLCLOCKTIME CurrentValue.
DateTime	DINT[7]	GSV SSV	Date et heure. Chaque valeur a une signification spécifique : <ul style="list-style-type: none"> <li>• DINT[0]. Année</li> <li>• DINT[1]. Mois (1 à 12)</li> <li>• DINT[2]. Jour (1 à 31)</li> <li>• DINT[3]. Heure (0 à 23)</li> <li>• DINT[4]. Minute (0 à 59)</li> <li>• DINT[5]. Secondes (0 à 59)</li> <li>• DINT[6]. Microsecondes (0 à 999 999)</li> </ul>
DSTAdjustment	INT	GSV SSV	Nombre de minutes pour l'ajustement à l'heure d'été.

LocalDateTime	DINT[7]	GSV SSV	<p>Heure locale actuelle ajustée. Chaque valeur a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• DINT[0]. Année</li> <li>• DINT[1]. Mois (1 à 12)</li> <li>• DINT[2]. Jour (1 à 31)</li> <li>• DINT[3]. Heure (0 à 23)</li> <li>• DINT[4]. Minute (0 à 59)</li> <li>• DINT[5]. Secondes (0 à 59)</li> <li>• DINT[6]. Microsecondes (0 à 999 999)</li> </ul>
TimeZoneString	INT	GSV SSV	Fuseau horaire de la valeur de temps.

**Voir aussi**

[Types et codes des défauts majeurs](#) sur la [page 163](#)

[Types et codes des défauts mineurs](#) sur la [page 169](#)


## Objets de Sécurité GSV/SSV

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Pour les tâches de sécurité, les instructions GSV et SSV sont plus limitées.

**Astuce :** Les instructions SSV pour les tâches de sécurité et standard ne peuvent pas définir le bit 0 (défaut majeur lors d’une erreur) dans l’attribut de mode d’un module d’E/S de sécurité.

Pour les objets de sécurité, le tableau suivant montre pour quels attributs vous pouvez obtenir des valeurs en utilisant l’instruction GSV et quels attributs vous pouvez définir en utilisant l’instruction SSV dans les tâches de sécurité et standard.

 **ATTENTION :** utilisez les instructions GSV/SSV avec prudence. Les changements apportés aux objets peuvent déclencher un fonctionnement inattendu de l’automate ou blesser les personnes.

Objet de sécurité	Nom d’attribut	Description d’attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
			GSV	SSV	GSV	SSV
Tâche de sécurité	Instance	Fournit le numéro d’instance de cet objet de tâche. Les valeurs valides sont comprises entre 0 et 31.	✓		✓	

Objet de sécurité	Nom d'attribut	Description d'attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
	MaximumInterval	L'intervalle de temps maximal entre deux exécutions successives de la tâche.			✓	✓
	MaximumScanTime	Temps d'exécution maximal enregistré (ms.) pour cette tâche.			✓	✓
	MinimumInterval	L'intervalle de temps minimal entre deux exécutions successives de la tâche.			✓	✓
	Priority	Priorité relative de cette tâche par rapport aux autres tâches. Les valeurs valides sont comprises entre 0 et 15.	✓		✓	
	Rate	Période de la tâche (en ms.), ou valeur du délai d'expiration de la tâche (en ms.).	✓		✓	
	Watchdog	Limite de temps (en ms.) pour l'exécution de tous les programmes associés à cette tâche.	✓		✓	
	<b>DisableUpdateOutputs</b>	Active ou désactive le traitement des sorties à la fin d'une tâche. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour activer le traitement des sorties à la fin d'une tâche.</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour désactiver le traitement des sorties à la fin d'une tâche.</li> </ul>			✓	
	<b>EnableTimeOut</b>	Active ou désactive la fonction de délai d'expiration d'une tâche. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour désactiver la fonction de délai d'expiration.</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour activer la fonction de délai d'expiration.</li> </ul>			✓	



Objet de sécurité	Nom d'attribut	Description d'attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
	<b>InhibitTask</b>	Empêche la tâche de s'accomplir. Si la tâche est inhibée, l'automate procède quand même à la pré-scrutation de la tâche lorsqu'il est en train de passer du mode Programme au mode Exécution ou Test. <ul style="list-style-type: none"> <li>• Définir l'attribut sur 0 pour activer la tâche</li> <li>• Définir l'attribut sur 1 (ou toute valeur différente de zéro) pour inhiber (désactiver) la tâche</li> </ul>			✓	
	<b>LastScanTime</b>	Durée de la dernière exécution du programme lors de son dernier lancement. Temps en microsecondes.			✓	
	<b>Nom (Name)</b>	Nom de la tâche				
	<b>OverlapCount</b>	Nombre de fois que la tâche a été déclenchée alors qu'elle était déjà en cours d'exécution. Valide pour un événement ou une tâche périodique. Pour effacer le compteur, définir l'attribut sur 0.			✓	
	<b>StartTime</b>	Valeur de WALLCLOCKTIME lors du démarrage de la dernière exécution de la tâche. DINT[0] contient les 32 bits inférieurs de la valeur ; DINT[1] contient les 32 bits supérieurs de la valeur.			✓	

Objet de sécurité	Nom d'attribut	Description d'attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
	<b>État (Status)</b>	<p>Fournit les informations d'état relatives à la tâche. Une fois que l'automate définit l'un de ces bits, il faut les mettre à zéro manuellement.</p> <p>Pour déterminer si :</p> <ul style="list-style-type: none"> <li>• une instruction EVENT a déclenché la tâche (uniquement la tâche d'événement), examiner le bit 0</li> <li>• un délai d'expiration a déclenché la tâche (uniquement la tâche d'événement), examiner le bit 1</li> <li>• un chevauchement s'est produit pour cette tâche, examiner le bit 2</li> </ul>			✓	
Programme de sécurité	Instance	Fournit le numéro d'instance de l'objet programme.	✓		✓	
	MajorFaultRecord	Enregistre les défauts majeurs pour ce programme.	✓	✓	✓	
	MaximumScanTime	Temps d'exécution maximal enregistré (ms.) pour ce programme.			✓	✓
	<b>Disable Flag</b>	<p>Contrôle l'exécution de ce programme. Chaque valeur a une signification spécifique :</p> <ul style="list-style-type: none"> <li>• 0. Exécution activée.</li> <li>• Différent de zéro. Exécution désactivée.</li> </ul>			✓	
	MaximumScanTime	Durée d'exécution maximale enregistrée (ms.) pour ce programme.			✓	
	Minor Fault Record	Enregistre les défauts mineurs pour ce programme.			✓	
	<b>LastScanTime</b>	Durée de la dernière exécution du programme lors de son dernier lancement. Temps en microsecondes.			✓	
	Nom (Name)	Nom de la tâche.				
Routine de sécurité	Instance	Fournit le numéro d'instance de cet objet routine. Les valeurs valides sont comprises entre 0 et 65 535.	✓			
Automates de sécurité	SafetyLockedState (SINT)	Indique si l'automate est verrouillée ou déverrouillée de sécurité.			✓	

Objet de sécurité	Nom d'attribut	Description d'attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
	SafetySILConfiguration (SINT)	Indique la configuration de sécurité SIL comme étant : <ul style="list-style-type: none"> <li>• 2 = SIL2/PLd</li> <li>• 3 = SIL3/PLe</li> </ul>			✓	
	SafetyStatus (INT) (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580).	Applications configurées pour SIL3/PLe, spécifiez l'état de sécurité suivant : <ul style="list-style-type: none"> <li>• Tâche de sécurité OK. (1100000000000000)</li> <li>• Tâche de sécurité inexploitable. (1100000000000011)</li> <li>• Partenaire manquant. (0100000000000000)</li> <li>• Partenaire indisponible. (0100000000000001)</li> <li>• Matériel incompatible (0100000000000010)</li> <li>• Firmware incompatible. (0100000000000011)</li> </ul> <b>Astuce :</b> pour les applications configurées pour SIL2/PLd, il ne faut pas tenir compte des bits 15, 0 et 1 lorsqu'ils peuvent prendre différentes valeurs basées sur le logement +1 de l'automate principal. Consultez la signification de l'état ci-dessus. Applications configurées pour SIL2/PLd, spécifiez la tâche de sécurité suivante : <ul style="list-style-type: none"> <li>• Tâche de sécurité OK (x1000000000000xx)</li> <li>• Tâche de sécurité inexploitable (x1000000000001xx)</li> </ul>			✓	

Objet de sécurité	Nom d'attribut	Description d'attribut	Accessible depuis la tâche de sécurité		Accessible depuis la tâche standard	
	SafetyStatus (INT) (uniquement applicable à Automates Compact GuardLogix 5370 et GuardLogix 5570).	Spécifie l'état de la sécurité comme étant : <ul style="list-style-type: none"> <li>• Tâche de sécurité OK. (1000000000000000)</li> <li>• Tâche de sécurité inexploitable. (1000000000000001)</li> <li>• Partenaire manquant. (0000000000000000)</li> <li>• Partenaire indisponible. (0000000000000001)</li> <li>• Matériel incompatible (0000000000000010)</li> <li>• Firmware incompatible. (0000000000000011)</li> </ul>			✓	
	SafetySignatureExists (SINT)	Indique la présence ou non d'une signature de sécurité.	✓		✓	
	SafetySignatureID (DINT) (uniquement applicable à Automates Compact GuardLogix 5370 et GuardLogix 5570)	Numéro d'identification 32 bits.			✓	
	SafetySignature (chaîne) (uniquement applicable à Automates Compact GuardLogix 5370 et GuardLogix 5570)	Numéro d'ID, la date et l'horodatage.			✓	
	SafetyTaskFaultRecord (DINT)	Enregistre les défauts de la tâche de sécurité.			✓	
	SafetySignatureIDLong SINT [33] (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580)	Le premier octet est la taille de l'ID de la signature de sécurité en octets, les 32 octets restants contiennent le contenu de l'ID de la signature de sécurité 32 octets.			✓	
	SafetySignatureIDHex(chaîne) (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580)	Représentation de chaîne hexadécimale 64 caractères de l'ID de signature			✓	
	SafetySignatureDateTime(chaîne) (uniquement applicable à Automates Compact GuardLogix 5380 et GuardLogix 5580)	Jour et heure de la signature de sécurité sur 27 caractères au format mm/dd/yyyy, hh:mm:ss.iii<AM ou PM>>			✓	

**Voir aussi**

[Instructions d'entrée/sortie](#) sur la [page 151](#)

**Contrôle des indicateurs d'état**

L'automate est compatible avec les mots-clés d'état que vous pouvez utiliser dans votre logique pour contrôler des événements spécifiques :

- Les mots-clés d'état ne sont *pas* sensibles à la casse.
- Comme les indicateurs d'état peuvent changer très rapidement, l'application Logix Designer n'affiche *pas* l'état des indicateurs (c-à-d., même lorsqu'un indicateur d'état est défini, l'instruction qui contient cet indicateur n'est pas mise en surbrillance).
- Vous *ne pouvez pas* définir un alias d'étiquette dans un mot-clé.

Vous pouvez utiliser les mots-clés suivants :

Pour déterminer si :	Utiliser :
la valeur que vous stockez ne peut pas être intégrée à destination, parce que soit : <ul style="list-style-type: none"> <li>• elle est plus grande que la valeur maximum de cette destination, soit</li> <li>• elle est plus inférieure à la valeur minimum de cette destination</li> </ul> <b>Important</b> : À chaque fois que S:V passe de mis à zéro à défini, cela génère un défaut mineur (type 4, code 4)	S:V
la valeur de destination de l'instruction est 0	S:Z
la valeur de destination de l'instruction est négative	S:N
Une opération arithmétique peut comporter des retenues et des poses qui tentent d'utiliser des bits qui se trouvent en dehors du type de données Par exemple : <ul style="list-style-type: none"> <li>• l'addition de 3 + 9 provoque une retenue de 1</li> <li>• la soustraction de 25 - 18 provoque une pose de 10</li> </ul>	S:C
c'est la première scrutation normale des routines du programme en cours	S:FS
au moins un défaut mineur a été généré : <ul style="list-style-type: none"> <li>• l'automate définit ce bit lorsqu'un défaut mineur survient à cause de l'exécution du programme.</li> <li>• L'automate ne définit pas ce bit pour les défauts mineurs qui ne sont pas liés à l'exécution du programme, comme l'indication de batterie faible.</li> </ul>	S:MINOR

**Sélection du type de message**

Après avoir entré l'instruction MSG et spécifié la structure MESSAGE, cliquer sur l'onglet Configuration (Configuration) de la boîte de dialogue Configuration du message (Message Configuration) pour spécifier les détails du message.

L'onglet Configuration (Configuration) contient également une case à cocher pour définir/mettre à zéro le bit .TO.

Les détails que vous configurez varient en fonction du type de message sélectionné.

Si le dispositif cible est :	Sélectionner l'un des types de message suivants :
Automate LOGIX 5000	Lecture de la table de données CIP (CIP data table read) Écriture de la table de données CIP (CIP data table write)
Module d'E/S que vous configurez en utilisant l'application Logix Designer	Reconfiguration module (Module Reconfigure) CIP générique (CIP Generic)
Automate PLC-5®	Lecture par PLC-5 (PLC-5 typed read) Écriture par PLC-5 (PLC-5 typed write) Lecture de gamme de mots PLC-5 (PLC-5 word range read) Écriture de gamme de mots PLC-5 (PLC-5 word range write)
Automate SLC™ Automate MicroLogix™	Lecture par SLC (SLC typed read) Écriture par SLC (SLC typed write)
Module de transfert par bloc	Lecture de transfert par bloc (block transfer read) Écriture de transfert par bloc (block transfer write)
processeur PLC-3®	Lecture par PLC-3 (PLC-3 typed read) Écriture par PLC-3 (PLC-3 typed write) Lecture de gamme de mots PLC-3 (PLC-3 word range read) Écriture de gamme de mots PLC-3 (PLC-3 word range write)
processeur PLC-2®	Lecture non protégée PLC-2 (PLC-2 unprotected read) Écriture non protégée PLC-2 (PLC-2 unprotected write)

Vous devez spécifier cette information de configuration :

Dans ce champ :	Spécifier :
Élément source (Source Element)	<p>si vous sélectionnez un type de message de lecture, l'Élément source est l'adresse des données que vous souhaitez lire dans le dispositif cible. Utiliser la syntaxe d'adresse du dispositif cible.</p> <p>Si vous sélectionnez un type de message d'écriture, l'étiquette Source est le premier élément de l'étiquette que vous souhaitez envoyer dans le dispositif cible.</p> <p>Les étiquettes de structure E/S et les booléens ne sont pas admis. Tous les autres types de données, par exemple, INT, DINT, peuvent être utilisés.</p>
Nombre d'éléments (Number of Elements)	Le nombre d'éléments en lecture/écriture dépend du type de message et du type de données que vous utilisez. Pour les messages « gammes de mots » et « non protégés », la taille d'un élément est indiquée dans la boîte de dialogue. Pour les messages CIP et « types », un élément est un élément simple d'un tableau que vous spécifié comme étant la source de l'écriture ou la destination de la lecture
Élément destinataire (Destination Element)	<p>Si vous sélectionnez un type de message de lecture, l'étiquette Destination est le premier élément de l'étiquette dans l'automate LOGIX 5000 où vous souhaitez stocker les données que vous lisez à partir du dispositif cible.</p> <p>Si vous sélectionnez un type de message d'écriture, l'Élément destinataire est l'adresse de l'emplacement dans le dispositif cible où vous souhaitez écrire les données.</p>

**Voir aussi**

[Spécification de messages CIP](#) sur la [page 288](#)

[Spécification de messages PLC-5](#) sur la [page 294](#)

[Spécification de messages SLC](#) sur la [page 189](#)

[Spécification de messages de bloc-transfert](#) sur la [page 189](#)

[Spécification de messages PLC-3](#) sur la [page 293](#)

[Spécification de messages PLC-2](#) sur la [page 295](#)

## Défauts du Module : **16#0000 - 16#00ff**

Voici les défauts du module : 16#0000 - 16#00ff

Code	Chaîne	Explication et causes/solutions possibles
16#0001	Erreur de connexion.	Échec de connexion à un module.
16#0002	Ressource indisponible.	<p>Soit :</p> <ul style="list-style-type: none"> <li>il n'existe pas suffisamment de connexions disponibles pour le Controller ou pour le module de communication utilisé pour la connexion. Vérifiez l'utilisation de la connexion du Controller ou du module de communication. Si toutes les connexions sont utilisées, essayez de libérer certaines des connexions utilisées ou ajoutez un autre module pour acheminer la connexion itinérante.</li> <li>les limites de mémoire I/O du Controller sont dépassées. Vérifiez la mémoire I/O disponible et modifiez le Program ou le tag, si nécessaire.</li> <li>le Module I/O ciblé ne comprend pas suffisamment de connexions disponibles. Vérifiez le nombre de Controllers établissant une connexion à ce module I/O et vérifiez que le nombre de connexions se situe à l'intérieur des limites du module I/O.</li> </ul>
16#0005	Erreur de demande de connexion : Classe incorrecte	<p>Le Controller essaie d'établir une connexion au Module et a reçu un message d'erreur.</p> <p>Soit :</p> <ul style="list-style-type: none"> <li>l'adresse configurée pour la connexion au module est incorrecte.</li> <li>le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p> <p>Si vous utilisez un module 1756-DHRIO, vérifiez que le type de voie sélectionné dans le logiciel (DH+ ou réseau I/O à distance) correspond aux paramètres du sélecteur rotatif du module.</p>

16#0006	Erreur de requête de connexion : classe incorrecte.	<p>Soit :</p> <ul style="list-style-type: none"> <li>la mémoire tampon de réponse est trop petite pour gérer les données de réponse.</li> <li>le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>
16#0007	Erreur de requête de connexion : classe incorrecte.	Une demande de service n'est pas connectée mais devrait l'être.
16#0008	Erreur de demande de service : Service non pris en charge	Le Controller tente de demander un service à partir du module non pris en charge.
16#0009	Configuration du module incorrecte : erreur de paramètre. <b>Astuce</b> : d'autres informations d'erreur relatives à ce défaut seront affichées sous la forme d'un code hexadécimal dans l'onglet Connexion.	La configuration du Module est incorrecte. La configuration du module a peut-être été modifiée dans la fenêtre Visualisation des données ou par programmation. S'il est disponible pour le module, accédez à l'onglet Connexions de la boîte de dialogue Propriétés du module pour le code de défaut supplémentaire. Le code de défaut supplémentaire indique le paramètre de configuration entraînant le défaut. Vous devrez peut-être corriger plusieurs paramètres avant que ce défaut ne soit effacé et que la connexion ne soit correctement établie.
16#000A	Un attribut dans Get_Attributes_List ou Set_Attributes_List est différent de zéro.	<p>Soit :</p> <ul style="list-style-type: none"> <li>une connexion est actuellement créée là où le type de connexion est incorrect.</li> <li>un attribut d'objet ou une valeur de tag est incorrect.</li> </ul> <p>Si un attribut d'objet ou un tag est incorrect, exportez le fichier Logix Designer puis réimportez-le. Replanifiez le réseau ControlNet après la réimportation si applicable.</p>
16#000C	Erreur de requête de service : Mode/état incorrect pour la requête de service.	<p>Le Controller essaie de demander un service au Module et a reçu un message d'erreur. Vérifiez d'abord que le module n'est pas défaillant.</p> <p>Pour un module I/O, ceci peut indiquer que le module comprend l'une des conditions suivantes :</p> <ul style="list-style-type: none"> <li>Communication limitée mais avec un défaut majeur</li> <li>Une mise en jour de firmware doit être effectuée ou est en cours.</li> </ul> <p>Reportez-vous à l'onglet Info. sur le Module pour déterminer la cause exacte du problème.</p>
16#000D	L'objet existe déjà.	Une instance de mappage I/O est créée là où l'instance est déjà utilisée.
16#000E	Impossible de définir la valeur d'attribut.	Une instruction MSG est configurée pour changer une valeur d'attribut non modifiable.
16#000F	Permission d'accès refusée pour le service demandé.	Une instruction MSG a été configurée pour supprimer un objet de mappage qui ne peut l'être.
16#0010	Le mode ou l'état du module n'autorise pas l'objet à exécuter le service demandé.	L'état du dispositif empêche le traitement d'une demande de service.



16#0011	Données de réponse trop longues.	La réponse à un message comporte une taille de données trop grande pour la destination. Changez la destination en tag qui peut gérer la taille de données et le type renvoyé.
16#0013	Configuration du module rejetée : taille des données trop faible.	La configuration du Module est incorrecte : un nombre insuffisant de données de configuration a été envoyé. Vérifiez que le module correct est ciblé.
16#0014	Attribut non défini ou non pris en charge.	Une instruction MSG est configurée pour changer un attribut inexistant.
16#0015	Configuration du module rejetée : taille des données trop élevée.	La configuration du Module est incorrecte : un nombre trop important de données de configuration a été envoyé. Vérifiez que le module correct est ciblé.

## Défauts du Module : Voici les défauts du module : 16#0100 - 16#01ff

### 16#0100 - 16#01ff

Code	Chaîne	Explication et causes/solutions possibles
16#0100	Erreur de demande de connexion : Module utilisé.	<ul style="list-style-type: none"> <li>• La connexion en cours d'accès est déjà utilisée.</li> </ul> Soit : <ul style="list-style-type: none"> <li>• Le Controller essaie d'établir une connexion spécifique avec un Module qui ne peut pas prendre en charge plus d'une de ces connexions.</li> <li>• La cible d'une connexion reconnaît que le propriétaire tente de rétablir une connexion déjà en cours.</li> </ul>
16#0103	Erreur de requête de service : classe de transport CIP non prise en charge.	Soit : <ul style="list-style-type: none"> <li>• Le Controller demande des services qui ne sont pas pris en charge par le Module.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte. Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder. Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.

16#0106	Erreur de demande de connexion : Module possédé et configuré par un autre Controller. Le module ne peut accepter qu'une connexion si l'option Unicast est utilisée.	<p>Il s'est produit un conflit de propriété pour la connexion. L'une des conditions suivantes existe :</p> <ul style="list-style-type: none"> <li>• La demande de connexion à ce module a été rejetée en raison d'un conflit de propriété avec un autre propriétaire (par exemple, un autre Controller). Ceci peut se produire avec des modules tels que des modules de sortie qui n'autorisent qu'un seul propriétaire pour configurer et contrôler ses sorties. Ce défaut peut également se produire si le module est configuré en tant qu'option Ecoute seule et ne prend en charge qu'une seule connexion.</li> <li>• Si le propriétaire est connecté au Module au moyen d'une connexion Unicast sur EtherNet/IP, les autres connexions au Module échouent car le propriétaire contrôle la connexion unique. Si le propriétaire est connecté au Module au moyen d'une connexion Multicast sur EtherNet/IP, les connexions Unicast au Module échouent car le propriétaire contrôle la connexion unique. Configurez la connexion du propriétaire et la connexion Ecoute seule en tant que Multicast.</li> </ul>
16#0107	Erreur de demande de connexion : Type inconnu.	Une connexion en cours d'accès est introuvable.
16#0108	Erreur de requête de connexion : type de connexion (Multicast/Unicast) non pris en charge.	<p>Le Controller demande un type de connexion qui n'est pas pris en charge par le Module. L'une des conditions suivantes existe :</p> <ul style="list-style-type: none"> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> <li>• Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Détrompage compatible dans la configuration du module au lieu de l'option Correspondance exacte. Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder. Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</li> <li>• Il est possible que vous ayez configuré un tag consommé pour utiliser une connexion Unicast, mais le Controller du producteur ne prend pas en charge les connexions Unicast.</li> </ul>

16#0109	<p>Erreur de requête de connexion : taille de connexion incorrecte.</p> <p><b>Astuce</b> : d'autres informations d'erreur relatives à ce défaut s'afficheront sous forme de nom de point associé au numéro d'instance de connexion défaillant.</p>	<p>La taille de connexion est incompatible avec celle prévue.</p> <p>Soit :</p> <ul style="list-style-type: none"> <li>• le Controllor tente en vain d'établir une connexion avec le module : la taille de la connexion est incorrecte.</li> <li>• il se peut que le Controllor tente de se connecter à un tag dans un Controllor de production dont la taille ne correspond pas au tag dans ce Controllor.</li> <li>• le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> <li>• il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Détrompage compatible dans la configuration du module au lieu de l'option Correspondance exacte.</li> </ul> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p> <p>Si le module est un module 1756 ControlNet, assurez-vous que la taille de châssis est correcte.</p> <p>Pour les adaptateurs I/O à distance, assurez-vous que la taille du rack et/ou la densité du rack sont correctes.</p>
16#0110	<p>Erreur de requête de connexion : module non configuré.</p>	<p>Le Controllor essaie en vain d'établir une connexion Ecoute seule avec le Module : celui-ci n'a pas été configuré ni connecté par un propriétaire (par exemple, un autre Controllor).</p> <p>Ce Controllor n'est pas le propriétaire de ce Module car il essaie d'établir une connexion Ecoute seule ne nécessitant aucune configuration de Module. La connexion ne peut pas être établie tant qu'un propriétaire n'aura pas configuré le Module avant de s'y connecter.</p>

16#0111	Intervalle de trame requis (RPI) hors limites.	<p>Soit :</p> <ul style="list-style-type: none"> <li>l'intervalle entre trames requis (RPI) spécifié est incorrect pour ce Module ou pour un Module se trouvant dans le chemin d'accès à ce Module. Voir l'onglet Avancé pour activer le RPI du producteur.</li> <li>le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p> <ul style="list-style-type: none"> <li>pour les connexions Ecoute seule : le RPI défini par le propriétaire de ce module est inférieur à celui demandé. Augmentez le RPI demandé ou diminuez le RPI utilisé par le Controller propriétaire.</li> </ul> <p>Voir l'onglet Connexion de la boîte de dialogue Propriétés du module pour obtenir les valeurs RPI correctes.</p>
16#0113	Erreur de demande de connexion : limite du module de connexion dépassée.	<p>Le nombre de connexions est supérieur au nombre disponible sur le module. Le nombre de connexions doit être réduit ou le matériel doit être mis à niveau.</p> <p>Pour réduire le nombre de connexion :</p> <ul style="list-style-type: none"> <li>Modifiez le format de communication de l'adaptateur Flex I/O d'une Configuration des entrées ou des sorties pour une <b>Optimisation du rack</b>. Lorsque le format de communication est modifié, l'adaptateur doit être supprimé et recréé dans l'arborescence de la configuration des I/O.</li> <li>Si la configuration utilise la messagerie sur ControlNet, organisez les messages pour réduire le nombre de messages s'exécutant simultanément ou réduisez le nombre de messages. Les messages (instructions MSG) utilisent également des connexions.</li> </ul>
16#0114	Discordance de détrompage électronique : discordance des codes de produit et/ou de l'identifiant du fournisseur du détrompage électronique.	<p>Le Code de produit du Module matériel ne correspond pas au Code de produit du Module créé dans le logiciel. Le détrompage électronique n'a pas abouti pour ce module. Le Module créé dans le logiciel ne correspond vraisemblablement pas au Module matériel.</p>

16#0115	Discordance de détrompage électronique : Discordance du type de produit de détrompage électronique.	Le Type de produit du Module matériel ne correspond pas au Type de produit du Module créé dans le logiciel. Le détrompage électronique n'a pas abouti pour ce module. Le Module créé dans le logiciel ne correspond vraisemblablement pas au Module matériel.
16#0116	Discordance de détrompage électronique : révision Major et/ou Minor non valable ou incorrecte.	Les Révisions majeures et/ou mineures du Module matériel ne correspondent pas aux Révisions majeures et/ou mineures du Module créé dans le logiciel. Vérifiez d'avoir indiqué la révision majeure ou mineure correcte si vous avez choisi le détrompage Module compatible ou Correspondance exacte. Le détrompage électronique n'a pas abouti pour ce module. Le Module créé dans le logiciel ne correspond vraisemblablement pas au Module matériel.
16#0117	<p>Erreur de demande de connexion : Point de connexion incorrect.</p> <p><b>Astuce :</b> d'autres informations d'erreur relatives à ce défaut apparaissent sous forme de nom de point associé au Controller à Controller (C2C) défaillant.</p>	<p>La connexion est établie avec un port incorrect ou un port déjà utilisé.</p> <p>L'une des conditions suivantes existe :</p> <ul style="list-style-type: none"> <li>• Un autre Controller possède ce module et s'est connecté en utilisant un format de communication : modules d'I/O différent de celui choisi par ce Controller. Vérifiez que le format de communications choisi est identique à celui sélectionné par le premier Controller propriétaire de ce Module.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p> <ul style="list-style-type: none"> <li>• Le Controller essaie peut-être de se connecter à un Tag inexistant dans un Controller de production.</li> </ul>

16#0118	Configuration du module rejetée : erreur de format.	<p>Un format de configuration incorrect est utilisé. L'une des conditions suivantes existe :</p> <ul style="list-style-type: none"> <li>• La classe de configuration indiquée ne correspond pas à la classe prise en charge par le module.</li> <li>• Le module ne reconnaît pas l'instance de connexion.</li> <li>• Le chemin d'accès indiqué pour la connexion est incompatible.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>
16#0119	Erreur de requête de connexion: Module non possédé.	<p>La connexion de contrôle n'est pas active. Là où une connexion Ecoute seule est demandée, la connexion de contrôle n'est pas active.</p>
16#011A	Erreur de requête de connexion : Ressources de connexion épuisées.	<p>Le Controller essaie en vain d'établir une connexion au Module : les ressources requises ne sont pas disponibles.</p> <p>Si le module est de type 1756 ControlNet, jusqu'à cinq Controllers peuvent établir des connexions Optimisation du rack avec le module. Vérifiez que cette limite n'est pas dépassée.</p> <p>Si le Module est un adaptateur 1794-ACN15, 1794-ACNR15 ou 1797-ACNR15, un seul Controller peut établir une connexion Optimisation du rack avec le Module. Vérifiez que cette limite n'est pas dépassée.</p>

## Défauts du Module : Voici les défauts du module : 16#0200 - 16#02ff. 16#0200 - 16#02ff

Code	Chaîne	Explication et causes/solutions possibles
16#0203	Dépassement de temps de la connexion.	<p>Le propriétaire ou l'expéditeur reconnaît que le dispositif cible figure sur le réseau ou dans le fond de panier. Toutefois, les données et les messages I/O ne reçoivent pas de réponses. En d'autres termes, la cible est accessible, mais sa réponse n'est pas celle prévue. Par exemple, il se peut que ce défaut soit indiqué là où les paquets Ethernet multicast ne sont pas renvoyés.</p> <p>Quand ce défaut se produit, le Contrôleur tente généralement de supprimer et de rétablir constamment la connexion.</p> <p>Si vous utilisez des modules FLEX I/O, veillez à utiliser le service de bornier de raccordement correct.</p>
16#0204	Erreur de timeout de la requête de connexion.	<p>Le Contrôleur essaie d'établir une connexion. Toutefois, le module cible ne répond pas. Le dispositif semble également être manquant du fond de panier ou du réseau.</p> <p>A des fins de récupération, effectuez ces actions :</p> <ul style="list-style-type: none"> <li>• Vérifiez que le module n'a pas été retiré, qu'il fonctionne encore et qu'il est sous tension.</li> <li>• Vérifiez que le numéro d'emplacement correct a été indiqué.</li> <li>• Vérifiez que le module est correctement connecté au réseau.</li> </ul> <p>Si vous utilisez les Modules Flex I/O, assurez-vous que le bornier de raccordement correct est utilisé.</p>
16#0205	Erreur de requête de connexion : paramètre incorrect.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur : un paramètre est erroné.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>
16#0206	Erreur de requête de connexion : taille requise trop importante.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur : la taille requise est trop importante.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>

## Défauts du Module : Voici les défauts du module : 16#0300 - 16#03ff 16#0300 - 16#03ff

Code	Chaîne	Explication et causes/solutions possibles
16#0301	Erreur de requête de connexion : mémoire tampon insuffisante.	<p>L'une des conditions suivantes peut exister :</p> <ul style="list-style-type: none"> <li>• Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur - la mémoire est insuffisante pour le module dans le chemin d'accès.</li> <li>• Le Contrôleur essaie peut-être de se connecter à un Tag présent dans un Contrôleur de production, mais ce Tag n'est pas marqué comme étant produit.</li> <li>• Le Contrôleur essaie peut-être de se connecter à un tag présent dans un Contrôleur de production. Il se peut que la configuration de ce Tag n'accepte pas suffisamment de consommateurs.</li> <li>• Diminuez la taille ou le nombre de connexions établies par ce module.</li> <li>• L'un des modules réseau se trouvant entre le module et le Contrôleur ne dispose peut-être pas d'une quantité suffisante de mémoire. Vérifiez la configuration réseau du système.</li> <li>• Le module ne dispose peut-être pas d'une quantité suffisante de mémoire. Vérifiez la configuration système et les fonctionnalités du Module.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>
16#0302	Erreur de requête de connexion : bande passante de communication insuffisante.	<p>Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur : un Module dans le chemin a dépassé la limite de bande passante de communication autorisée.</p> <p>Augmentez l'intervalle entre trames requis (RPI) et reconfigurez votre réseau avec RSNetWorx.</p> <p>Répartissez la charge sur un autre module de pont.</p>
16#0303	Erreur de requête de connexion : aucune passerelle n'est disponible.	<p>Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur : un Module dans le chemin a dépassé la limite de bande passante de communication autorisée.</p> <p>Répartissez la charge sur un autre module de pont.</p>



16#0304	Non configuré pour envoyer des données scrutées.	Le module ControlNet n'est pas planifié pour envoyer des données. Utilisez le logiciel RSNetWorx for ControlNet pour planifier ou planifier à nouveau le réseau ControlNet.
16#0305	Erreur de requête de connexion : la configuration ControlNet dans le Controller ne correspond pas à la configuration dans la passerelle.	La configuration ControlNet du Controller ne correspond pas à la configuration du Module de pont. Cela peut être dû au fait que le Module ControlNet a été modifié depuis la planification du réseau ou qu'un nouveau programme de commande a été chargé dans le Controller. Utilisez le logiciel RSNetWorx for ControlNet pour planifier à nouveau les connexions.
16#0306	Aucun gestionnaire de configuration ControlNet n'est disponible.	Le gestionnaire de configuration ControlNet est introuvable. Seuls les Modules 1756-CNB et PLC-5C peuvent être un gestionnaire de configuration ControlNet (CCM) et ce dernier doit être la station numéro 1. Vérifiez que le numéro de station du module 1756-CNB ou PLC-5C est 1 et qu'il fonctionne correctement. Ce défaut peut se produire temporairement lors de la mise en marche du système, mais il disparaît dès que le gestionnaire de configuration ControlNet est localisé.
16#0311	Erreur de requête de connexion : port incorrect.	Le Controller essaie d'établir une connexion au Module et a reçu un message d'erreur. Vérifiez que tous les modules de l'arborescence de configuration I/O sont ceux appropriés.
16#0312	Erreur de requête de connexion : adresse de liaison incorrecte.	Le Controller essaie d'établir une connexion au Module et a reçu un message d'erreur : une adresse de liaison incorrecte a été spécifiée. Une adresse de liaison peut être un numéro d'emplacement, une adresse réseau ou le numéro de châssis I/O à distance et le groupe de démarrage. Vérifiez que le numéro d'emplacement choisi pour ce module n'est pas supérieur à la taille du rack. Vérifiez que le numéro de station ControlNet n'est pas supérieur au nombre de stations maximum configuré pour le réseau dans le logiciel RSNetWorx for ControlNet.
16#0315	Erreur de requête de connexion : type de segment incorrect.	Le type de segment ou la route est incorrect. Soit : <ul style="list-style-type: none"> <li>le Controller essaie d'établir une connexion au Module et a reçu un message d'erreur : la demande de connexion est incorrecte.</li> <li>le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte. Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder. Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.

16#0317	Erreur de requête de connexion : connexion non scrutée.	Le Contrôleur essaie d'établir une connexion ControlNet au Module et a reçu un message d'erreur. Utilisez le logiciel RSNetWorx for ControlNet pour planifier ou planifier à nouveau la connexion à ce module.
16#0318	Erreur de requête de connexion : adresse de liaison incorrecte - Acheminement impossible.	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur - l'adresse de liaison est incorrecte. Vérifiez que le numéro d'emplacement et/ou de station sélectionné pour le module ControlNet est correct.
16#0319	Erreur de requête de connexion : pas de ressources secondaires disponibles dans le châssis redondant.	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur : le Module redondant ne dispose pas des ressources appropriées pour prendre en charge la connexion. Diminuez la taille ou le nombre de connexions pour ce Module ou ajoutez un autre Contrôleur ou Module ControlNet au système.
16#031a	Erreur de demande de connexion : connexion du rack refusée.	Le Contrôleur essaie d'établir une connexion directe au Module et a reçu un message d'erreur. Une connexion Optimisation du rack à ce Module a déjà été établie dans le même châssis via le Module 1756-CNB/R. <ul style="list-style-type: none"> <li>• Connectez-vous à ce Module via un module 1756-CNB/R dans le même châssis.</li> <li>• Connectez-vous à ce Module via un module 1756-CNB/R différent pour pouvoir utiliser une connexion directe.</li> <li>• Faites passer la première connexion de Optimisation du rack à Connexion directe, puis rétablissez la deuxième connexion directe.</li> <li>• Connectez-vous à ce module à partir d'un Contrôleur présent dans le même châssis que le module (sans passer par 1756-CNB/R).</li> </ul>
16#031e	Erreur de requête de connexion: Impossible de consommer le Tag.	<ul style="list-style-type: none"> <li>• Le Contrôleur essaie d'établir une connexion au Tag dans un Contrôleur de production et a reçu un message d'erreur.</li> <li>• Le Contrôleur essaie d'établir une connexion avec un tag dans un Contrôleur de production et ce tag a déjà été utilisé par un nombre trop important de consommateurs. Augmentez le nombre maximum de consommateurs sur le Tag.</li> </ul>
16#031f	Erreur de requête de connexion: Impossible de consommer le Tag.	Aucun objet de connexion SC (contrôleur de maintenance) n'a été trouvé correspondant à une instance de symbole.
16#0322	Erreur de demande de connexion : Incompatibilité entre points de connexion	Incompatibilité entre points de connexion. Soit : <ul style="list-style-type: none"> <li>• une nouvelle connexion demandée ne correspond pas à la connexion existante. Vérifiez les Contrôleurs qui utilisent la connexion et vérifiez que toutes les configurations sont identiques.</li> <li>• la connexion demandée n'est pas un port d'écoute ni un type de connexion de contrôle.</li> </ul>

## Défauts du Module : Voici les défauts du module : 16#0800 - 16#08ff

### 16#0800 - 16#08ff

Code	Chaîne	Explication et causes/solutions possibles
16#0800	Le lien réseau du chemin vers le module est hors ligne.	Aucune interprétation disponible.
16#0801	RPI de distribution multiple non compatible.	Aucune interprétation disponible.
16#0810	Aucune donnée d'application cible disponible.	L'application de commande n'a pas initialisé les données devant être produites par le périphérique cible. Il se peut que les connexions d'envoi des données soient configurées dans un périphérique cible et que l'application de commande de ce périphérique n'ait pas initialisé les données à produire. Pour le périphérique cible associé à la connexion d'envoi des données qui renvoie cette erreur de connexion, démarrez l'application de commande et effectuez au moins une écriture de données. Reportez-vous à la documentation du périphérique cible et de l'application de commande pour obtenir des informations sur cette opération.
16#0814	Erreur de requête de connexion : discordance de type de données.	Informations d'état de connexion incorrectes trouvées.

## Défauts du Module : Les défauts du module : 16#fd00 - 16#fdff.

### 16#fd00 - 16#fdff

Code	Chaîne	Explication et causes/solutions possibles
16#fd03	Erreur de demande de connexion : Connexion requise manquante	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur ; ce Module requiert un groupe particulier de connexions et de types de connexion, mais un de ces types est manquant. <ul style="list-style-type: none"> <li>• Appelez le support technique</li> <li>• <a href="http://www.support.rockwellautomation.com">http://www.support.rockwellautomation.com</a></li> </ul>
16#fd04	Erreur de demande de connexion : Aucun maître CST détecté	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur ; ce Module requiert un CST maître dans le châssis. <ul style="list-style-type: none"> <li>• Configurez un module (généralement un Contrôleur) dans ce châssis en tant que CST maître.</li> <li>• Appelez le support technique</li> <li>• <a href="http://www.support.rockwellautomation.com">http://www.support.rockwellautomation.com</a></li> </ul>
16#fd05	Erreur de demande de connexion : Aucun Axis ni groupe assigné.	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur ; ce Module requiert l'affectation d'un axis ou d'une table de groupe. <ul style="list-style-type: none"> <li>• Affectez un groupe ou un axe.</li> <li>• Appelez le support technique</li> <li>• <a href="http://www.support.rockwellautomation.com">http://www.support.rockwellautomation.com</a></li> </ul>
16#fd06	Défaut de transition	Le Contrôleur ordonne la transition de la boucle SERCOS à une nouvelle phase qui renvoie une erreur depuis le module. Vérifiez les stations de variateurs doubles.
16#fd07	Vitesse de données SERCOS incorrecte	La tentative de configuration de la boucle SERCOS a échoué. La vitesse de transmission de tous les dispositifs doit être la même et doit être prise en charge par les variateurs et par le module SERCOS.

16#fd08	Défaut Comm. de SERCOS	<p>Deux groupes de défaut principaux peuvent provoquer une Comm. Défaut - Défauts physiques et d'interface.</p> <p>Une source possible des défauts physiques est :</p> <ul style="list-style-type: none"> <li>• une boucle brisée ;</li> <li>• un connecteur lâche ;</li> <li>• des fibres optiques souillées ;</li> <li>• un bruit électrique dû à une mise à la terre inappropriée des variateurs ;</li> <li>• trop de stations sur la boucle.</li> </ul> <p>Les erreurs d'interface surviennent lorsque vous configurez des variateurs pour un tiers.</p> <p>Une source possible des erreurs d'interface est :</p> <ul style="list-style-type: none"> <li>• pas de MST SERCOS (erreur de protocole) ;</li> <li>• AT manqué (le variateur n'a pas envoyé les données lorsqu'il aurait dû le faire) ;</li> <li>• erreur de temporisation SERCOS dans la phase 3 ;</li> <li>• erreur dans le renvoi des données variateur au module SERCOS.</li> </ul>
16#fd09	Défaut d'initialisation de station	La tentative du Controller de configurer la station pour une opération cyclique a renvoyé une erreur.
16#fd0a	Erreur d'attribut Axis	Réponse incorrecte reçue depuis un module de mouvement.
16#fd0c	Erreur de maîtres différents	Le dispositif final possède un maître différent que le Controller.
16#fd1f	Format de protocole de sécurité incorrect	Une erreur s'est produite lors de l'ajout du segment réseau de sécurité à une route.
16#fd20	Aucune Task de sécurité	Aucune Task de sécurité ne semble en cours.
16#fd22	Discordance de taille de châssis	Vérifiez le nombre de Modules d'extension I/O physiques configurés pour le Controller puis mettez à jour le nombre de Modules sélectionnés dans la liste I/O d'extension de la page Général de la boîte de dialogue Propriétés du Controller.
16#fd23	Dépassement de la taille du châssis	<p>Pour vérifier le nombre d'I/O d'extension que le Controller prend en charge, ouvrez la boîte de dialogue Propriétés du Controller et développez la liste I/O d'extension de la page Général.</p> <p>Configurez le nombre de modules d'extension I/O physiques pour qu'il corresponde à la sélection de la liste I/O d'extension.</p>

**Défauts du Module :** Les défauts du module : 16#fe00 - 16#feff.

**16#fe00 - 16#feff**

Code	Chaîne	Explication et causes/solutions possibles
16#fe01		Un format de configuration non valide a été trouvé.
16#fe02	Intervalle entre paquets demandé (RPI) hors limites.	<p>L'Intervalle entre paquets demandé (RPI) spécifié est non valide pour ce module.</p> <ul style="list-style-type: none"> <li>• Reportez-vous à l'onglet Connexion pour connaître les valeurs RPI valides.</li> </ul>
16#fe03		Le point de connexion d'entrée n'a pas été défini.
16#fe04	Erreur de requête de connexion : pointeur de données d'entrée incorrect.	L'automate essaie d'établir une connexion au module et a reçu un message d'erreur.

16#fe05	Erreur de requête de connexion : taille des données d'entrée incorrecte.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• L'automate essaie d'établir une connexion au module et a reçu un message d'erreur.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration E/S et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration E/S et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration E/S de l'application Logix Designer.</p>
16#fe06		Le point de forçage d'entrée n'a pas été défini.
16#fe07		Le point de connexion de sortie n'a pas été défini.
16#fe08	Erreur de requête de connexion : pointeur de données de sortie incorrect.	L'automate essaie d'établir une connexion au module et a reçu un message d'erreur.
16#fe09	Erreur de requête de connexion : taille des données de sortie incorrecte.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• L'automate essaie d'établir une connexion au module et a reçu un message d'erreur.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration E/S et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration E/S et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration E/S de l'application Logix Designer.</p>
16#fe0a		Le point de forçage de sortie n'a pas été défini.

16#fe0b	Chaîne de symbole non valide.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• L'étiquette devant être consommé sur ce module est non valide. Vérifiez que l'étiquette est marqué comme étant produit.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration E/S et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration E/S et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration E/S de l'application Logix Designer.</p>
16#fe0c	Numéro d'instance de PLC-5 non valide.	<p>L'automate essaie d'établir une connexion à PLC-5 et a reçu un message d'erreur.</p> <p>Vérifiez que le numéro d'instance indiqué a été correctement spécifié dans le module PLC-5.</p>
16#fe0d	L'étiquette n'existe pas dans l'automate de même niveau.	Le numéro d'instance du symbole a été trouvé non défini.
16#fe0e	Mise à jour automatique du firmware en cours.	Le module est en cours de mise à jour.
16#fe0f	Echec de la mise à jour automatique du firmware : fichier firmware incompatible avec le module.	L'utilitaire de supervision de firmware a tenté de mettre à jour un module non pris en charge.
16#fe10	Echec de la mise à jour automatique du firmware : fichier firmware introuvable.	Le fichier de firmware pour mettre à jour le module est introuvable.
16#fe11	Echec de la mise à jour automatique du firmware : fichier firmware non valide.	Le fichier firmware est endommagé.
16#fe12	Echec de la mise à jour automatique du firmware.	Une erreur s'est produite lors de la mise à jour du module.
16#fe13	Echec de la mise à jour automatique du firmware : connexions actives détectées.	Impossible d'établir une connexion active avec le module cible.
16#fe14	Mise à jour automatique du firmware en attente : recherche du fichier NVS pour obtenir l'identité du module approprié.	Le fichier de firmware est en cours de lecture.
16#fe22		Le type de connexion netparams cible-vers-expéditeur est non valide.
16#fe23		La connexion netparams cible-vers-expéditeur n'indique pas si l'option monodiffusion est autorisée.

## Défauts du Module : Voici les défauts du module : 16#ff00 - 16#ffff

### 16#ff00 - 16#ffff

Code	Chaîne	Explication et causes/solutions possibles
16#ff00	Erreur de requête de connexion : aucune instance de connexion.	<p>Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur.</p> <p>Vérifiez que le module physique est du même type que le module créé dans le logiciel (ou est compatible avec lui).</p> <p>S'il s'agit d'un module 1756-DHRIO présent dans un châssis à distance (connecté via un réseau ControlNet), vérifiez que le réseau a été planifié à l'aide du logiciel RSNetWorx.</p> <p>Même après la planification du réseau avec le logiciel RSNetWorx for ControlNet, si vous êtes en ligne et que le module 1756-DHRIO est configuré pour un réseau DH+ uniquement, un défaut de Module #ff00 (aucune instance de connexion) peut se produire. Le module communique correctement bien que son état soit Anormal dans la boîte de dialogue Propriétés du module. Ignorez le message d'erreur et l'état de défaut et poursuivez.</p>
16#ff01	Erreur de requête de connexion : chemin du module trop long.	<p>Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur.</p> <p>Vérifiez que la longueur du chemin de ce module est correcte.</p>
16#ff04		L'instance de mappage du contrôleur à distance a tenté d'accéder à une connexion tout en étant dans un état incorrect.
16#ff08	Erreur de requête de connexion : chemin du module incorrect.	<p>Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur.</p> <p>Vérifiez que la longueur du chemin de ce module est correcte.</p>
16#ff0b	Configuration du module incorrecte : format incorrect.	<p>Soit :</p> <ul style="list-style-type: none"> <li>• La configuration du Module est incorrecte.</li> <li>• Le module utilisé (c'est-à-dire, le module physique) est différent de celui indiqué dans l'arborescence de configuration I/O et entraîne donc l'échec de la connexion ou du service.</li> </ul> <p>Il se peut que le défaut se produise même quand le module a réussi le test de détrompage électronique. Ceci peut résulter de l'utilisation des options Désactiver détrompage ou Module compatible dans la configuration du module au lieu de l'option Correspondance exacte.</p> <p>Bien qu'il ait réussi le test de détrompage électronique, le module connecté n'a pas les mêmes fonctionnalités ou paramètres que le module spécifié dans l'arborescence de configuration d'entrée et de sortie et ne prend pas en charge la connexion ou le service auquel vous tentez d'accéder.</p> <p>Vérifiez le Module utilisé et assurez-vous qu'il correspond exactement à celui indiqué dans l'arborescence de configuration I/O de l'application Logix Designer.</p>
16#ff0e	Erreur de requête de connexion : aucune connexion acceptée pour la passerelle.	Le Contrôleur essaie d'établir une connexion au Module et a reçu un message d'erreur.

## Spécification de messages CIP

Les types de messages en lecture et écriture de la table de données CIP transfèrent les données entre les automates LOGIX 5000.

Sélectionner cette commande	Si vous voulez
Lecture de la table de données CIP (CIP Data Table Read)	Lire les données d'un autre automate. Les types de Source et de Destination doivent correspondre.
Écriture de la table de données CIP (CIP Data Table Write)	Écrire les données sur un autre automate. Les types de Source et de Destination doivent correspondre.

### Reconfiguration d'un module d'E/S

Utiliser le message Reconfiguration module pour envoyer de nouvelles informations de configuration à un module d'E/S.

Pendant la reconfiguration, la situation suivante se produit :

- Les modules d'entrée continuent d'envoyer des données d'entrée à l'automate.
- Les modules de sortie continuent de contrôler leurs dispositifs de sortie.

Un message Reconfiguration de module requiert la propriété suivante de configuration.

Dans cette propriété	Sélectionner
Type de message	Reconfiguration de module

### Exemple

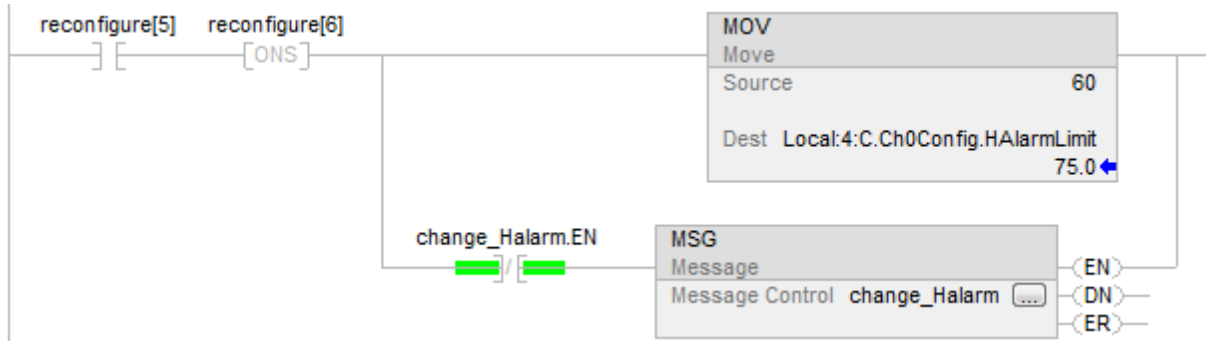
Suivre ces étapes pour reconfigurer le module d'E/S.

1. Définir la nouvelle valeur du membre requis de l'étiquette de configuration du module.
2. Envoyer un message Reconfiguration de module au module.  
Lorsque reconfigure[5] est défini, définir l'alarme haute à 60 pour le module local du logement 4. Le message Reconfiguration de module envoie alors la nouvelle valeur d'alarme au module. Cette instruction d'impulsion évite que l'échelon envoie des messages multiples au module tout le temps que reconfigure[5] est actif.

**Astuce :** Nous vous recommandons d'inclure toujours un XIO du bit MSG.EN en tant que précondition d'échelon MSG en série.



### Logique à relais



### Texte structuré

```
IF reconfigure[5] AND NOT reconfigure[6] THEN
  Local:4:C.Ch0Config.HAlarmLimit := 60;
```

```
IF NOT change_Halarm.EN THEN MSG(change_Halarm);
```

```
END_IF; END_IF;
```

```
reconfigure[6] := reconfigure[5];
```

### Spécifier les messages génériques CIP

**Important :** Les modules ControlLogix disposent de services qui peuvent être appelés en utilisant une instruction MSG et en choisissant le type de message CIP générique.

Si vous voulez	Dans cette propriété	Saisir ou sélectionner	
Exécuter un test d'impulsion sur un module de sortie numérique	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Test d'impulsion (Pulse Test)	
	Source (Source)	tag_name de type INT [5]	
		Ce tableau contient	Description
		tag_name[0]	Masque de bit de points à tester (test d'un point à la fois uniquement)
		tag_name[1]	Réservé, laisser à 0
		tag_name[2]	Largeur d'impulsion (centaines de $\mu$ , généralement 20)
	tag_name[3]	Retard de passage par zéro pour ControlLogix E/S (centaines de $\mu$ , généralement 40)	
tag_name[4]	Vérifier le retard		
Destination (Destination)	En blanc		

Obtenir la valeur de l'audit	Type de message (Message Type)	CIP Générique (CIP Generic)		
	Type de service (Service Type)	Obtention de la valeur de l'audit (Audit Value Get)		
	Élément source (Source Element)	Champ non modifiable, en blanc		
	Longueur Source (Source Length)	Champ non modifiable, définir sur 0 octets		
	Élément destinataire (Destination Element)	Ce tableau contient	Description	
tag_name de type DINT[2] ou LINT		Cette étiquette contient la Valeur de l'audit pour l'automate. <b>Important</b> : Rockwell Automation recommande d'utiliser le type de données DINT[2] pour éviter les limitations lorsque vous travaillez avec des types de données LINT dans les automates Allen-Bradley®.		
Obtenir les événements de l'automate surveillés pour les changements	Type de message (Message Type)	CIP Générique (CIP Generic)		
	Type de service (Service Type)	Obtenir les changements à détecter (Changes to Detect Get)		
	Élément source (Source Element)	Champ non modifiable, en blanc		
	Longueur Source (Source Length)	Champ non modifiable, définir sur 0 octets		
	Élément destinataire (Destination Element)	Ce tableau contient	Description	
tag_name de type DINT[2] ou LINT		Cette étiquette représente un masque de bit des changements surveillés pour l'automate. <b>Important</b> : Rockwell Automation conseille d'utiliser le type de données DINT[2] pour éviter les limitations lorsque vous travaillez avec des types de données LINT dans les automates Allen-Bradley.		
Définir les événements de l'automate surveillés pour les changements	Type de message (Message Type)	CIP Générique (CIP Generic)		
	Type de service (Service Type)	Définir les changements à détecter (Changes to Detect Set)		
	Élément source (Source Element)	Ce tableau contient	Description	
		tag_name de type DINT[2] ou LINT	Cette étiquette représente un masque de bit des changements surveillés pour l'automate. <b>Important</b> : Rockwell Automation conseille d'utiliser le type de données DINT[2] pour éviter les limitations lorsque vous travaillez avec des types de données LINT dans les automates Allen-Bradley.	
	Longueur Source (Source Length)	Champ non modifiable, définir sur 8 octets		
Élément destinataire (Destination Element)	Champ non modifiable, en blanc			

Réinitialiser les fusibles électroniques sur un module de sorties numériques	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Réinitialiser les fusibles électroniques (Reset Electronic Fuse)	
	Source (Source)	Nom d'étiquette de type DINT Cette étiquette représente un masque de bit des points sur lesquels les fusibles doivent être réinitialisés.	
	Destination (Destination)	Laisser en blanc	
Réinitialiser des diagnostics de verrouillage sur un module numérique d'entrée	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Réinitialiser des diagnostics de verrouillage (I) (Reset Latched Diagnostics (I))	
	Source (Source)	tag_name de type DINT Cette étiquette représente un masque de bit des points sur lesquels les diagnostics doivent être réinitialisés.	
Réinitialiser des diagnostics de verrouillage sur un module numérique de sortie	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Réinitialiser les diagnostics verrouillés (O) (Reset Latched Diagnostics (O))	
	Source (Source)	tag_name de type DINT Cette étiquette représente un masque de bit des points sur lesquels les diagnostics doivent être réinitialisés.	
Déverrouiller l'alarme d'un module d'entrée analogique	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Sélectionner l'alarme que vous souhaitez déverrouiller. <ul style="list-style-type: none"> <li>• Déverrouiller toutes les alarmes (I)</li> <li>• Déverrouiller l'alarme haute analogique (I)</li> <li>• Déverrouiller l'alarme haute haute analogique (I)</li> <li>• Déverrouiller l'alarme basse analogique (I)</li> <li>• Déverrouiller l'alarme basse basse analogique (I)</li> <li>• Déverrouiller l'alarme de vitesse (I)</li> </ul>	
	Instance (Instance)	Canal de l'alarme à déverrouiller.	
Déverrouiller l'alarme d'un module de sortie analogique	Type de message (Message Type)	CIP Générique (CIP Generic)	
	Type de service (Service Type)	Sélectionner l'alarme que vous souhaitez déverrouiller. <ul style="list-style-type: none"> <li>• Déverrouiller toutes les alarmes (O)</li> <li>• Déverrouiller alarme haute (O)</li> <li>• Déverrouiller alarme basse (O)</li> <li>• Déverrouiller alarme de rampe (O)</li> </ul>	
	Instance (Instance)	Canal de l'alarme à déverrouiller.	

**Définitions de bits de obtention/définition des événements de l'automate surveillés pour les changements**

Noms d'étiquettes	Type de données	Définition de bit
<p>Obtenir les événements de l'automate surveillés pour les changements</p> <p>Définir les événements de l'automate surveillés pour les changements</p>	<p>DINT[0]</p>	<p>Chaque bit a une signification spécifique :</p> <ul style="list-style-type: none"> <li>0 Stocker sur un support amovible avec l'application Logix Designer</li> <li>1 Des modifications en ligne ont été acceptées, testées ou assemblées</li> <li>2 Transaction en ligne d'importation partielle terminée</li> <li>3 Forçages SFC ont été activés</li> <li>4 Forçages SFC ont été désactivés</li> <li>5 Forçages SFC ont été supprimés</li> <li>6 Forçages SFC ont été modifiés</li> <li>7 Forçages E/S ont été activés</li> <li>8 Forçages E/S ont été désactivés</li> <li>9 Forçages E/S ont été supprimés</li> <li>10 Forçages E/S ont été modifiés</li> <li>11 Mise à jour du firmware depuis une source non connectée</li> <li>12 Mise à jour du firmware par un support amovible</li> <li>13 Changement de mode par une station de travail</li> <li>14 Changement de mode avec l'interrupteur de mode</li> <li>15 Un défaut majeur s'est produit</li> <li>16 Les défauts majeurs ont été effacés</li> <li>17 Les défauts majeurs ont été effacés avec l'interrupteur de mode</li> <li>18 Les propriétés Tâche ont été modifiés</li> <li>19 Les propriétés Programme ont été modifiées</li> <li>20 Les options de tranche de temps de l'automate ont été modifiés</li> <li>21 Le support amovible a été retiré</li> <li>22 Le support amovible a été inséré</li> <li>23 Signature de sécurité créée</li> <li>24 Signature de sécurité supprimée</li> <li>25 Verrouillage de la sécurité</li> <li>26 Déverrouillage de la sécurité</li> <li>27 La valeur de l'étiquette constante modifiée</li> <li>28 Valeurs multiples de l'étiquette constante modifiées</li> <li>29 Attribut d'étiquette constante mis à zéro</li> <li>30 Étiquette définie à valeur constante</li> <li>31 Entrée de journal personnalisé ajoutée</li> </ul>
	<p>DINT[1]</p>	<ul style="list-style-type: none"> <li>32 Changements qui affectent la corrélation</li> <li>33 Attribut Aide à protéger la signature en mode Exécution défini</li> <li>34 Attribut Aide à protéger la signature en mode Exécution effacé</li> <li>35...63 Inutilisés</li> </ul>

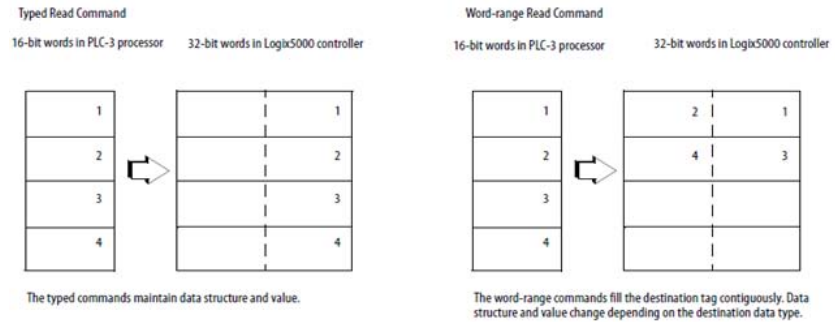
<b>Astuces :</b>	<ul style="list-style-type: none"> <li>• La sélection du type de message <b>CIP générique</b> (CIP Generic) active l'option <b>Grande connexion</b> (Large Connection) sur l'onglet <b>Communication</b> (Communication). Utiliser des grandes connexions CIP Générique lorsqu'un message est plus grand que 480 octets. 500 octets est la règle générale, mais il y a des entêtes au début du message. Les grandes connexions CIP sont destinées aux messages occupant jusqu'à 3980 octets.</li> <li>• La case <b>Grande connexion</b> (Large Connection) n'est activée que lorsque la case <b>Connecté</b> (Connected) est cochée et que le type de message sélectionné sur l'onglet <b>Configuration</b> (Configuration) est <b>CIP générique</b> (CIP Generic).</li> <li>• L'option <b>Grande connexion</b> (Large Connection) est uniquement disponible dans l'application Logix Designer version 21.00.00 ou ultérieure et dans le logiciel RSLogix 5000, version 20.00.00 ou ultérieure.</li> </ul>
------------------	--

## Spécification de messages PLC-3

Les types de message PLC-3 sont conçus pour les processeurs PLC-3.

Sélectionner cette commande :	Pour :
PLC3 Lecture par type	<p>Lire des données de type entier ou REAL.</p> <p>Pour les entiers, cette commande lit les entiers de 16 bits depuis le processeur PLC-3 et les stocke dans des tableaux de données SINT, INT ou DINT dans l'automate LOGIX 5000, et préserve l'intégrité des données.</p> <p>Cette commande lit également les données à virgule flottante depuis le PLC-3, et les stocke dans une étiquette de type de données REAL dans l'automate LOGIX 5000.</p>
PLC3 Écriture par type	<p>Écrire des données de type entier ou REAL.</p> <p>Cette commande écrit des données SINT ou INT dans le fichier d'entiers PLC-3, et préserve l'intégrité des données. Vous pouvez écrire des données DINT tant qu'elles peuvent être contenues dans un type de données INT (<math>-32\ 768 \geq \text{données} \leq 32\ 767</math>).</p> <p>Cette commande écrit également des types de données REAL depuis l'automate LOGIX 5000 vers un fichier à virgule flottante PLC-3.</p>
PLC3 Lecture de la plage de mots	<p>Lire une plage continue de mots de 16 bits dans la mémoire PLC-3 quel que soit le type de données.</p> <p>Cette commande commence à l'adresse spécifiée comme élément source, et lit de façon séquentielle le nombre de mots de 16 bits requis.</p> <p>Les données provenant de l'Élément source sont stockées, en commençant à l'adresse donnée comme Étiquette destination.</p>
PLC3 Écriture de la plage de mots	<p>Écrire une plage continue de mots de 16 bits dans la mémoire PLC-3, depuis la mémoire LOGIX 5000, quel que soit le type de données.</p> <p>Cette commande commence à l'adresse spécifiée comme Étiquette source, et lit de façon séquentielle le nombre de mots de 16 bits requis.</p> <p>Les données provenant de l'Étiquette source sont stockées, en commençant à l'adresse donnée comme Élément destinataire dans le processeur PLC-3.</p>

Les schémas suivants présentent la différence entre la commande par type et la commande de la plage de mots. L'exemple se base sur les commandes de lecture depuis un processeur PLC-3 vers un automate LOGIX 5000.



## Spécification de messages PLC-5

Utilisez les types de message PLC-5 pour communiquer avec les automates PLC-5.

Sélectionner cette commande :	Pour :
PLC-5 Lecture par type	Lire des données de type chaîne, à virgule flottante ou entières 16 bits, et préserver l'intégrité des données.
PLC-5 Écriture par type	Écrire des données de type chaîne, à virgule flottante ou entières 16 bits, et préserver l'intégrité des données.
PLC-5 Lecture de la plage de mots	Lire une plage continue de mots de 16 bits dans la mémoire PLC-5 quel que soit le type de données. Cette commande commence à l'adresse spécifiée comme élément source, et lit de façon séquentielle le nombre de mots de 16 bits requis. Les données provenant de l'Élément source sont stockées, en commençant à l'adresse donnée comme Étiquette destination.
PLC-5 Écriture de la plage de mots	Écrire une plage continue de mots de 16 bits dans la mémoire PLC-5, depuis la mémoire LOGIX 5000, quel que soit le type de données. Cette commande commence à l'adresse spécifiée comme Étiquette source, et lit de façon séquentielle le nombre de mots de 16 bits requis. Les données provenant de l'étiquette source sont stockées, en commençant à l'adresse donnée comme élément destinataire dans le processeur PLC-5.

### Types de données pour les messages PLC-5 Lecture et écriture par type

Le tableau suivant indique les types de données à utiliser avec les messages PLC-5 Lecture par type et PLC-5 Écriture par type.

Pour ce type de données PLC-5 :	Utilisez ce type de données LOGIX 5000 :
B	INT
F	REAL
N	INT DINT (écrire uniquement des valeurs DINT dans un automate PLC-5 si la valeur est $\geq -32\,768$ et $\leq 32\,767$ )
S	INT
ST	STRING

Les commandes Lecture par type et Écriture par type fonctionnent également avec les processeurs SLC 5/03 (OS303 et versions supérieures), SLC 5/04 (OS402 et versions supérieures) et SLC 5/05.

Les types de messages PLC-2 sont conçus pour les processeurs PLC-2.

## Spécification de messages PLC-2

Sélectionner cette commande :	Pour :
Lecture non protégée PLC2	Lire des mots de 16 bits de n'importe quel emplacement de la table de données PLC-2 ou du fichier de compatibilité PLC-2 d'un autre processeur.
Écriture non protégée PLC2	Écrire des mots de 16 bits dans n'importe quel emplacement de la table de données PLC-2 ou du fichier de compatibilité PLC-2 d'un autre processeur.

Le transfert de messages utilisant des mots de 16 bits, vous devez vous assurer que l'étiquette LOGIX 5000 stocke correctement les données transférées, généralement sous forme d'un tableau INT.





## Instructions de comparaison

### Instructions de comparaison

Les instructions de comparaison comparent les valeurs en utilisant une expression ou une instruction de comparaison spécifique.

#### Instructions disponibles

#### Diagramme à relais

<a href="#">CMP</a>	<a href="#">EQU</a>	<a href="#">GEQ</a>	<a href="#">GRT</a>	<a href="#">LEQ</a>	<a href="#">LES</a>	<a href="#">LIM</a>	<a href="#">MEQ</a>	<a href="#">NEQ</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

#### Diagramme de bloc fonctionnel

#### Bloc FBD

<a href="#">EQU</a>	<a href="#">GEQ</a>	<a href="#">GRT</a>	<a href="#">LEQ</a>	<a href="#">LES</a>	<a href="#">LIM</a>	<a href="#">MEQ</a>	<a href="#">NEQ</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

#### Fonction FBD

$=_f$	$\geq_f$	$>_f$	$\leq_f$	$<_f$	$LIM_f$	$MEQ_f$	$\neq_f$
<a href="#">EQU</a>	<a href="#">GEQ</a>	<a href="#">GRT</a>	<a href="#">LEQ</a>	<a href="#">LES</a>	<a href="#">LIM</a>	<a href="#">MEQ</a>	<a href="#">NEQ</a>

#### Texte structuré

Indisponible

Si vous voulez :	Utilisez cette instruction :
Compare les valeurs d'après une expression	CMP
Vérifie l'égalité entre deux valeurs	EQU
Vérifie qu'une valeur est supérieure ou égale à la seconde valeur	GEQ
Vérifie qu'une valeur est supérieure à la seconde valeur	GRT
Vérifie qu'une valeur est inférieure ou égale à la seconde valeur	LEQ
Vérifie qu'une valeur est inférieure à la seconde valeur	LES
Vérifie qu'une valeur se trouve entre deux autres valeurs	LIM
Passe deux valeurs dans un masque et vérifie leur égalité	MEQ
Vérifie qu'une valeur est différente de la seconde valeur	NEQ

Compare les valeurs de différents types de données, telles que la virgule flottante ou le nombre entier.

Les types de données en caractères gras correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses paramètres de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

**Voir aussi**

[Instructions de calcul/math](#) sur la [page 373](#)

## Comparaison (CMP)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Définir l'expression CMP en utilisant des opérateurs, des étiquettes et des valeurs immédiates. Utiliser les parenthèses ( ) pour définir des membres d'expressions plus complexes.

L'avantage de l'instruction CMP est qu'elle permet d'inclure des expressions complexes dans une même instruction.

Lors de l'évaluation de l'expression, tous les opérandes non-REAL seront convertis en REAL avant de procéder aux calculs, si au moins une des conditions suivantes est vraie.

- Toutes les opérandes dans l'expression sont REAL.
- L'expression contient SIN, COS, TAN, ASN, ACS, ATN, LN, LOG, DEG ou RAD.

Il existe des règles pour les opérateurs admissibles dans les opérations de sécurité. Reportez-vous à la section *Opérateurs valides*.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Voici les opérandes pour l'instruction CMP.

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*.

### Diagramme à relais

Ce qui suit est l'opérande pour le diagramme à relais.

Opérande	Type de données	Format	Description
Expression	SINT INT DINT REAL Type de chaînes	Immédiate étiquette	Une expression composée d'étiquettes et/ou de valeurs immédiates séparées par des opérateurs

### Expressions de formatage

Pour chaque opérateur utilisé dans une expression, il faut fournir un ou deux opérandes (étiquettes ou valeurs immédiates). Utiliser le tableau suivant pour formater les opérateurs et opérandes au sein d'une expression.

Pour les opérateurs qui opèrent sur :	Utiliser ce format :	Exemple
Un opérande	operator(operand)	ABS(tag)
Deux opérandes	operand_a operator operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

### Définir l'ordre des opérations

Les opérations d'une expression sont exécutées par instruction dans un ordre prédéfini qui n'est pas nécessairement leur ordre d'apparition dans l'expression. L'ordre des opérations peut être spécifié en regroupant certains termes dans des parenthèses, obligeant ainsi l'instruction à exécuter les opérations entre parenthèses avant les autres.

Les opérations de rang égal sont exécutées de gauche à droite.

Order	Fonctionnement
1	( )
2	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3	**
4	- (négation), NOT
5	*, /, MOD
6	- (soustraction), +
7	AND
8	XOR
9	OR
10	<, <=, >, >=, =, <>

### Utilisation des chaînes dans une expression

Pour utiliser des chaînes de caractères ASCII dans une expression, observer ces lignes directrices :

- Une expression peut comparer deux étiquettes de chaînes.
- On ne peut pas entrer directement des caractères ASCII dans une expression.
- Les opérateurs suivants sont admis :

Opérateur	Description
=	Égal
<	Inférieur à
<=	Inférieur à ou égal
>	Supérieur à
>=	Supérieur à ou égal
<>	Différent

- Les chaînes sont égales si leurs caractères correspondent.
- Les caractères ASCII sont sensibles à la casse. Le A majuscule (\$41) n'est pas égal au a minuscule (\$61).

- Les valeurs hexadécimales des caractères permettent de définir si une chaîne est plus ou moins longue qu'une autre.
- Lorsque deux chaînes sont triées comme dans un annuaire téléphonique, l'ordre des chaînes détermine celle qui est supérieure.

ASCII Characters	Hex Codes
tab	\$31\$61\$62
tb	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

**Affecte les indicateurs d'état mathématique**

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Non
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	L'instruction CMP affecte les indicateurs d'état mathématique si l'expression contient un opérateur (par exemple, +, -, *, /), qui affecte les indicateurs d'état mathématique.

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

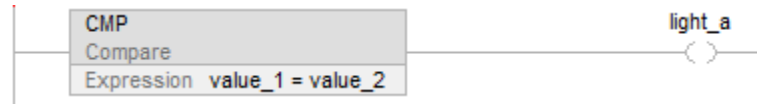
**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Si une expression est évaluée à faux La condition de sortie d'échelon est définie sur faux
Post-scrutation	N/A.

**Exemple**

**Diagramme à relais**



Si value\_1 est égale à value\_2, light\_a est défini sur vrai. Si value\_1 n'est pas égale à value\_2, light\_a est défini sur faux.

**Voir aussi**

[Instructions de comparaison](#) sur la [page 297](#)

[Opérateurs valides](#) sur la [page 371](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

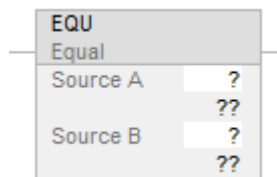
**Égal à (EQU)**

Cette instruction s'applique aux automates CompactLogix 5370, ControlLogix 5570 et Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

Si activé, l'instruction EQU et l'opérateur = testent si la Source A est égale à la Source B.

**Langages disponibles**

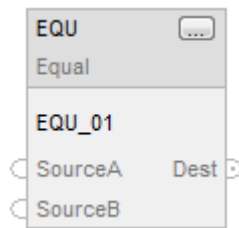
**Diagramme à relais**



**Diagramme de bloc fonctionnel**

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur « = » avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

#### Comparaison numérique

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source A

#### Comparaison de chaînes

**Astuce :** Les valeurs littérales de chaîne immédiates ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A



## Diagramme de bloc fonctionnel

### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
EQU	FBD_COMPARE	étiquette	Structure EQU

### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB
SourceB	REAL	Valeur à tester avec la SourceA

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir sur vrai lorsque la SourceA est égale à la SourceB. Définir sur faux lorsque la SourceA n'est pas égale à la SourceB.

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB.

SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA
---------------	---	---------------------------------

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir sur vrai lorsque la SourceA est égale à la SourceB. Définir sur faux lorsque la SourceA n'est pas égale à la SourceB.

Reportez-vous aux *fonctions FBD*.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Reportez-vous à *Organigramme de comparaison de chaînes EQU* pour consulter les défauts.

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<b>Comparaison numérique :</b> Si la Source A et la Source B ne sont pas des NAN et que la Source A est égale à la Source B. Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux.

	<b>Comparaison de chaînes :</b> Reportez-vous à Organigramme de comparaison de chaînes EQU. Si la sortie est fausse Définir la condition de sortie d'échelon sur faux sinon Définir la condition de sortie d'échelon sur vrai
Post-scrutation	N/A

### Diagramme de bloc fonctionnel

#### Bloc FBD

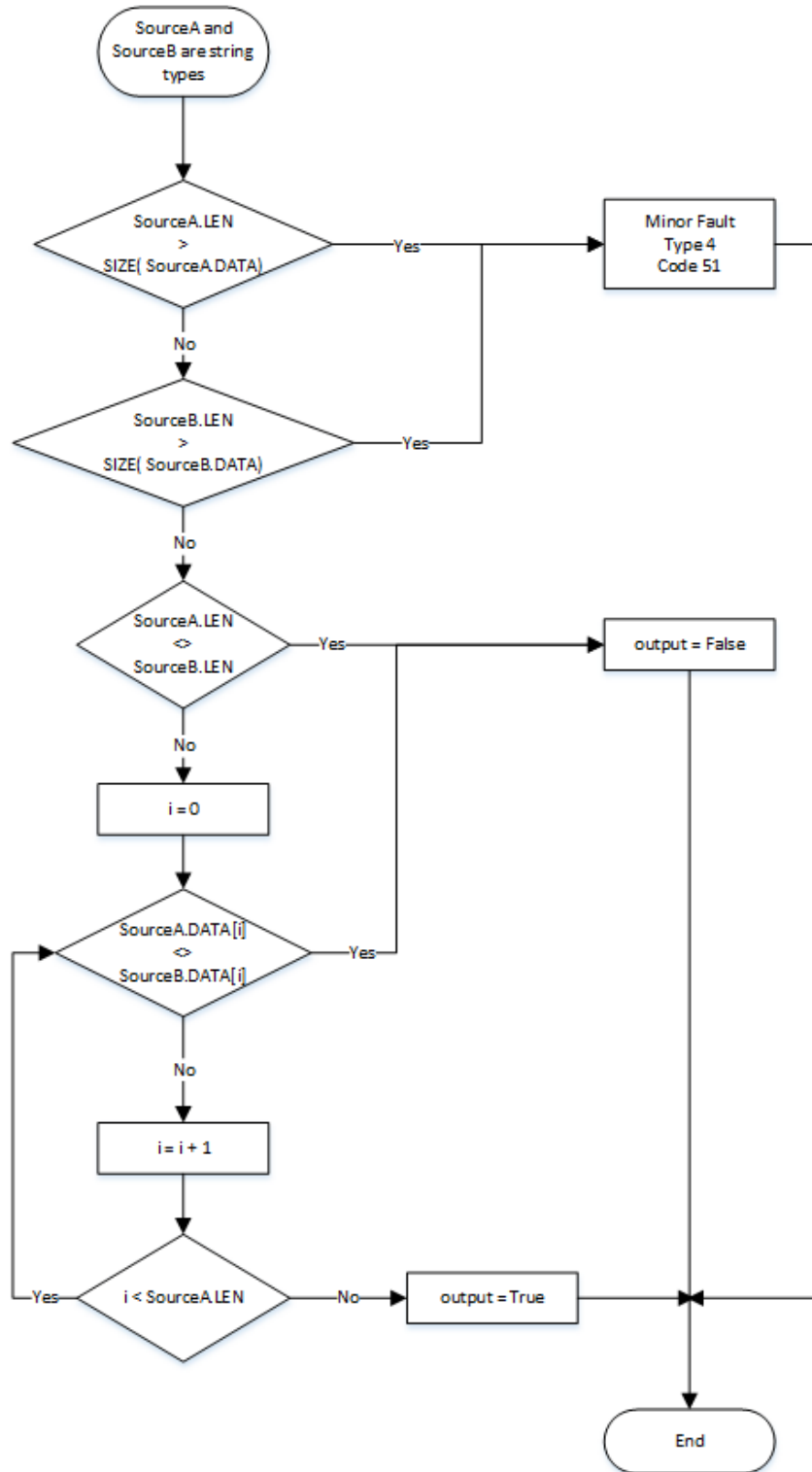
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<b>Comparaison numérique :</b> Définir EnableOut sur EnableIn Si SourceA et SourceB ne sont pas NAN et SourceA est égale à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

#### Fonction FBD

**Astuce :** La fonction FBD est applicable à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

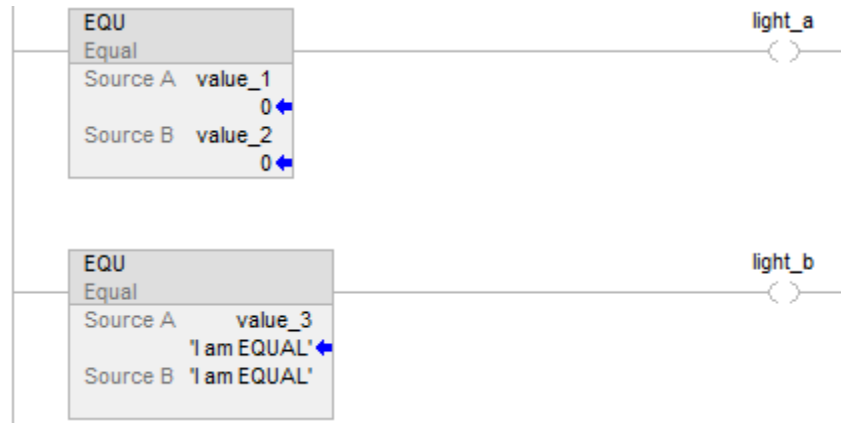
Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<b>Comparaison numérique :</b> Si SourceA et SourceB ne sont pas NAN et SourceA est égale à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

### Organigramme de comparaison de chaînes EQU



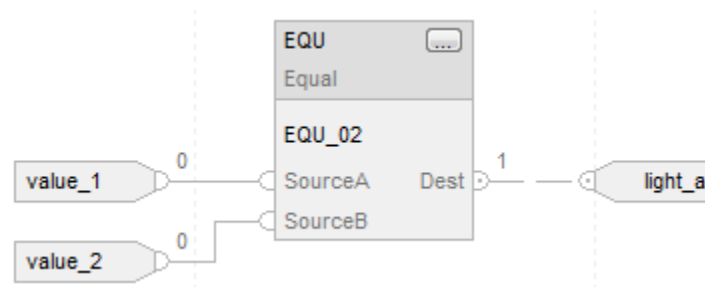
**Exemples**

**Diagramme à relais**



**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Texte structuré**

```

if value_1 = value_2 then
    light_a := 1;
sinon
    light_a := 0;
end_if;

if value_3 = 'I am EQUAL' then

```

```
light_b := 1;
```

```
sinon
```

```
light_b := 0;
```

```
end_if;
```

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

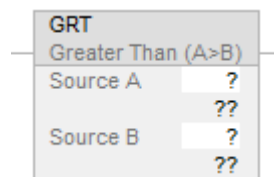
## Supérieur à (GRT)

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l’instruction GRT et l’opérateur  $>$  testent si la Source A est supérieure à la Source B.

### Langages disponibles

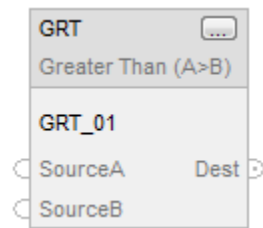
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utilisez l'opérateur  $>$  avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

**Comparaison numérique**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source A

**Comparaison de chaînes**

**Astuce :** Les valeurs littérales de chaîne immédiates ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A



## Diagramme de bloc fonctionnel

### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
GRT	FBD_COMPARE	étiquette	Structure GRT

### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB
SourceB	REAL	Valeur à tester avec la SourceA

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir sur vrai lorsque la SourceA est supérieure à la SourceB. Définir sur faux lorsque la SourceA n'est pas supérieure à la SourceB.

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB
SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description (Description)
Dest	BOOL	Définir sur vrai lorsque la SourceA est supérieure à la SourceB. Définir sur faux lorsque la SourceA n'est pas supérieure à la SourceB.

Reportez-vous aux *fonctions FBD*.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Reportez-vous à *Organigramme de comparaison de chaînes GRT* pour consulter les défauts.

Reportez-vous à *Indexer via des tableaux* pour consulter les défauts d'indexation par tableau.

## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<p><b>Comparaison numérique :</b> Si la Source A et la Source B ne sont pas des NAN et que la Source A est supérieure à la Source B. Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux.</p> <p><b>Comparaison de chaînes :</b> Reportez-vous à <i>Organigramme de comparaison de chaînes GRT</i>. Si la sortie est fausse Définir la condition de sortie d'échelon sur faux ou bien Définir la condition de sortie d'échelon sur vrai</p>
Post-scrutation	N/A

### Diagramme de bloc fonctionnel

#### Bloc FBD

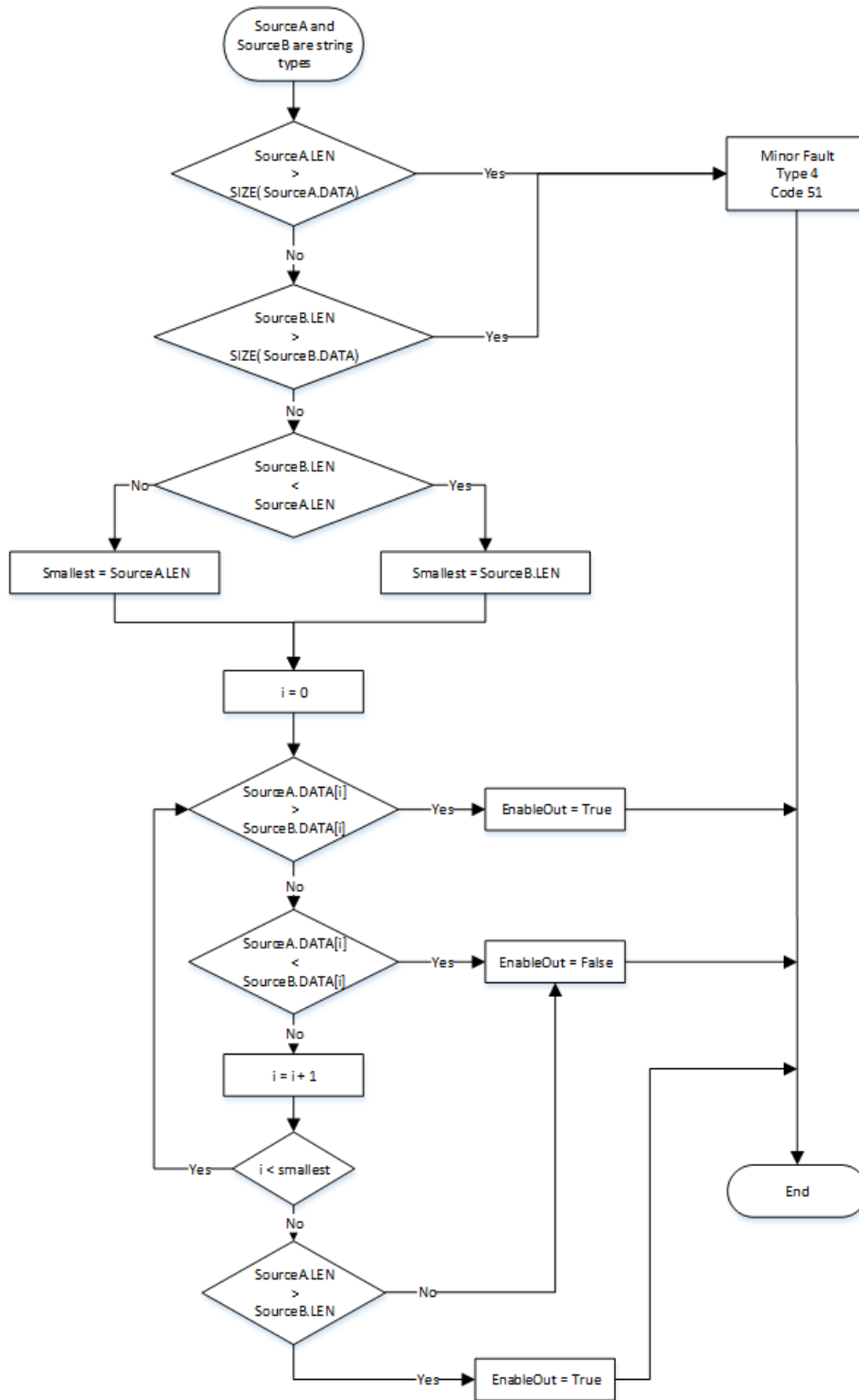
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<p><b>Comparaison numérique :</b> Définir EnableOut sur EnableIn Si SourceA et SourceB ne sont pas NAN et SourceA est supérieure à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.</p>
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<b>Comparaison numérique :</b> Si SourceA et SourceB ne sont pas NAN et SourceA est supérieure à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

Organigramme de comparaison de chaînes GRT



Exemple

Diagramme à relais

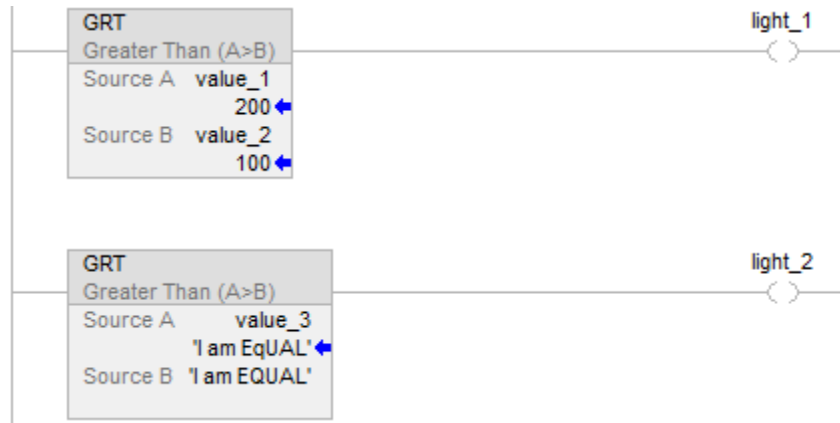
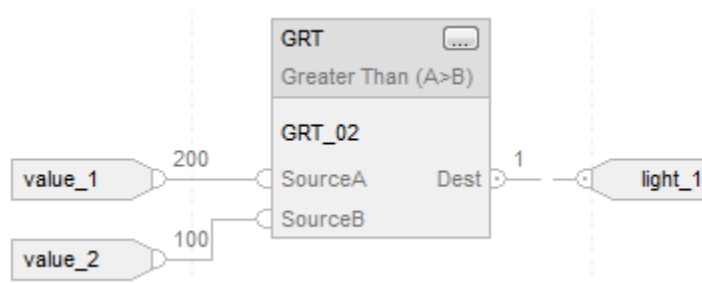
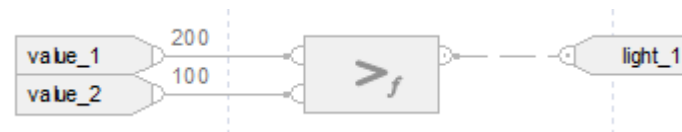


Diagramme de bloc fonctionnel

Bloc FBD



Fonction FBD



Texte structuré

```

if value_1 > value_2 then
    light_1 := 1;
sinon
    light_1 := 0;
end_if;
    
```

```
if value_3 > 'I am EQUAL' then
```

```
    light_2 := 1;
```

```
sinon
```

```
    light_2 := 0;
```

```
end_if;
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

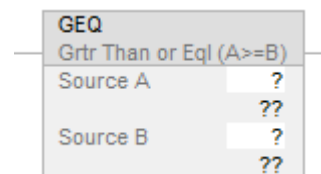
## Supérieur ou égal à (GEQ)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction GEQ et l'opérateur  $\geq$  testent si la Source A est supérieure ou égale à la Source B.

### Langages disponibles

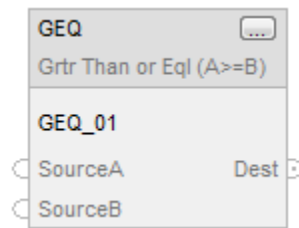
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utilisez l'opérateur  $\geq$  avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*



## Diagramme à relais

### Comparaison numérique

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Immédiate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur à tester avec la Source A

### Comparaison de chaînes

**Astuce :** Les valeurs littérales de chaîne immédiates ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
GEQ	FBD_COMPARE	étiquette	Structure GEQ

#### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB
SourceB	REAL	Valeur à tester avec la SourceA

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir sur vrai lorsque la SourceA est supérieure ou égale à la SourceB. Définir sur faux lorsque la SourceA est inférieure à la SourceB.

#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB.

SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA.
---------------	---	----------------------------------

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir sur vrai lorsque la SourceA est supérieure ou égale à la SourceB. Définir sur faux lorsque la SourceA est inférieure à la SourceB.

Reportez-vous aux *fonctions FBD*.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Reportez-vous à *Organigramme de comparaison de chaînes GEQ* pour consulter les défauts.

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<p><b>Comparaison numérique :</b></p> <p>Si la Source A et la Source B ne sont pas des NAN et que la Source A est supérieure ou égale à la Source B.</p> <p>Définir la condition de sortie d'échelon sur vrai</p> <p>sinon</p> <p>Définir la condition de sortie d'échelon sur faux.</p>

	<p><b>Comparaison de chaînes :</b>                  Reportez-vous à Organigramme de comparaison de chaînes GEQ.                  Si la sortie est fausse                      Définir la condition de sortie d'échelon sur faux                  sinon                      Définir la condition de sortie d'échelon sur vrai</p>
Post-scrutation	N/A

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<p><b>Comparaison numérique :</b>                  Définir EnableOut sur EnableIn                  Si SourceA et SourceB ne sont pas NAN et SourceA est supérieure ou égale à SourceB.                      Définir Dest sur vrai                  sinon                      Définir Dest sur faux.</p>
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

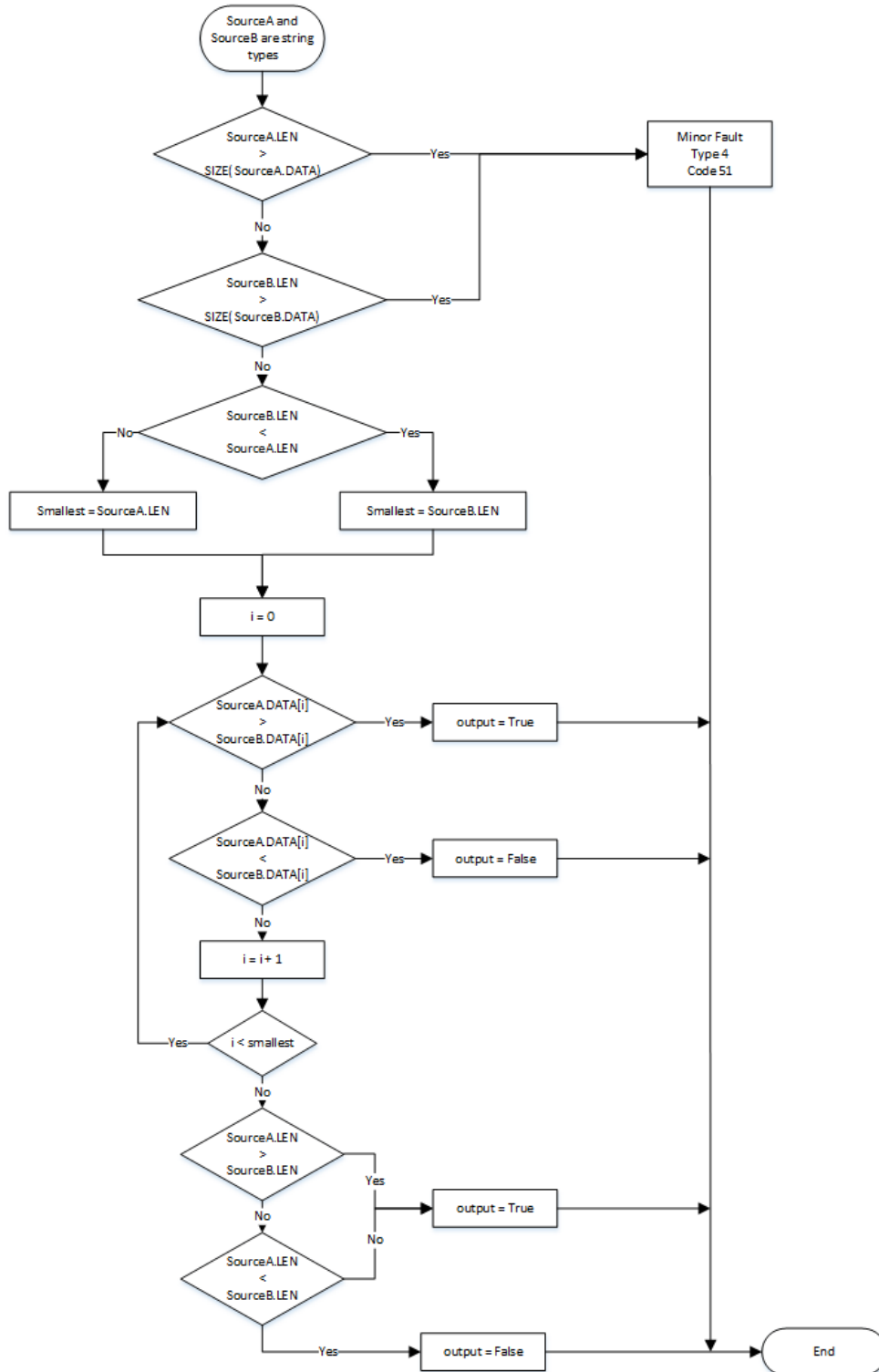
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<p><b>Comparaison numérique :</b>                  Si SourceA et SourceB ne sont pas NAN et SourceA est supérieure ou égale à SourceB.                      Définir Dest sur vrai                  sinon                      Définir Dest sur faux.</p>
Première exécution de l'instruction	N/A

Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Organigramme de comparaison de chaînes GEQ**



Exemple

Diagramme à relais

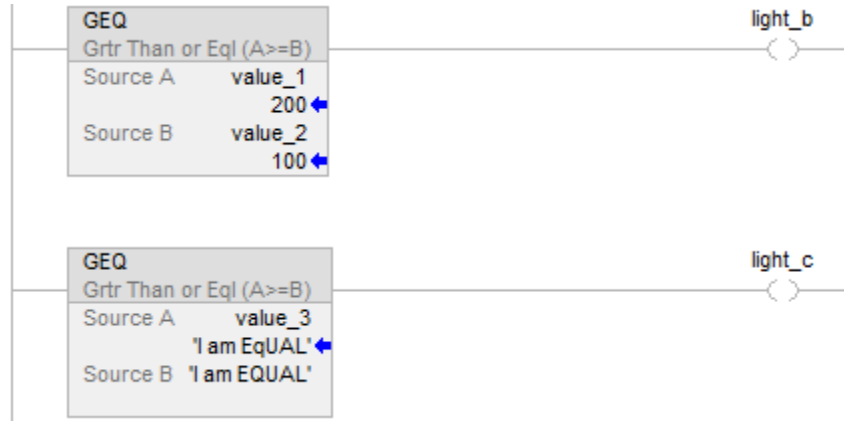
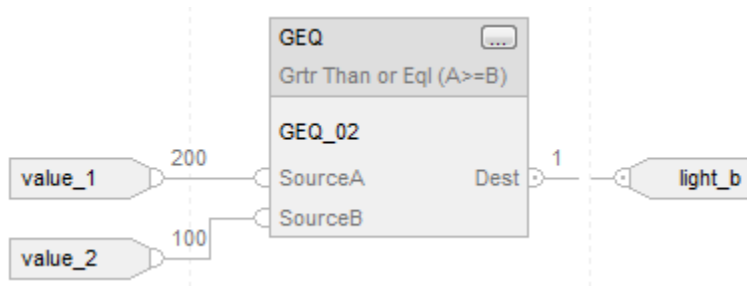


Diagramme de bloc fonctionnel

Bloc FBD



Fonction FBD



Texte structuré

if value\_1 >= value\_2 then

light\_b := 1;

sinon

light\_b := 0;

```

end_if;

if value_3 >= 'I am EQUAL' then

    light_c := 1;

sinon

    light_c := 0;

end_if;

```

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

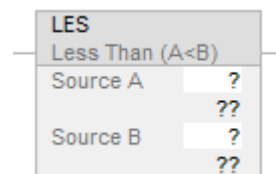
[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

**Inférieur à (LES)**

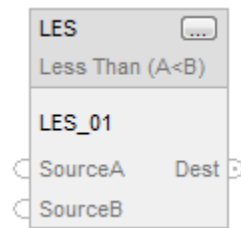
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction LES et l'opérateur  $<$  testent si la Source A est inférieure à la Source B.

**Langages disponibles****Diagramme à relais****Diagramme de bloc fonctionnel**

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utilisez l'opérateur < avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*



## Diagramme à relais

## Comparaison numérique

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur à tester avec la Source A

## Comparaison de chaînes

**Astuce :** Les valeurs littérales de chaîne immédiates s'appliquent à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A

## Diagramme de bloc fonctionnel

### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
LES	FBD_COMPARE	étiquette	Structure LES

### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB
SourceB	REAL	Valeur à tester avec la SourceA

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir sur vrai lorsque la SourceA est inférieure à la SourceB. Définir sur faux lorsque la SourceA n'est pas inférieure à la SourceB.

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB.
SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir sur vrai lorsque la SourceA est inférieure à la SourceB. Définir sur faux lorsque la SourceA n'est pas inférieure à la SourceB.

Reportez-vous aux *fonctions FBD*.

#### Affecter les indicateurs d'état mathématique

Non

#### Défauts majeurs/mineurs

Reportez-vous à *Organigramme de comparaison de chaînes LES* pour consulter les défauts.

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<p><b>Comparaison numérique :</b>                      Si la Source A et la Source B ne sont pas des NAN et que la Source A est inférieure à la Source B.                      Définir la condition de sortie d'échelon sur vrai                      sinon                      Définir la condition de sortie d'échelon sur faux.</p>
	<p><b>Comparaison de chaînes :</b>                      Reportez-vous à Organigramme de comparaison de chaînes LES.                      Si la sortie est fausse                      Définir la condition de sortie d'échelon sur faux                      sinon                      Définir la condition de sortie d'échelon sur vrai</p>
Post-scrutation	N/A

**Diagramme de bloc fonctionnel**

**Bloc FBD**

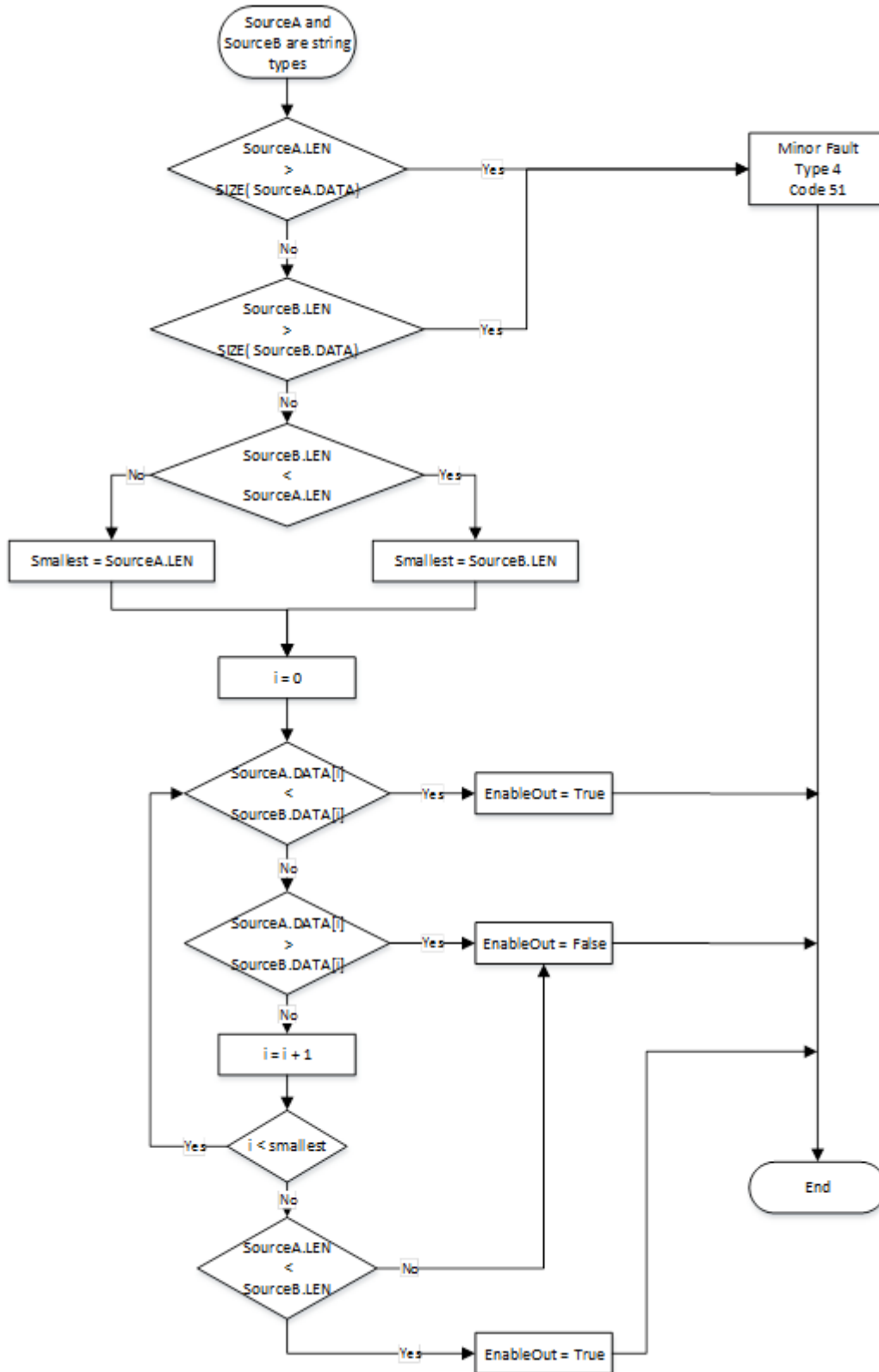
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<p><b>Comparaison numérique :</b>                      Si SourceA et SourceB ne sont pas NAN et SourceA est inférieure à SourceB.                      Définir Dest sur vrai                      sinon                      Définir Dest sur faux.</p>
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

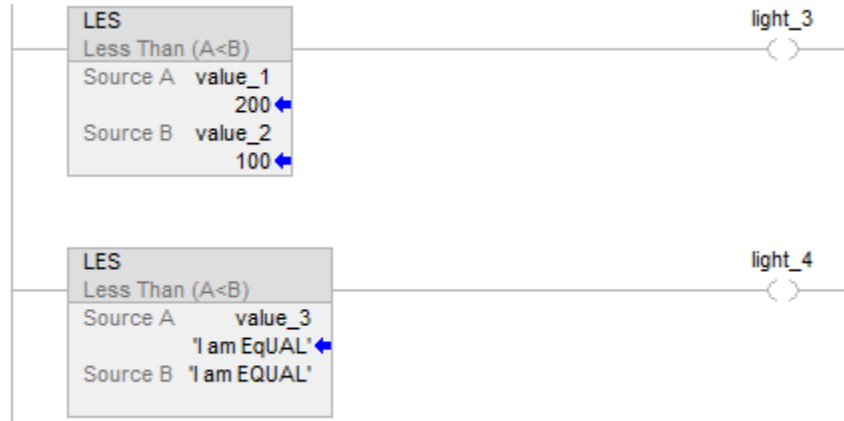
Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<b>Comparaison numérique :</b> Définir EnableOut sur EnableIn Si SourceA et SourceB ne sont pas NAN et SourceA est inférieure à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

Organigramme de comparaison de chaînes LES



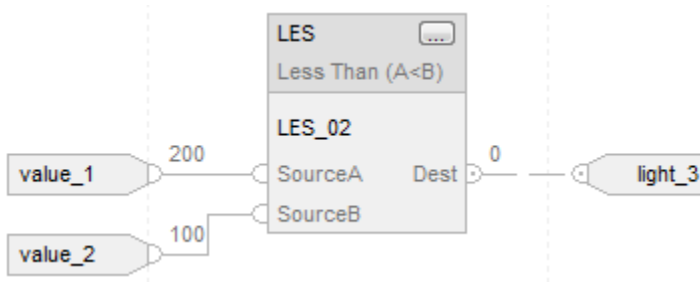
**Exemple**

**Diagramme à relais**



**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Texte structuré**

```

if value_1 < value_2 then
    light_3 := 1;
sinon
    light_3 := 0;
end_if;
    
```

```

if value_3 < 'I am EQUAL' then

    light_4 := 1;

sinon

    light_4 := 0;

end_if;

```

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

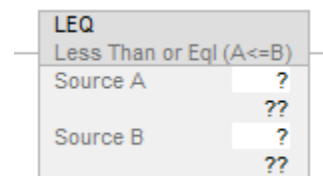
[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

**Inférieur ou égal à (LEQ)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

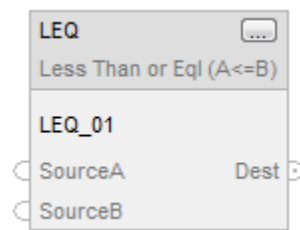
Si activé, l'instruction LEQ et l'opérateur  $\leq$  testent si la Source A est inférieure ou égale à la Source B.

**Langages disponibles****Diagramme à relais****Diagramme de bloc fonctionnel**

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :



### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utilisez l'opérateur  $\leq$  avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

**Comparaison numérique**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur à tester avec la Source A

**Comparaison de chaînes**

**Astuce :** Les valeurs littérales de chaîne immédiates ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A

## Diagramme de bloc fonctionnel

### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
LEQ	FBD_COMPARE	étiquette	Structure LEQ

### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB
SourceB	REAL	Valeur à tester avec la SourceA

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Défini sur vrai lorsque la SourceA est inférieure ou égale à la SourceB. Définir sur faux lorsque la SourceA est supérieure à la SourceB.

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB.
SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Défini sur vrai lorsque la SourceA est inférieure ou égale à la SourceB. Définir sur faux lorsque la SourceA est supérieure à la SourceB.

Reportez-vous aux *fonctions FBD*.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Reportez-vous à *Organigramme de comparaison de chaînes LEQ* pour consulter les défauts.

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<p><b>Comparaison numérique :</b> Si la Source A et la Source B ne sont pas des NAN et que la Source A est inférieure ou égale à la Source B. Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux.</p> <p><b>Comparaison de chaînes :</b> Reportez-vous à <i>Organigramme de comparaison de chaînes LEQ</i>. Si la sortie est fausse Définir la condition de sortie d'échelon sur faux sinon Définir la condition de sortie d'échelon sur vrai</p>
Post-scrutation	N/A

### Diagramme de bloc fonctionnel

#### Bloc FBD

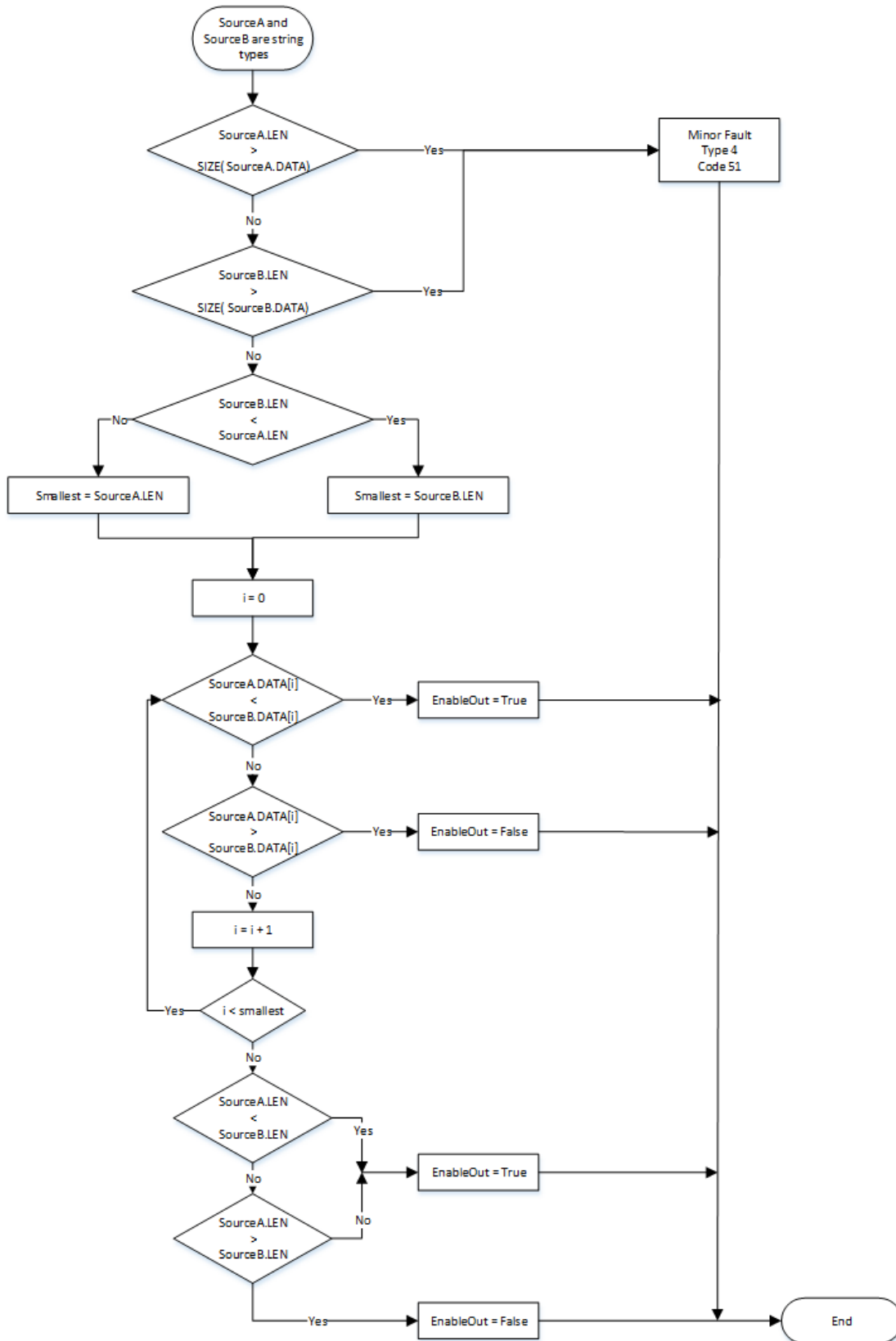
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<p><b>Comparaison numérique :</b> Définir EnableOut sur EnableIn Si SourceA et SourceB ne sont pas NAN et SourceA est inférieure ou égale à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.</p>
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

**Astuce :** La fonction FBD est applicable à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 .

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<b>Comparaison numérique :</b> Si SourceA et SourceB ne sont pas NAN et SourceA est inférieure ou égale à SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

Organigramme de comparaison de chaînes LEQ



Exemple

Diagramme à relais

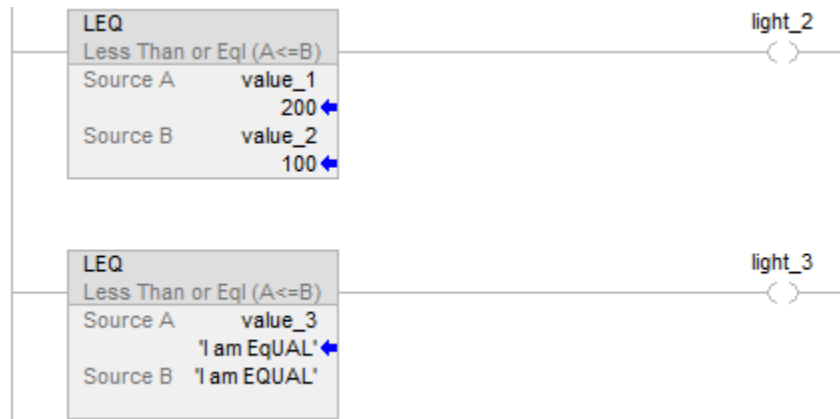
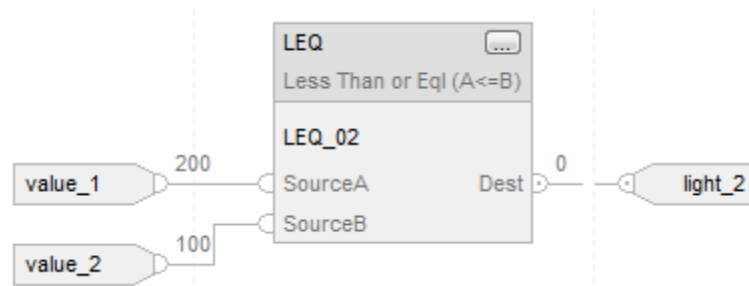


Diagramme de bloc fonctionnel

Bloc FBD



Fonction FBD



Texte structuré

```

if value_1 <= value_2 then
    light_2 := 1;
sinon
    light_2 := 0;
end_if;
    
```



```

if value_3 <= 'I am EQUAL' then
    light_3 := 1;
sinon
    light_3 := 0;
end_if;

```

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

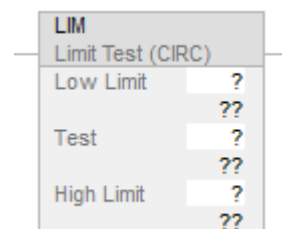
[Fonctions FBD](#) sur la [page 430](#)

**Limite (LIM)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

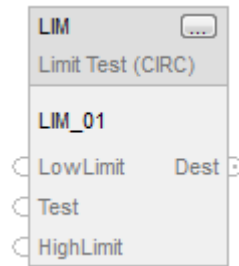
L'instruction LIM teste si la valeur Test se trouve dans la plage de Low Limit et de High Limit telle qu'indiquée dans l'organigramme LIM (Vrai).

Si l'un des opérande n'est pas un numéro (NAN), le .EnableOut est défini sur faux.

**Langages disponibles****Diagramme à relais****Diagramme de bloc fonctionnel**

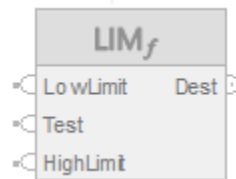
Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

## Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Low Limit	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur de la limite basse.
Test	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec les limites.
High Limit	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur de la limite haute.

## Diagramme de bloc fonctionnel

## Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
LIM	FBD_LIMIT	étiquette	Structure LIM

**Structure FBD\_LIMIT**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
LowLimit	REAL	Valeur de la limite basse.
Test	REAL	Valeur à tester avec les limites.
HighLimit	REAL	Valeur de la limite haute.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir sur vrai si le test de limite est vrai. Définir sur faux si le test de limite est faux.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
Low Limit	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur de la limite basse

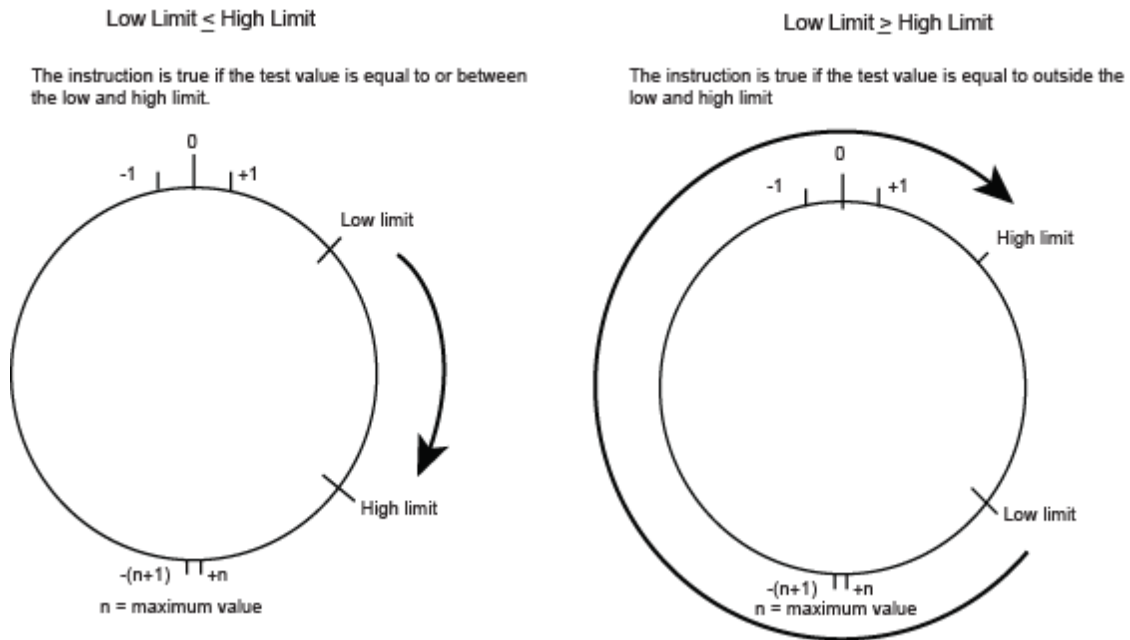
Test	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec les limites.
High Limit	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur de la limite haute.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir sur vrai si le test de limite est vrai. Définir sur faux si le test de limite est faux.

Reportez-vous aux *fonctions FBD*.

### Fonctionnement

Cette section illustre le fonctionnement de l'instruction LIM.



Si Low Limit :	Et que la valeur du test est :	Alors EnableOut est :
< or = to High Limit	égal à ou à l'intérieur des limites différent de ou en dehors des limites	vrai faux
> High Limit	égal à ou en dehors des limites différent de ou à l'intérieur des limites	vrai faux

La transition d'entiers signés d'un nombre positif maximal vers un nombre négatif maximal lorsque le bit le plus significatif est vrai. Par exemple, dans les entiers 16 bit (type INT), l'entier positif maximal est 32 767, ce qui est représenté, en hexadécimal par 16#7FFF (les bits 0 à 14 sont tous vrais). Si ce nombre s'incrémente de un, le résultat est 16#8000 (le bit 15 est vrai). Pour les entiers signés, le nombre hexadécimal 16#8000 est égal à -32 768 en décimal. Si on incrémente à partir de là jusqu'à ce que les 16 bits soient tous définis, on arrive à 16#FFFF, ce qui est égal à -1 en décimal.

On peut représenter cela comme une ligne circulaire de nombres. L'instruction LIM démarre à Low Limit et s'incrémente dans le sens horaire jusqu'à atteindre High Limit. Tout valeur Test dans la gamme horaire située entre Low Limit et High Limit définit EnableOut sur vrai. Tout valeur Test dans la gamme horaire située entre High Limit et Low Limit définit EnableOut sur faux.

Si l'un des opérande n'est pas un numéro (NaN), le .EnableOut est défini sur faux.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Reportez-vous à <i>Organigramme LIM (vrai)</i> Si la sortie est vraie Définir la condition de sortie d'échelon sur vrai. sinon Définir la condition de sortie d'échelon sur faux.
Post-scrutation	N/A

**Diagramme de bloc fonctionnel****Bloc FBD**

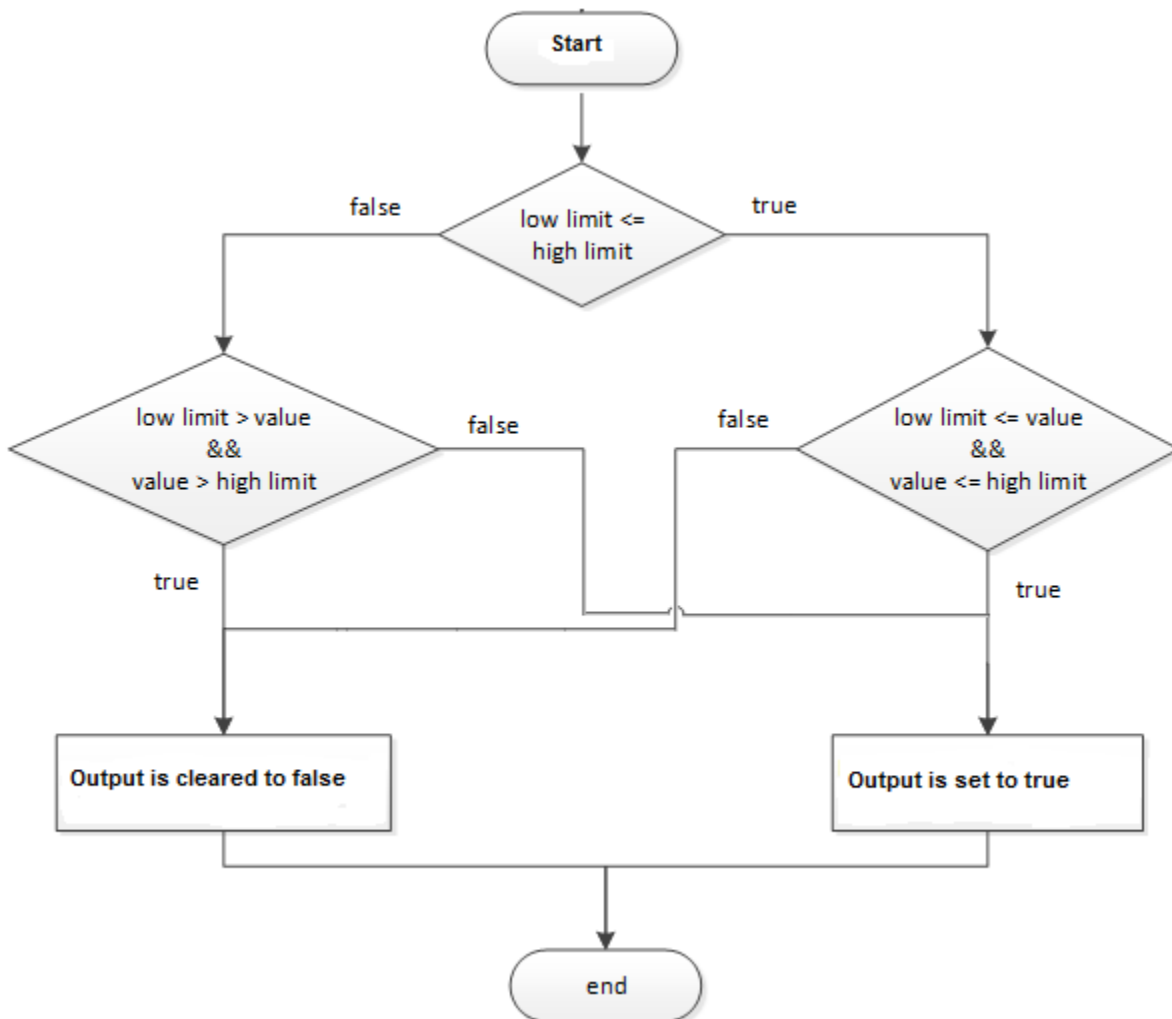
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn.
EnableIn est vrai	Définir EnableOut sur EnableIn. Reportez-vous à Organigramme LIM (vrai) Dest = output
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Reportez-vous à Organigramme LIM (vrai) Dest = sortie
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Organigramme LIM (vrai)**



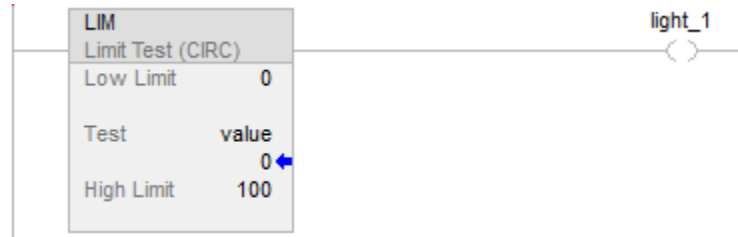


**Exemples**

**Exemple 1: Limite basse <= Limite haute**

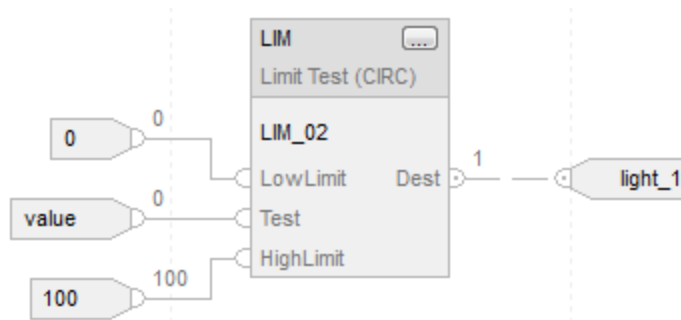
Lorsque la valeur Test est égale ou supérieure à Limite basse et que la valeur Test est inférieure ou égale à Limite haute, light\_1 est défini.

**Diagramme à relais**

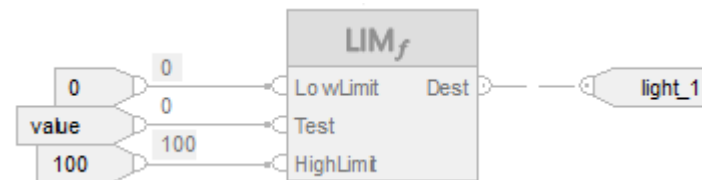


**Diagramme de bloc fonctionnel**

**Bloc FBD**



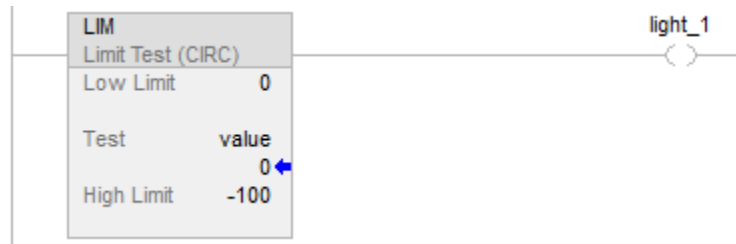
**Fonction FBD**



**Exemple 2: Limite basse > Limite haute**

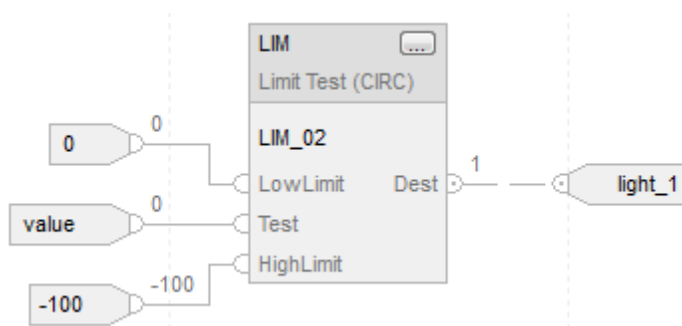
Lorsque la valeur > ou = à 0 ou que la valeur < ou = à -100, définir light\_1 sur vrai. Lorsque la valeur < 0 et la valeur > -100, définir light\_1 sur faux.

**Diagramme à relais**

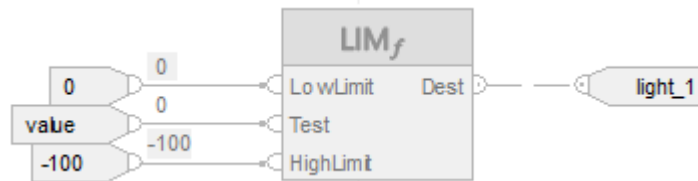


**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Voir aussi**

[Instructions de comparaison](#) sur la [page 297](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

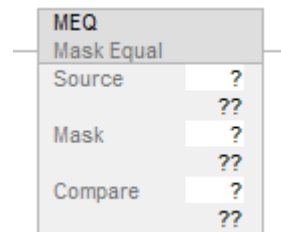
**Masque égal à (MEQ)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction MEQ fait passer les valeurs Source et Compare dans un Masque et compare les résultats.

### Langages disponibles

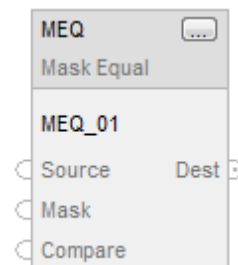
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

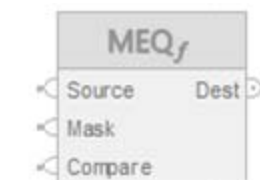
Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source (Source)	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	immediate étiquette	Valeur à tester avec Compare.
Masque (Mask)	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	immediate étiquette	Quels bits il faut bloquer ou laisser passer.
Compare	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	immediate étiquette	Valeur à tester avec Source.

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
MEQ	FBD_MASK_EQUAL	étiquette	Structure MEQ

**Structure FBD\_MASK\_EQUAL**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	DINT	Valeur à tester avec Compare.
Masque (Mask)	DINT	Définir les bits à bloquer, comme le masque.
Compare	DINT	Valeur à tester avec Source.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	BOOL	Définir sur vrai si le résultat est vrai. Définir sur faux si le résultat est faux.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
Source (Source)	SINT INT DINT LINT USINT UINT UDINT ULINT	Valeur à tester avec Compare.

Masque (Mask)	SINT INT DINT LINT USINT UINT UDINT ULINT	Quels bits il faut bloquer ou laisser passer.
Compare	SINT INT DINT LINT USINT UINT UDINT ULINT	Valeur à tester avec Source.
		L'étiquette SINT ou INT est convertie en valeur DINT par un remplissage avec des zéros.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir sur vrai si le résultat est vrai. Définir sur faux si le résultat est faux.

Reportez-vous aux fonctions FBD.

**Fonctionnement**

Un « 1 » dans le masque signifie que le bit de données est passé. Un « 0 » dans le masque signifie que le bit de données est bloqué. En général, les valeurs Source, Mask et Compare sont toutes du même type de données.

Si avec les types de données SINT ou INT, l'instruction remplit les bits supérieurs de cette valeur avec des 0 pour lui donner la même taille que le type de données DINT.

**Saisir une valeur de masque immédiate**

Lorsque vous entrez un masque, le logiciel de programmation utilise des valeurs décimales par défaut. Pour entrer un masque avec un autre format, il faut faire précéder la valeur par le préfixe qui convient.

Préfixe	Description (Description)
16#	Hexadécimal, tel que 16#0F0F
8#	octal, tel que 8#16
2#	binaire, tel que 2#00110011

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme MEQ (vrai). Si la sortie est vraie Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux
Post-scrutation	N/A

**Diagramme de bloc fonctionnel****Bloc FBD**

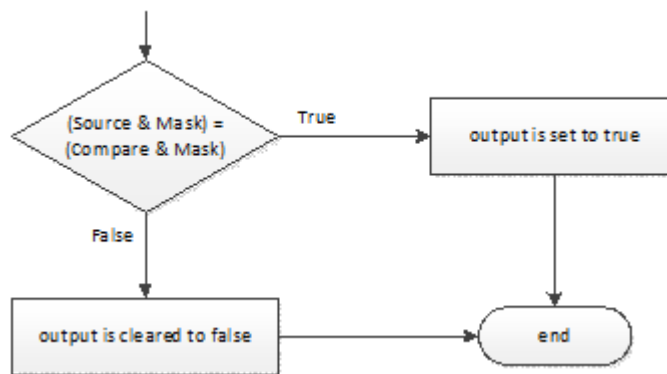
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn.
EnableIn est vrai	Définir EnableOut sur EnableIn. Reportez-vous à l'organigramme MEQ (vrai). Si la sortie est vraie Définir Dest sur vrai sinon Définir Dest sur faux
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Reportez-vous à l'organigramme MEQ (vrai). Si la sortie est vraie Définir Dest sur vrai sinon Définir Dest sur faux
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Organigramme MEQ (vrai)**



**Exemples**

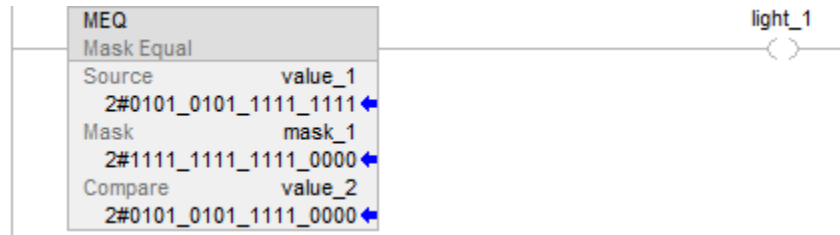
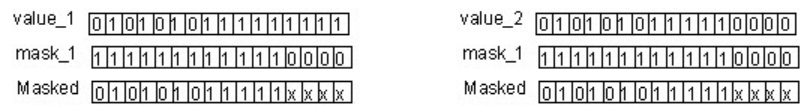
**Exemple 1**

Si value\_1 masqué est égal à value\_2 masqué, définir light\_1 sur vrai. Si value\_1 masqué est différent de value\_2 masqué, définir light\_1 sur faux.

Dans l'exemple suivant, les valeurs masquées sont égales. Un 0 dans le masque empêche l'instruction de comparer ce bit (indiqué à l'aide d'un x dans l'exemple).

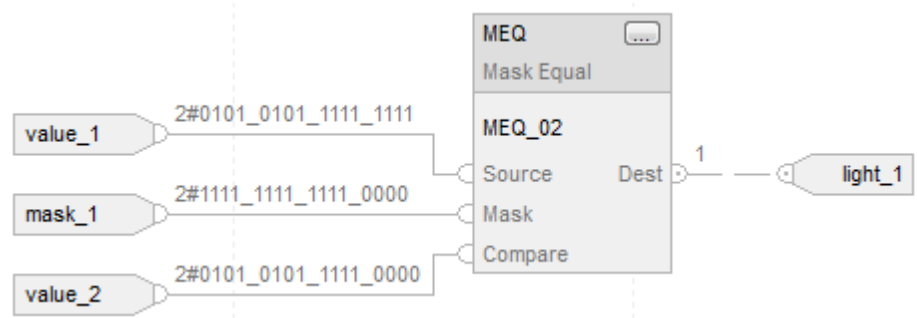


### Diagramme à relais

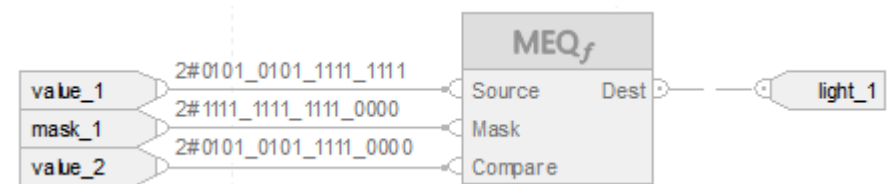


### Diagramme de bloc fonctionnel

#### Bloc FBD



#### Fonction FBD

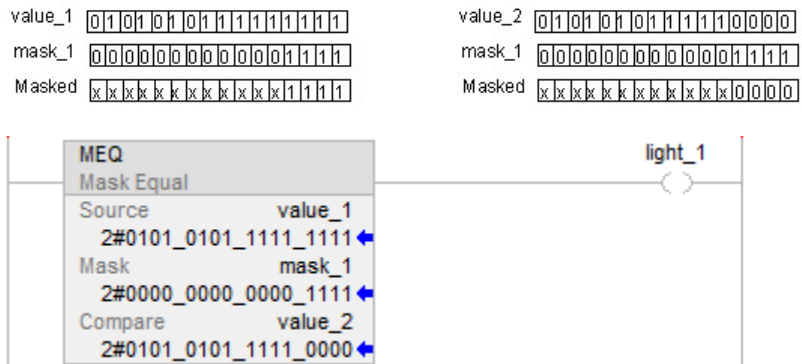


### Exemple 2

Si value\_1 masqué est égal à value\_2 masqué, définir light\_1 sur vrai. Si value\_1 masqué est différent de value\_2 masqué, définir light\_1 sur faux.

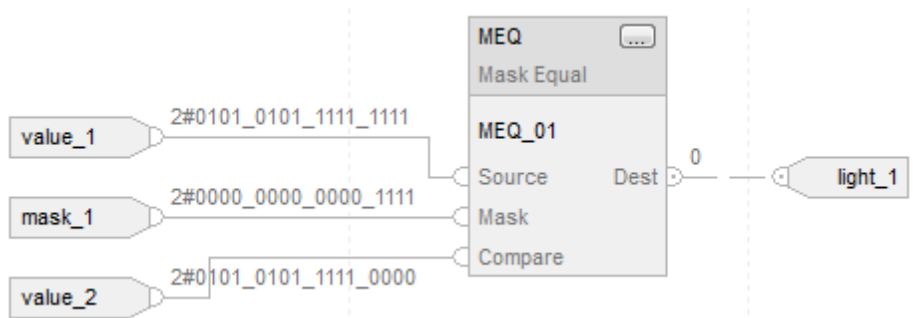
Dans l'exemple suivant, les valeurs masquées sont différentes. Un 0 dans le masque empêche l'instruction de comparer ce bit (indiqué à l'aide d'un x dans l'exemple).

**Diagramme à relais**

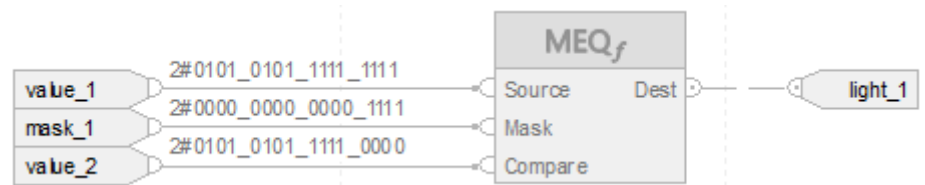


**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Voir aussi**

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Conversions de données](#) sur la [page 889](#)

[Qu'est-ce qu'un remplissage avec des zéros ?](#) sur la [page 372](#)

[Fonctions FBD](#) sur la [page 430](#)

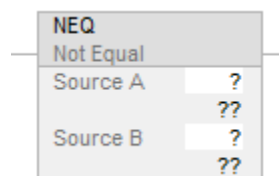
## Différent de (NEQ)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction NEQ et l'opérateur  $\neq$  déterminent si la Source A est différente de la Source B.

### Langages disponibles

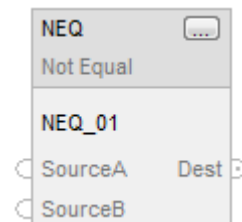
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utilisez l'opérateur  $\lt \gt$  avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

**Opérandes**

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

**Comparaison numérique**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source B
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à tester avec la Source A

### Comparaison de chaînes

**Astuce :** Les valeurs littérales de chaîne immédiates ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande	Type de donnée	Format	Description (Description)
Source A	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source B
Source B	Type de chaînes	Valeur littérale immédiate étiquette	Chaîne à tester avec la Source A

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
NEQ	FBD_COMPARE	étiquette	Structure NEQ

#### Structure FBD\_COMPARE

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à tester avec la SourceB.
SourceB	REAL	Valeur à tester avec la SourceA.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	BOOL	Définir à vrai lorsque la SourceA est différente de la SourceB. Définir à faux lorsque la SourceA est égale à la SourceB.

#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
SourceA (haut)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceB
SourceB (bas)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Valeur à tester avec la SourceA.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	BOOL	Définir à vrai lorsque la SourceA est différente de la SourceB. Définir à faux lorsque la SourceA est égale à la SourceB.

Reportez-vous aux fonctions FBD

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Reportez-vous à *Organigramme de comparaison de chaînes NEQ* pour consulter les défauts.

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	<b>Comparaison numérique :</b> Si la Source A ou la Source B est NAN et que la Source A est différente de la Source B. Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux.
	<b>Comparaison de chaînes :</b> Reportez-vous à <i>Organigramme de comparaison de chaînes NEQ</i> . Si la sortie est fausse Définir la condition de sortie d'échelon sur faux ou bien Définir la condition de sortie d'échelon sur vrai
Post-scrutation	N/A

### Diagramme de bloc fonctionnel

#### Bloc FBD

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	<b>Comparaison numérique :</b> Définir EnableOut sur EnableIn Si la SourceA ou la SourceB est NAN et que la SourceA est différente de la SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

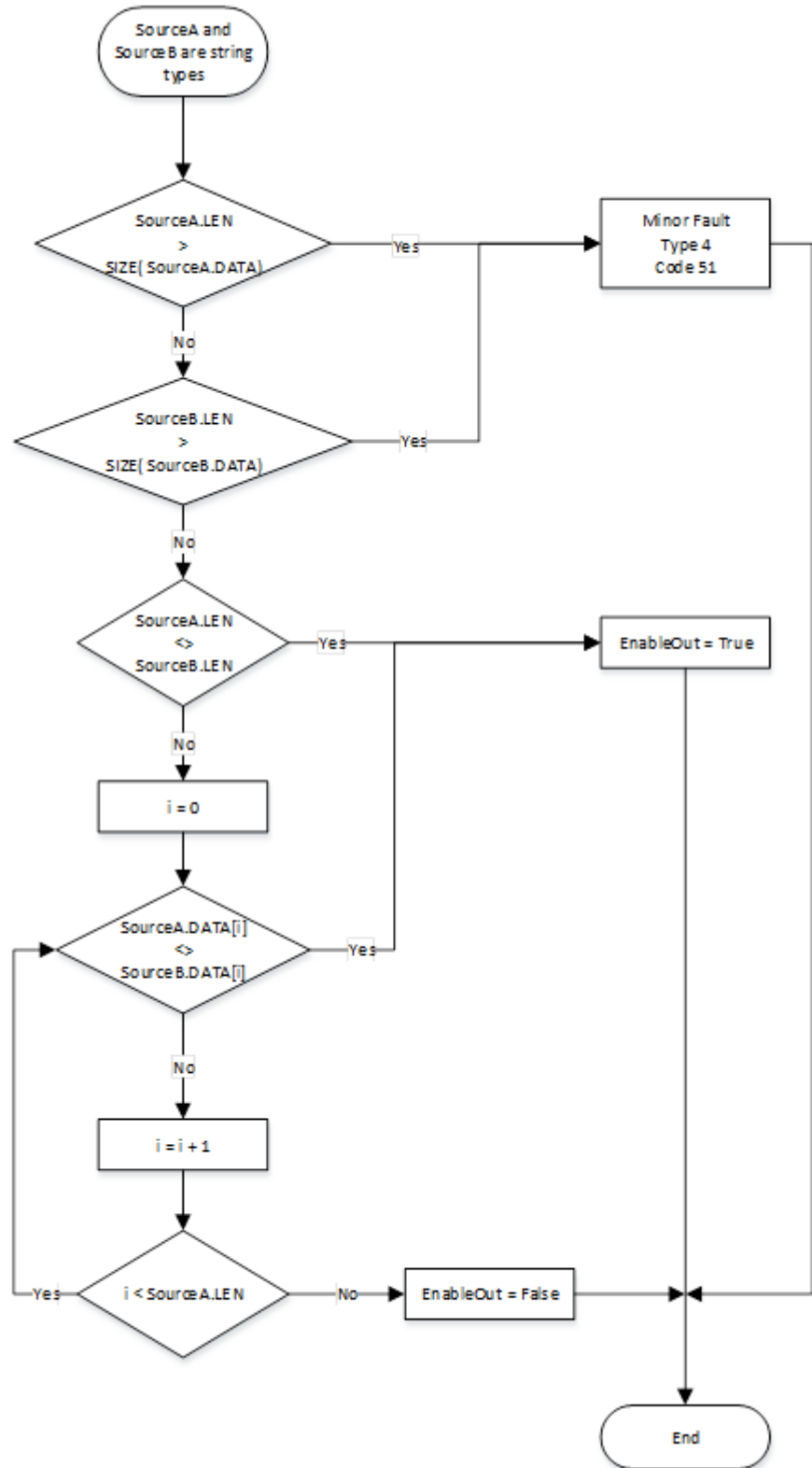
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	<b>Comparaison numérique :</b> Si la SourceA ou la SourceB est NAN et que la SourceA est différente de la SourceB. Définir Dest sur vrai sinon Définir Dest sur faux.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

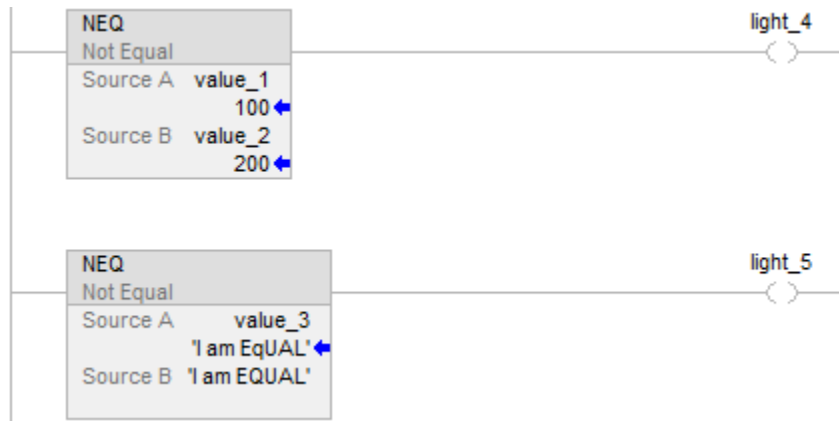


Organigramme de comparaison de chaînes NEQ



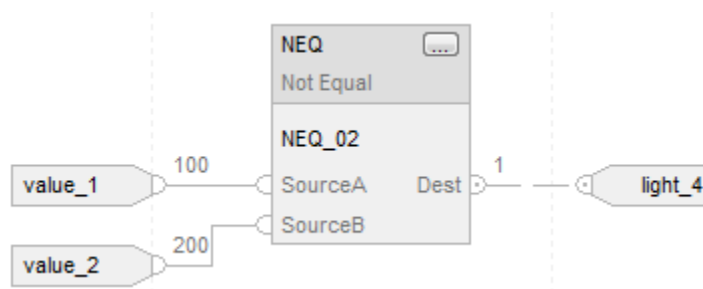
### Exemples

#### Diagramme à relais

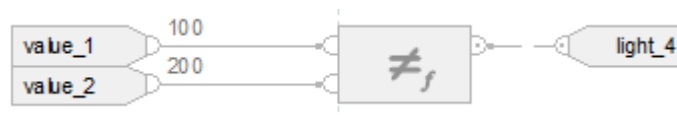


#### Diagramme de bloc fonctionnel

##### Bloc FBD



##### Fonction FBD



##### Texte structuré

```

if value_1 <> value_2 then
    light_4 := 1;
sinon
    light_4 := 0;
end_if;

if value_3 <> 'I am EQUAL' then

```

```
light_5 := 1;
```

```
sinon
```

```
light_5 := 0;
```

```
end_if;
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

## Opérateurs valides

Les opérateurs valides sont les suivants.

Opérateur	Description (Description)	Autorisé dans					
		Index du tableau	FSC	CMP	FAL	CPT	Safety
+	ajouter	X	X	X	X	X	X
-	soustraire/mettre au négatif	X	X	X	X	X	X
*	multiplier	X	X	X	X	X	X
/	diviser	X	X	X	X	X	X
=	égal		X	X			X
<	inférieur à		X	X			X
<=	inférieur à ou égal		X	X			X
>	supérieur à		X	X			X
>=	supérieur à ou égal		X	X			X
<>	différent		X	X			X
**	exposant (x à y)		X	X	X	X	
ABS	valeur absolue		X	X	X	X	X
ACS	Cosinus d'arc		X	X	X	X	
AND	ET au niveau du bit	X	X	X	X	X	X
ASN	Sinus d'arc		X	X	X	X	
ATN	Tangente d'arc		X	X	X	X	
COS	Cosinus		X	X	X	X	

DEG	Radians vers degrés		X	X	X	X	
FRD	BCD à nombre entier	X	X	X	X	X	
LN	Logarithme naturel		X	X	X	X	
LOG	Logarithme décimal		X	X	X	X	
MOD	Diviser modulo		X	X	X	X	X
NOT	NON au niveau du bit	X	X	X	X	X	X
OU	OU au niveau du bit	X	X	X	X	X	X
RAD	Degrés vers radians		X	X	X	X	
SIN	Sinus		X	X	X	X	
SQR	racine carrée	X	X	X	X	X	
TAN	Tangente		X	X	X	X	
TOD	nombre entier à BCD	X	X	X	X	X	
TRN	Tronquer		X	X	X	X	
XOR	OU exclusif au niveau du bit	X	X	X	X	X	X

## Qu'est-ce qu'un remplissage avec des zéros ?

Il existe deux méthodes pour convertir un type d'entier petit en un entier plus grand :

- Remplissage avec des zéros
- Extension de signe

La méthode employée dépend de l'instruction qui utilise l'opérande.

Pour le remplissage avec des zéros, tous les bits au-dessus de la plage du type le plus petit sont remplis de 0.

Par exemple, SINT : 16#87 = -121 converti en DINT renvoie  
16#00000087 = 135

Pour l'extension de signe, tous les bits au-dessus de la plage du type le plus petit sont remplis avec le bit de signe du type le plus petit.

Par exemple, SINT : 16#87 = -121 converti en DINT renvoie  
16#FFFFFF87 = -121

**Voir aussi**

[Masque égal à \(MEQ\)](#) sur la [page 354](#)

## Instructions de calcul/Math

### Instructions de calcul/math

Les instructions calcul/math évaluent les opérations arithmétiques en utilisant une expression ou une instruction arithmétique spécifique.

#### Instructions disponibles

##### Diagramme à relais

<a href="#">CPT</a>	<a href="#">ADD</a>	<a href="#">SUB</a>	<a href="#">MUL</a>	<a href="#">DIV</a>	<a href="#">MOD</a>	<a href="#">SQR</a>	<a href="#">SQRT</a>	<a href="#">NEG</a>	<a href="#">ABS</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	----------------------	---------------------	---------------------

##### Diagramme de bloc fonctionnel

##### Bloc FBD

<a href="#">ADD</a>	<a href="#">SUB</a>	<a href="#">MUL</a>	<a href="#">DIV</a>	<a href="#">MOD</a>	<a href="#">SQR</a>	<a href="#">SQRT</a>	<a href="#">NEG</a>	<a href="#">ABS</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	----------------------	---------------------	---------------------

##### Fonction FBD

$+_f$	$\times_f$	$\div_f$	$\%_f$	$\sqrt{x}_f$	$-x_f$	$ x _f$
<a href="#">ADD</a>	424	<a href="#">DIV</a>	<a href="#">MOD</a>	<a href="#">SQR/SQRT/</a>	411	<a href="#">ABS</a>

##### Texte structuré

<a href="#">SQR</a>	<a href="#">SQRT</a>	<a href="#">ABS</a>
---------------------	----------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
évalue une expression	CPT
ajoute deux valeurs	ADD
soustrait deux valeurs	SUB
multiplie deux valeurs	MUL
divise deux valeurs	DIV
détermine le restant après la division d'une valeur par une autre	MOD

calcule la racine carrée d'une valeur	SQR
prend le signe opposé à celui de la valeur	NEG
prend la valeur absolue d'une valeur	ABS

Vous pouvez mélanger les types de données, mais cela pourrait entraîner une perte en précision et des erreurs d'arrondi et l'instruction prendra plus de temps à s'exécuter. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données en caractères gras correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

L'exécution d'une instruction calcul/math a lieu lors de chaque scrutation de cette instruction, à condition que la condition d'entrée d'échelon soit vraie. Si vous souhaitez que l'expression ne soit évaluée qu'à une fois, utiliser toute instruction d'impulsion pour déclencher l'instruction.

#### Voir aussi

[Instructions de comparaison](#) sur la [page 297](#)

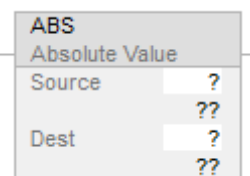
## Valeur absolue (ABS)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Lorsqu'elle est activée, l'instruction ABS et l'opérateur prennent la valeur absolue de la Source. L'instruction mémorise le résultat dans Dest tandis que l'opérateur affiche tout simplement le résultat. Un débordement est indiqué si le résultat est la valeur entière négative maximal (par exemple, -128 pour SINT, -32 768 pour INT et -2 147 483 648 pour DINT).

#### Langages disponibles

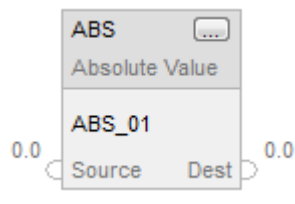
#### Diagramme à relais



## Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser ABS en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à partir de laquelle prendre la valeur absolue.
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
ABS	FBD_MATH_ADVANCED	étiquette	Structure ABS

**Structure FBD\_MATH\_ADVANCED**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	REAL	Valeur à partir de laquelle prendre la valeur absolue.



Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande d'entrée (broche gauche)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
Source (Source)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à partir de laquelle prendre la valeur absolue.

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
Dest	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

**Affecter les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecter les indicateurs d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Dest = valeur absolue de la Source.
Post-scrutation	N/A

**Diagramme de bloc fonctionnel**

**Bloc FBD**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
EnableIn est faux.	Définir EnableOut sur EnableIn.

EnableIn est vrai	Dest = valeur absolue de la Source. En cas de débordement Définir EnableOut sur faux. sinon Définir EnableOut sur vrai.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

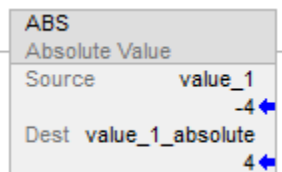
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = valeur absolue de la Source
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

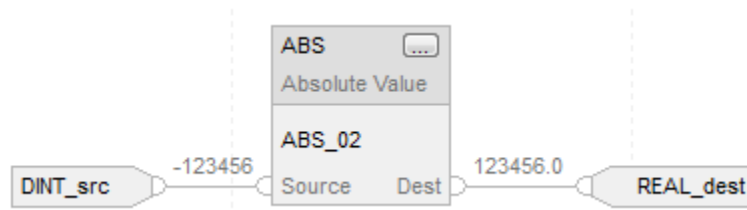
**Exemples**

**Diagramme à relais**



## Diagramme de bloc fonctionnel

### Bloc FBD



### Fonction FBD



### Texte structuré

```
DINT_dest := ABS(DINT_src);
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

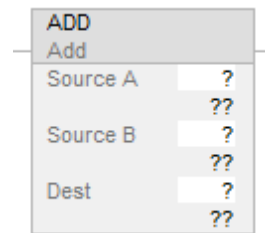
## Addition (ADD)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction ADD et l'opérateur « + » ajoutent Source A à Source B.

## Langages disponibles

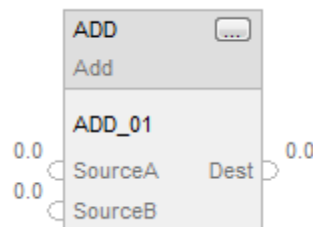
### Diagramme à relais



### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** L'élément Fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur « + » dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
SourceA	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Immédiate étiquette	Valeur à ajouter à la Source B
SourceB	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	Immédiate étiquette	Valeur à ajouter à la Source A

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction
------	-----------------------------	---	-----------	---

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
ADD	FBD_MATH	étiquette	Structure ADD

#### Structure FBD\_MATH

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à ajouter à SourceB.
SourceB	REAL	Valeur à ajouter à SourceA.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

**Fonction FBD**

**Astuce :** L'élément Fonction FBD est applicable uniquement aux automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement	Description (Description)
SourceA (haut)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à ajouter à SourceB.
SourceB (bas)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à ajouter à SourceA.

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement	Description (Description)
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux *fonctions FBD*.



### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

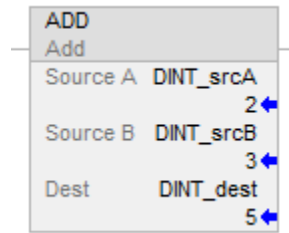
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon, Dest = Source A + Source B
Post-scrutation	N/A

#### Diagramme de bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Dest = SourceA + SourceB En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

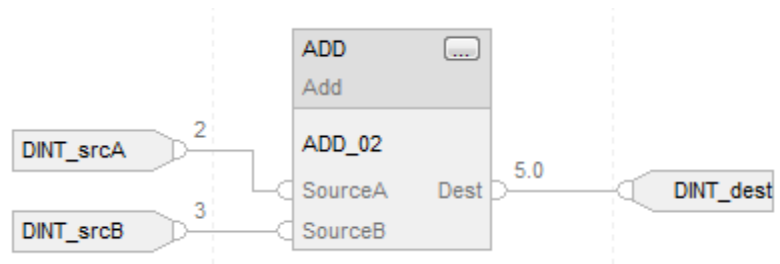
**Exemple**

**Diagramme à relais**



**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Texte structuré**

DINT\_dest := DINT\_srcA + DINT\_srcB;

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

## Calcul (CPT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Lorsqu'elle est activée, l'instruction CPT évalue l'expression et place le résultat dans Dest.

L'instruction CPT permet d'utiliser des expressions complexes dans la même instruction.

Lors de l'évaluation de l'expression, tous les opérandes non-LREAL seront convertis en LREAL avant de procéder aux calculs, si l'une des conditions suivantes est vraie :

- Tous les opérandes dans l'expression sont LREAL.
- L'expression contient SIN, COS, TAN, ASN, ACS, ATN, LN, LOG, DEG ou RAD.
- Dest est LREAL

Il existe des règles pour les opérateurs admissibles dans les opérations de sécurité. Reportez-vous à la section *Opérateurs valides*.

### Langages disponibles

#### Diagramme à relais

CPT	
Compute	
Dest	?
	??
Expression	?

#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée	Format	Description (Description)
Dest	SINT INT DINT REAL	étiquette	Étiquette où enregistrer le résultat
Expression	SINT INT DINT REAL	immediate étiquette	Une expression comportant des étiquettes et/ou des valeurs immédiates séparées par des opérateurs.

### Expressions de formatage

Pour chaque opérateur utilisé dans une expression, il faut fournir un ou deux opérandes (étiquettes ou valeurs immédiates). Utiliser le tableau suivant pour formater les opérateurs et opérandes au sein d'une expression.

Pour les opérateurs qui opèrent sur :	Utiliser ce format :	Exemple
Un opérande	operator(operand)	ABS(tag)
Deux opérandes	operand_a operator operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

### Définir l'ordre des opérations

L'instruction effectue les opérations dans les expressions selon un ordre prédéfini. Spécifiez l'ordre d'opération en groupant les termes entre des parenthèses. Cela force l'expression à effectuer une opération entre parenthèses en préalable aux autres opérations.

Les opérations de rang égal sont exécutées de gauche à droite.

Ordre	Fonctionnement
1	( )
2	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3	**
4	- (négation), NOT
5	*, /, MOD
6	- (soustraction), +
7	AND
8	XOR
9	OU

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon L'instruction évalue l'expression et place le résultat dans Dest.
Post-scrutation	N/A

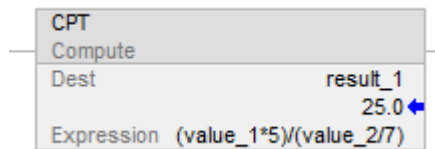
**Exemples**

**Diagramme à relais**

**Exemple 1**

Lorsqu'elle est activée, l'instruction CPT évalue value\_1 multipliée par 5 et divise le résultat par le résultat de value\_2 divisée par 7, puis place le résultat final dans result\_1.

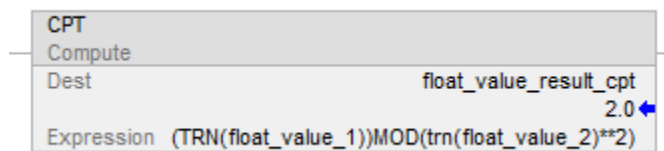
result_1	25.0		Float	REAL
+ value_1	10		Decimal	DINT
+ value_2	14		Decimal	DINT



**Exemple 2**

Lorsqu'elle est activée, l'instruction CPT tronque float\_value\_1 et float\_value\_2 à la puissance deux et divise float\_value\_1 tronquée par ce résultat, le chiffre restant de la division est placé dans float\_value\_result.cpt.

**Diagramme à relais**



float_value_result_cpt	2.0		Float	REAL
float_value_1	10.5		Float	REAL
float_value_2	2.5		Float	REAL

**Voir aussi**

[Instructions Calcul](#) sur la [page 373](#)

[Opérateurs valides](#) sur la [page 371](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

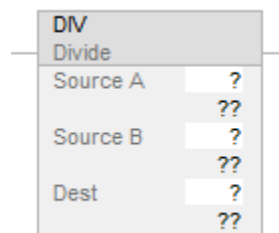
## Division (DIV)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction DIV et l'opérateur « / » divisent Source A par Source B.

### Langages disponibles

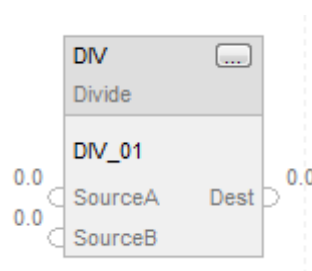
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur « / » dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
SourceA	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur du dividende



SourceB	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur du diviseur
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
DIV	FBD_MATH	étiquette	Structure DIV

**Structure FBD\_MATH**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source A	REAL	Valeur du dividende.
Source B	REAL	Valeur du diviseur.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
SourceA (haut)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du dividende.
SourceB (bas)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du diviseur

Opérandes de sortie (broche droite)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction

Reportez-vous aux *fonctions FBD*.

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
Source_B = 0	4	4

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest = Source A / Source B <sup>1,2</sup>
Post-scrutation	N/A

#### Diagramme de bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Dest = SourceA / SourceB <sup>1,2</sup> En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A

Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

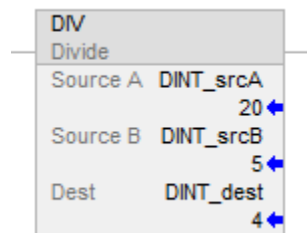
Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = SourceA / SourceB <sup>1,2</sup>
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

<sup>1</sup> Si Source B est 0, le résultat est Source A et un défaut mineur est signalé.

<sup>2</sup> Si Destination et les opérandes source sont des nombres entiers, le résultat est tronqué.

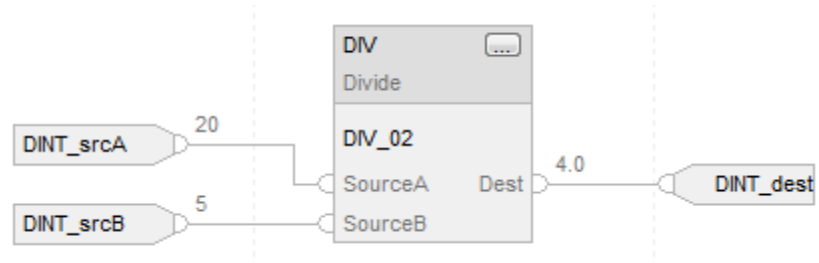
**Exemples**

**Diagramme à relais**



## Diagramme de bloc fonctionnel

### Bloc FBD



### Fonction FBD



### Texte structuré

DINT\_dst := DINT\_srcA / DINT\_srcB;

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

## Modulo (MOD)

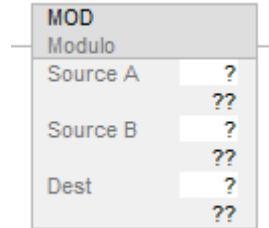
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, L'instruction MOD et l'opérateur divisent Source A par Source B et place le reste dans Dest. Cette opération s'effectue en utilisant l'algorithme suivant :

$$\text{Dest} = \text{Source A} - (\text{truncate}(\text{Source A} / \text{Source B}) * \text{Source B})$$

### Langages disponibles

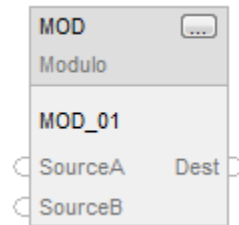
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

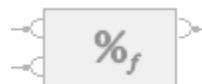
Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

##### Bloc FBD



##### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



##### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser MOD en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Voici les opérandes pour le diagramme à relais.

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur du dividende.

Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Valeur du diviseur.
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
MOD	FBD_MATH	étiquette	Structure MOD

**Structure FBD\_MATH**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur du dividende.
SourceB	REAL	Valeur du diviseur.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
SourceA (haut)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du dividende.
SourceB (bas)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du diviseur

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux *fonctions FBD*.

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
Source B = 0	4	4

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest est défini (jusqu'au reste) comme indiqué à la section Description.
Post-scrutation	N/A

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn

EnableIn est vrai	Dest est défini (jusqu'au reste) comme indiqué à la section Description. Lorsqu'un débordement se produit Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Fonction FBD**

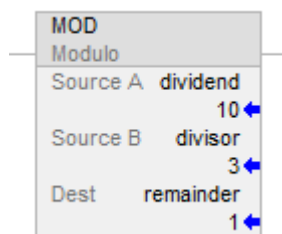
**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest est défini (jusqu'au reste) comme indiqué à la section Description.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Astuce :** Si Source B est 0, le résultat est 0 et un défaut mineur est signalé.

**Exemples**

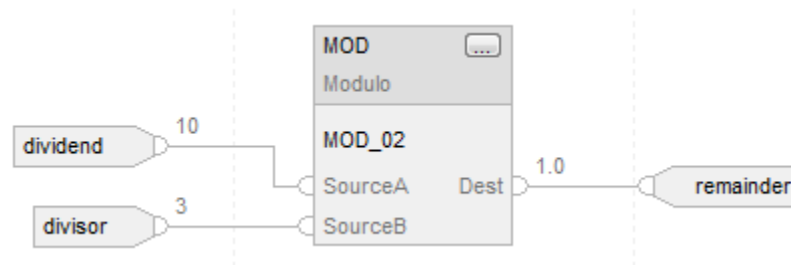
**Diagramme à relais**



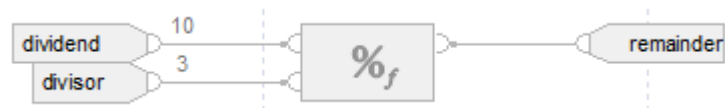
Diviser le Dividend par le Divisor et placer le reste dans Remainder. Dans cet exemple, 3 entre dans 10, à 3 reprises, avec un reste égal à 1.

## Diagramme de bloc fonctionnel

### Bloc FBD



### Fonction FBD



### Texte structuré

remainder := dividend MOD divisor;

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

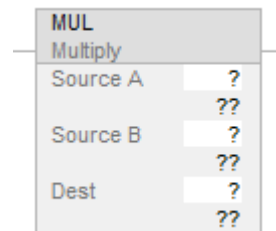
## Multiplication (MUL)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction MUL et l'opérateur « \* » multiplie Source A par Source B.

## Langages disponibles

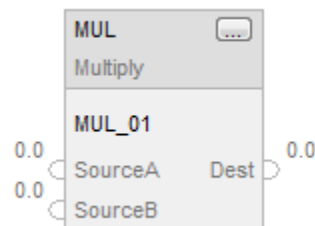
### Diagramme à relais



### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur « \* » dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

## Opérandes

- 
- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.
- 

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

## Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur du multiplicande.
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur du multiplicateur.

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.
------	-----------------------------	---	-----------	--

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
MUL	FBD_MATH	étiquette	Structure MUL

#### Structure FBD\_MATH

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur du multiplicande.
SourceB	REAL	Valeur du multiplicateur.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
SourceA (haut)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du multiplicande.
SourceB (bas)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur du multiplicateur.

Opérande de sortie (broche droite)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux *fonctions FBD*.



### Affecter les indicateurs d'état mathématique

Automates	Affecte l'indicateur d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest = Source A x Source B
Post-scrutation	N/A

#### Diagramme de bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Dest = SourceA x SourceB En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

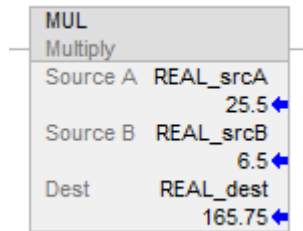
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = Source A x Source B
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

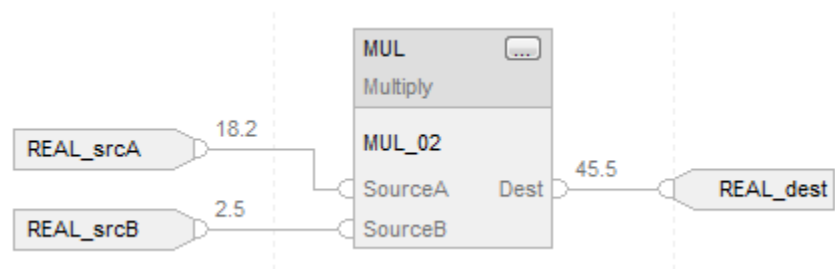
**Exemples**

**Diagramme à relais**



**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Texte structuré**

REAL\_dest := REAL\_srcA \* REAL\_srcB;

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

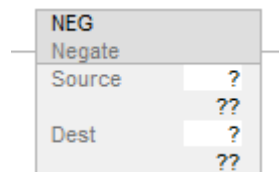
[Valeurs immédiates](#) sur la [page 888](#)

[Fonctions FBD](#) sur la [page 430](#)

**Négation (NEG)**

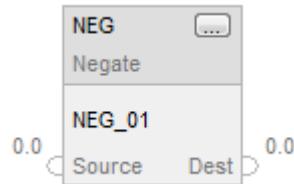
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

En cas d'activation, l'instruction NEG et l'opérateur soustraient de zéro la valeur Source.

**Langages disponibles****Diagramme à relais****Diagramme de bloc fonctionnel**

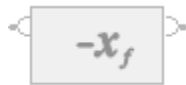
Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur « - » dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à mettre au négatif
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
NEG	FBD_MATH_ADVANCED	étiquette	Structure NEG

**Structure FBD\_MATH\_ADVANCED**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	REAL	Valeur à mettre au négatif.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande d'entrée (broche gauche)	Type de donnée	Description
Source (Source)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à mettre au négatif.

Opérande de sortie (broche droite)	Type de donnée	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux *fonctions FBD*.

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Dest = 0 - Source.
Post-scrutation	N/A

#### Diagramme de bloc fonctionnel

##### Bloc FBD

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn.
EnableIn est vrai	Dest = 0 - Source. En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

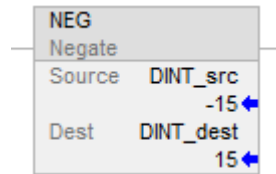
##### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = 0 - Source.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

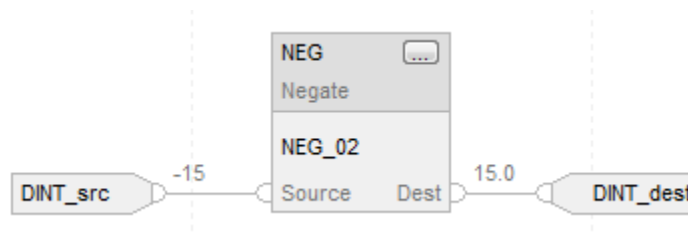
**Exemples**

**Diagramme à relais**



**Diagramme de bloc fonctionnel**

**Bloc FBD**



**Fonction FBD**



**Texte structuré**

DINT\_dest := -DINT\_src;

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)



[Conversions de données](#) sur la [page 889](#)

[Fonctions FBD](#) sur la [page 430](#)

[Valeurs immédiates](#) sur la [page 888](#)

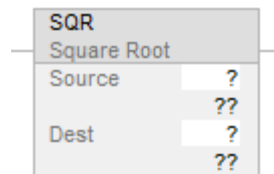
## Racine carrée (SQR/SQRT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction SQR et l'opérateur calculent la racine carrée de Source et placent le résultat dans Dest.

### Langages disponibles

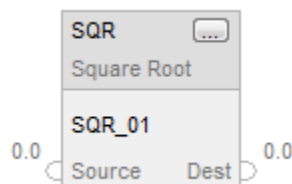
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser SQRT en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immédiate étiquette	Calcule la racine carrée de cette valeur.

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.
------	-----------------------------	---	-----------	--

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
SQR	FBD_MATH_ADVANCED	étiquette	Structure SQR

#### Structure FBD\_MATH\_ADVANCED

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	REAL	Trouver la racine carrée de cette valeur.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérande d'entrée (broche gauche)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
SourceA	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Calcule la racine carrée de cette valeur.

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux fonctions FBD.

**Description (Description)**

Si Dest n'est pas une valeur LREAL/REAL, l'instruction traite de la manière suivante la partie fractionnaire du résultat :

Si Source est :	(pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570) La partie fractionnaire du résultat :	Exemple			(pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580) La partie fractionnaire du résultat :	Exemple		
toute étiquette/valeur	Tronque	Source (Source)	DINT	3	Arrondit	Source (Source)	DINT	3

Si Source est :	(pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570) La partie fractionnaire du résultat :	Exemple			(pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580) La partie fractionnaire du résultat :	Exemple		
entière élémentaire		Dest	DINT	1		Dest	DINT	2
toute étiquette/valeur entière à valeur flottante	Arrondi	Source (Source)	REAL	3,0	Arrondi	Source (Source)	REAL	3,0
		Dest	DINT	2		Dest	DINT	2

Si Source est négative, cette instruction prend la valeur absolue de Source avant de calculer la racine carrée.

Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570, si Source est une donnée de type nombre entier et si Dest est elle aussi une donnée de type nombre entier, l'instruction tronque le résultat. Par exemple, si la valeur Source est le nombre entier 3, le résultat est 1,732 et la valeur Dest devient 1.

Si Source est une donnée de type réel et si Dest est de type à nombre entier, l'instruction arrondit le résultat. Par exemple, si Source a la valeur réelle 3,0, le résultat est 1,732 et la valeur Dest devient 2.

SQR s'utilise en tant qu'opérateur dans les expressions à diagramme à relais ; SQRT s'utilise en tant qu'opérateur dans les instructions à texte structuré.

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Dest = racine carrée de Source.
Post-scrutation	N/A

**Diagramme de bloc fonctionnel****Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn.
EnableIn est vrai	Dest. = racine carrée de Source. En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

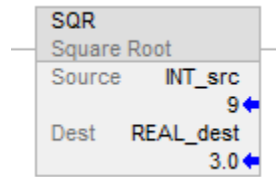
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = racine carrée de Source
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

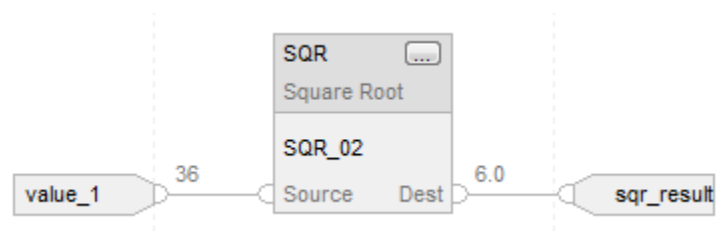
## Exemples

### Diagramme à relais



### Diagramme de bloc fonctionnel

#### Bloc FBD



#### Fonction FBD



#### Texte structuré

```
REAL_dest := SQRT(INT_src);
```

#### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Fonctions FBD](#) sur la [page 430](#)

[Valeurs immédiates](#) sur la [page 888](#)

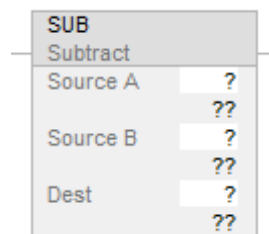
## Soustraction (SUB)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Si activé, l'instruction SUB et l'opérateur « - » soustraient Source A de Source B.

### Langages disponibles

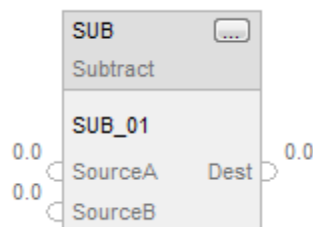
#### Diagramme à relais



#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.



**Astuce :** Utiliser l'opérateur « - » dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

**Opérandes**

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à partir de laquelle est soustraite Source B.
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à soustraire de Source A.

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.
------	-----------------------------	---	-----------	--

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
SUB	FBD_MATH	étiquette	Structure SUB

**Structure FBD\_MATH**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	REAL	Valeur à partir de laquelle est soustraite SourceB.
SourceB	REAL	Valeur à soustraire de SourceA.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	REAL	Résultat de l'instruction.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
SourceA (haut)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à partir de laquelle est soustraite SourceB.
SourceB (bas)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	Valeur à soustraire de SourceA.

Opérande de sortie (broche droite)	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 Type de donnée	Description
Dest	DINT UDINT LINT ULINT REAL LREAL	Résultat de la fonction.

Reportez-vous aux *fonctions FBD*.

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest = Source A - Source B
Post-scrutation	N/A

#### Diagramme de bloc fonctionnel

##### Bloc FBD

Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Dest = SourceA - SourceB En cas de débordement Définir EnableOut sur faux sinon Définir EnableOut sur vrai

Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

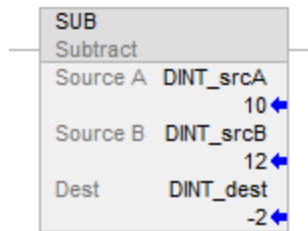
**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Dest = SourceA - SourceB
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

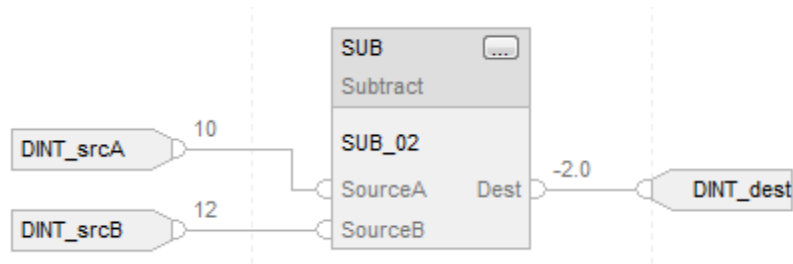
**Exemples**

**Diagramme à relais**

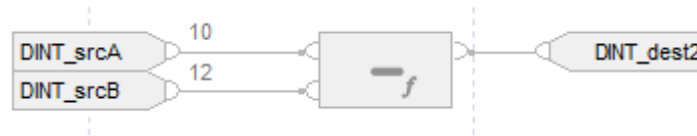


**Diagramme de bloc fonctionnel**

**Bloc FBD**



### Fonction FBD



### Texte structuré

DINT\_dest := DINT\_srcA - DINT\_srcB;

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Fonctions FBD](#) sur la [page 430](#)

[Valeurs immédiates](#) sur la [page 888](#)

## Fonctions FBD

Ces informations s'appliquent aux automates Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580..

Les fonctions FBD sont implémentées conformément à la norme IEC 61131-3 Edition 3. Les fonctions arithmétiques et numériques sont fournies dans le langage de diagramme de bloc fonctionnel. Les langages Diagramme à relais et Texte structuré incluent des éléments arithmétiques et numériques en tant qu'opérateurs et fonctions.

Les fonctions FBD possèdent une ou plusieurs entrées et une entrée. Les fonctions FBD sont implémentées pour l'efficacité, ont une empreinte plus petite et utilisent moins de ressources système pour fonctionner que les blocs fonctionnels FBD.

### Fonctions FBD

- Nécessite toutes les entrées et sorties. Toutes les entrées doivent être d'un type de données prises en charge.
- Ne pas avoir d'étiquettes de sauvegarde ou de types de données prédéfinis. Les valeurs d'entrée connectées ne se convertissent pas en types de données prédéfinis.
- Ne possèdent pas de bits EnableIn et sont toujours exécutés.

**Exemple : fonction Ajouter****Voir aussi**

[Débordement de fonction](#) sur la [page 431](#)

[Conversions de données](#) sur la [page 889](#)

## Débordement de fonction

Ces informations s'appliquent aux automates Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

Le débordement de fonction définit au moins deux fonctions possédant le même nom, mais avec une signature différente, telles que l'argument ou le type de retour. Les fonctions FBD qui prennent en charge le débordement acceptent une plage de types de données d'entrée. Les types de données de sortie dépendent des types de données d'entrée.

Les fonctions FBD suivent les règles ci-dessous :

- Promotion du type de données d'entrée
  - Promotion du type de données d'entrée
    - Classement des types de données de la priorité la plus élevée à la plus faible :  
LREAL, REAL, ULINT, LINT, UDINT, DINT, UINT, INT, USINT, SINT
    - Toutes les entrées sont promues vers le type de données de l'entrée qui possède le rang le plus élevé avant l'exécution
    - Si toutes les entrées ont une valeur de classement DINT ou inférieure, toutes les entrées sont promues vers le type DINT avant l'exécution
  - Le type de sortie dépend du type d'entrée  
Le type de sortie de la fonction est le type d'entrée promu

Par exemple, fonction Ajouter,

- Les entrées SINT + UINT sont promues vers les entrées DINT + DINT.  
Les sorties sont DINT
- Les entrées USINT + LINT sont promues vers les entrées LINT + LINT.  
Les sorties sont LINT

- Les entrées UNIT + LREAL sont promues vers les entrées LREAL + LREAL. Les sorties sont LREAL

**Voir aussi**

[Fonctions FBD](#) sur la [page 430](#)

[Conversions de données](#) sur la [page 889](#)



## Instructions de mouvement/logique

### Instructions de mouvement/logique

Les instructions de mouvement modifient les bits et les déplacent.

#### Instructions disponibles

##### Diagramme à relais

<a href="#">MOV</a>	<a href="#">MVM</a>	<a href="#">AND</a>	<a href="#">OU</a>	<a href="#">XOR</a>	<a href="#">NOT</a>	<a href="#">SWPB</a>	<a href="#">CLR</a>	<a href="#">BTD</a>
---------------------	---------------------	---------------------	--------------------	---------------------	---------------------	----------------------	---------------------	---------------------





##### Diagramme de bloc fonctionnel

##### Bloc FBD

<a href="#">MVMT</a>	<a href="#">AND</a>	<a href="#">OU</a>	<a href="#">XOR</a>	<a href="#">NOT</a>	<a href="#">BTD</a>	<a href="#">BAND</a>	<a href="#">BXOR</a>
----------------------	---------------------	--------------------	---------------------	---------------------	---------------------	----------------------	----------------------

<a href="#">BNOT</a>	<a href="#">BOR</a>
----------------------	---------------------

##### Fonction FBD

			
<a href="#">BNOT</a>	<a href="#">BOR</a>	<a href="#">BAND</a>	<a href="#">BXOR</a>

##### Texte structuré

<a href="#">MVMT</a>	<a href="#">SWPB</a>	<a href="#">BTD</a>
----------------------	----------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
Copier une valeur ou déplacer des chaînes	MOV
Copier une partie spécifique d'un nombre entier	MVM

Copier une partie spécifique d'un nombre entier dans un bloc fonctionnel	MVMT
Déplacer des bits à l'intérieur d'un nombre entier ou entre des nombres entiers	BTD
Déplacer des bits à l'intérieur d'un nombre entier ou entre des nombres entiers dans un bloc fonctionnel	BTDI
Effacer une valeur	CLR
Réorganiser les octets d'une étiquette INT, DINT ou REAL	SWPB

Les instructions logiques exécutent des opérations logiques sur des bits.

Si vous voulez :	Utilisez cette instruction :
Exécuter une opération sur ET au niveau du bit	AND
Exécuter une opération sur OU au niveau du bit.	OR
Exécuter une opération sur OU au niveau du bit exclusif	XOR
Exécuter une opération sur NON au niveau du bit	NOT

Vous pouvez mélanger des types de données, mais cela peut entraîner une perte de précision et une erreur d'arrondi et l'exécution de l'instruction prendra plus de temps. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données **en caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

L'exécution d'une instruction de déplacement/logique a lieu lors de chaque scrutation de cette instruction, à condition que la condition d'entrée d'échelon soit vraie. Si vous souhaitez que l'expression ne soit évaluée qu'à une seule reprise, il convient de faire appel à n'importe quelle instruction du type Impulsion pour déclencher l'instruction de déplacement/logique.

#### Voir aussi

[Instructions de conversion mathématique](#) sur la [page 767](#)

[Instructions d'entrée/sortie](#) sur la [page 151](#)

[Instructions de fin/rupture](#) sur la [page 669](#)

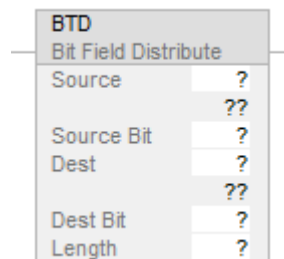
## Répartition champ de bits (BTD)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction BTD copie les bits spécifiés depuis Source, transfère ces bits à la position appropriée et inscrit ces bits dans Destination.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT	immédiate étiquette	Étiquette qui contient les bits à déplacer.
Source bit	DINT	immédiat (0-31)	Numéro du bit (numéro de bit le plus bas) à partir duquel démarre le mouvement Doit se situer dans la plage valide pour les données de type Source
Destination	SINT INT DINT	étiquette	Étiquette de l'emplacement de déplacement des bits

Destination bit	DINT	immédiat (0-31)	Le numéro du bit auquel les données devraient être transférées doit se situer dans la plage valide pour les données de type Destination.
Length	DINT	immédiat (1-32)	Nombre de bits à déplacer

### Description

En cas d'activation, l'instruction BTM copie un groupe de bits de Source à Destination. Ce groupe de bits est identifié par le Source bit (bit qui a le numéro le plus faible de Source) et Length (le nombre de bits à copier). Destination bit identifie le numéro du bit le plus faible avec lequel commencer à Destination. Source reste inchangée.

Si la longueur du champ de bits va au-delà de Destination, l'instruction ne mémorise pas les bits supplémentaires. Les bits supplémentaires éventuels ne passeront pas au mot suivant.

L'étiquette SINT ou INT est convertie en valeur DINT par un remplissage avec des zéros.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse.	N/A
Condition d'entrée d'échelon est vraie.	Cette instruction copie et transfère les bits Source à Destination.
Post-scrutation	N/A

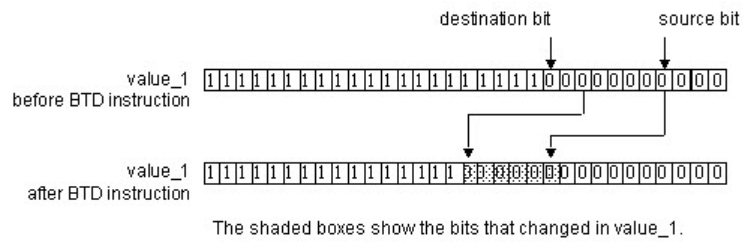
**Exemples**

**Exemple 1**

**Diagramme à relais**

BTD	
Bit Field Distribute	
Source	value_1
	2#1111_1111_1111_1111_1111_1000_0000_0000 ←
Source Bit	3
Dest	value_1
	2#1111_1111_1111_1111_1111_1000_0000_0000 ←
Dest Bit	10
Length	6

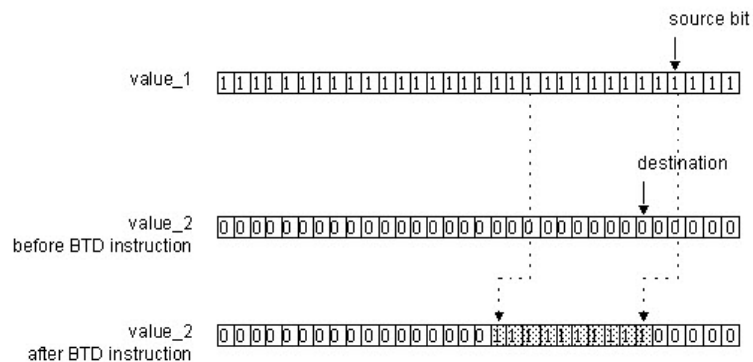
En cas d'activation, l'instruction BTD achemine les bits dans value\_1.



**Exemple 2**

BTD	
Bit Field Distribute	
Source	value_1
	2#1111_1111_1111_1111_1111_1111_1111 ←
Source Bit	3
Dest	value_2
	2#0000_0000_0000_0000_0000_0000_0000 ←
Dest Bit	5
Length	10

En cas d'activation, l'instruction BTDT achemine 10 bits de value\_1 à value\_2.



The shaded boxes show the bits that changed in value\_2.

### Voir aussi

[Instruction de déplacement](#) sur la [page 433](#)

[Effacer \(CLR\)](#) sur la [page 479](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Transfert avec masque \(MVM\)](#) sur la [page 481](#)

## Répartition champ de bits avec cible (BTDT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

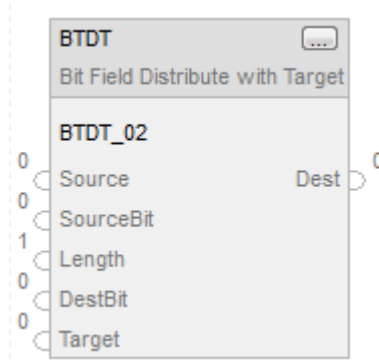
L'instruction BTDT copie tout d'abord Target dans Destination. Ensuite, cette instruction copie les bits spécifiés depuis Source, transfère ces bits à la position appropriée et inscrit ces bits dans Destination. Target et Source restent inchangées.

### Langages disponibles

### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

**Bloc fonctionnel**



**Texte structuré**

BTDT(BTDT\_tag);

**Opérandes**

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
BTDT tag	FBD_BIT_FIELD_DISTRIBUTE	structure	Structure BTDT

**Texte structuré**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
Source (Source)	DINT	Valeur d'entrée contenant les bits à transférer à Destination. Valide = n'importe quel nombre entier
SourceBit	DINT	La position du bit dans Source (le numéro de bit le plus faible à partir duquel doit commencer le transfert). Valide = 0-31
Longueur (Length)	DINT	Nombre de bits à déplacer. Valide = 1-32
DestBit	DINT	La position du bit dans Dest (le numéro de bit le plus faible à partir duquel doit commencer la copie des bits). Valide = 0-31
Target	DINT	Déplacement de la valeur entrée vers Dest avant le déplacement des bits depuis Source. Valide = n'importe quel nombre entier

Paramètre de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	DINT	Résultat de l'opération de déplacement de bits.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description (Description)

Lorsqu'elle est vraie, cette instruction BTDT copie tout d'abord Target sur Destination puis copie un groupe de bits de Source à Destination. Ce groupe de bits est identifié par Source bit (bit qui a le numéro le plus faible de ce groupe) et Length (le nombre de bits à copier). Destination bit identifie le bit qui a le numéro le plus faible avec lequel commencer à Destination. Source et Target restent inchangées.

Si la longueur du champ de bits va au-delà de Destination, l'instruction ne mémorise pas les bits supplémentaires. Les bits supplémentaires éventuels ne passeront pas au mot suivant.

### Affecter les indicateurs d'état mathématique

Automates	Indicateur d'état mathématique affecté
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Oui
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Non

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.



**Exécution**

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

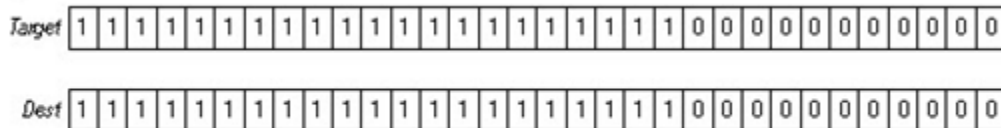
**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemple**

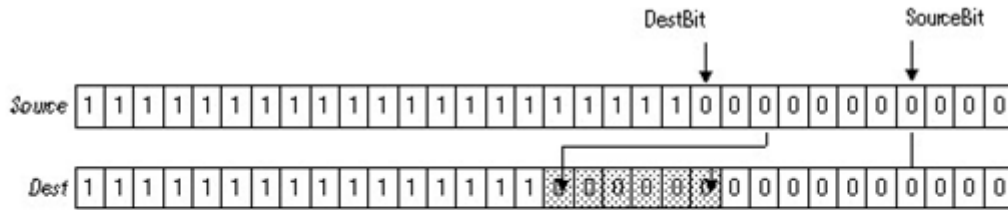
**Étape 1**

L'automate copie Target dans Dest.



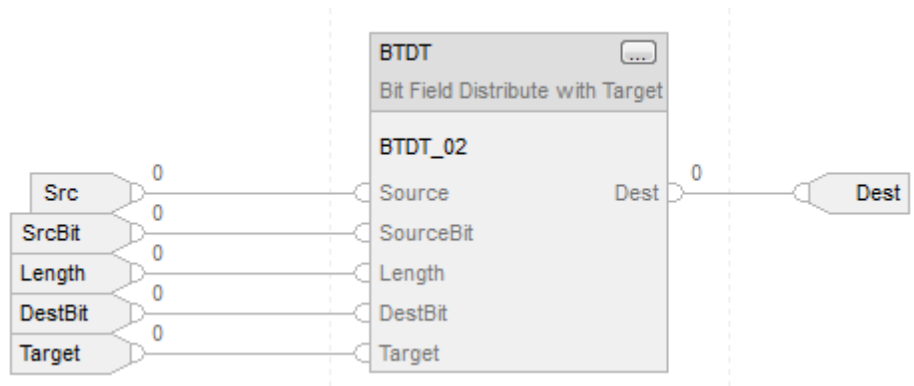
**Étape 2**

SourceBit et Length précisent les bits de Source à copier dans Dest. À commencer par DestBit, Source et Target restent inchangées.



The shaded boxes show the bits that changed.

**Bloc fonctionnel**



**Texte structuré**

```

BDT_01.Source := sourceSTX;
BDT_01.SourceBit := source_bitSTX;
BDT_01.Length := LengthSTX;
BDT_01.DestBit := dest_bitSTX;
BDT_01.Target := TargetSTX;
BDT(BDT_01);
distributed_value := BDT_01.Dest;
    
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

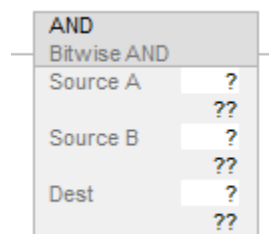
## ET au niveau du bit (AND)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

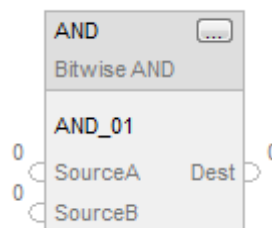
L'instruction AND exécute une opération sur ET au niveau du bit en faisant appel aux bits de Source A et de Source B et place le résultat dans Dest.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser l'opérateur AND (ou « & ») dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*.

### Diagramme à relais

Opérande	Type de données	Format	Description
Source A	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à AND avec Source B. <b>Astuce :</b> si les données ont le type REAL, la valeur d'entrée va être convertie en DINT, ce qui risque de provoquer un débordement.
Source B	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à AND avec Source A. <b>Astuce :</b> si les données ont le type REAL, la valeur d'entrée va être convertie en DINT, ce qui risque de provoquer un débordement.
Dest	SINT INT DINT REAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction. <b>Astuce :</b> si les données ont le type REAL, la valeur DINT qui en résulte va être convertie en REAL.

**Astuce :** L'instruction AND opère sur les DINT. Les opérandes des sources INT ou SINT sont convertis en DINT en remplissant les bits supérieurs de 0.

### Bloc fonctionnel

Opérande	Type de données	Format	Description
AND	FBD_LOGICAL	étiquette	Structure AND

**Structure FBD\_LOGICAL**

Membres d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	DINT	Valeur par rapport à AND avec SourceB.
SourceB	DINT	Valeur par rapport à AND avec SourceA.

Membres de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction a été exécutée sans faille après son activation.
Dest	DINT	Résultat de l'instruction.

**Description**

En cas d'activation, cette instruction évalue l'opération sur ET au niveau du bit :  
 $Dest = A \text{ AND } B$

Si le bit de la Source A est :	Et si le bit de la Source B est :	Le bit de la Dest est :
0	0	0
0	1	0
1	0	0
1	1	1

**Affecte les indicateurs d'état mathématique**

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

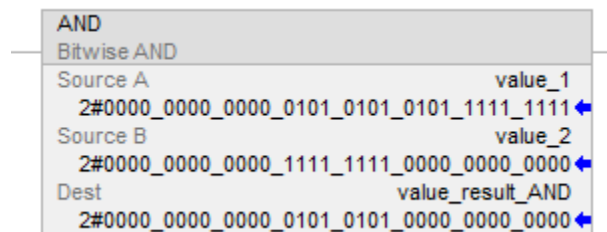
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest est défini comme indiqué à la section Description.
Post-scrutation	N/A

#### Bloc fonctionnel

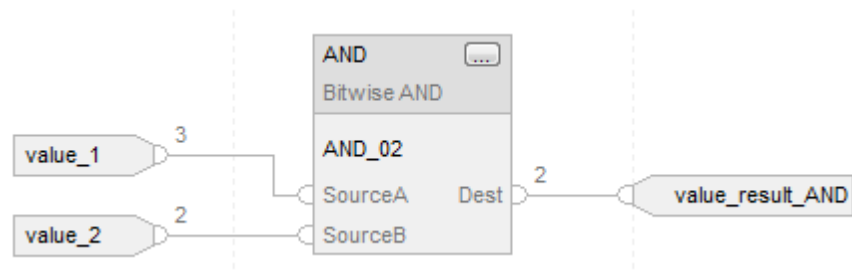
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Définir EnableOut sur EnableIn Dest est défini comme indiqué à la section Description.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

### Exemples

#### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

```
value_result_and := value_1 AND value_2;
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Instruction de déplacement](#) sur la [page 433](#)

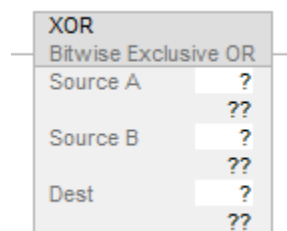
## Ou exclusif au niveau du bit (XOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction XOR exécute une opération sur XOR au niveau du bit en faisant appel aux bits de Source A et de Source B et place le résultat dans Dest.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser XOR en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*.

### Diagramme à relais

Opérande	Type de données	Format	Description
Source A	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à XOR avec Source B.  <b>Astuce :</b> si le type est REAL, la valeur d'entrée va être convertie en DINT (ce qui risque de provoquer un débordement).



Source B	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à XOR avec Source A.  <b>Astuce</b> : si le type est REAL, la valeur d'entrée va être convertie en DINT (ce qui risque de provoquer un débordement).
Dest	SINT INT DINT REAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.  <b>Astuce</b> : si le type est REAL, la valeur DINT qui en résulte va être convertie en REAL.

**Astuce** : L'instruction XOR opère sur les DINT. Les opérandes des sources INT ou SINT sont convertis en DINT en remplissant les bits supérieurs de 0.

### Bloc fonctionnel

Opérande	Type de données	Format	Description
XOR	FBD_LOGICAL	étiquette	Structure XOR

### Structure FBD\_LOGICAL

Membres d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	DINT	Valeur par rapport à XOR avec SourceB.
SourceB	DINT	Valeur par rapport à XOR avec SourceA.

Membres de sortie	Type de données	Description
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	DINT	Résultat de l'instruction.

### Description

En cas d'activation, cette instruction évalue l'opération sur XOR au niveau du bit :

$$\text{Dest} = \text{Source A XOR Source B}$$

Si le bit de la Source A est :	Et si le bit de la Source B est :	Le bit de Dest est :
0	0	0
0	1	1
1	0	1
1	1	0

### Affecte les indicateurs d'état mathématique

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via le tableau* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

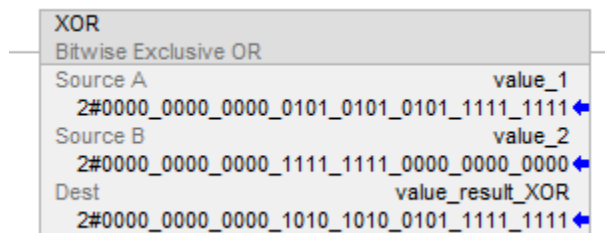
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest est défini comme indiqué à la section Description.
Post-scrutation	N/A

**Bloc fonctionnel**

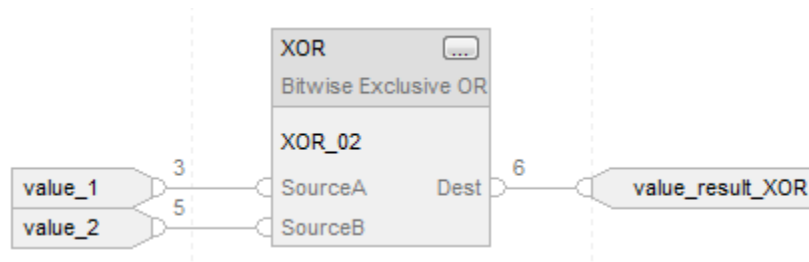
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Définir EnableOut sur EnableIn Dest est défini comme indiqué à la section Description.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Exemples**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

value\_result\_XOR := value\_1 XOR value\_2;

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Instruction de déplacement](#) sur la [page 433](#)

[Conversions de données](#) sur la [page 889](#)

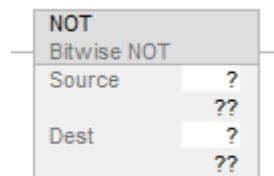
## Non au niveau du bit (NOT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

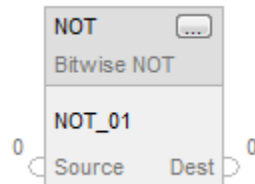
L'instruction NOT exécute une inversion de bits de la Source et place le résultat dans Dest.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser NOT en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

## Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

### Diagramme à relais

Opérande	Type de donnée	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	immediate étiquette	Valeur par rapport à NOT  <b>Astuce :</b> si le type est REAL, la valeur d'entrée sera convertie en DINT (ce qui risque de provoquer un débordement).
Dest	SINT INT DINT REAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction.  <b>Astuce :</b> si le type est REAL, la valeur DINT qui en résulte sera convertie en REAL.

**Astuce :** L'instruction NOT opère sur les DINT. Les opérandes des sources INT ou SINT sont convertis en DINT en remplissant les bits supérieurs de 0.

### Bloc fonctionnel

Opérande	Type de donnée	Format	Description (Description)
NOT	FBD_CONVERT	étiquette	Structure NOT

**Structure FBD\_CONVERT**

Membres d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	DINT	Valeur par rapport à NOT

Membres de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Indique si une instruction a été exécutée sans défauts après son activation.
Dest	DINT	Résultat de l'instruction

**Description (Description)**

En cas d'activation, cette instruction évalue l'opération sur NOT au niveau du bit :

$$\text{Dest} = \text{NOT Source}$$

Si le bit de Source est :	Le bit de Dest est :
0	1
1	0

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

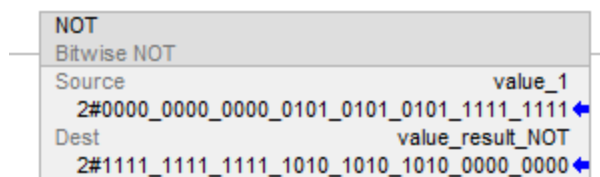
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest est défini comme indiqué à la section Description.
Post-scrutation	N/A

**Bloc fonctionnel**

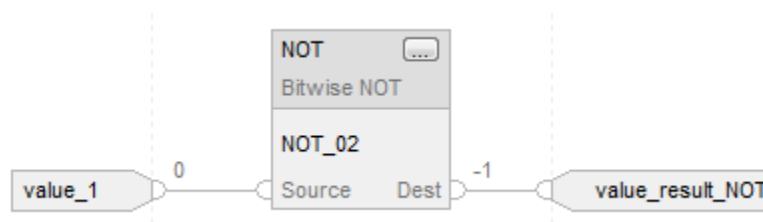
Condition/État	Action entreprise
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Définir EnableOut sur EnableIn Dest est défini comme indiqué à la section Description.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Exemples**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

```
value_result_NOT := NOT value_1;
```

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

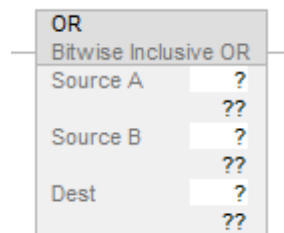
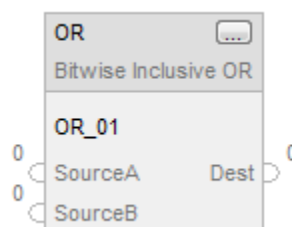
[Conversions de données](#) sur la [page 889](#)

[Instruction de déplacement](#) sur la [page 433](#)

**Ou au niveau du bit  
(OR)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction OR exécute une opération sur OU au niveau du bit en faisant appel aux bits de Source A et de Source B et place le résultat dans Dest.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser OR en tant qu'opérateur dans une expression pour calculer le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*.

### Diagramme à relais

Opérande	Type	Format	Description
Source A	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à OR avec Source B. <b>Astuce :</b> si le type est REAL, la valeur d'entrée va être convertie en DINT (ce qui risque de provoquer un débordement).
Source B	SINT INT DINT REAL	immédiate étiquette	Valeur par rapport à OR avec Source A. <b>Astuce :</b> si le type est REAL, la valeur d'entrée va être convertie en DINT (ce qui risque de provoquer un débordement).
Dest	SINT INT DINT REAL	étiquette	Étiquette pour mémoriser le résultat de cette instruction. <b>Astuce :</b> si le type est REAL, la valeur DINT qui en résulte va être convertie en REAL.

**Astuce :** L’instruction OR opère sur les DINT. Les opérandes des sources INT ou SINT sont convertis en DINT en remplissant les bits supérieurs de 0.

**Bloc fonctionnel**

Opérande	Type	Format	Description
OR	FBD_LOGICAL	étiquette	Structure OR

**Structure FBD\_LOGICAL**

Membres d’entrée	Type de données	Description
EnableIn	BOOL	Activer l’entrée. Si défini sur faux, l’instruction n’est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
SourceA	DINT	Valeur par rapport à OR avec SourceB.
SourceB	DINT	Valeur par rapport à OR avec SourceA.

Membres de sortie	Type de données	Description
EnableOut	BOOL	Indique si l’instruction a été exécutée avec succès après son activation.
Dest	DINT	Résultat de l’instruction.

**Description**

En cas d’activation, cette instruction évalue l’opération sur OU au niveau du bit :

$$\text{Dest} = \text{Source A OR Source B}$$

Si le bit de la Source A est :	Et si le bit de la Source B est :	Le bit de Dest est :
0	0	0
0	1	1
1	0	1
1	1	1

**Affecte les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecte l'indicateur d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

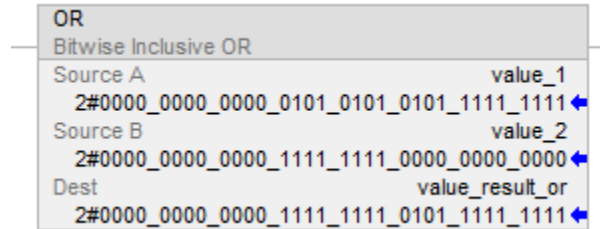
<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon Dest est défini comme indiqué à la section Description.
Post-scrutation	N/A

**Bloc fonctionnel**

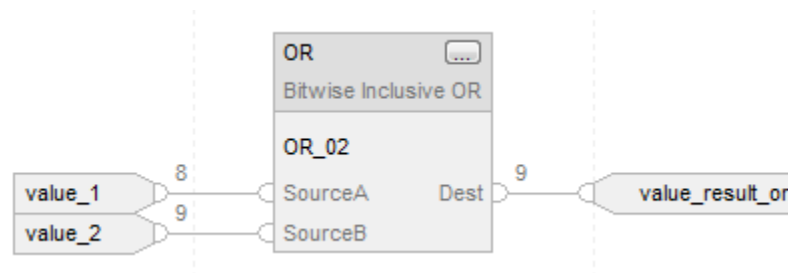
<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
EnableIn est faux	Définir EnableOut sur EnableIn
EnableIn est vrai	Définir EnableOut sur EnableIn Dest est défini comme indiqué à la section Description.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

## Exemples

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

```
value_result_or := value_1 OR value_2;
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Instruction de déplacement](#) sur la [page 433](#)

## ET booléen (BAND)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction BAND exécute, logiquement, des opérations ET pour un maximum de huit entrées booléennes. Pour réaliser un ET au niveau du bit, reportez-vous à *Et au niveau du bit (AND)*.

**Langages disponibles**

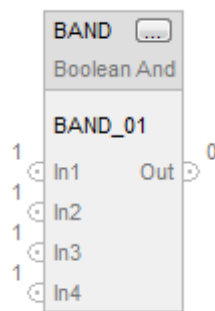
**Diagramme à relais**

Cette instruction n'est pas disponible dans un diagramme à relais.

**Diagramme de bloc fonctionnel**

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

**Bloc FBD**



**Fonction FBD**

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
BAND tag	FBD_BOOLEAN_AND	structure	Structure BAND

**Structure FBD\_BOOLEAN\_AND**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
In1	BOOL	Première entrée booléenne. Définir sur 1 lors du premier téléchargement.
In2	BOOL	Deuxième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In3	BOOL	Troisième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In4	BOOL	Quatrième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In5	BOOL	Cinquième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In6	BOOL	Sixième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In7	BOOL	Septième entrée booléenne. Définir sur 1 lors du premier téléchargement.
In8	BOOL	Huitième entrée booléenne. Définir sur 1 lors du premier téléchargement.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Out	BOOL	La sortie de l'instruction

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
In1	BOOL	Première entrée booléenne
In2	BOOL	Deuxième entrée booléenne

Opérande de sortie (broche droite)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
Out	BOOL	La sortie de l'instruction

Reportez-vous aux fonctions FBD.

### Fonctionnement

#### Bloc FBD

L'instruction BAND exécute des opérations ET pour un maximum de huit entrées booléennes. Lorsqu'une entrée n'est pas utilisée, elle passe par défaut à (1).

$$\text{Out} = \text{In1 AND In2 AND In3 AND In4 AND In5 AND In6 AND In7 AND In8}$$


---

**Important :** Lorsque vous retirez un fil d'entrée de l'instruction BAND lors d'une modification, vous devez vous assurer que l'entrée est définie (1).

---

#### Fonction FBD

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

La fonction FBD exécute des opérations ET pour deux entrées booléennes.

$$\text{Out} = \text{In1 AND In2}$$

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction.

**Exécution****Diagramme de bloc fonctionnel****Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction est exécutée de la manière décrite dans la section Fonctionnement.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Out = In1 AND In2
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A



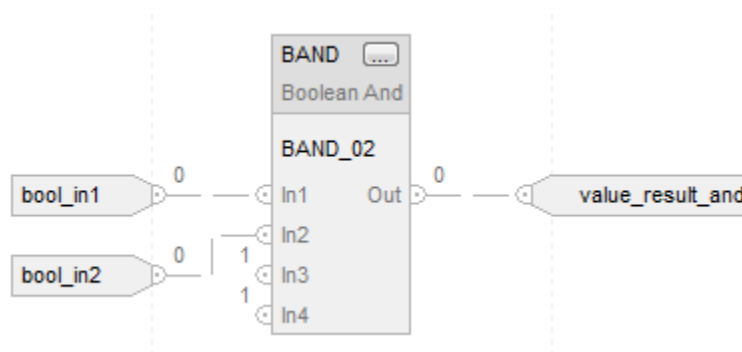
**Exemple**

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Dans cet exemple, bool\_in1 est copié dans BAND\_02.In1, bool\_in2 est copié dans BAND\_02.In2, le résultat de l'exécution des opérations ET de toutes les entrées BAND\_02 est placé dans BAND\_02.Out, puis BAND\_02.Out est copié dans value\_result\_and.

Si bool_In1 est :	Si bool_In2 est :	De ce fait, value_result_and est :
0	0	0
0	1	0
1	0	0
1	1	1



**Fonction FBD**

Cet exemple illustre l'exécution d'un ET sur bool\_in1 et bool\_in2 et place le résultat dans value\_result\_and.



**Voir aussi**

[ET au niveau du bit \(AND\)](#) sur la [page 443](#)

[Fonctions FBD](#) sur la [page 430](#)

## OU exclusif booléen (BXOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction BXOR exécute un OU exclusif sur deux entrées booléennes.

### Langages disponibles

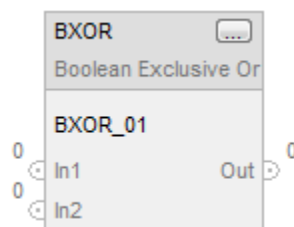
#### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

### Diagramme de bloc fonctionnel

#### Bloc FBD

Opérande	Type de donnée	Format	Description (Description)
BXOR tag	FBD_BOOLEAN_XOR	Structure	Structure BXOR

#### Structure FBD\_BOOLEAN\_XOR

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
In1	BOOL	Première entrée booléenne. La valeur par défaut est mise à zéro.
In2	BOOL	Deuxième entrée booléenne. La valeur par défaut est mise à zéro.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Out	BOOL	La sortie de l'instruction

#### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580

Opérandes d'entrée (broches gauches)	Type de donnée	Description
	<b>Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580</b>	
In1	BOOL	Première entrée booléenne.
In2	BOOL	Deuxième entrée booléenne.

Opérandes de sortie (broche droite)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
Out	BOOL	La sortie de l'instruction

Reportez-vous aux fonctions FBD.

**Fonctionnement**

L'instruction BXOR exécute un OU exclusif sur deux entrées booléennes.

$$\text{Out} = \text{In1 XOR In2}$$

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction.

**Exécution**

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction est exécutée de la manière décrite dans la section Fonctionnement.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Out = In1 XOR In2
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Exemple**

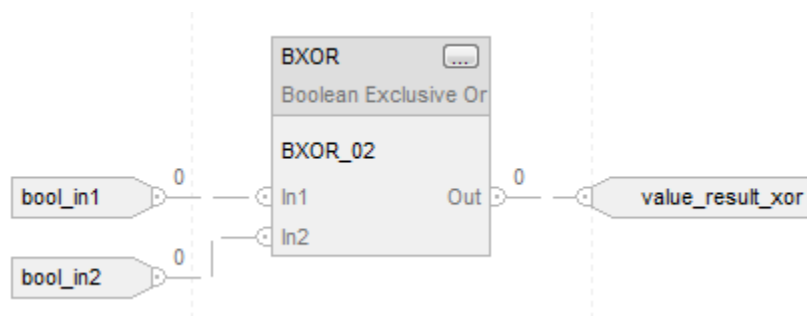
**Diagramme de bloc fonctionnel**

Dans cet exemple, bool\_in1 est copié dans BXOR\_02.In1, bool\_in2 est copié dans BXOR\_02.In2, le résultat de l'exécution d'une opération OU exclusif sur BXOR\_02.In1 et BXOR\_02.In2 est placé dans BXOR\_02.Out, puis BXOR\_02.Out est copié dans value\_result\_xor.

Si bool_In1 est :	Si bool_In2 est :	Alors, value_result_xor est :
0	0	0
0	1	1
1	0	1
1	1	0

**Bloc FBD**

Cet exemple illustre l'exécution d'un OU exclusif sur bool\_in1 et sur bool\_in2 et place le résultat dans value\_result\_xor.



### Fonction FBD



### Voir aussi

[Ou exclusif au niveau du bit \(XOR\)](#) sur la [page 447](#)

[Fonctions FBD](#) sur la [page 430](#)

## NON booléen (BNOT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction BNOT est le complément d'une entrée booléenne. Pour réaliser un NON au niveau du bit, reportez-vous à *Non au niveau du bit (NOT)*.

### Langages disponibles

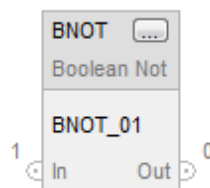
#### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

#### Diagramme de bloc fonctionnel

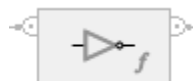
Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

#### Bloc FBD



### Fonction FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes****Diagramme de bloc fonctionnel****Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
BNOT tag	FBD_BOOLEAN_NOT	structure	Structure BNOT

**Structure FBD\_BOOLEAN\_NOT**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
In	BOOL	Entrée dans cette instruction. Définir sur 1 lors du premier téléchargement

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Out	BOOL	La sortie de l'instruction

**Fonction FBD**

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
In	BOOL	Entrée de l'instruction.

Opérande de sortie (broche droite)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
Out	BOOL	La sortie de l'instruction

Reportez-vous aux fonctions FBD.

**Fonctionnement**

L'instruction BNOT est le complément d'une entrée booléenne.

Out = NOT In

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction.

**Exécution**

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.



Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction est exécutée de la manière décrite dans la section Fonctionnement.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

### Fonctions FBD

**Astuce :** La fonction FBD est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	L'instruction est exécutée de la manière décrite dans la section Fonctionnement.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

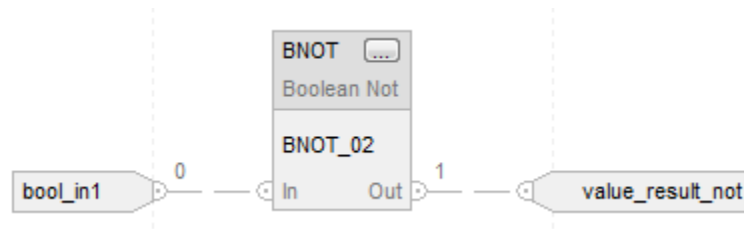
### Exemple

#### Diagramme de bloc fonctionnel

#### Bloc FBD

Dans cet exemple, bool\_in1 est copié dans BNOT\_02.In, le résultat du complément de BNOT\_02.In est placé dans BNOT\_02.Out et BNOT\_02.Out est copié dans value\_result\_not.

Si bool_In1 est :	Par conséquent, value_result_not est :
0	1
1	0



### Fonction FBD

Dans cet exemple, le résultat du complément de `bool_in1` est placé dans `value_result_not`.



### Voir aussi

[Non au niveau du bit \(NOT\)](#) sur la [page 452](#)

[Fonctions FBD](#) sur la [page 430](#)

## OU booléen (BOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction BOR exécute, logiquement, des opérations OU pour un maximum de huit entrées booléennes. Pour réaliser un OU au niveau du bit, reportez-vous à *Ou au niveau du bit (OR)*.

### Langages disponibles

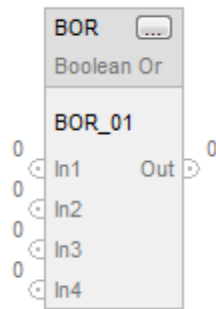
#### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.

#### Diagramme de bloc fonctionnel

Le diagramme de bloc fonctionnel prend en charge les éléments suivants :

**Bloc FBD**



**Fonction FBD**

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.



**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**

**Diagramme de bloc fonctionnel**

**Bloc FBD**

Opérande	Type de donnée	Format	Description (Description)
BOR tag	FBD_BOOLEAN_OR	structure	Structure BOR

**Structure FBD\_BOOLEAN\_OR**

Membres d'entrée	Type de donnée	Description
EnableIn	BOOL	Activer l'entrée. En cas de désactivation, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Définir sur 0 lors du premier téléchargement.
In1	BOOL	Première entrée booléenne. Définir sur 0 lors du premier téléchargement.

In2	BOOL	Deuxième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In3	BOOL	Troisième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In4	BOOL	Quatrième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In5	BOOL	Cinquième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In6	BOOL	Sixième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In7	BOOL	Septième entrée booléenne. Définir sur 0 lors du premier téléchargement.
In8	BOOL	Huitième entrée booléenne. Définir sur 0 lors du premier téléchargement.

Membres de sortie	Type de donnée	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Out	BOOL	La sortie de l'instruction

**Fonction FBD**

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

Opérandes d'entrée (broches gauches)	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Description
In1	BOOL	Première entrée booléenne.
In2	BOOL	Deuxième entrée booléenne.

Opérande de sortie (broche droite)	Type de donnée	Description
	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	
Out	BOOL	La sortie de l'instruction

Reportez-vous aux fonctions FBD.

### Fonctionnement

#### Bloc FBD

L'instruction BOR exécute des opérations OU pour un maximum de huit entrées booléennes. Lorsqu'une entrée n'est pas utilisée, elle mise à zéro par défaut (0).

$$\text{Out} = \text{In1 OR In2 OR In3 OR In4 OR In5 OR In6 OR In7 OR In8}$$


---

**Important :** Lorsque vous retirez un fil d'entrée de l'instruction BOR lors d'une modification, vous devez vous assurer que l'entrée est effacée (0).

---

#### Fonction FBD

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

La fonction FBD exécute des opérations OR pour deux entrées booléennes.

$$\text{Out} = \text{In1 OR In2}$$

#### Affecter les indicateurs d'état mathématique

Non

#### Défauts majeurs/mineurs

Aucun spécifique à cette instruction.

**Exécution****Diagramme de bloc fonctionnel****Bloc FBD**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction est exécutée de la manière décrite dans la section Fonctionnement.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Fonction FBD**

**Astuce :** Les fonctions FBD prennent en charge seulement deux entrées et sont applicables uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.

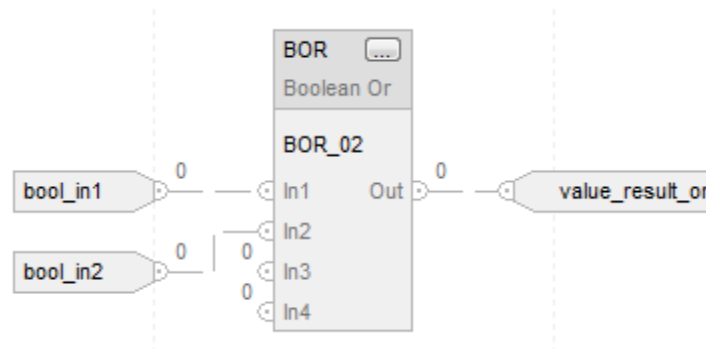
Condition/État	Action entreprise
Pré-scrutation	N/A
Scrutation normale	Out = In1 OR In2
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	N/A

**Exemple****Diagramme de bloc fonctionnel****Bloc FBD**

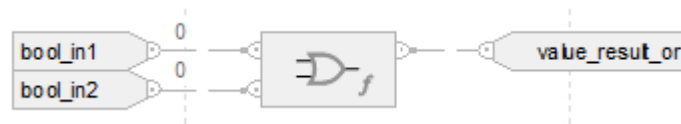
Dans cet exemple, bool\_in1 est copié dans BOR\_02.In1, bool\_in2 est copié dans BOR\_02.In2, le résultat de l'exécution des opérations OU de toutes les entrées

BOR\_02 est placé dans BOR\_02.Out, puis BOR\_02.Out est copié dans value\_result\_or.

Si bool_In1 est :	Si bool_In2 est :	De ce fait, value_result_or est :
0	0	0
0	1	1
1	0	1
1	1	1



**Fonction FBD**



**Voir aussi**

[Ou au niveau du bit \(OR\)](#) sur la [page 456](#)

[Fonctions FBD](#) sur la [page 430](#)

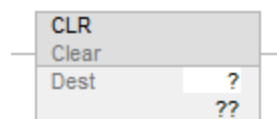
**Effacer (CLR)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction CLR efface tous les bits de Dest.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

L'instruction CLR accepte les données de types élémentaires. Reportez-vous *Types de données élémentaires*.

**Diagramme à relais**

Opérande	Type de donnée	Format	Description (Description)
Dest	SINT INT DINT REAL	étiquette	Étiquette à mettre à zéro.

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.



### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

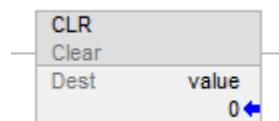
### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Mettre Dest à 0.
Post-scrutation	N/A

### Exemple

#### Diagramme à relais



### Voir aussi

[Instruction de déplacement](#) sur la [page 433](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Types de données élémentaires](#) sur la [page 893](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

## Transfert avec masque (MVM)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction MVM copie Source sur Destination et permet le masquage de certaines portions des données.

L’instruction MVM utilise un Masque pour laisser passer ou bloquer des bits de données de Source. La présence d’un « 1 » dans le masque signifie que le passage du bit de données est autorisé alors que la présence d’un « 0 » signifie que ce bit de données est bloqué.

Si vous mélangez des types de données à nombre entier, l’instruction remplit les bits supérieurs des types de données à petits nombres entiers avec des 0 afin qu’ils aient la même taille que le type de données le plus grand.

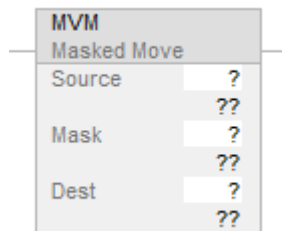
**Saisie d'une valeur de masque immédiate**

Lorsque vous entrez un masque, le logiciel de programmation passe par défaut à des valeurs décimales. Pour entrer un masque avec un autre format, il faut faire précéder la valeur par le préfixe qui convient.

Préfixe	Description
16#	Hexadécimal (par ex., 16#0F0F)
8#	Octal (par ex., 8#16)
2#	Binaire (par ex., 2#00110011)

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion des données*.

## Diagramme à relais

Opérande	Type de données	Format	Description
Source	SINT INT DINT	immédiate étiquette	Valeur à déplacer
Mask	SINT INT DINT	immédiate étiquette	Bits à bloquer ou à laisser passer
Dest	SINT INT DINT	étiquette	Étiquette où enregistrer le résultat

## Affecte les indicateurs d'état mathématique

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Non
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

## Défauts majeurs/mineurs

Automates	Un défaut mineur surviendra si :	Type de défaut	Code de défaut
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Cette fonctionnalité est activée et un débordement est détecté.	4	4

Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	N/A	N/A	N/A
--	-----	-----	-----

Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Cette instruction fait passer Source par le Masque et copie le résultat dans Destination. Les bits sans masque de Destination restent inchangés.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



The shaded boxes show the bits that changed in value\_b

- Ligne 1 : value\_b avant MVM
- Ligne 2 : value\_a
- Ligne 3 : mask\_2

Ligne 4 : value\_b après MVM

MVM	
Masked Move	
Source	value_a
2#0101_0101_0101_0101_0101_0101_0101_0101	←
Mask	mask_2
2#1111_0000_1111_0000_1111_0000_1111_0000	←
Dest	value_b
2#1111_1111_1111_1111_1111_1111_1111_1111	←

Copier des données de value\_a à value\_b, tout en permettant le masquage de données (un 0 masque les données dans value\_a).

**Voir aussi**

[Instruction de déplacement](#) sur la [page 433](#)

[Conversions de données](#) sur la [page 889](#)

[Indexer via des tableaux](#) sur la [page 899](#)

## Transfert masqué avec cible (MVMT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

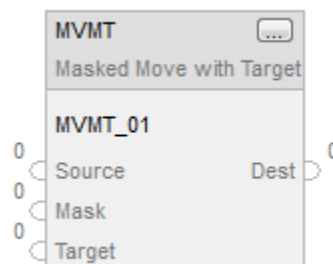
L'instruction MVMT copie Source sur Destination et permet le masquage de certaines portions des données.

**Langages disponibles**

**Diagramme à relais**

Cette instruction n'est pas disponible dans un diagramme à relais.

**Bloc fonctionnel**



**Texte structuré**

MVMT(MVMT\_tag);

**Opérandes**

**Texte structuré**

Variable	Type (Type)	Format	Description (Description)
MVMT tag	FBD_MASKED_MOVE	Structure	Structure MVMT

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
MVMT tag	FBD_MASKED_MOVE	Structure	Structure MVMT

**Structure FBD\_MASKED\_MOVE**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Si mis à zéro, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. Si définir, l'instruction s'exécute. La valeur par défaut est définie.
Source (Source)	DINT	Déplacement de la valeur entrée vers Destination sur la base de la valeur de Mask. Valide = n'importe quel nombre entier
Masque (Mask)	DINT	Masquage de bits pour un déplacement de Source à Dest. Tous les bits sont définis sur un ce qui fait passer les bits correspondants de Source à Dest. Tous les bits sont définis sur zéro, ce qui empêche les bits correspondants de passer de Source à Dest. Valide = n'importe quel nombre entier
Target	DINT	Déplacement de la valeur d'entrée vers Dest avant le déplacement des bits Source en les faisant passer par le Masque. Valide = n'importe quel nombre entier

Paramètre de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	DINT	Résultat de l'opération de déplacement avec masque.

### Description (Description)

En cas d'activation, l'instruction MVMT utilise un Masque pour laisser passer ou bloquer des bits de données de Source. Un « 1 » dans le masque signifie que le bit de données est passé. Un « 0 » dans le masque signifie que le bit de données est bloqué.

Si vous mélangez des types de données à nombre entier, l'instruction remplit les bits supérieurs des types de données à petits nombres entiers avec des 0 afin qu'ils aient la même taille que le type de données le plus grand.

### Saisie d'une valeur immédiate de masque en faisant appel à une référence d'entrée

Lorsque vous entrez un masque, le logiciel de programmation passe par défaut à des valeurs décimales. Si vous souhaitez entrer un masque en faisant appel à un autre format, la valeur doit être précédée du préfixe correct.

Préfixe	Description (Description)
16#	hexadécimal (par ex., 16#0F0F)
8#	octal (par ex., 8#16)
2#	binaire (par ex., 2#00110011)

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Non
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui pour la sortie

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est faux	Les bits EnableIn et EnableOut sont définis sur faux.
Tag.EnableIn est vrai	Les bits EnableIn et EnableOut sont définis sur vrai. L'instruction s'exécute.
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Les bits EnableIn et EnableOut sont définis sur faux.

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Bloc fonctionnel.
Exécution normale	Reportez-vous à Tag.EnableIn est vrai dans le tableau Bloc fonctionnel.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Bloc fonctionnel.

**Exemples**

**Étape 1**

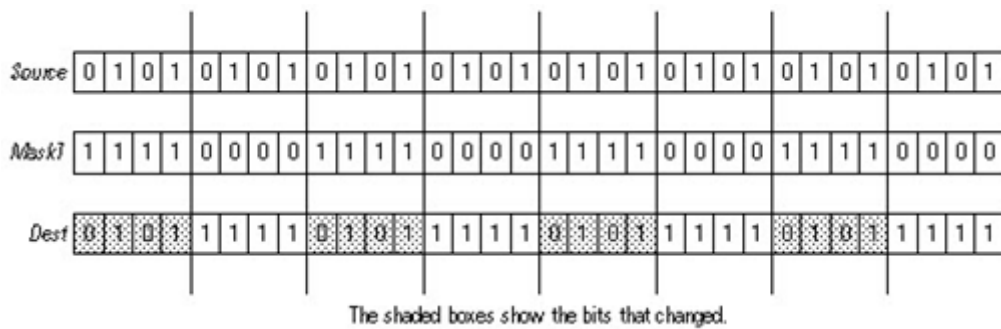
L'automate copie Target dans Dest.



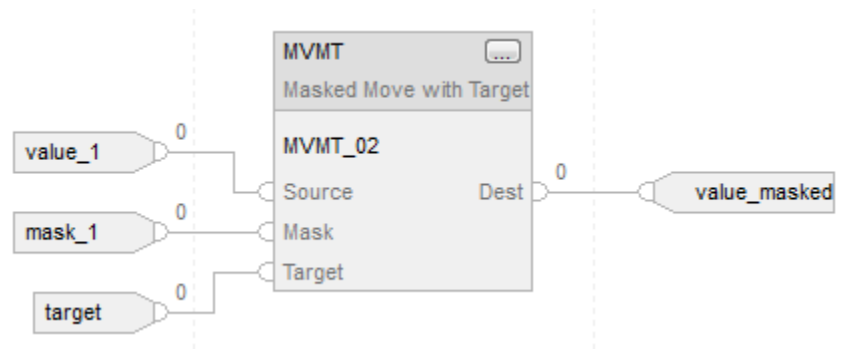


## Étape 2

L'instruction masque Source et la compare à Dest. Les changements éventuels nécessaires sont exécutés dans Dest qui devient alors un paramètre d'entrée de value\_masked. Source et Target restent inchangés. Un 0 dans le masque empêche l'instruction de comparer ce bit.



## Bloc fonctionnel



## Texte structuré

```
MVMT_01.Source := value_1;
MVMT_01.Mask := mask_1;
MVMT_01.Target := target;
```

```
MVMT(MVMT_01);
```

```
value_masked := MVMT_01.Dest;
```

## Voir aussi

[Transfert avec masque \(MVM\)](#) sur la [page 481](#)

[Conversions de données](#) sur la [page 889](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Attributs communs](#) sur la [page 885](#)

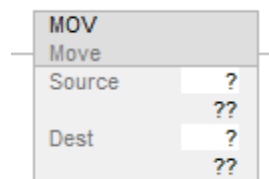
## Transfert (MOV)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction MOV transfère une copie de Source à Dest. Source reste inchangée.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

**Astuce :** Utiliser une affectation « := » avec une expression pour obtenir le même résultat. Pour plus d'informations sur la syntaxe des expressions et des affectations dans le texte structuré, reportez-vous à la section *Syntaxe du texte structuré*.

### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*

**Diagramme à relais**

**Numérique**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	immediate étiquette	Valeur à déplacer
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	étiquette	Étiquette où enregistrer le résultat

**Chaîne (pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement)**

Opérande	Type de donnée	Format	Description (Description)
Source (Source)	Type de chaînes	immediate étiquette	Chaîne à déplacer
Dest	Type de chaînes	étiquette	Étiquette où enregistrer le résultat

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La fonction de détection du débordement est activée et la valeur Source est en dehors de la plage du type Dest.	4	4

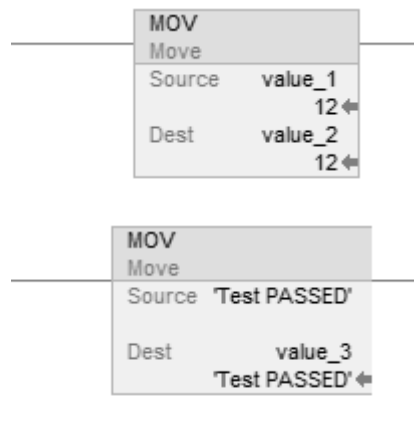
**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. Cette instruction copie Source dans Dest. <b>Opérandes de chaîne :</b> Lorsque Source.LEN > SIZE( Dest.DATA) La chaîne est tronquée à ce qui va s'adapter S:V est défini.
Post-scrutation	N/A

## Exemples

### Diagramme à relais



### Texte structuré

```
value_2 := value_1;
```

```
value_3 := 'Test PASSED';
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Instruction de déplacement](#) sur la [page 433](#)

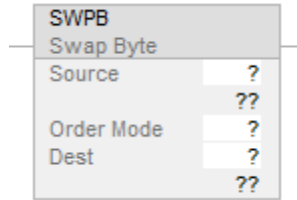
## Permutation d'octet (SWPB)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction SWPB réorganise l'ordre des octets de Source. Elle place le résultat dans Destination.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

SWPB(Source, Order Mode, Dest);

#### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à *Conversion de données*.

#### Un diagramme à relais et texte structuré

Opérande	Type de donnée	Format	Description (Description)
Source (Source)	INT DINT	étiquette	Étiquette qui contient les octets à réorganiser.
Order Mode		élément de liste	Cet opérande précise les principes de la réorganisation. Reportez-vous au tableau Mode d'ordonnement.
Dest	INT DINT	étiquette	Étiquette de mémorisation des octets dans un ordre différent. Reportez-vous au tableau Dest.

En cas de sélection du mode d'ordonnement HIGH/LOW, entrer cette option sous le format HIGHLOW (sans tiret). Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Order Mode

Si Source est :	Et si vous souhaitez changer l'ordre des octets en fonction de ce modèle (chaque lettre représente un octet différent)	Vous devez choisir
INT	AB => BA	N'importe quelle option
DINT	ABCD => DCBA	REVERSE
	ABCD => CDAB	WORD
	ABCD => BADC	HIGH/LOW

### Dest

Si Source est :	Par conséquent, Destination doit être un
INT	INT, DINT Si la destination est un DINT, le résultat s'accompagne d'un signe après la permutation des octets.
DINT	DINT

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Cette instruction réorganise les octets spécifiés.
Post-scrutation	N/A

#### Texte structuré

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.

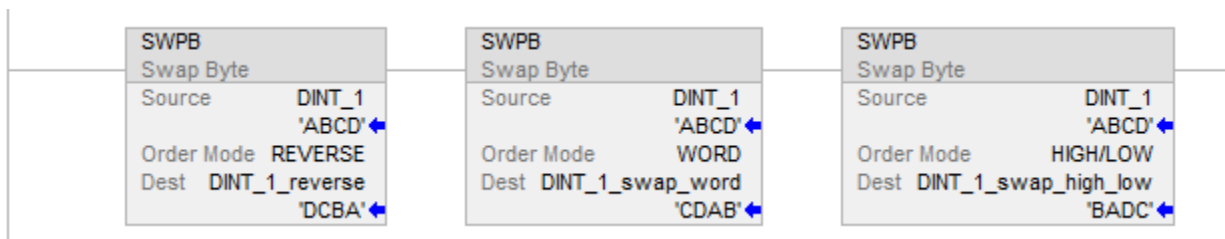
Condition/État	Action entreprise
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

**Exemples**

**Exemple 1 - Permutation des octets d'une étiquette DINT**

Les trois instructions SWPB réorganisent les octets de DINT\_1 en fonction d'un autre mode d'ordonnancement. Le style d'affichage est ASCII et chaque caractère représente un octet. Chaque instruction place les octets dans le nouvel ordre et à Destination différent.

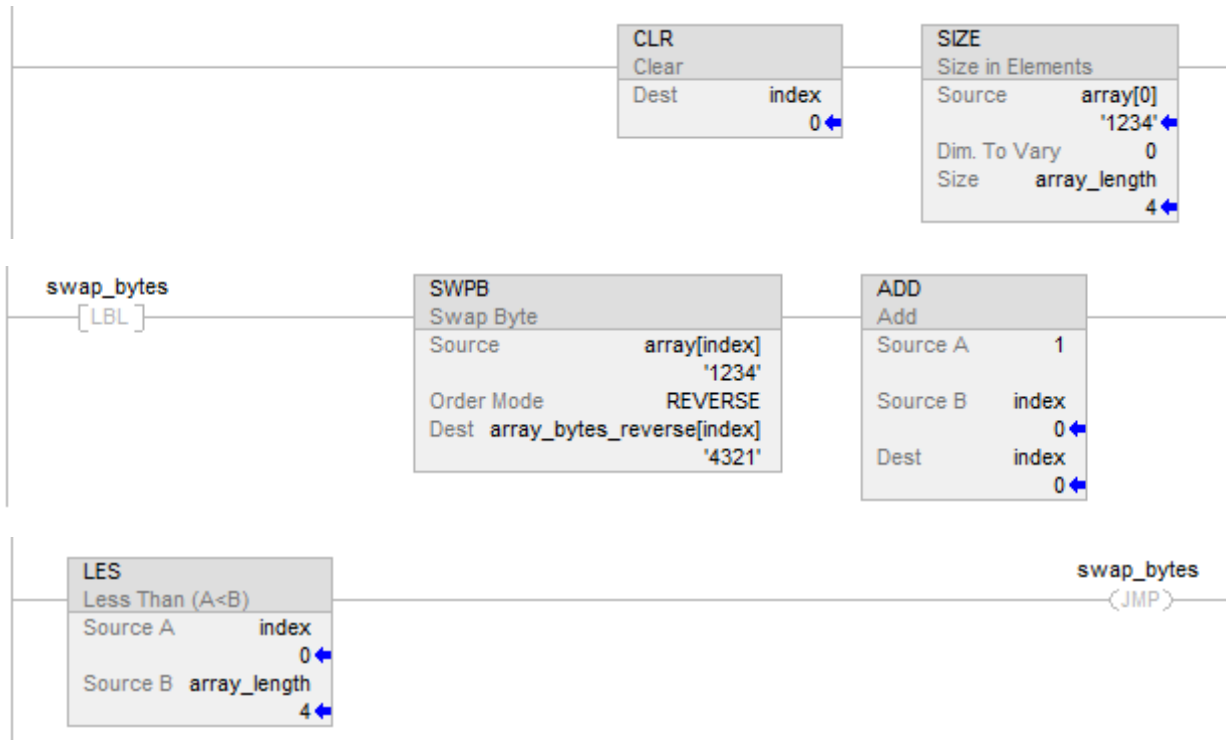
**Diagramme à relais**



**Exemple 2 - Permutation des octets dans tous les éléments d'un tableau**



## Diagramme à relais



## Exemple 3: SWPB sur texte structuré

## Texte structuré

```
index := 0;
```

```
SIZE (array[0],0,array_length);
```

```
REPEAT
```

```
    SWPB(array[index],REVERSE,array_bytes_reverse[index]);
```

```
    index := index + 1;
```

```
UNTIL(index >= array_length)END_REPEAT;
```

## Voir aussi

[Instruction de déplacement](#) sur la [page 433](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)



## Instructions de diverse/tableau (fichier)

### Tableau (Fichier)/Instructions diverses

Les instructions de fichiers/diverses agissent sur des tableaux de données.

Instructions disponibles

Diagramme à relais

<a href="#">FAL</a>	<a href="#">FSC</a>	<a href="#">COP</a>	<a href="#">CPS</a>	<a href="#">FLL</a>	<a href="#">AVE</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

Bloc fonctionnel

Indisponible

Texte structuré

<a href="#">SIZE</a>	<a href="#">FSC</a>	<a href="#">COP</a>	<a href="#">CPS</a>
----------------------	---------------------	---------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
Exécuter des opérations arithmétiques, logiques, de décalage et fonctionnelles sur des valeurs dans des tableaux	FAL
Rechercher la présence de valeurs dans des tableaux et les comparer	FSC
Copier le contenu d'un tableau dans un autre tableau	COP
Copier la ou les valeurs de Source dans Destination	CPS
Remplir un tableau de données spécifiques	FLL
Calculer la moyenne d'un tableau de valeurs	AVE
Trier une dimension de donnée de tableau par ordre ascendant	SRT
Calculer la déviation standard d'un tableau de valeurs	STD
Trouver la taille d'une dimension d'un tableau	SIZE

Vous pouvez mélanger les types de données, mais cela pourrait entraîner une perte en précision et des erreurs d'arrondi et l'instruction prendra plus de temps à s'exécuter. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données **en caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

### Sélection du mode de l'opération

En ce qui concerne les instructions FAL et FSC, ce mode indique à l'automate comment distribuer l'opération du tableau.

Si vous voulez :	Sélectionnez ce mode :
opérer avec tous les éléments spécifiés dans un tableau avant de passer à l'instruction suivante	Mode Tous
distribuer l'opération du tableau sur un certain nombre de scrutations entrer le nombre d'éléments à faire opérer par scrutation (1-2147483647)	Mode Numérique
manipuler un élément du tableau chaque fois que la condition d'entrée d'échelon passe de faux à vrai	Mode Incrémentiel

### Voir aussi

[Mode Tous](#) sur la [page 562](#)

[Mode Numérique](#) sur la [page 563](#)

[Mode Incrémentiel](#) sur la [page 566](#)

## Copie de fichier (COP), Copie synchrone de fichier (CPS)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Les instructions COP et CPS copient les valeurs dans la source vers les valeurs dans la destination. Source reste inchangée.

## Langages disponibles

### Diagramme à relais

COP	
Copy File	
Source	?
Dest	?
Length	?

CPS	
Synchronous Copy File	
Source	?
Dest	?
Length	?

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

COP(Source, Dest, Length);

CPS(Source, Dest, Length);

### Opérandes

---

<b>Important :</b>	Une opération inattendue peut se produire dans les cas suivants : <ul style="list-style-type: none"> <li>• Les opérandes d'étiquette de sortie sont remplacés.</li> <li>• Les membres d'un opérande de structure sont remplacés.</li> <li>• Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.</li> </ul>
--------------------	---

---

### Diagramme à relais

Opérande	Type de données	Format	Description
Source	SINT INT DINT LINT REAL Type de chaînes structure	étiquette	Élément initial à copier

Dest	SINT INT DINT LINT REAL Type de chaînes structure	étiquette	Élément initial à remplacer par la source
Length	SINT INT DINT	immédiate étiquette	Nombre d'éléments destinataires à copier

**Texte structuré**

Opérande	Type de données	Format	Description
Source	SINT INT DINT LINT REAL Type de chaînes structure	étiquette	Élément initial à copier
Dest	SINT INT DINT LINT REAL Type de chaînes structure	étiquette	Élément initial à remplacer par la source
Length	SINT INT DINT	immédiate étiquette	Nombre d'éléments destinataires à copier

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. L'instruction copie les données.
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

Pendant l'exécution des instructions COP et CPS, d'autres actions de l'automate peuvent tenter d'interrompre la copie et de modifier la source :

Si la source ou la destination est :	Et doit :	Sélectionnez :	Remarques
<ul style="list-style-type: none"> <li>• étiquette produite</li> <li>• étiquette consommée</li> <li>• données E/S</li> <li>• données qu'une autre tâche peut remplacer</li> </ul>	Empêcher la modification des données source pendant la copie	CPS	Les tâches qui tentent d'interrompre une instruction CPS sont repoussées jusqu'à ce que l'instruction soit terminée. Pour évaluer le temps d'exécution de l'instruction CPS, reportez-vous au Manuel utilisateur du système ControlLogix, publication 1756-UM 001.
	Autoriser la modification des données source pendant la copie	COP	
Autre	----->	COP	

Les instructions COP et CPS fonctionnent sur une mémoire contiguë et exécutent une copie mémoire directe d'octet à octet.

Lorsque les types de données source et de destination sont différents, le nombre d'octets copiés est égal à la plus petite des valeurs suivantes :

- La quantité demandée est égale à la longueur x (nombre d'octets dans un élément destinataire)
- Le nombre d'octets dans l'étiquette destinataire
- Pour la Automate Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 ou GuardLogix 5580 : le nombre d'octets dans l'étiquette source

**Astuce :** La fin de l'étiquette source ou destinataire est définie comme le dernier octet de l'étiquette de base. Si l'étiquette est une structure, la fin de l'étiquette correspond au dernier octet du dernier élément de la structure. Cela signifie que les instructions COP et CPS peuvent écrire après la fin d'un tableau de membre, mais pas après la fin de l'étiquette de base.

---

**Important :** Tester et confirmer que l'instruction ne modifie pas des données qu'elle ne devrait pas modifier.

---

## Exemples

### Exemple 1

Copier un tableau.

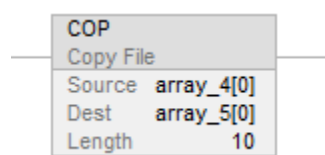
Si activée, l'instruction COP copie 40 octets depuis le tableau array\_4 vers le tableau array\_5.

array\_4 est un DINT (4 octets par élément) et contient 10 éléments (taille totale = 40 octets)

array\_5 est un DINT (4 octets par élément) et contient 10 éléments (taille totale = 40 octets).

Length indique que 10 éléments destinataires doivent être copiés, donc 40 octets sont copiés.

### Diagramme à relais





**Texte structuré**

```
COP(array_4[0],array_5[0],10);
```

**Exemple 2**

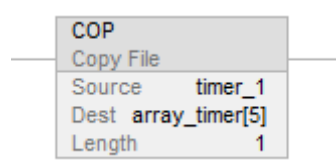
Copier une structure.

Si activée, l'instruction COP copie la structure timer\_1 dans l'élément 5 de array\_timer.

timer\_1 est un TIMER (taille totale = 12 octets).

array\_timer est un TIMER (12 octets par élément) et contient 10 éléments (taille totale = 120 octets).

Length indique que 1 élément destinataire doit être copié, donc 12 octets sont copiés.

**Diagramme à relais****Texte structuré**

```
COP(timer_1,array_timer[5],1);
```

**Exemple 3**

Copier les données du tableau tout en empêchant la modification des données tant que la copie n'est pas terminée.

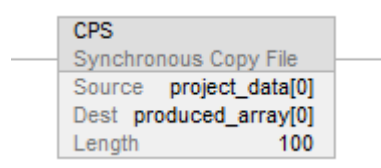
Le tableau project\_data (100 éléments) stocke diverses valeurs qui changent à différents moments dans l'application. Pour envoyer une image complète du tableau project\_data dans une instance à temps vers un autre automate, l'instruction CPS copie project\_data dans produced\_array. Pendant que l'instruction CPS copie les données, aucune mise à jour E/S ni autre tâche ne peut modifier les données. L'étiquette produced\_array génère les données sur un réseau ControlNet pour utilisation par d'autres automates.

project\_data est un DINT (4 octets par élément) et contient 100 éléments (taille totale = 400 octets)

produced\_array est un DINT (4 octets par élément) et contient 100 éléments (taille totale = 400 octets).

Length indique que 100 éléments destinataires doivent être copiés, donc 400 octets sont copiés.

#### Diagramme à relais



#### Texte structuré

```
CPS(project_data[0],produced_array[0],100);
```

#### Exemple 4

Copier les données vers une étiquette produite tout en empêchant l'envoi des données tant que la copie n'est pas terminée.

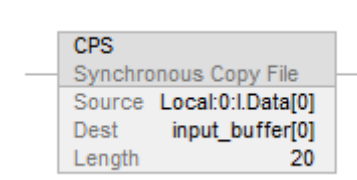
Local:0:I.Data stocke les données d'entrée pour le réseau DeviceNet connecté au module 1756-DNB dans le logement 0. Pour synchroniser les entrées avec l'application, l'instruction CPS copie les données d'entrées dans input\_buffer. Pendant que l'instruction CPS copie les données, aucune mise à jour E/S ne peut modifier les données. Pendant que l'application s'exécute, elle utilise comme entrées les données d'entrées contenues dans input\_buffer.

Local:0:I.Data est un DINT (4 octets par élément) et contient 2 éléments (taille totale = 8 octets)

input\_buffer est un DINT (4 octets par élément) et contient 20 éléments (taille totale = 80 octets).

Length indique que 20 éléments destinataires doivent être copiés (4 X 20 = 80 octets). Toutefois, la source ne peut fournir que 8 octets, donc 8 octets sont copiés.

#### Diagramme à relais

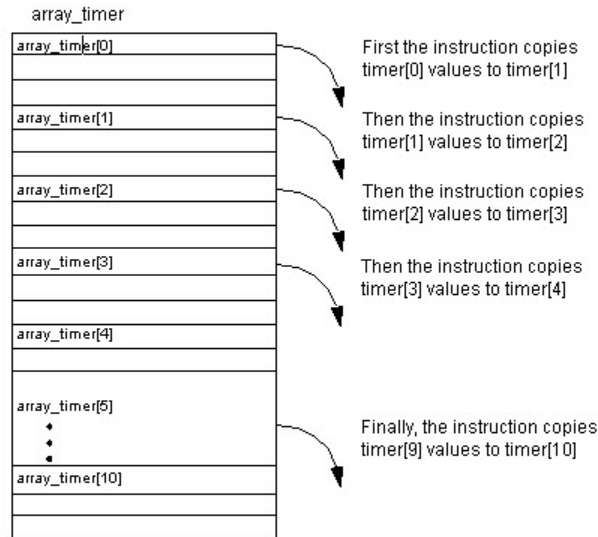


#### Texte structuré

```
CPS(Local:0:I.Data[0], input_buffer[0], 20);
```

### Exemple 5

Initialiser une structure de tableau, initialiser le premier élément, puis utiliser COP pour le répliquer dans le reste du tableau.

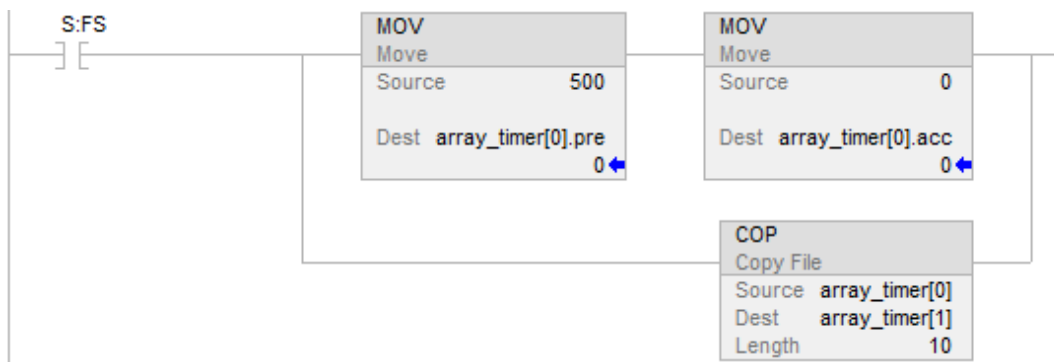


Cet exemple initialise des structures de temporisateur ou de tableau. Si activées, les instructions MOV initialisent les valeurs .PRE et .ACC du premier élément array\_timer. Si activée, l’instruction COP copie un bloc contigu d’octets, en commençant à array\_timer[0]. La longueur est de neuf structures de temporisateur.

array\_timer est un TIMER (12 octets par élément) et contient 15 éléments (taille totale = 180 octets)

Length indique que 10 élément destinataire doit être copié, donc 120 octets sont copiés.

### Diagramme à relais



**Texte structuré**

```
IF S:FS THEN  
  
array_timer[0].pre := 500;  
  
array_timer[0].acc := 0;  
  
COP(array_timer[0],array_timer[1],10);  
  
END_IF;
```

**Exemple 6**

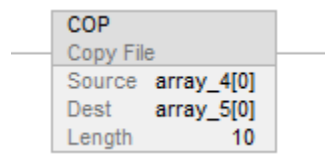
Copier des tableaux de différentes tailles.

Si activée, l'instruction COP copie les octets depuis le tableau SINT array\_6 vers le tableau DINT array\_7.

array\_6 est un SINT (1 octet par élément) et contient 5 éléments (taille totale = 5 octets)

array\_7 est un DINT (4 octets par élément) et contient 10 éléments (taille totale = 40 octets).

Length indique que 20 éléments destinataires doivent être copiés (4 X 20 = 80 octets). Toutefois, la destination ne peut accepter que 40 octets, et la source ne peut fournir que 5 octets, donc 5 octets sont copiés.

**Diagramme à relais****Texte structuré**

```
COP(array_4[0],array_5[0],10);
```

**Voir aussi**

[Indexer via des tableaux](#) sur la [page 899](#)

[Instructions File/Misc](#) sur la [page 499](#)

[Instructions Mouvement/Logical](#) sur la [page 433](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

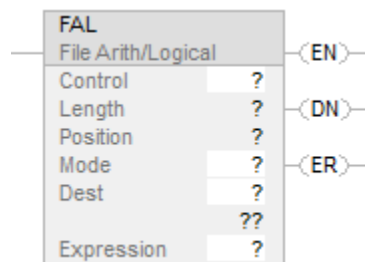
## Arithmétique et logique des fichiers (FAL)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

Cette instruction FAL effectue des opérations de copie, arithmétiques, logiques et fonctionnelles sur les données mémorisées dans un tableau. Lorsque la condition d'entrée d'échelon de l'instruction FAL passe de faux à vrai, l'expression donnée va être exécutée durant le mode spécifié d'itération.

### Langages disponibles

#### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section Conversion des données.

**Diagramme à relais**

Opérande	Type de donnée	Format	Description (Description)
Control	CONTROL	Étiquette (Tag)	Structure de contrôle de l'opération
Longueur (Length)	DINT	Immédiate	Nombre d'éléments dans le tableau à manipuler
Position (Position)	DINT	Immédiate	Décalage dans le tableau Valeur initiale typique : 0
Mode	DINT	Immédiate	Montre comment distribuer cette opération Sélectionner INC ou ALL, ou saisir un nombre compris entre 1 et 2 147 483 647
Expression	SINT INT DINT REAL	Immédiate Étiquette (Tag)	Une expression comportant des étiquettes et/ou des valeurs immédiates séparées par des opérateurs.
Destination	SINT INT DINT REAL	Étiquette (Tag)	La valeur de cette expression va être mémorisée dans destination.

**Structure CONTROL**

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction FAL a été activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction s'est exécutée sur le dernier élément (.POS = .LEN).
.ER	BOOL	Lorsqu'un débordement se produit, les deux plates-formes définissent .ER et arrêtent l'exécution de l'instruction. Les automates suivants vont générer un débordement : <ul style="list-style-type: none"> <li>• CompactLogix 5370</li> <li>• ControlLogix 5570</li> </ul>
.LEN	DINT	La longueur spécifie le nombre d'éléments du tableau sur lesquels opère l'instruction FAL.
.POS	DINT	La position contient la position de l'élément actuel auquel l'instruction est en train d'accéder.

Reportez-vous à Syntaxe du texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré[2].

La valeur de cette expression est mémorisée dans l'étiquette de destination spécifiée. Lorsqu'un débordement se produit, cela définit le bit ER et arrête

l'exécution. Dès que FAL a terminé toutes les itérations configurées, le bit .DN est défini.

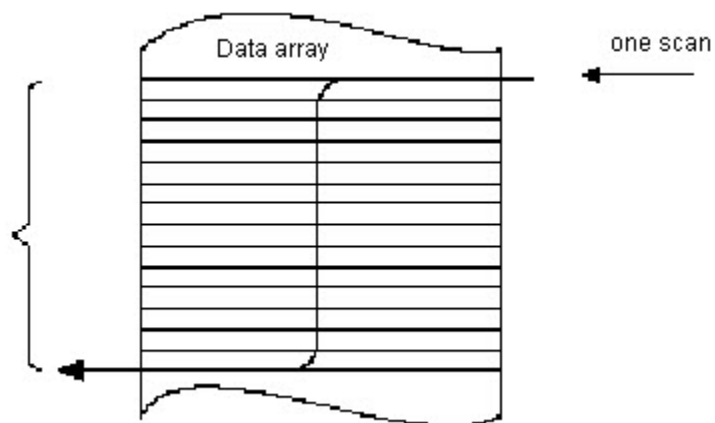
### Sélectionner le mode de fonctionnement

En ce qui concerne les instructions FAL, le mode indique à l'automate comment distribuer l'opération sur le tableau.

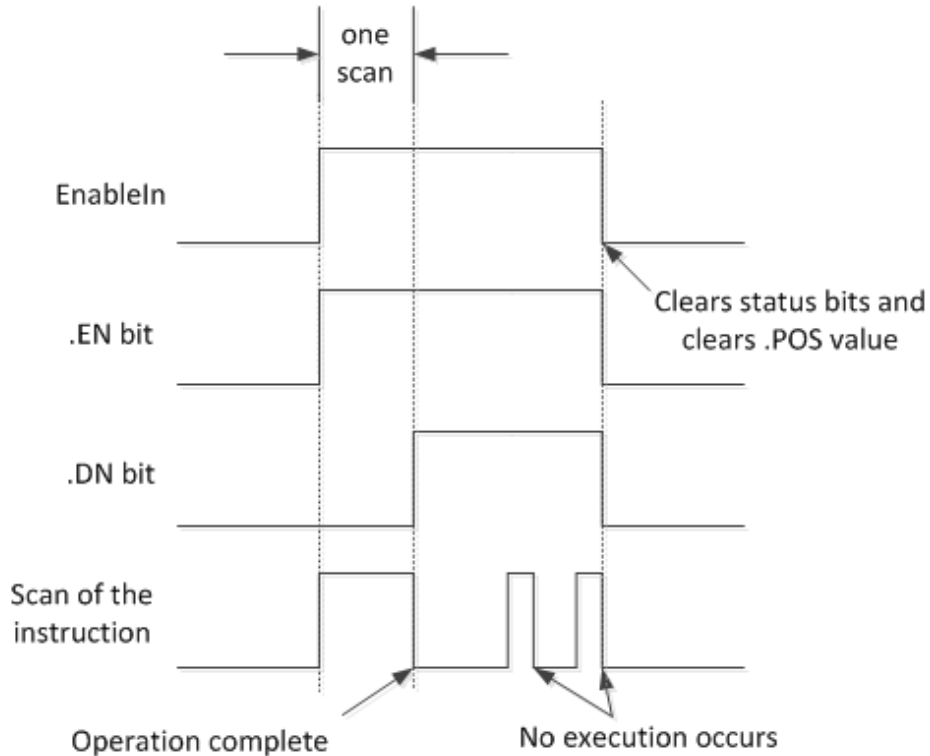
Si :	Sélectionnez ce mode :
Opérant sur tous les éléments spécifiés dans un tableau avant de passer à l'instruction suivante.	Tout (All)
Répartition de l'opération de tableau sur un certain nombre de scrutations. Saisir le nombre d'éléments par scrutation sur lesquels procéder à l'exécution (1-2 147 483 647)	Numérique
Manipulation d'un élément du tableau chaque fois que l'option EnableIn passe de faux à vrai.	Incrémentiel

### Mode Tous

Dans le mode Tous, cette instruction agit sur tous les éléments spécifiés du tableau avant de passer à l'instruction suivante. Cette opération commence lorsque l'option EnableIn d'activation de l'instruction passe de faux à vrai. La valeur de la position (.POS) de la structure de contrôle indique l'élément du tableau qu'utilise en ce moment cette instruction. Cette opération s'arrête lorsque la valeur .POS est égale à la valeur .LEN ou la dépasse et lorsqu'un débordement se produit dans l'expression et lorsque le bit .ER est défini sur Vrai.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est vrai. Le bit .DN, le bit .EN et la valeur .POS sont mis à zéro lorsque EnableIn est faux. Ce n'est qu'à ce moment-là qu'une autre exécution de cette instruction peut être déclenchée par une transition d'EnableIn de faux à vrai.

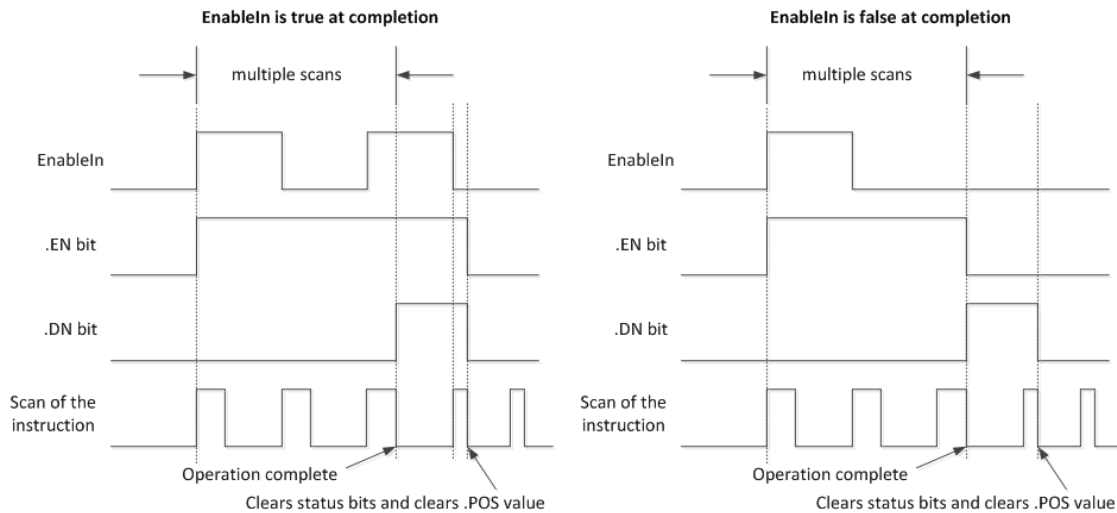


**Mode Numérique**

Le mode Numérique répartit l'opération de tableau sur un certain nombre de scrutations. Ce mode s'utilise lorsqu'on travaille avec des données qui ne sont pas critiques sur le plan temporel ou avec des volumes importants de données. Entrer le nombre d'éléments sur lesquels va agir chaque scrutation, ce qui permet de réduire la durée des scrutations.

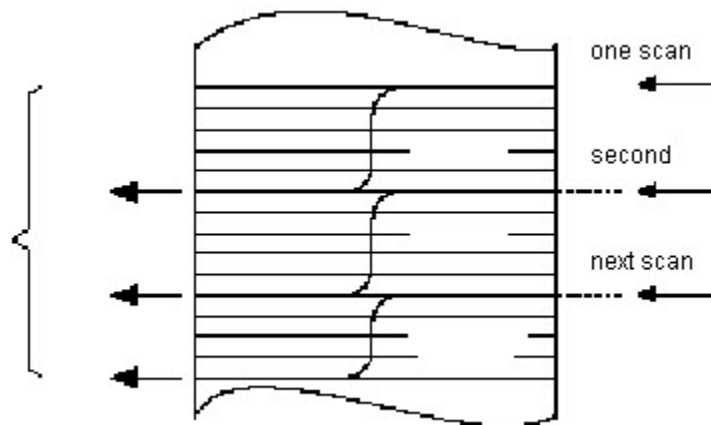


Cette exécution se déclenche lorsque l'option EnableIn d'activation passe de faux à vrai. Dès que cette instruction se déclenche, elle est exécutée chaque fois qu'elle fait l'objet d'une scrutation et pendant le nombre de scrutations qui s'avèrent nécessaires pour exploiter la totalité du tableau. Dès que l'option EnableIn se déclenche, elle peut changer de manière répétée, sans interrompre l'exécution de cette instruction.



Éviter l'utilisation des résultats d'une instruction de fichier opérant dans le mode Numérique tant que le bit .DN n'est pas défini.

Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est défini.

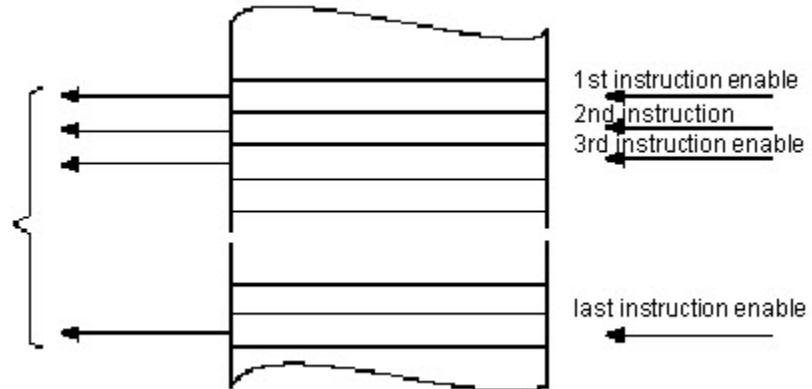


Si l'option EnableIn est vraie lors de l'exécution, les bits .EN et .DN sont vrais jusqu'à ce que l'option EnableIn devienne fausse. Lorsque l'option EnableIn devient fausse, ces bits sont mis à zéro, de même que la valeur de .POS.

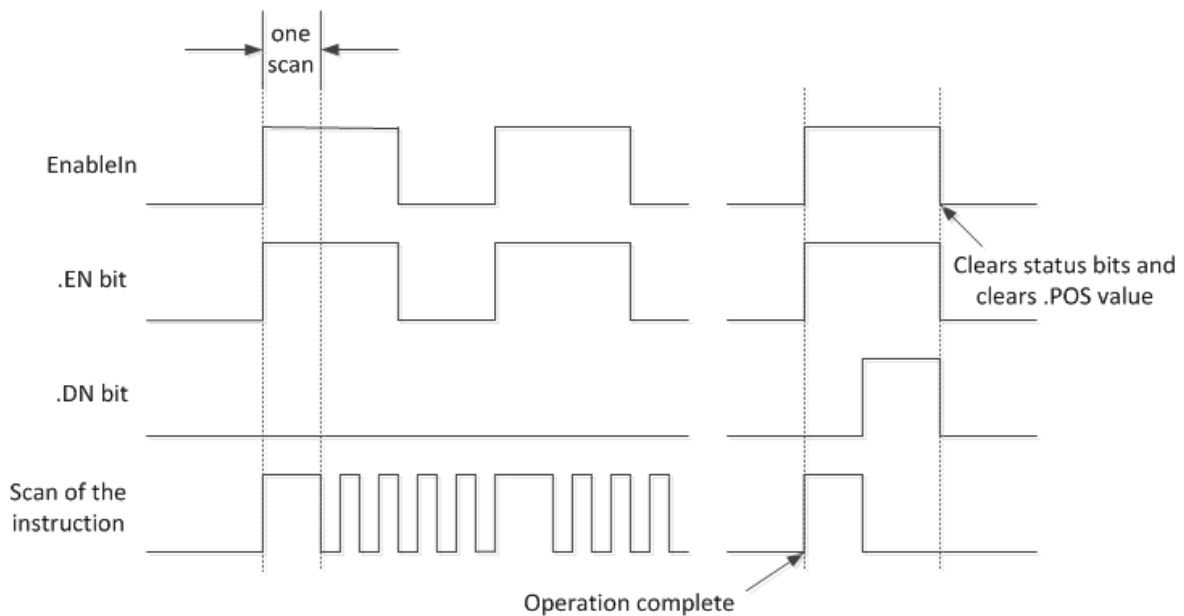
Si, à la fin de cette exécution, l'option EnableIn d'activation de l'entrée est fausse, le bit .EN est immédiatement mis à zéro. Une scrutation après l'effacement du bit .EN, le bit .DN et la valeur de .POS sont eux aussi mis à zéro.

**Mode Incrémentiel**

Le mode Incrémentiel manipule un élément du tableau chaque fois que l'option EnableIn de l'instruction passe de faux à vrai.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. L'exécution n'a lieu que lors d'une scrutation durant laquelle l'option EnableIn passe de faux à vrai. Chaque fois que cela se produit, un seul élément du tableau est manipulé. Si l'option EnableIn reste vraie pendant plus d'une scrutation, l'exécution de cette instruction n'a lieu que lors de la première scrutation



Le bit .EN est défini lorsque EnableIn est vrai. Le bit .DN est défini lorsque le dernier élément du tableau a été manipulé. Après la manipulation du dernier élément et après que EnableIn soit devenue fausse, le bit .EN, le bit .DN et la valeur de .POS sont mis à zéro.

La différence entre le mode Incrémentiel et le mode Numérique, au rythme d'un élément par scrutation, est la suivante :

Dans le mode Numérique, quel que soit le nombre d'éléments par scrutation, il faut une seule transition de faux à vrai de l'option EnableIn pour lancer l'exécution. L'exécution de cette instruction se poursuit afin de réaliser le nombre spécifié d'éléments lors de chaque scrutation jusqu'à leur achèvement final, quel que soit l'état de l'option EnableIn.

Lors du mode Incrémentiel, l'option EnableIn doit au préalable passer de faux à vrai afin de manipuler un élément du tableau.

### Expressions de formats

Pour chaque opérateur que vous utilisez dans le cadre d'une expression, vous devez fournir un ou deux opérandes (étiquettes ou valeurs immédiates). Utiliser le tableau suivant pour formater les opérateurs et opérandes au sein d'une expression.

Pour les opérateurs qui opèrent sur :	Utiliser ce format :	Exemple
Un opérande	operator(operand)	ABS(tag)
Deux opérandes	operand_a operator operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

### Définir l'ordre des opérations

Les opérations que vous rédigez dans l'expression sont exécutées par l'instruction dans un ordre bien précis, qui n'est pas forcément l'ordre dans lequel vous les avez rédigées. Vous pouvez outrepasser outre l'ordre de réalisation de cette opération en groupant plusieurs termes en les mettant entre parenthèses, ce qui oblige cette instruction à exécuter une opération entre parenthèses avant les autres opérations.

Les opérations de rang égal sont exécutées de gauche à droite.

Ordre	Fonctionnement
1	( )
2	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3	**
4	- (négation), NOT

	*, /, MOD
6	- (soustraction), +
7	AND
8	XOR
9	OU

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Non
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
.POS < 0 ou .LEN < 0	4	21

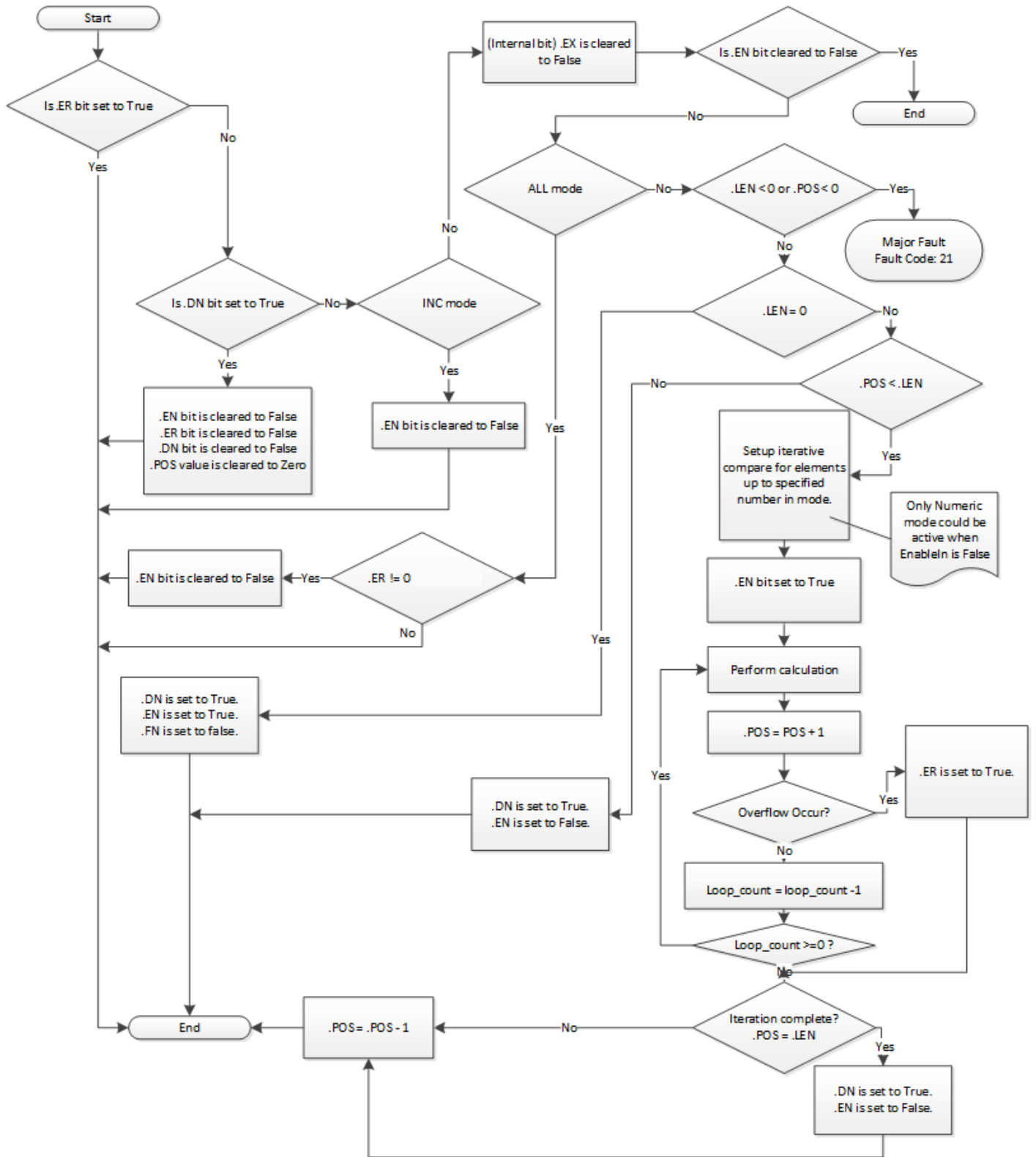
Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution**

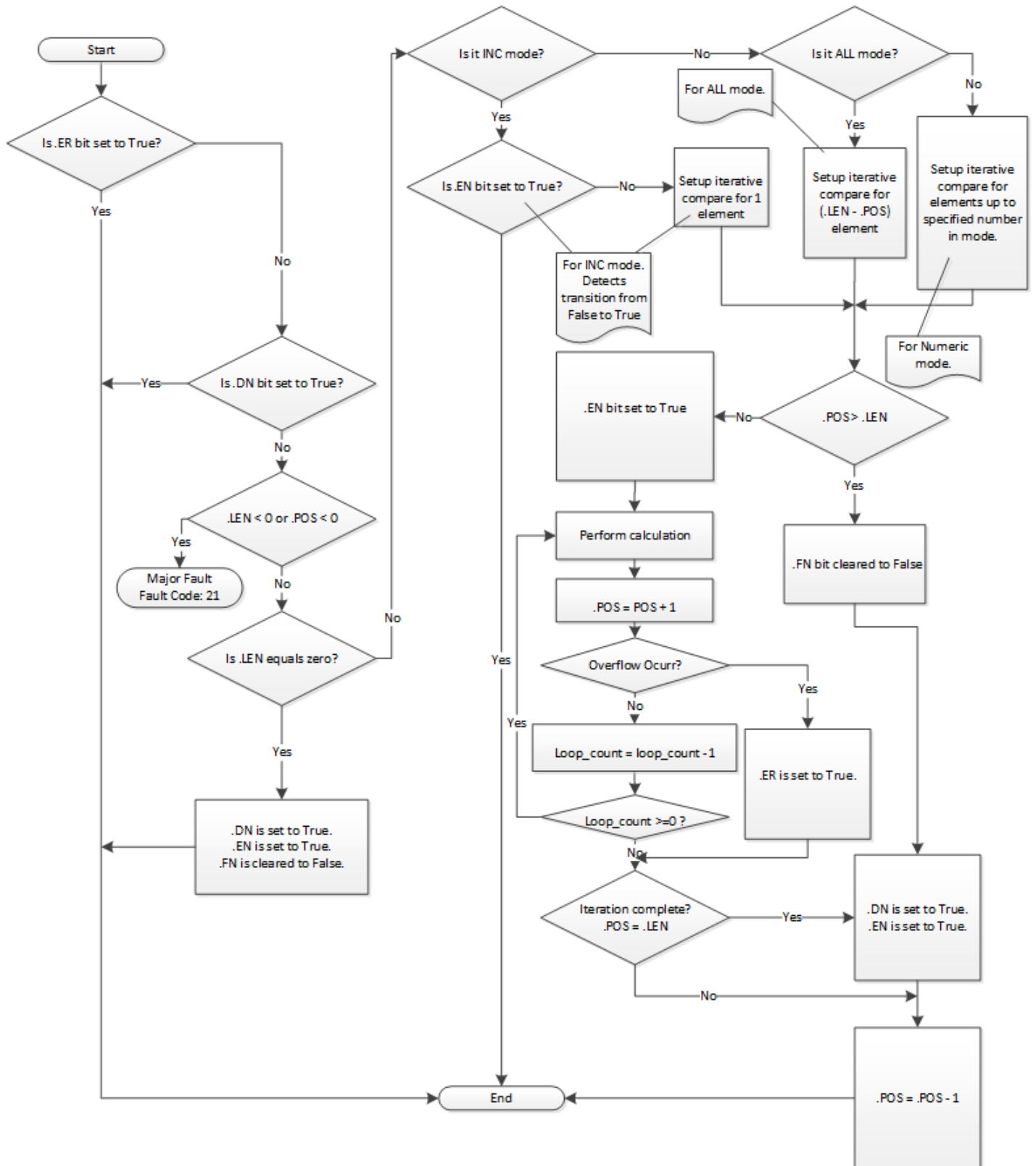
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme FAL (condition de sortie d'échelon est fausse)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme FAL (condition de sortie d'échelon est vraie)
Post-scrutation	N/A

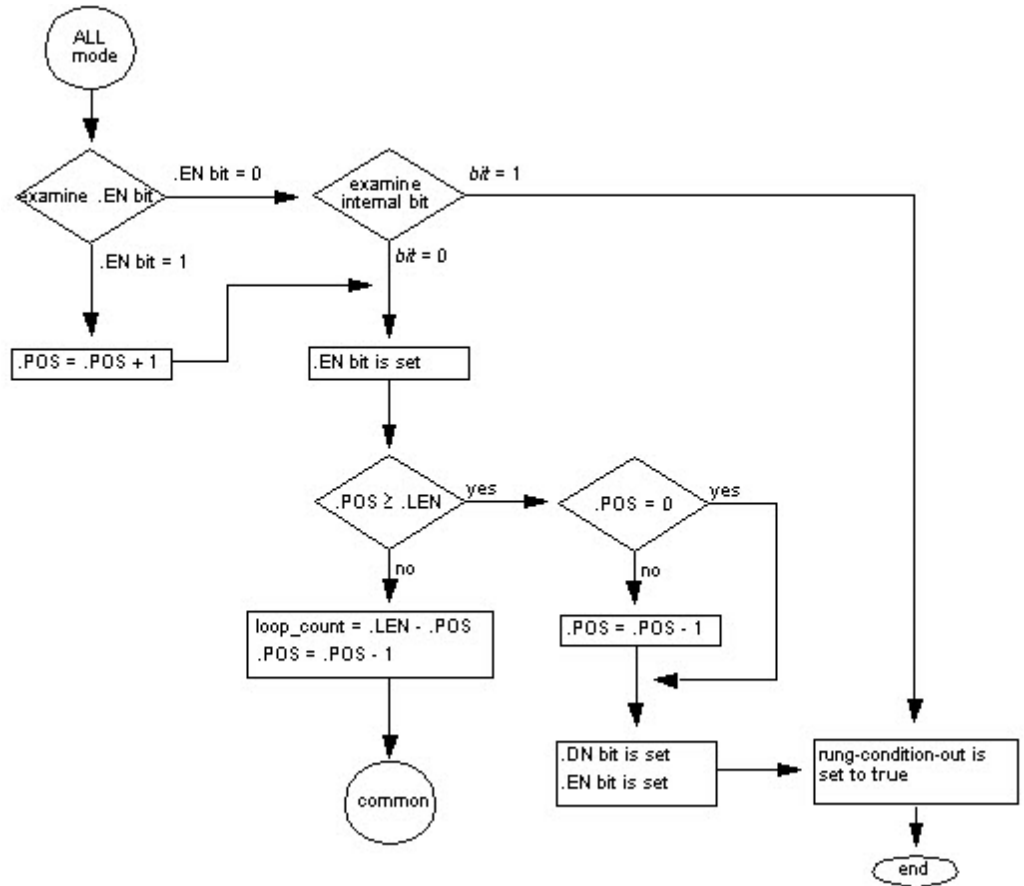
Organigramme FAL (condition de sortie d'échelon est fausse)



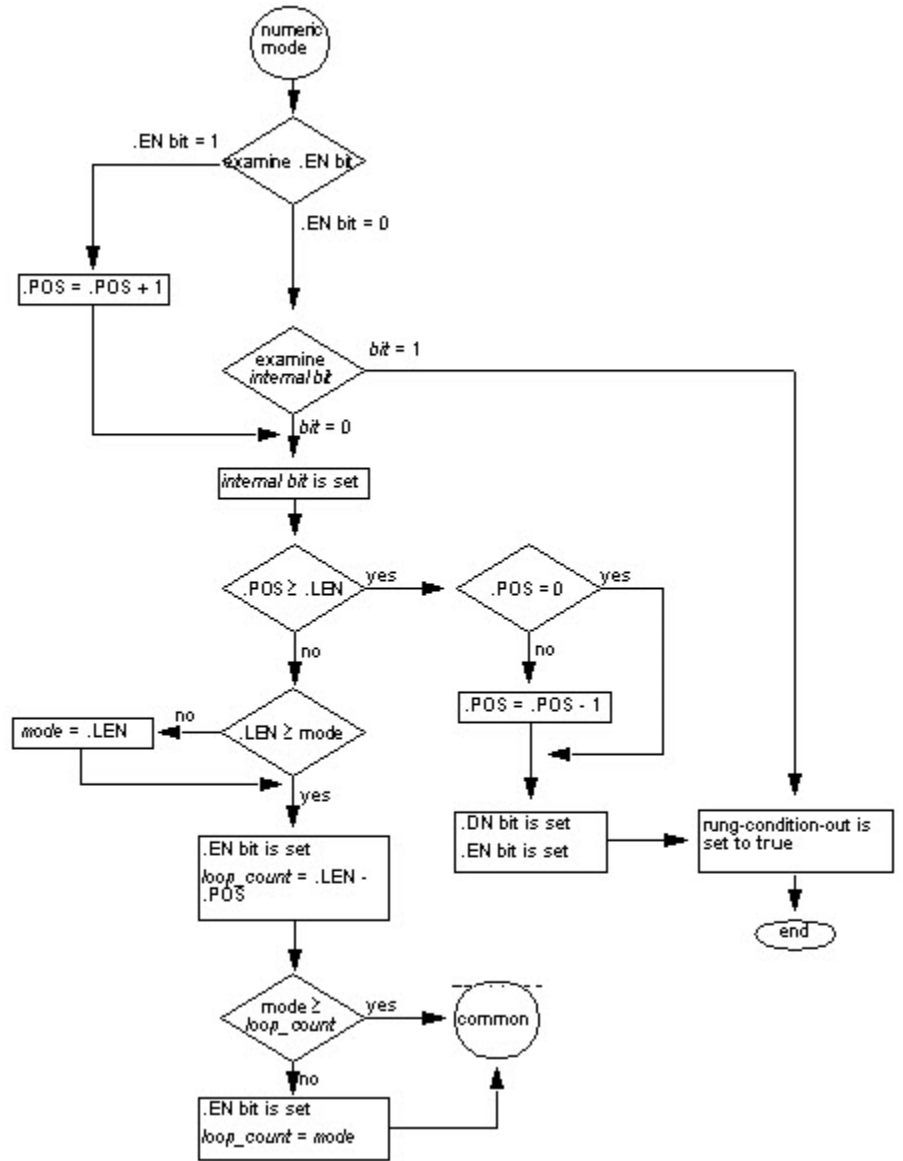
Organigramme FAL (condition de sortie d'échelon est vraie)



Organigramme FAL (Mode Tous)

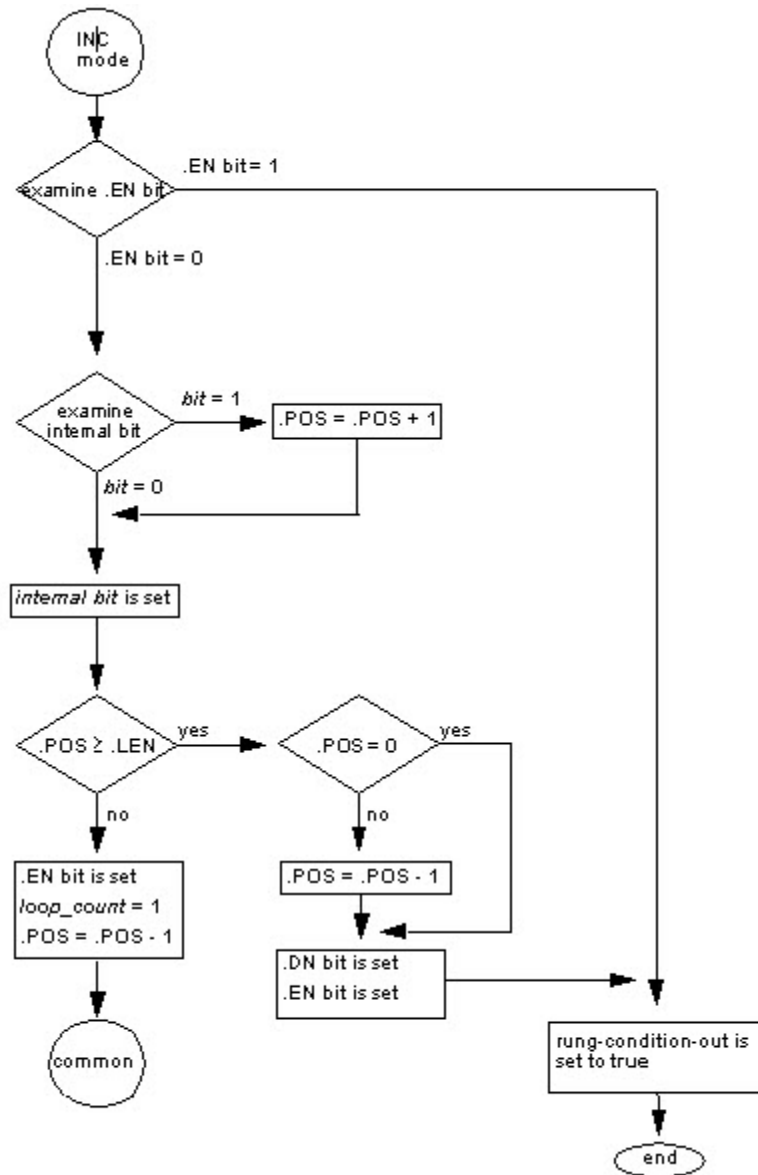


Organigramme FAL (Mode Numérique)





Organigramme FAL (Mode Incrémentiel)

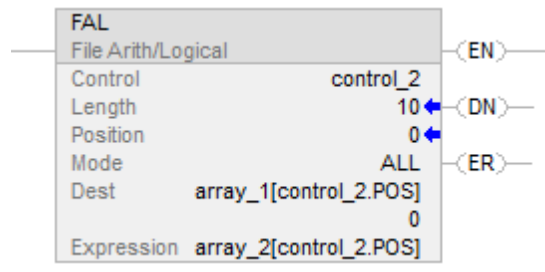


Exemples

Exemple 1

Tableau à tableau.

**Diagramme à relais**



En cas d'activation, l'instruction FAL copie chaque élément de array\_2 à la même position dans array\_1.

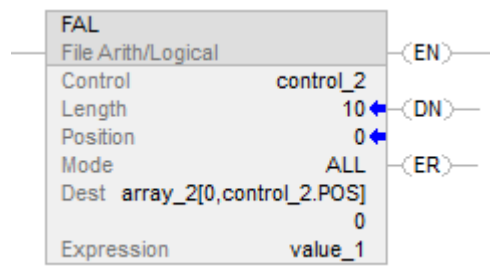


Expression: array\_2[control\_2.pos]      Destination: array\_1[control\_2.pos]

**Exemple 2**

Copie élément à tableau.

**Diagramme à relais**



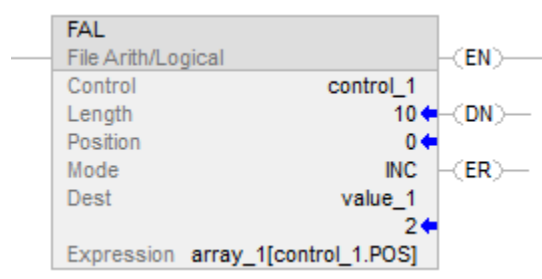
En cas d'activation, l'instruction FAL copie value\_1 aux 10 premières positions de la deuxième dimension de array\_2.



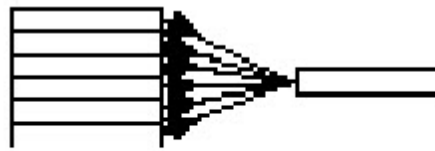
Expression: value\_1      Destination: array\_2[0,control\_2.pos]

**Exemple 3 :**

Copie tableau à élément.



Chaque activation de l’instruction FAL entraîne la copie par cette dernière de la valeur actuelle de array\_1 sur value\_1. L’instruction FAL utilise le mode Incrémentiel et, de ce fait, une seule valeur de tableau est copiée lors de chaque activation de cette instruction. Lors de l’activation suivante de cette instruction, cette dernière remplace value\_1 par la valeur suivante du array\_1.



Expression: array\_1[control\_1.pos]      Destination: value\_1

**Exemple 4 :**

Opération arithmétique : tableau/tableau à tableau



Lorsqu'elle est activée, l'instruction FAL divise la valeur dans la position actuelle d'array\_2 par la valeur dans la position actuelle d'array\_3 et mémorise le résultat dans la position actuelle d'array\_1.

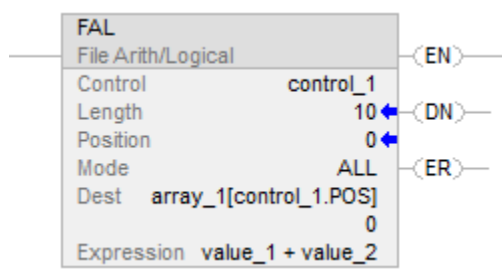


Expression:  
array\_2[control\_2.pos] /  
array\_3[control\_2.pos]

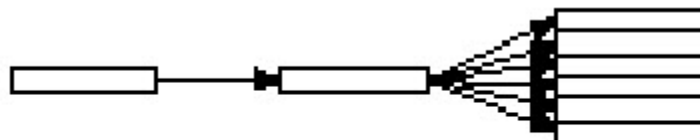
Destination:  
array\_1[control\_2.pos]

**Exemple 5 :**

Opération arithmétique : tableau/tableau à tableau



Lorsqu'elle est activée, l'instruction FAL additionne value\_1 et value\_2 et mémorise le résultat dans la position actuelle d'array\_1.

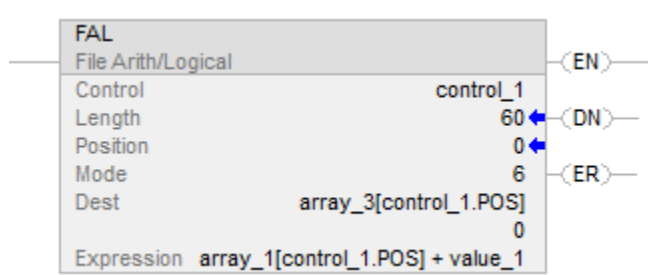


Expression:  
value\_1 + value\_2

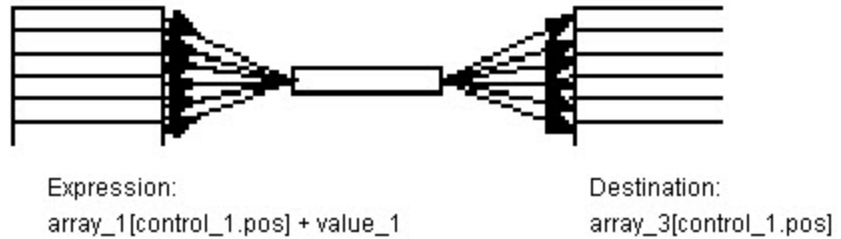
Destination:  
array\_1[control\_1.pos]

**Exemple 6 :**

Opération arithmétique : tableau + élément à tableau

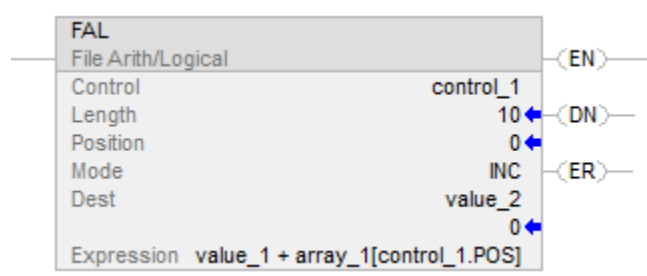


Lorsqu'elle est activée, l'instruction FAL ajoute la valeur dans la position actuelle d'array\_1 à value\_1 et mémorise le résultat dans la position actuelle d'array\_3. Cette instruction doit être exécutée à 10 reprises avant d'obtenir la manipulation de la totalité d'array\_1 et d'array\_3.

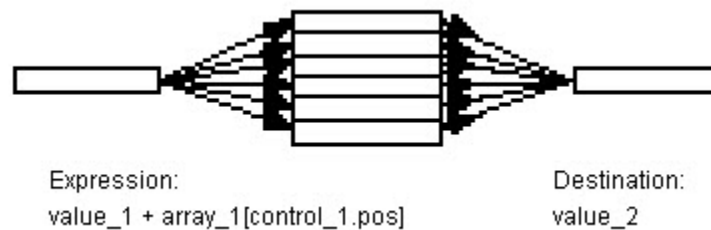


**Exemple 7 :**

Opération arithmétique : (élément + tableau) à élément

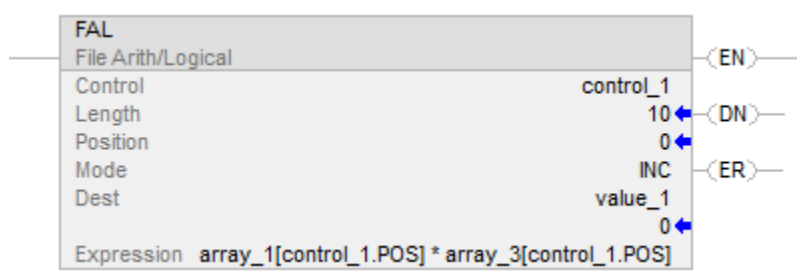


Chaque activation de l'instruction FAL, ajoute value\_1 à la valeur actuelle d'array\_1 et mémorise le résultat dans value\_2. L'instruction FAL utilise le mode Incrémentiel et, de ce fait, une seule valeur de tableau vient s'ajouter à value\_1 lors de chaque activation de cette instruction. Lors de l'activation suivante de cette instruction, cette dernière remplace value\_2.

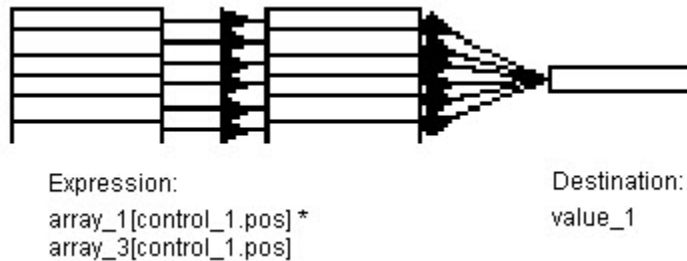


**Exemple 8 :**

Opération arithmétique : (tableau \* tableau) à élément



Lorsqu'elle est activée, l'instruction FAL multiplie la valeur actuelle d'array\_1 par la valeur actuelle d'array\_3 et mémorise le résultat dans value\_1. L'instruction FAL utilise le mode Incrémentiel et, de ce fait, une seule paire de valeurs de tableau est multipliée lors de chaque activation de cette instruction. Lors de l'activation suivante de cette instruction, cette dernière remplace value\_1.



**Voir aussi**

[Instructions File/Misc](#) sur la [page 499](#)

[Opérateurs valides](#) sur la [page 371](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

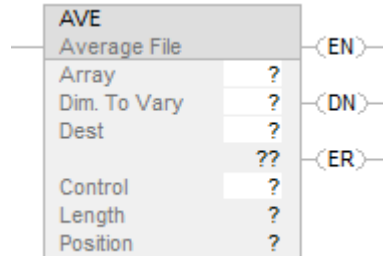
**Moyenne de fichier (AVE)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction AVE calcule la moyenne d'une série de valeurs.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Diagramme à relais

Opérande	Type	Format	Description
Array Tag	SINT INT DINT REAL	étiquette	Trouver la moyenne des valeurs de ce tableau préciser le premier élément du groupe d'éléments pour déterminer la moyenne Ne pas utiliser CONTROL.POS dans cet indice
Dimension to vary	DINT	immédiate (0, 1, 2)	De quelle dimension se servir l'ordre des dimensions est le suivant : array[0,1,2]
Destination	SINT INT DINT REAL	étiquette	Résultat de l'opération
Control	CONTROL	étiquette	Structure de contrôle de l'opération
Length	DINT	immédiate	Nombre d'éléments du tableau dont il faut déterminer la moyenne
Position	DINT	immédiate	Décalage dans le tableau spécifié qui identifie l'élément actuel auquel l'instruction est en train d'accéder. la valeur initiale type est 0

## Description

L'instruction AVE calcule la moyenne d'une série de valeurs.

---

**Important :** S'assurer que Length ne provoque pas un dépassement par l'instruction de Dimension to vary spécifiée. Si cela se produit, Destination sera incorrecte. Pour de plus amples informations, reportez-vous à la section Consultation d'un tableau en tant que bloc de mémoire.

---

En cas de débordement durant l'évaluation d'une expression, la lecture de l'instruction va au-delà de la fin d'un tableau et l'instruction définit le bit ER et arrête l'instruction

## Affecte les indicateurs d'état mathématique

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section Indicateurs d'état mathématique.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

## Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

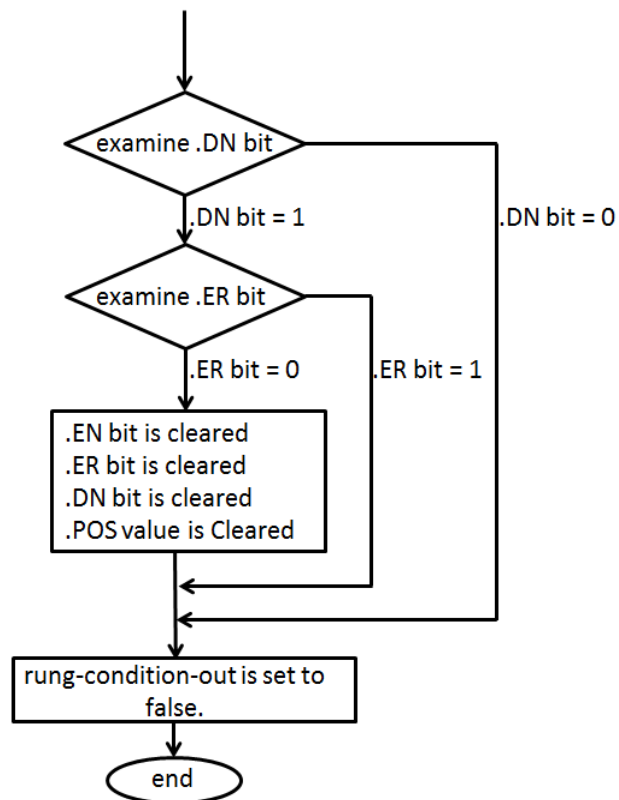
## Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Le bit .EN est mis à zéro. Le bit .DN est mis à zéro. Si le bit .ER est égal à zéro pendant la pré-scrutation, tous les bits de contrôle (.DN, .EN, .EU, .EM, .UL, .IN et .FD) sont mis à zéro.
Condition d'entrée d'échelon est fausse.	Reportez-vous à l'organigramme AVE (faux)
Condition d'entrée d'échelon est vraie.	L'instruction AVE calcule la moyenne en additionnant tous les éléments spécifiés dans le tableau et en divisant ce total par le nombre d'éléments.
Post-scrutation	N/A.



organigramme AVE (faux)



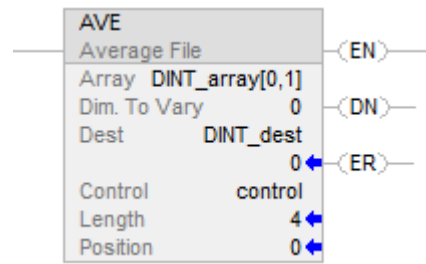
Exemple 1

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$AVE = \frac{19 + 14 + 9 + 4}{4} = \frac{46}{4} = 11.5$$

dint\_ave = 12

Diagramme à relais



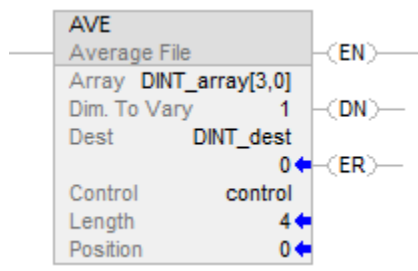
**Exemple 2**

		dimension 1				
	subscript	0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$AVE = \frac{5+4+3+2+1}{5} = \frac{15}{5} = 3$$

dint\_ave = 3

**Diagramme à relais**



**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

**Remplissage de fichier (FLL)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction FLL remplit un bloc de mémoire avec la valeur source fournie. Source reste inchangée.

Si le tableau de destination est SINT, INT, DINT ou REAL et si le type de valeur source est différent, cette valeur source va être convertie au type de destination avant sa mémorisation. Les types à petits nombres entiers vont être convertis en types à grands nombres entiers par le biais d'une extension de signe.

Si le tableau de destination est une structure, la valeur source va être rédigée sans conversion.

## Langages disponibles

### Diagramme à relais

FLL	
Fill File	
Source	?
Dest	?
Length	?

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

---

<b>Important :</b>	Une opération inattendue peut se produire dans les cas suivants :
	<ul style="list-style-type: none"> <li>• Les opérandes d'étiquette de sortie sont remplacés.</li> <li>• Les membres d'un opérande de structure sont remplacés.</li> <li>• Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.</li> </ul>

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section Conversion des données.

### Diagramme à relais

Opérande	Type de données	Format	Description
Source	SINT INT DINT REAL	immédiate étiquette	Élément à copier
Destination	SINT INT DINT REAL structure	étiquette	Élément initial à remplacer par Source.
Length	DINT INT SINT	immédiate étiquette	Nombre d'éléments destinataires à remplir.

Le nombre d'octets remplis est le plus petit de :

- Quantité demandée = Longueur x (nombre d'octets dans un élément de destination)
- Le nombre d'octets dans l'étiquette destinataire

**Astuce :** L'extrémité de l'étiquette de destination est définie comme étant le dernier octet de l'étiquette de base. Si l'étiquette est une structure, la fin de l'étiquette correspond au dernier octet du dernier élément de la structure. Cela signifie que l'instruction FLL risque de continuer à écrire après la fin d'un tableau membre mais ne continuera jamais d'écrire après l'extrémité de l'étiquette de base. Tester et confirmer que l'instruction FLL ne modifie pas des données qui ne devraient pas être modifiées.

Pour obtenir les meilleurs résultats, Source et Destination devraient être du même type. Utiliser FLL pour remplir une structure en faisant appel à une constante comme, par exemple, des 0.

En cas d'initialisation d'une structure, s'assurer qu'il y a une instance contenant les valeurs initiales et utiliser COP pour la reproduire. FLL peut s'utiliser, par exemple, pour mettre à zéro la totalité de la structure.

Si Source est :	Et Destination est :	Source est convertie en :
SINT, INT, DINT ou REAL	SINT	SINT
SINT, INT, DINT ou REAL	INT	INT
SINT, INT, DINT ou REAL	DINT	DINT
SINT, INT, DINT ou REAL	REAL	REAL

La conversion de gros nombres entiers en petits nombres entiers va entraîner une troncature (les bits hauts sont ignorés). Dès que la source est convertie, elle est rédigée dans la destination à N reprises, où N = nombre d'octets. L'extension de signe est le résultat d'une conversion de petits nombres entiers en gros nombres entiers. Les nombres REAL vont être arrondis lors d'une conversion en nombres entiers.

**Affecte les indicateurs d'état mathématique**

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

### Exécution

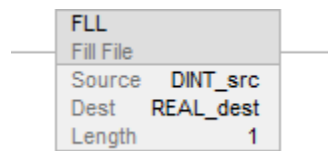
#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Cette instruction remplit la mémoire.
Post-scrutation	N/A

### Exemple

L'instruction FLL copie le nombre d'éléments de destination spécifié par Length depuis l'opérande source de type DINT\_src dans une destination de type REAL\_dest.

#### Diagramme à relais



### Voir aussi

[Instructions File/Misc](#) sur la [page 499](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

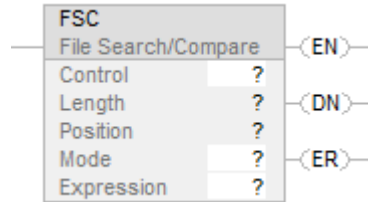
## Recherche et comparaison de fichier (FSC)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction FSC compare les valeurs d'un tableau, élément par élément.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

#### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Control	CONTROL	Étiquette (Tag)	Structure de contrôle de l'opération
Longueur (Length)	DINT	Immédiate	Nombre d'éléments dans le tableau à manipuler
Position (Position)	DINT	Immédiate	Décalage dans le tableau Valeur initiale typique : 0
Mode	DINT	Immédiate	Distribution de l'opération Sélectionner INC ou ALL, ou saisir un nombre compris entre 1 et 2 147 483 647
Expression	SINT INT DINT REAL STRING	Immédiate Étiquette (Tag)	Une expression composée d'étiquettes et/ou de valeurs immédiates séparées par des opérateurs

### Structure CONTROL

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction FSC est activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction s'est exécutée sur le dernier élément (.POS = .LEN).
.ER	BOOL	Le bit d'erreur n'est pas modifié.
.IN	BOOL	Le bit d'inhibition indique que l'instruction FSC a détecté une vraie comparaison. Vous devez effacer ce bit pour poursuivre la recherche.
.FD	BOOL	Le bit de détection indique que l'instruction FSC a détecté une vraie comparaison.
.LEN	DINT	La longueur indique le nombre d'éléments du tableau sur lesquels l'instruction s'exécute.
.POS	DINT	La position contient la position de l'élément actuel auquel l'instruction est en train d'accéder.

### Description (Description)

Lorsque EnableIn de l'instruction FSC passe de faux à vrai, l'expression est évaluée pour le mode d'itération spécifié.

Si le résultat de l'évaluation est vrai, l'instruction définit le bit .FD, et la valeur .POS reflète la position dans tableau où l'instruction a détecté la vraie comparaison. L'instruction définit le bit .IN pour empêcher toute itération ultérieure.

### Sélectionner le mode de fonctionnement

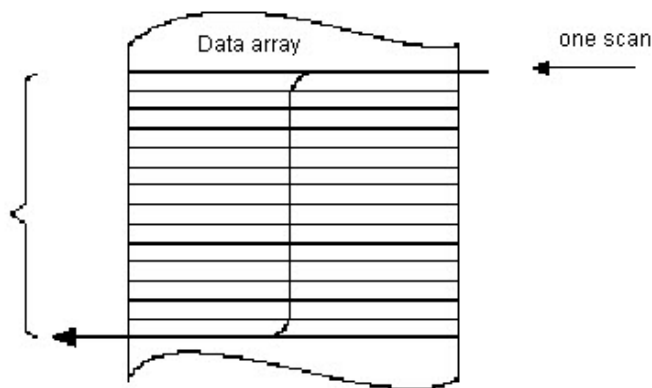
Pour les instructions FSC, le mode indique à l'automate comment distribuer le fonctionnement du tableau.

Si vous voulez :	Sélectionnez ce mode :
Procéder à l'exécution sur tous les éléments spécifiés d'un tableau avant de passer à l'instruction suivante.	Tout (All)
Distribuer le fonctionnement du tableau sur un certain nombre de scrutations. Saisir le nombre d'éléments par scrutation sur lesquels procéder à l'exécution (1-2 147 483 647)	Numérique

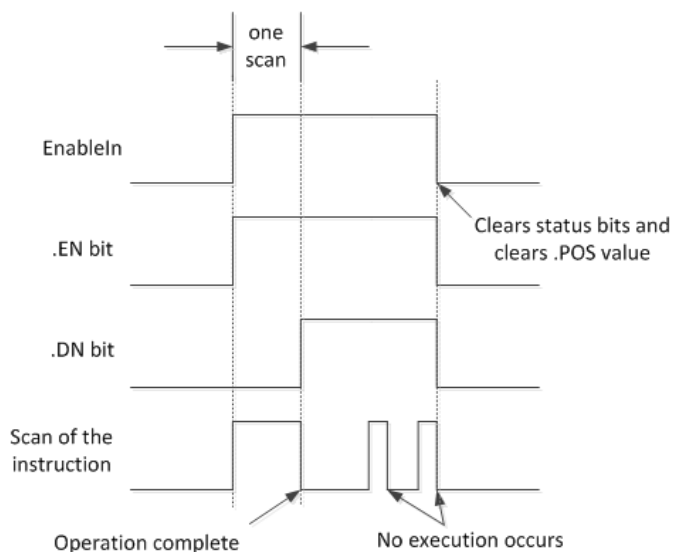
Manipuler un élément du tableau à chaque fois que EnableIn passe de faux à vrai.	Incrémentiel
--	--------------

**Mode Tous**

Dans le mode Tous, tous les éléments spécifiés dans le tableau sont exploités avant de passer à l'instruction suivante. Cette opération commence lorsque l'option EnableIn d'activation de l'instruction passe de faux à vrai. La valeur de la position (.POS) de la structure de contrôle indique l'élément du tableau qu'utilise en ce moment cette instruction. L'opération s'arrête lorsque les deux conditions suivantes sont présentes. Lorsque la valeur .POS est égale ou supérieure à la valeur .LEN ET lorsque l'expression correspond à vrai.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est vrai. Le bit .DN, le bit .EN et la valeur .POS sont mis à zéro lorsque EnableIn est faux. Ce n'est qu'à ce moment-là qu'une autre exécution de cette instruction peut être déclenchée par une transition d'EnableIn de faux à vrai.

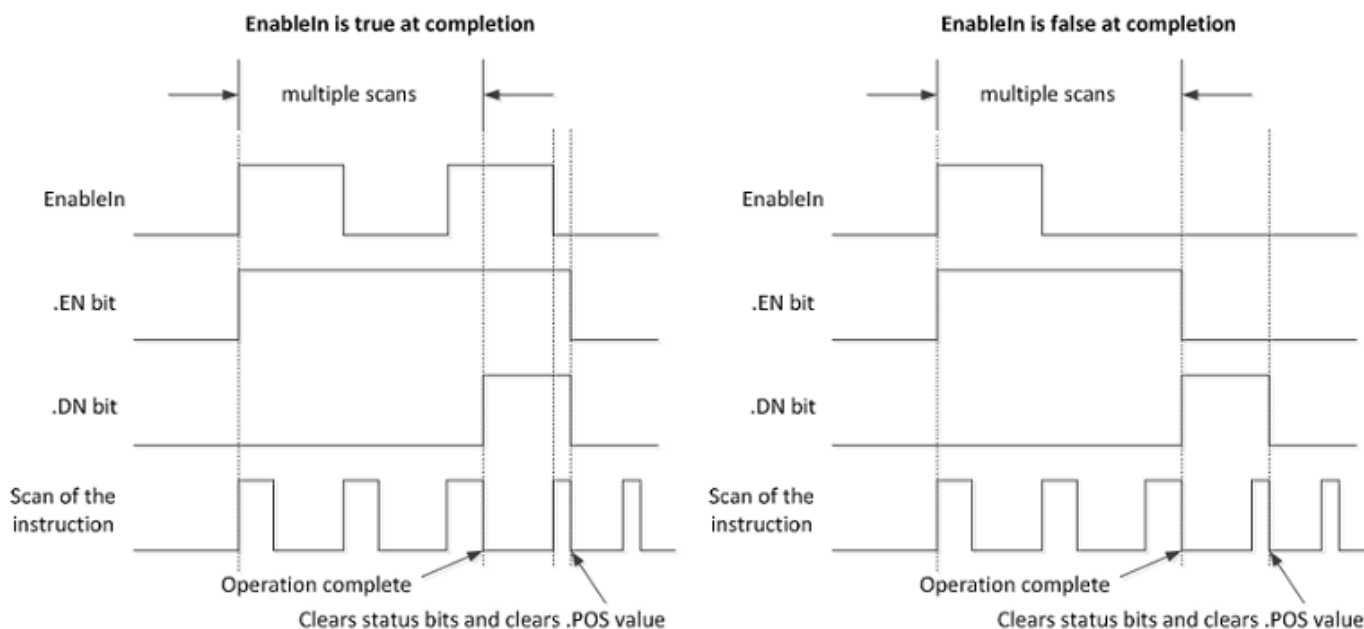




### Mode Numérique

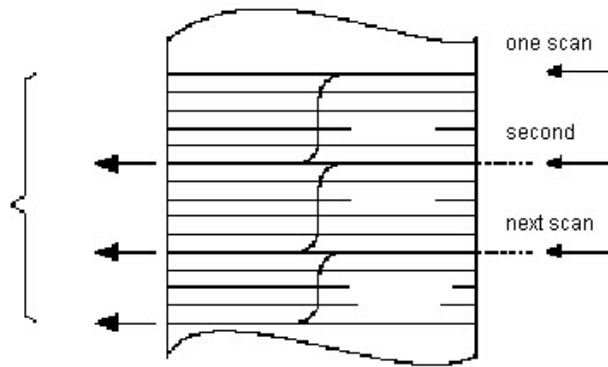
Le mode Numérique répartit l'opération de tableau sur un certain nombre de scrutations. Ce mode est utile lorsqu'on travaille avec des données qui ne sont pas critiques sur le plan temporel ou avec des volumes importants de données. Vous entrez le nombre d'éléments sur lesquels va agir chaque scrutation, ce qui permet de réduire la durée des scrutations.

Cette exécution se déclenche lorsque l'option EnableIn d'activation passe de faux à vrai. Dès que cette instruction se déclenche, elle est exécutée chaque fois qu'elle fait l'objet d'une scrutation et pendant le nombre de scrutations qui s'avèrent nécessaires pour exploiter la totalité du tableau. Dès que l'option EnableIn se déclenche, elle peut changer de manière répétée, sans interrompre l'exécution de cette instruction.



Évitez l'utilisation des résultats d'une instruction de fichier opérant dans le mode Numérique tant que le bit .DN ou .IN n'est pas vrai.

Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est vrai.

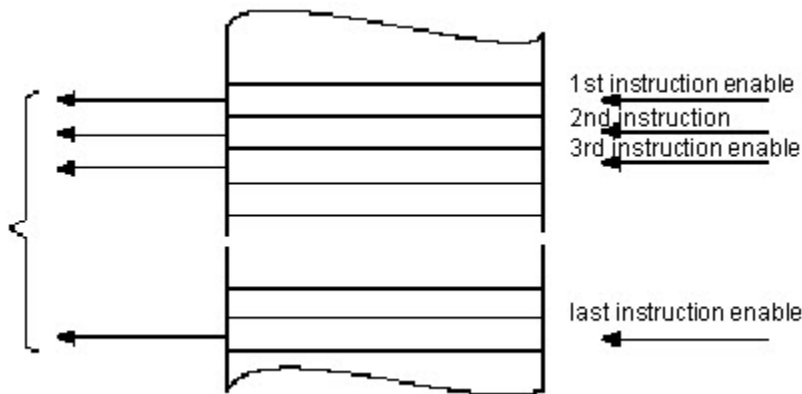


Si l'option EnableIn est vraie lors de l'exécution, les bits .EN et .DN sont vrais jusqu'à ce que l'option EnableIn devienne fausse. Lorsque l'option EnableIn devient fausse, ces bits sont mis à zéro, de même que la valeur de .POS.

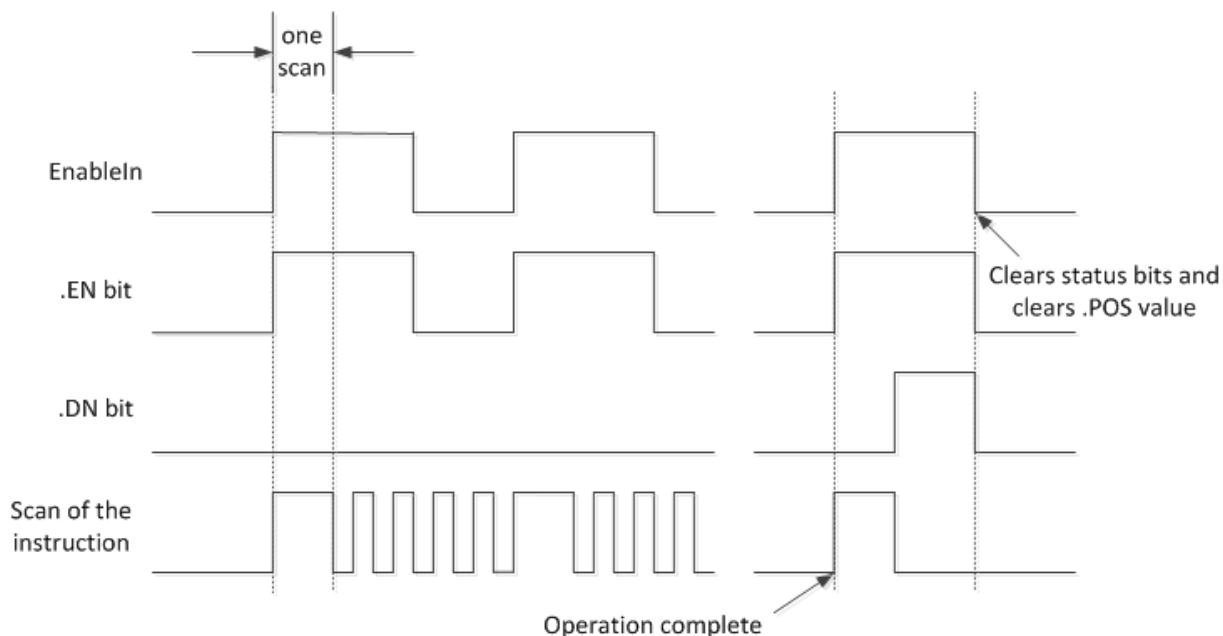
Si, à la fin de cette exécution, l'option EnableIn d'activation de l'entrée est fausse, le bit .EN est immédiatement mis à zéro. Une scrutation après l'effacement du bit .EN, le bit .DN et la valeur de .POS sont eux aussi mis à zéro.

**Mode Incrémentiel**

Le mode Incrémentiel manipule un élément du tableau chaque fois que l'option EnableIn de l'instruction passe de faux à vrai.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. L'exécution n'a lieu que lors d'une scrutation durant laquelle l'option EnableIn passe de faux à vrai. Chaque fois que cela se produit, un seul élément du tableau est manipulé. Si l'option EnableIn reste vraie pendant plus d'une scrutation, l'exécution de cette instruction n'a lieu que lors de la première scrutation.



Le bit .EN est défini lorsque la condition d'entrée d'échelon est vraie. Le bit .DN est défini lorsque le dernier élément du tableau a été manipulé. Après la manipulation du dernier élément et après que la condition d'entrée d'échelon soit devenue fausse, le bit .EN, le bit .DN et la valeur de .POS sont mis à zéro.

La différence entre le mode Incrémentiel et le mode Numérique, au rythme d'un élément par scrutation, est la suivante :

Dans le mode Numérique, quel que soit le nombre d'éléments par scrutation, il faut une seule transition de faux à vrai de l'option EnableIn pour lancer l'exécution. L'exécution de cette instruction se poursuit afin de réaliser le nombre spécifié d'éléments lors de chaque scrutation jusqu'à leur achèvement final, quel que soit l'état de l'option EnableIn.

Lors du mode Incrémentiel, l'option EnableIn doit au préalable passer de faux à vrai afin de manipuler un élément du tableau.

### Expressions de formats

Pour chaque opérateur que vous utilisez dans le cadre d'une expression, vous devez fournir un ou deux opérandes (étiquettes ou valeurs immédiates). Utiliser le tableau suivant pour formater les opérateurs et opérandes au sein d'une expression.

Pour les opérateurs qui opèrent sur :	Utiliser ce format :	Exemple
Un opérande	operator(operand)	ABS(tag)
Deux opérandes	operand_a operator operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

### Définir l'ordre des opérations

Les opérations que vous rédigez dans l'expression sont exécutées par l'instruction dans un ordre bien précis, qui n'est pas forcément l'ordre dans lequel vous les avez rédigées. Vous pouvez outrepasser l'ordre de réalisation de cette opération en groupant plusieurs termes en les mettant entre parenthèses, ce qui oblige cette instruction à exécuter une opération entre parenthèses avant les autres opérations.

Les opérations de rang égal sont exécutées de gauche à droite.

Ordre	Fonctionnement
1	( )
2	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3	**
4	- (négation), NOT
5	*, /, MOD
6	- (soustraction), +
7	AND
8	XOR
9	OU
10	<, <=, >, >=, =, <>

### Utiliser des chaînes dans une expression.

Pour utiliser des chaînes de caractères ASCII dans une expression, observer ces lignes directrices :

Une expression vous permet de comparer deux étiquettes de chaînes.

Vous ne pouvez pas entrer des caractères ASCII directement dans cette expression.

Seuls les opérateurs suivants sont autorisés :

Opérateur	Description (Description)
=	Égal
<	Inférieur à
<=	Inférieur à ou égal
>	Supérieur à
>=	Supérieur à ou égal
<>	Différent

Les chaînes sont égales si leurs caractères correspondent.

Les caractères ASCII sont sensibles à la casse. Le A majuscule (\$41) n'est pas égal au a minuscule (\$61).

Les valeurs hexadécimales des caractères permettent de définir si une chaîne est plus ou moins longue qu'une autre.

Lorsque deux chaînes sont triées comme dans un annuaire téléphonique, l'ordre des chaînes détermine celle qui est supérieure.

ASCII Characters	Hex Codes
tab	\$31\$61\$62
tb	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
ControlLogix 5580	Non
CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
.POS < 0 ou .LEN < 0	4	21

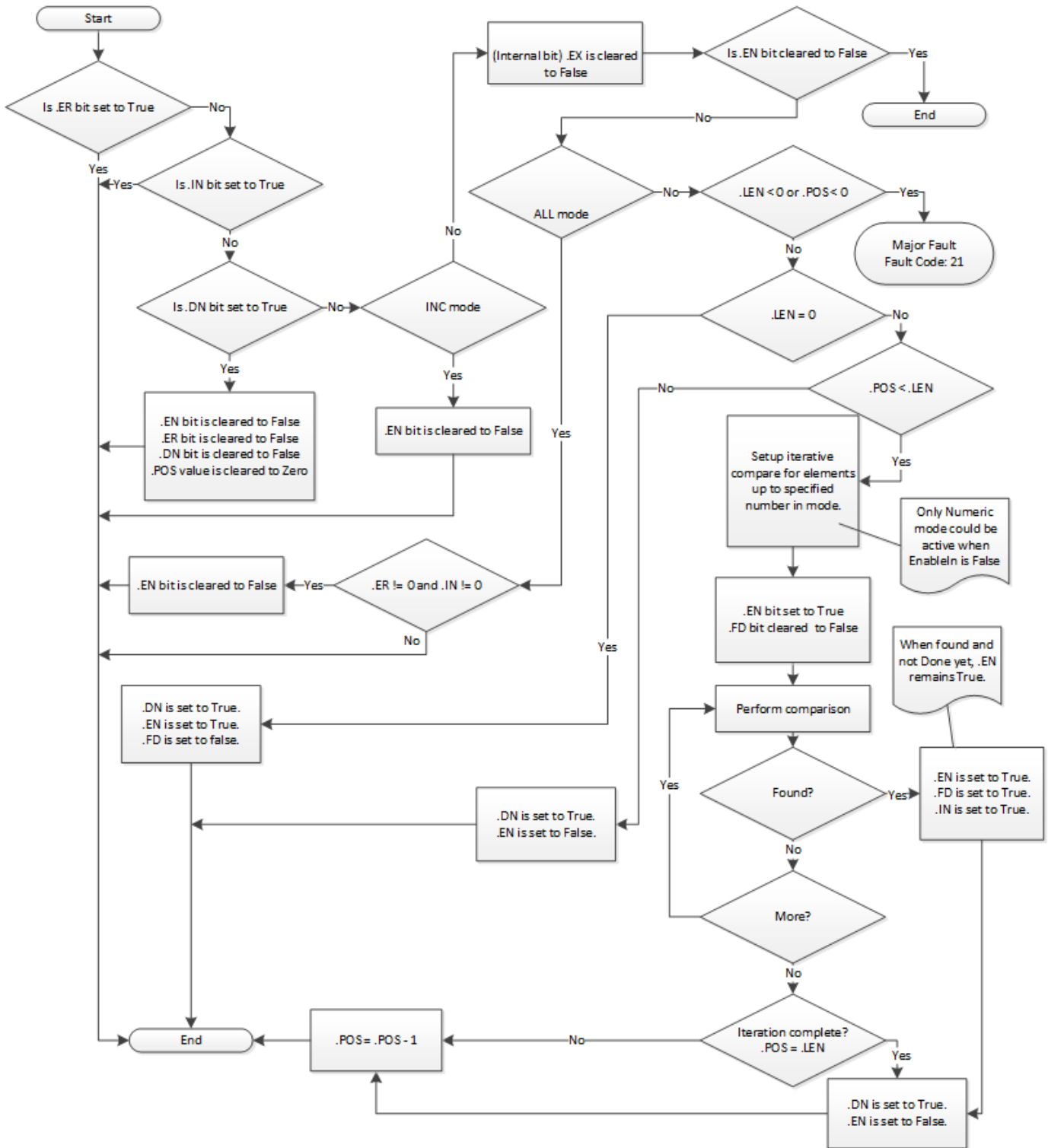
Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution**

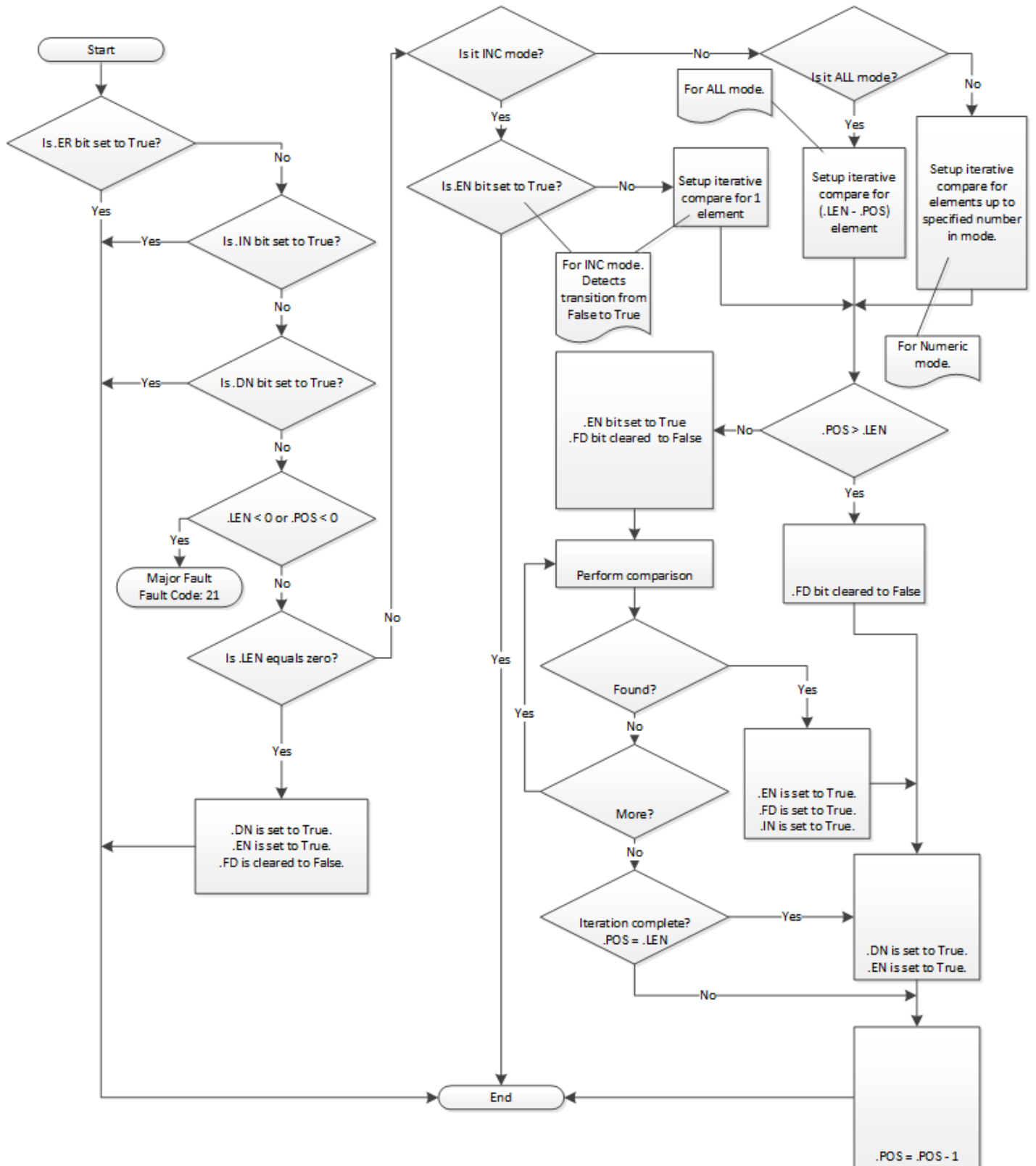
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme FSC (condition de sortie d'échelon est fausse)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme FSC (condition de sortie d'échelon est vraie)
Post-scrutation	N/A

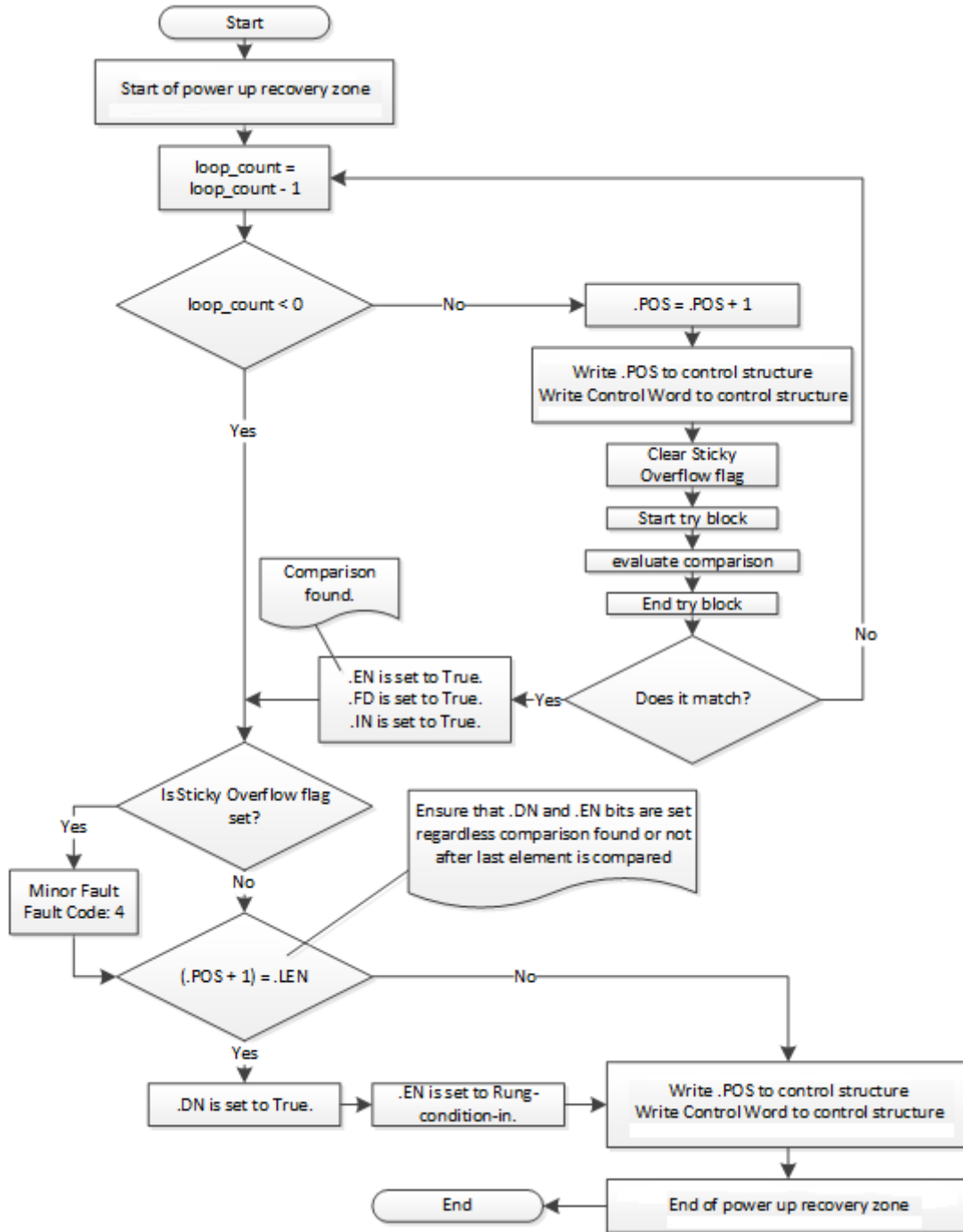
Organigramme FSC (condition de sortie d'échelon est fausse)



Organigramme FSC (condition de sortie d'échelon est vraie)

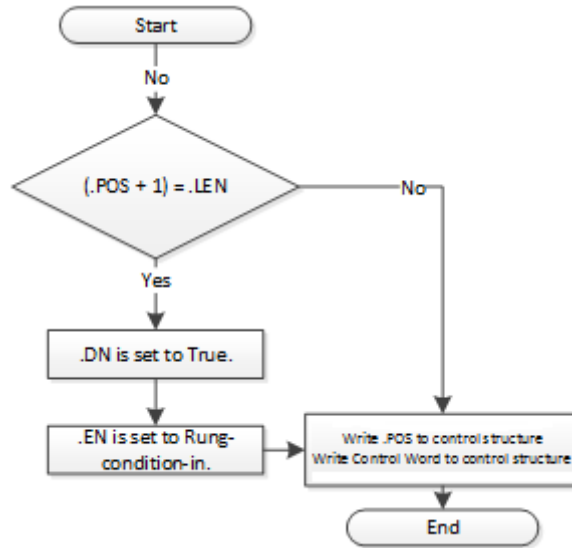


Organigramme FSC (sous-flux commun FSC)





**Organigramme FSC (sous-flux commun d'exception FSC)**

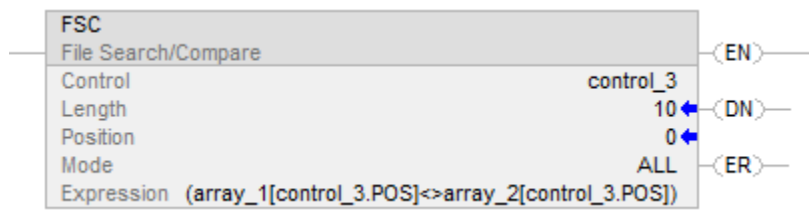


**Exemples**

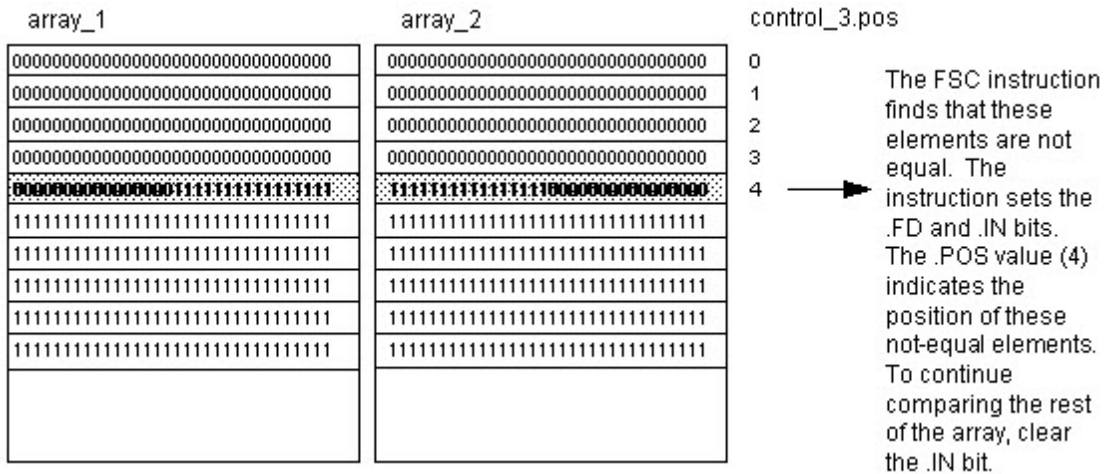
**Exemple 1**

Procéder à des recherches entre deux tableaux DINT afin d'identifier les éléments qui sont inégaux.

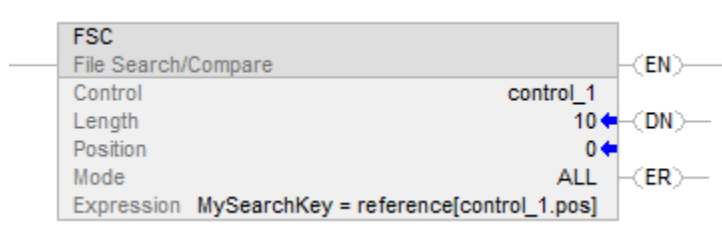
**Diagramme à relais**



Lorsque la fonction FSC est activée, elle compare chacun des 10 premiers éléments d'array\_1 aux éléments correspondants d'array\_2. Lorsqu'un élément inégal est ainsi identifié, les bits FD et IN sont définis. POS identifie l'emplacement des éléments inégaux. Mettre le bit IN à zéro pour continuer ces recherches sur le reste du tableau.



**Exemple 2**



Rechercher la présence d'une correspondance dans un tableau de structures.

**Exemple 3**



Rechercher la présence d'une correspondance dans un tableau de chaînes.

Lorsqu'elle est activée, l'instruction FSC compare les caractères du code à 10 éléments de code\_table.

**Voir aussi**

[Instructions File/Misc](#) sur la [page 499](#)

[CMP](#) sur la [page 298](#)

[FAL](#) sur la [page 509](#)

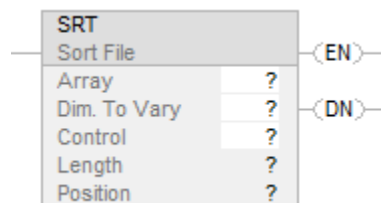
[Indexer via des tableaux](#) sur la [page 899](#)

[Opérateurs valides](#) sur la [page 371](#)

**Tri sur fichier (SRT)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction SRT trie une série de valeurs dans une dimension (Dim to vary) du tableau dans un ordre ascendant.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

```
SRT(Array,Dimtovary,Control);
```

### Opérandes

#### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Tableau	SINT INT DINT REAL	Étiquette de tableau	tableau à trier spécifier le premier élément du groupe d'éléments à trier
Dimension to vary	DINT	Immédiate (0, 1, 2)	de quelle dimension se servir l'ordre des dimensions est le suivant : array[0,1,2]
Control	CONTROL	Étiquette (Tag)	structure de contrôle de l'opération
Longueur (Length)	DINT	Immédiate	nombre d'éléments du tableau à trier
Position (Position)	DINT	Immédiate	élément actuel du tableau valeur initiale typique : 0

#### Texte structuré

Opérande	Type (Type)	Format	Description (Description)
Tableau	SINT INT DINT REAL	Étiquette de tableau	tableau à trier spécifier le premier élément du groupe d'éléments à trier
Dimension to vary	DINT	Immédiate (0, 1, 2)	de quelle dimension se servir l'ordre des dimensions est le suivant : array[0,1,2]
Control	CONTROL	Étiquette (Tag)	structure de contrôle de l'opération
Longueur (Length)	DINT	Immédiate	Nombre d'éléments du tableau à trier. L'accès aux valeurs spécifiées de Length et de Position se fait depuis les membres .LEN et .POS de la structure CONTROL.
Position (Position)	DINT	Immédiate	élément actuel du tableau valeur initiale typique : 0 L'accès aux valeurs spécifiées de Length et de Position se fait depuis les membres .LEN et .POS de la structure CONTROL.

Reportez-vous à Syntaxe du texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Structure CONTROL

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction SRT a été activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction a agi sur le dernier élément du Tableau.
.ER	BOOL	Le bit d'erreur est défini lorsque .LEN < 0 ou lorsque .POS < 0. En outre, l'une ou l'autre de ces conditions déclenche un défaut majeur. Lorsque le bit .ER est défini, cette instruction n'effectue pas d'exécution.
.LEN	DINT	Le mot de longueur spécifie le nombre d'éléments du tableau sur lesquels opère cette instruction.
.POS	DINT	Le mot de position identifie l'élément actuel auquel cette instruction est en train d'accéder.

### Description (Description)

L'instruction SRT trie une série de valeurs dans une dimension (Dim to vary) du Tableau dans un ordre ascendant.

---

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

L'instruction SRT agit sur une mémoire de données contiguës. Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570 uniquement, la portée de cette instruction est limitée par l'étiquette de base. L'instruction SRT ne va pas rédiger des données en dehors de l'étiquette de base mais peut traverser des limites de membres. Lorsque vous spécifiez un tableau qui est membre d'une structure et lorsque la longueur dépasse la taille de ce tableau, vous devez tester et confirmer que l'instruction SRT ne modifie pas des données que vous ne souhaitez pas modifier.

Dans la Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580, les données sont limitées par le membre spécifié.

Dans cette instruction de transition, le logique à relais fait basculer la condition d'entrée d'échelon de faux à vrai afin de permettre l'exécution de cette instruction.

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Non
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
.POS < 0 ou .LEN < 0	4	21
Dimension to vary > nombre de dimensions	4	20
Length > fin de tableau	4	20

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	Le bit .EN est défini sur faux Le bit .EN est défini sur faux Le bit .DN est défini sur faux
Condition d'entrée d'échelon est vraie	L'instruction s'exécute
Post-scrutation	N/A.

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Étant donné que cette instruction a besoin d'une transition pour son exécution, cette exécution est d'abord fausse puis vraie. Reportez-vous au table de Diagramme à relais pour plus de détails.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

### Exemples

#### Exemple 1

Trier DINT\_array, qui est DINT[4,5].

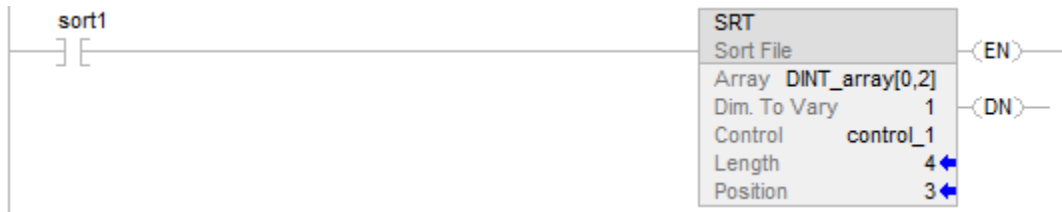
		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

**Before**

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	7	17	16
	1	15	14	8	12	11
	2	10	9	13	7	6
	3	5	4	18	2	1

**After**

#### Diagramme à relais



#### Texte structuré

IF sort1 then

control\_1.LEN := 4;

control\_1.POS := 0;

SRT(DINT\_array[0,2],0, control\_1);

END\_IF;

#### Exemple 2

Trier DINT\_array, qui est DINT[4,5].

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

**Before**

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	6	7	8	9	10
	3	5	4	3	2	1

**After**

## Diagramme à relais



## Texte structuré

```
ctrl.LEN := 4;
```

```
ctrl.POS := 0;
```

```
SRT(DINT_array[0,2],0, ctrl);
```

## Voir aussi

[Instructions File/Misc](#) sur la [page 499](#)

[Moyenne de fichier \(AVE\)](#) sur la [page 526](#)

[Conversions de données](#) sur la [page 889](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Écart standard de fichier (STD)

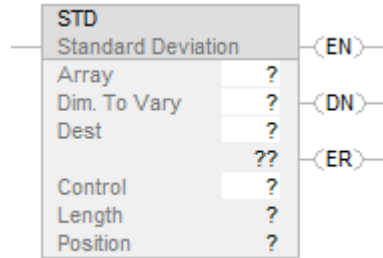
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction STD calcule l'écart standard d'une série de valeurs dans une dimension d'Array et mémorise le résultat dans Destination.



**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Tableau	SINT INT DINT REAL	étiquette de tableau	Trouver l'écart standard des valeurs de ce tableau spécifier le premier élément du groupe d'éléments à utiliser lors du calcul de l'écart standard
Dimension to vary	DINT	immediate (0, 1, 2)	de quelle dimension se servir l'ordre des dimensions est le suivant : array[0,1,2]
Destination	REAL	étiquette	résultat de l'opération
Control	CONTROL	étiquette	Structure de contrôle de l'opération
Longueur (Length)	DINT	immediate	nombre d'éléments du tableau à utiliser lors du calcul de l'écart standard
Position (Position)	DINT	immediate	Décalage dans le tableau spécifié qui identifie l'élément actuel auquel l'instruction est en train d'accéder. la valeur initiale type est 0

**Structure CONTROL**

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction STD a été activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction a agi sur le dernier élément du Tableau.
.ER	BOOL	Le bit d'erreur est défini lorsque l'instruction entraîne un débordement. L'instruction arrête l'exécution jusqu'à ce que le programme mets le bit .ER à zéro. La valeur .POS mémorise la position de l'élément à l'origine du débordement.
.LEN	DINT	Le mot de longueur spécifie le nombre d'éléments du tableau sur lesquels opère cette instruction.
.POS	DINT	Le mot de position est un décalage qui existe dans le tableau spécifié et qui identifie l'élément actuel auquel l'instruction est en train d'accéder.

**Description (Description)**

L'écart standard se calcule en fonction de cette formule :

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^N [(X_{(start+i)} - AVE)^2]}{(N-1)}}$$

Où :

start = indice dimension-to-vary de l'opérande du tableau

xi = élément variable du tableau

N = nombre d'éléments spécifiés dans le tableau

$$AVE = \frac{\sum_{i=1}^N x_{(start+i)}}{N}$$

---

**Important :** S'assurer que Length ne provoque pas un dépassement par l'instruction de Dimension to vary spécifiée. Si cela se produit, Destination sera incorrecte.

---

En cas de débordement durant l'évaluation d'une expression ou si la lecture de l'instruction va au-delà de la fin d'un tableau, l'instruction définit le bit ER et arrête l'exécution.

**Affecter les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecter les indicateurs d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, en fonction de la langage de programmation. Reportez-vous à la section Indicateurs d'état mathématique.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

<b>Un défaut majeur se produira si :</b>	<b>Type de défaut</b>	<b>Code de défaut</b>
.POS < 0 ou .LEN < 0	4	21
Dimension to vary > nombre de dimensions	4	20

Reportez-vous à Attributs communs pour les défauts liés aux opérandes.

**Exécution****Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Le bit .EN est mis à zéro. Le bit .DN est mis à zéro. Le bit .ER est mis à zéro.
Condition d'entrée d'échelon est fausse	Le bit .EN est mis à zéro. Le bit .ER est mis à zéro. Le bit .DN est mis à zéro. La valeur .POS est mise à zéro. La condition de sortie d'échelon est fausse.
Condition d'entrée d'échelon est vraie	Sur le plan interne, cette instruction utilise une instruction FAL pour calculer la moyenne : Expression = calcul de l'écart standard Mode = TOUS
Post-scrutation	N/A.

**Exemples**

**Exemple 1**

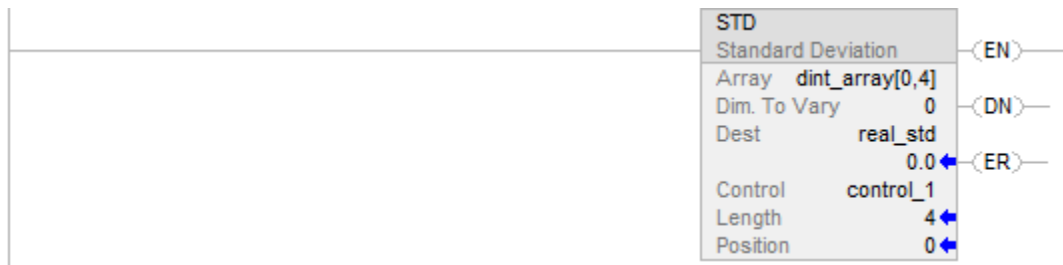
Calculer l'écart standard de arrayDint, à savoir DINT[4,5].

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$STD = \sqrt{\frac{\langle 16 - 8.5 \rangle^2 + \langle 11 - 8.5 \rangle^2 + \langle 6 - 8.5 \rangle^2 + \langle 1 - 8.5 \rangle^2}{\langle 4 - 1 \rangle}} = 6.454972$$

*real\_std* = 6.454972

**Diagramme à relais**



**Exemple 2**

Calculer l'écart standard de dint\_array, à savoir DINT[4,5].

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

**Diagramme à relais**



**Voir aussi**

[Instructions File/Misc](#) sur la [page 499](#)

[AVE](#) sur la [page 526](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

**Taille en éléments (SIZE)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

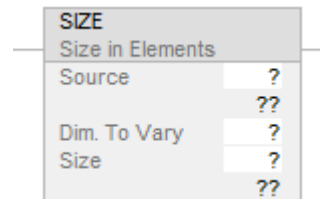
L'instruction SIZE trouve le nombre d'éléments (taille) dans la dimension désignée de tableau Source ou opérande de chaîne et place le résultat dans l'opérande Size. Cette instruction trouve la taille d'une dimension d'un tableau.

Cette instruction opère sur :

- Les tableaux
- Les tableaux d'une structure
- Les tableaux qui font partie d'un tableau plus gros
- Les étiquettes de chaînes

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

```
SIZE(Source,Dimtovary,Size);
```

#### Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
- Les membres d'un opérande de structure sont remplacés.
- Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.

---

Il existe des règles de conversion de données pour combiner des types de données numériques dans une instruction. Reportez-vous à la section *Conversion des données*.

### Diagramme à relais

Opérande	Type de données	Format	Description	
Source	SINT INT DINT REAL structure Type de chaînes	Étiquette de tableau	Premier élément du tableau sur lequel doit opérer cette instruction. Les étiquettes qui ne sont pas un tableau ne sont pas acceptées durant la vérification.	
Dimension to Vary	DINT	immédiate (0, 1, 2)	Dimension à utiliser :	
			<b>Pour la taille de :</b> <b>Entrer :</b>	
			première dimension	0
			deuxième dimension	1
troisième dimension	2			
Size	SINT INT DINT REAL	étiquette	Étiquette de mémorisation du nombre d'éléments dans la dimension spécifiée du tableau	

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à *Indexer via les tableaux* pour consulter les défauts d'indexation par tableau.

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon.
Condition d'entrée d'échelon est vraie	Définir la condition de sortie d'échelon sur la condition d'entrée d'échelon. L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

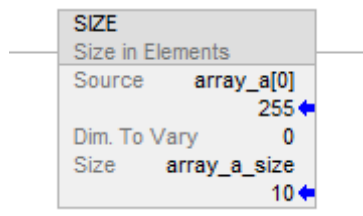
Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

**Exemples**

**Exemple 1**

Trouver le nombre d'éléments dans la dimension 0 (première dimension) d'array\_a. Mémoriser la taille dans array\_a\_size. Dans cet exemple, la dimension 0 d'array\_a a 10 éléments.

**Diagramme à relais**



**Texte structuré**

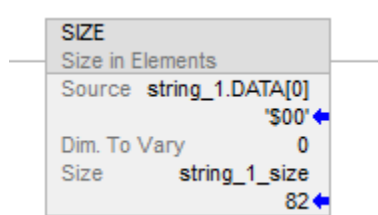
```
SIZE(array_a,0,array_a_size);
```

**Exemple 2**

Trouver le nombre d'éléments dans le membre DATA de string\_1, qui est une chaîne. Mémorise la taille dans string\_1\_size.

Dans cet exemple, le membre DATA de string\_1 a 82 éléments. La chaîne utilise le type de données par défaut STRING. Étant donné que chaque élément contient un caractère, string\_1 peut contenir un maximum de 82 caractères.

**Diagramme à relais**





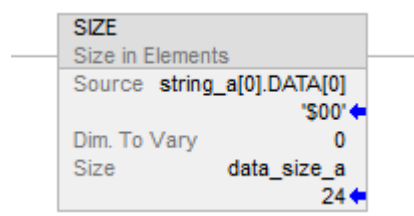
**Texte structuré**

```
SIZE(string_1.DATA[0],0,string_1_size);
```

**Exemple 3**

String\_a est un tableau de structures à chaînes. L'instruction SIZE trouve le nombre d'éléments dans le membre DATA de la structure à chaînes et mémorise la taille dans data\_size\_a.

Dans cet exemple, le membre DATA a 24 éléments. La structure à chaînes a une longueur spécifiée par l'utilisateur de 24.

**Diagramme à relais****Texte structuré**

```
SIZE(string_a.[0].DATA[0],0,data_size_a);
```

**Voir aussi**

[Instructions File/Misc](#) sur la [page 499](#)

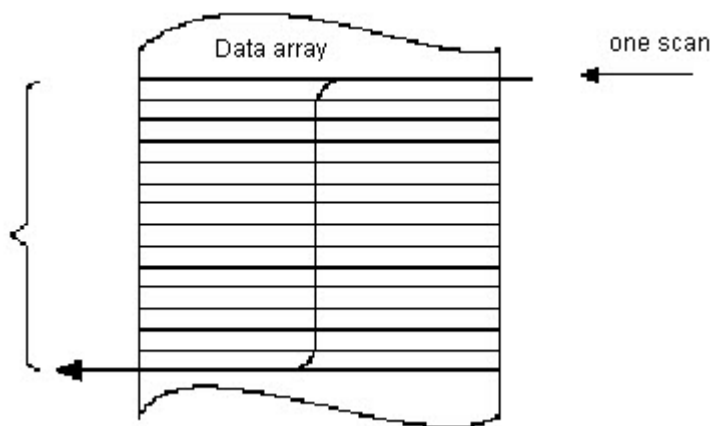
[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

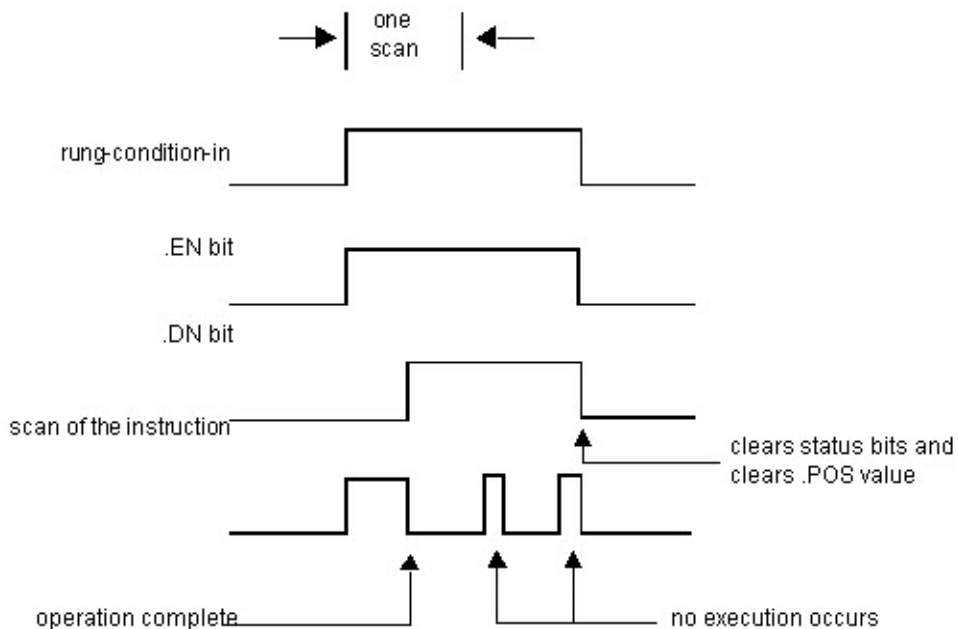
[Syntaxe du texte structuré](#) sur la [page 917](#)

## Mode Tous

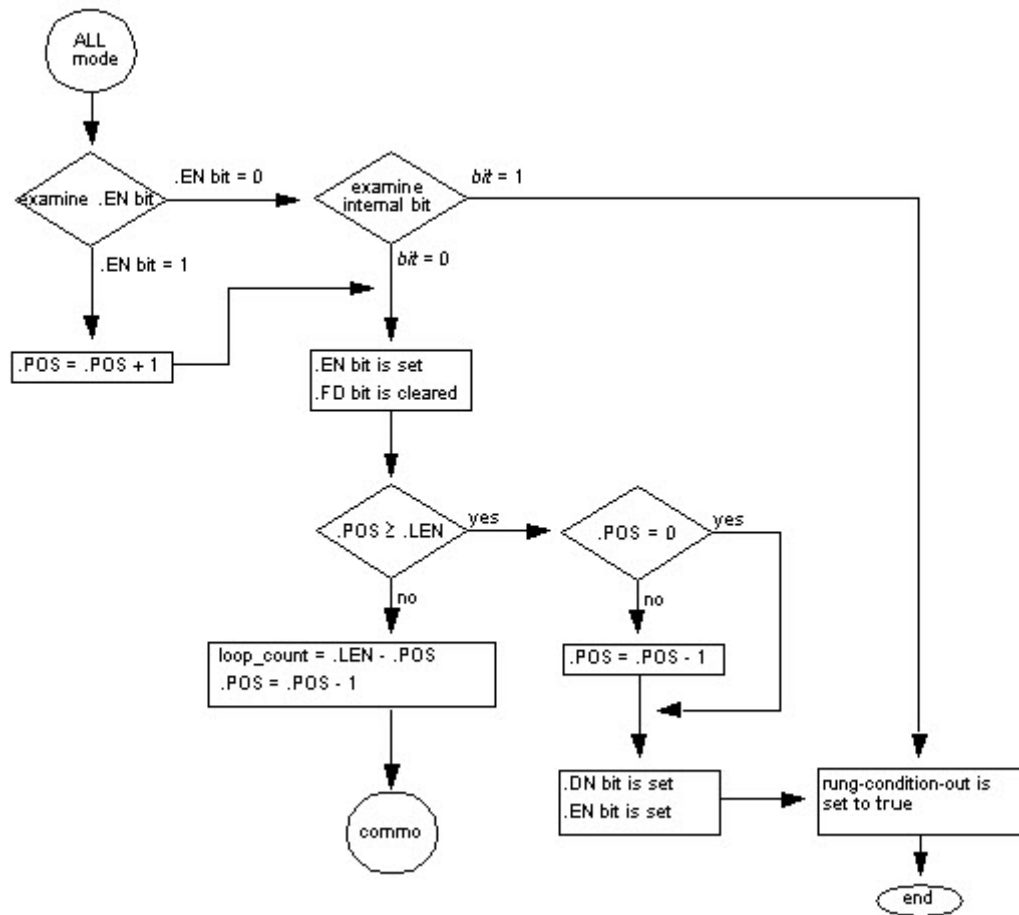
Dans le mode Tous, tous les éléments spécifiés dans le tableau sont exploités avant de passer à l'instruction suivante. Cette opération commence lorsque la condition d'entrée d'échelon de l'instruction passe de faux à vrai. La valeur de la position (.POS) de la structure de contrôle indique l'élément du tableau qu'utilise en ce moment cette instruction. Cette opération s'arrête lorsque la valeur .POS est égale à la valeur .LEN.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est défini. Le bit .DN, le bit .EN et la valeur .POS sont mis à zéro lorsque la condition d'entrée d'échelon est fausse. Ce n'est qu'à ce moment-là qu'une autre exécution de cette instruction peut être déclenchée par une transition de condition d'entrée d'échelon de faux à vrai.



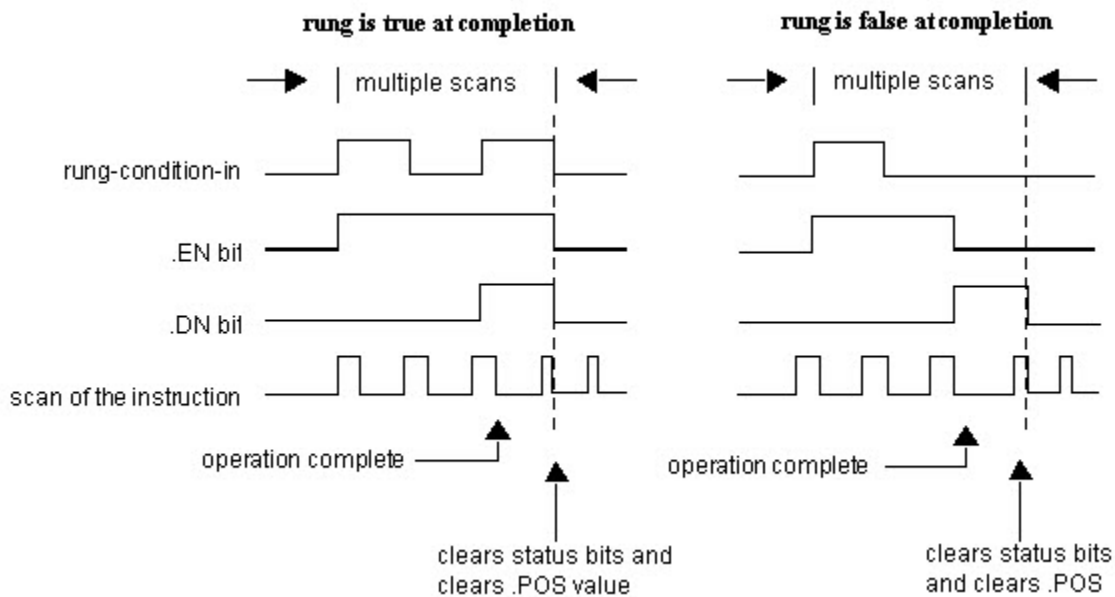
## Organigramme du mode Tous (FSC)



## Mode Numérique

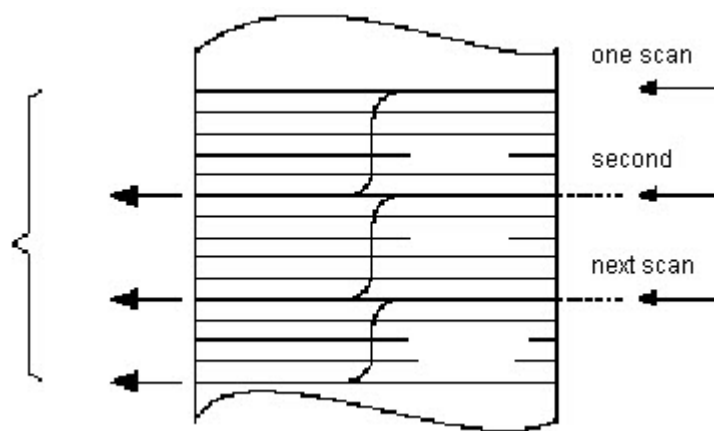
Le mode Numérique répartit l'opération de tableau sur un certain nombre de scrutations. Ce mode est utile lorsqu'on travaille avec des données qui ne sont pas critiques sur le plan temporel ou avec des volumes importants de données. Vous entrez le nombre d'éléments sur lesquels va agir chaque scrutation, ce qui permet de réduire la durée des scrutations.

L'exécution se déclenche lorsque la condition d'entrée d'échelon passe de faux à vrai. Dès que cette instruction se déclenche, elle est exécutée chaque fois qu'elle fait l'objet d'une scrutation et pendant le nombre de scrutations qui s'avèrent nécessaires pour exploiter la totalité du tableau. Dès ce déclenchement, la condition d'entrée d'échelon changent, de manière répétée, sans interrompre l'exécution de l'instruction.



Éviter l'utilisation des résultats d'une instruction de fichier opérant dans le mode Numérique tant que le bit .DN n'est pas défini.

Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. Lorsque l'exécution de l'instruction est terminée, le bit .DN est défini.

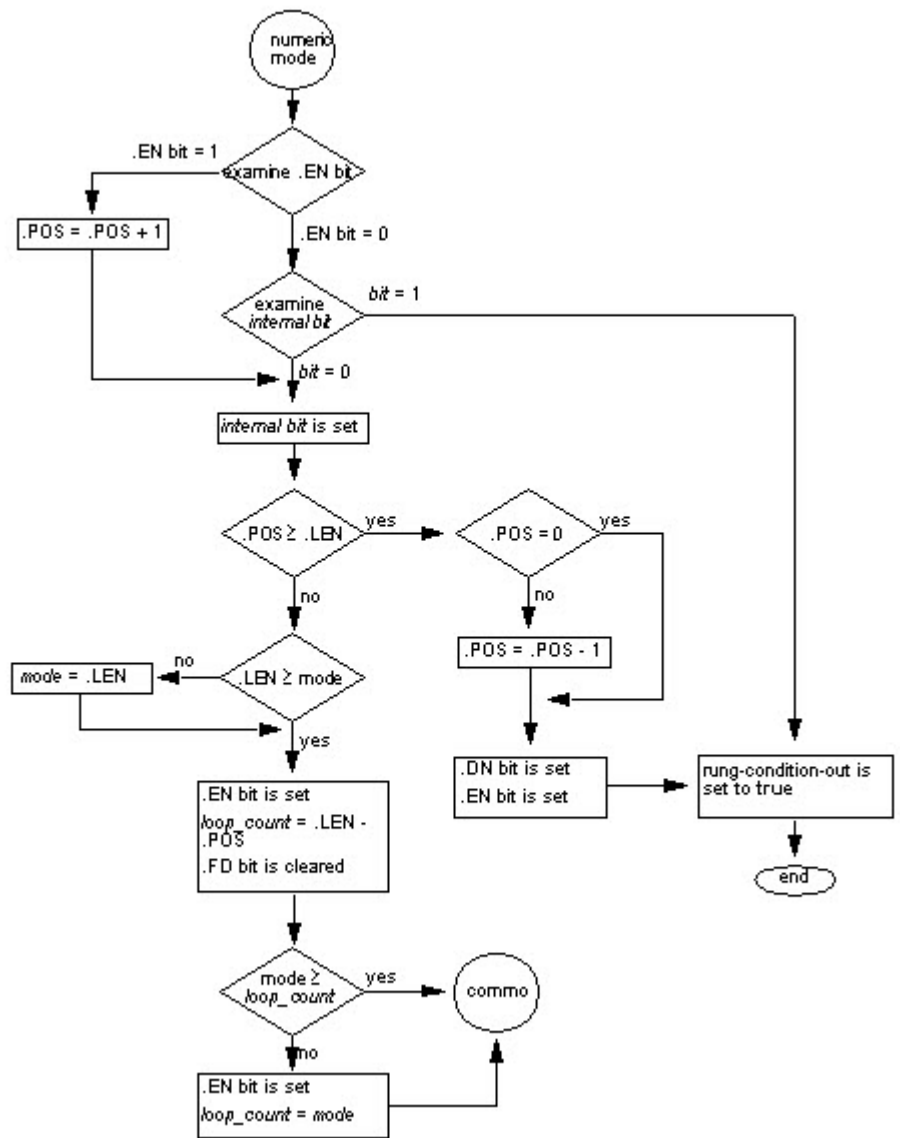


Si la condition d'entrée d'échelon est vraie, une fois cette exécution terminée, les bits .EN et .DN sont définis jusqu'à ce que la condition d'entrée d'échelon devienne

fausse. Lorsque la condition d'entrée d'échelon devient fausse, ces bits sont mis à zéro ainsi que la valeur .POS.

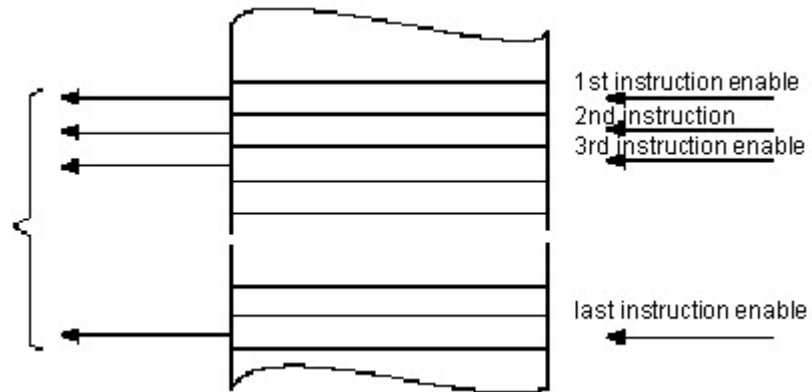
Si, une fois cette exécution terminée, la condition d'entrée d'échelon est fausse, le bit .EN est immédiatement mis à zéro. Une scrutation après l'effacement du bit .EN, le bit .DN et la valeur .POS sont eux aussi mis à zéro.

### Organigramme du mode Numérique (FSC)

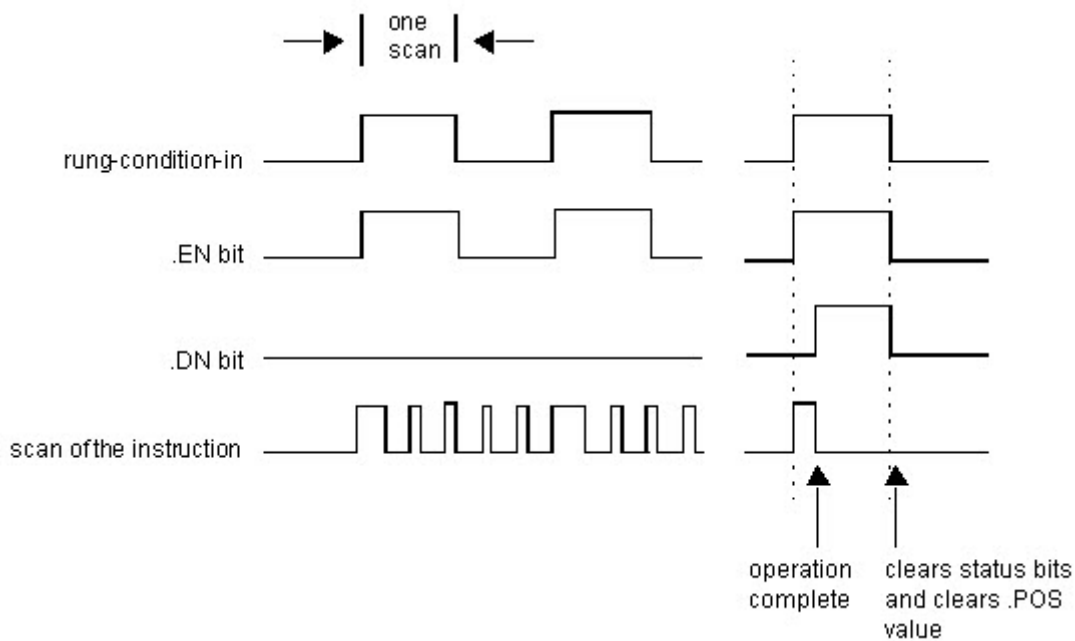


## Mode Incrémentiel

Le mode Incrémentiel manipule un élément du tableau chaque fois que la condition d'entrée d'échelon de l'instruction passe de faux à vrai.



Le diagramme temporel suivant montre les rapports entre les bits d'état et l'exécution de l'instruction. L'exécution ne se fait que durant une scrutation dont la condition d'entrée d'échelon passe de faux à vrai. Chaque fois que cela se produit, un seul élément du tableau est manipulé. Lorsque la condition d'entrée d'échelon reste vraie pendant plus d'une scrutation, cette instruction n'est exécutée que lors de la première scrutation

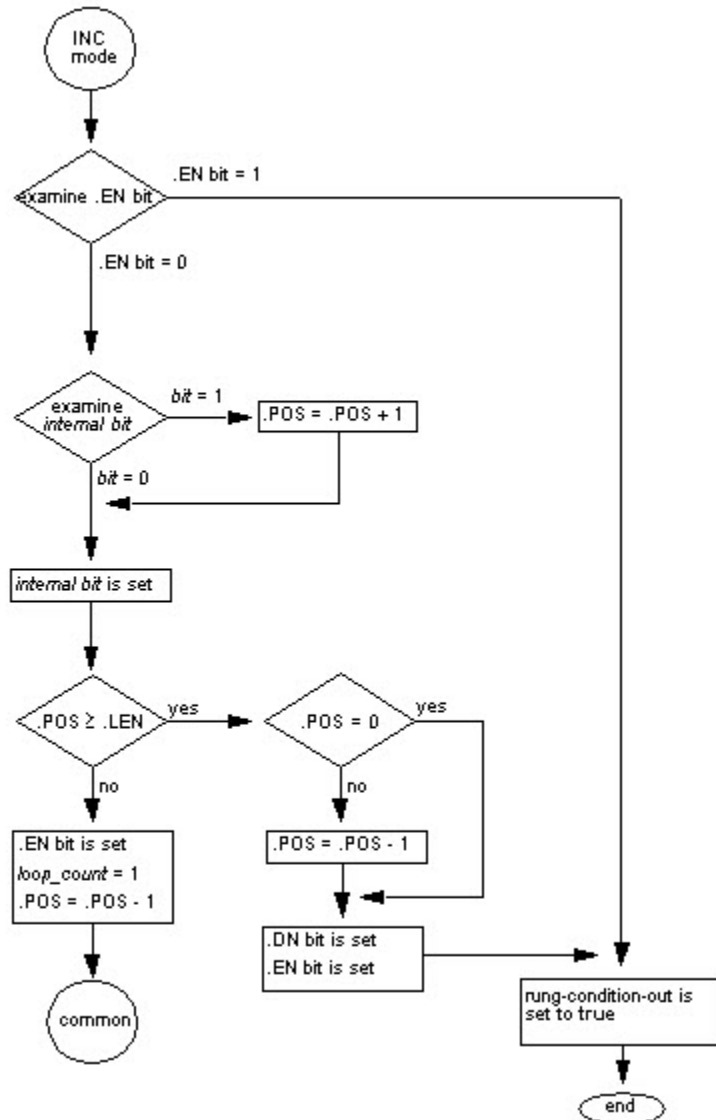


Le bit .EN est défini lorsque la condition d'entrée d'échelon est vraie. Le bit .DN est défini lorsque le dernier élément du tableau a été manipulé. Après la manipulation du dernier élément et après que la condition d'entrée d'échelon soit devenue fausse, le bit .EN, le bit .DN et la valeur de .POS sont mis à zéro.

La différence entre le mode Incrémentiel et le mode Numérique, au rythme d'un élément par scrutation, est la suivante :

- Le mode Numérique, quel que soit le nombre d'éléments par scrutation, n'a besoin que d'une transition de faux à vrai de condition d'entrée d'échelon pour entamer une exécution. Cette instruction continue d'exécuter le nombre spécifié d'éléments lors de chaque scrutation, jusqu'à l'achèvement de cette exécution, quel que soit l'état de condition d'entrée d'échelon.
- Le mode Incrémentiel a besoin du passage de faux à vrai de condition d'entrée d'échelon avant de pouvoir manipuler un élément du tableau.

## Organigramme du mode Incrémentiel (FSC)



## Étiquette de tableau

Lors de l'entrée dans une étiquette de tableau, il convient de spécifier le premier élément du tableau à manipuler. Il ne faut pas utiliser CONTROL.POS pour identifier l'élément initial parce que l'instruction modifie la valeur .POS pendant son fonctionnement, ce qui risque de donner un résultat corrompu.

## Écart standard

L'écart standard se calcule en fonction de cette formule :

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^N [(X_{(start+i)} - AVE)^2]}{(N-1)}}$$

Où :

- start = indice dimension-to-vary de l'opérande du tableau
- xi = élément variable du tableau
- N = nombre d'éléments spécifiés dans le tableau

$$\bullet \text{ AVE} = \frac{\sum_{i=1}^N x_{(start+i)}}{N}$$



## Instructions de décalage/tableau (fichier)

### Instructions de décalage/tableau (fichier)

Utiliser les instructions de tableau (fichier)/décalage pour modifier l'emplacement des données à l'intérieur des tableaux.

#### Instructions disponibles

#### Diagramme à relais

<a href="#">BSL</a>	<a href="#">BS</a> <a href="#">R</a>	<a href="#">FFL</a>	<a href="#">FF</a> <a href="#">U</a>	<a href="#">LFL</a>	<a href="#">LFU</a>
---------------------	---	---------------------	---	---------------------	---------------------

#### Bloc fonctionnel

Indisponible

#### Texte structuré

Indisponible

Si vous voulez :	Utilisez cette instruction :
À raison d'un bit de tableau à la fois, charger des bits dans un tableau de bits, faire passer les bits par ce tableau et décharger les bits de ce tableau.	BSL BSR
Charger et décharger des valeurs, dans le même ordre.	FFL FFU
Charger et décharger des valeurs, dans l'ordre inverse.	LFL LFU

Vous pouvez mélanger des types de données, mais cela peut entraîner une perte de précision et une erreur d'arrondi.

Les types de données en caractères gras correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

**Voir aussi**

[Instructions de conversion ASCII](#) sur la [page 851](#)

[Instructions de port série ASCII](#) sur la [page 831](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

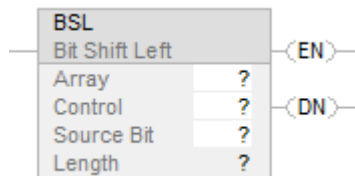
**Décalage binaire à gauche (BSL)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L’instruction BSL déplace d’une position vers la gauche les bits spécifiés dans Array.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n’est pas disponible dans le texte structuré.

**Opérandes**

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Tableau	DINT ARRAY	étiquette	Tableau à modifier spécifier le premier élément où doit commencer ce déplacement
Control	CONTROL	étiquette	Structure de contrôle de l’opération
Source Bit	BOOL	étiquette	Bit à déplacer pour l’amener à la position évacuée.
Longueur (Length)	DINT	immediate	Nombre de bits dans le tableau à déplacer

### Structure CONTROL

Mnémonique	Type de donnée	Description
.EN	BOOL	Le bit d'activation indique que l'instruction BSL a été activée.
.DN	BOOL	Le bit de fin est défini pour indiquer que les bits se sont déplacés d'une position vers la gauche.
.UL	BOOL	Le bit de déchargement est la sortie de cette instruction. Le bit .UL mémorise le statut du bit qui a été déplacé en dehors de la plage de bits.
.ER	BOOL	Le bit d'erreur est défini lorsque .LEN < 0.
.LEN	DINT	La longueur spécifie le nombre de bits de tableau à déplacer.

### Description

En cas d'activation, cette instruction décharge le bit le plus haut des bits spécifiés sur le bit .UL, déplace les bits restants d'une position vers la gauche et charge l'adresse Bit dans le bit 0 de Array.

---

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

L'instruction BSL agit sur une mémoire de données contiguës. L'instruction BSL agit sur une mémoire de données contiguës. Pour les automates CompactLogix 5370 et ControlLogix 5570 uniquement, la portée de cette instruction est limitée par l'étiquette de base. L'instruction BSL ne va pas rédiger des données en dehors de l'étiquette de base mais peut traverser des limites de membres. Lorsque vous spécifiez un tableau qui est membre d'une structure et lorsque la longueur dépasse la taille de ce tableau, vous devez tester et confirmer que l'instruction BSL ne modifie pas des données que vous ne souhaitez pas modifier.

Pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580, les données sont limitées par le membre spécifié.

Dans cette instruction de transition, le logique à relais fait basculer la condition d'entrée d'échelon de faux à vrai afin de permettre l'exécution de cette instruction.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produit si	Type de défaut	Code de défaut
LEN dépasse la taille du tableau.	4	20

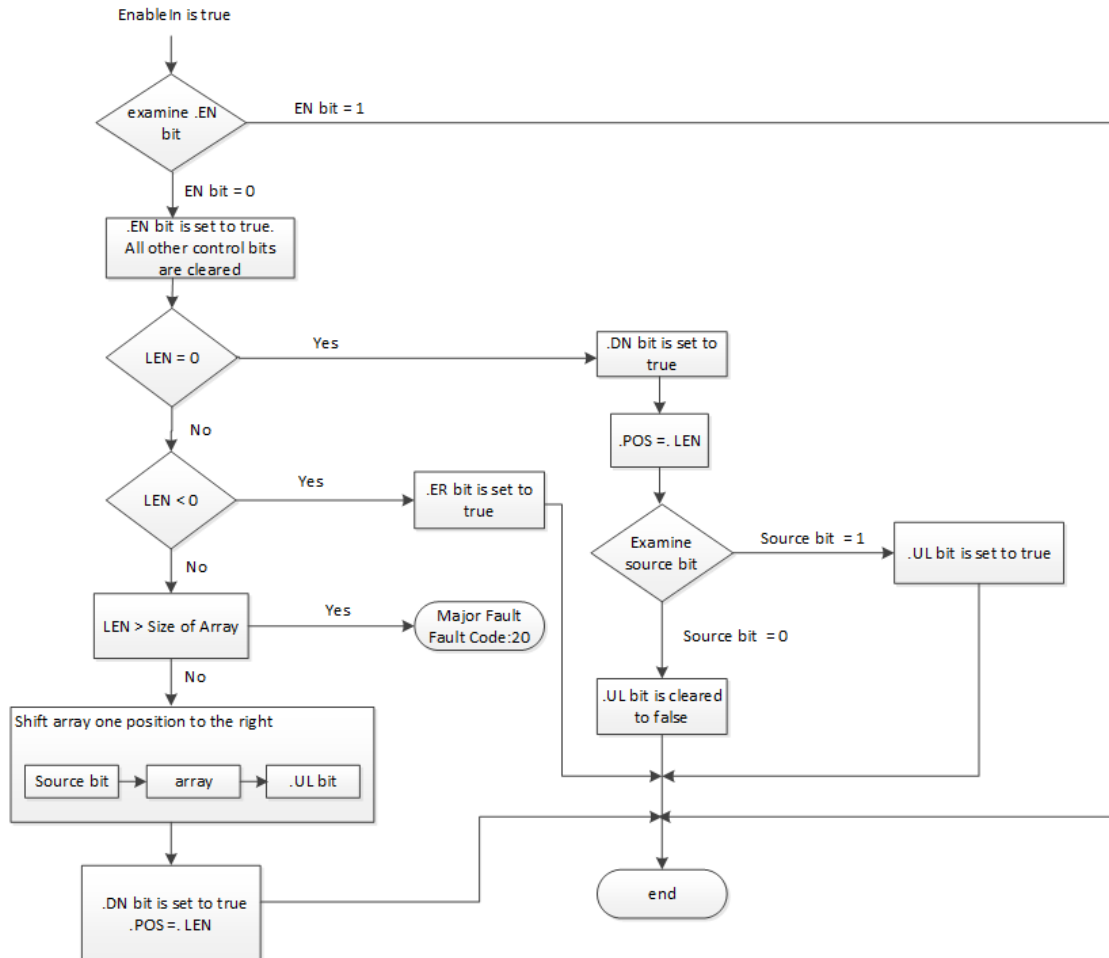
Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit .EN est défini sur faux. Le bit .DN est défini sur faux. Le bit .ER est défini sur faux. La valeur .POS est mise à zéro
Condition d'entrée d'échelon est fausse	Le bit .EN est défini sur faux. Le bit .DN est défini sur faux. Le bit .ER est défini sur faux. La valeur .POS est mise à zéro.
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme BSL (vrai)
Post-scrutation	N/A

**Organigramme BSL (vrai)**

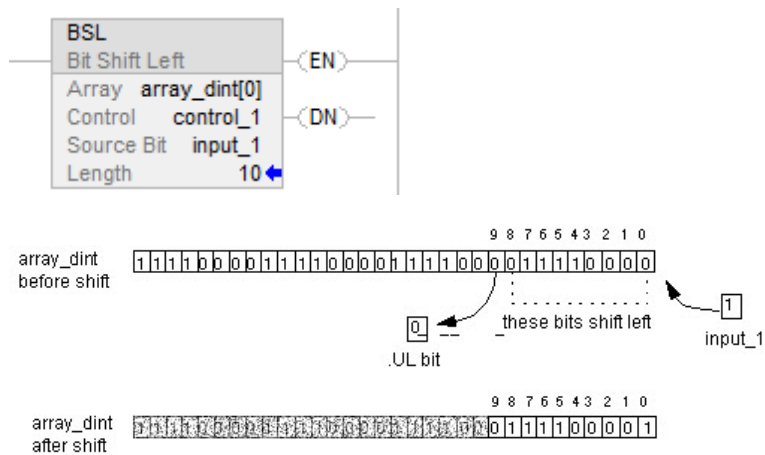


**Exemples**

**Exemple 1**

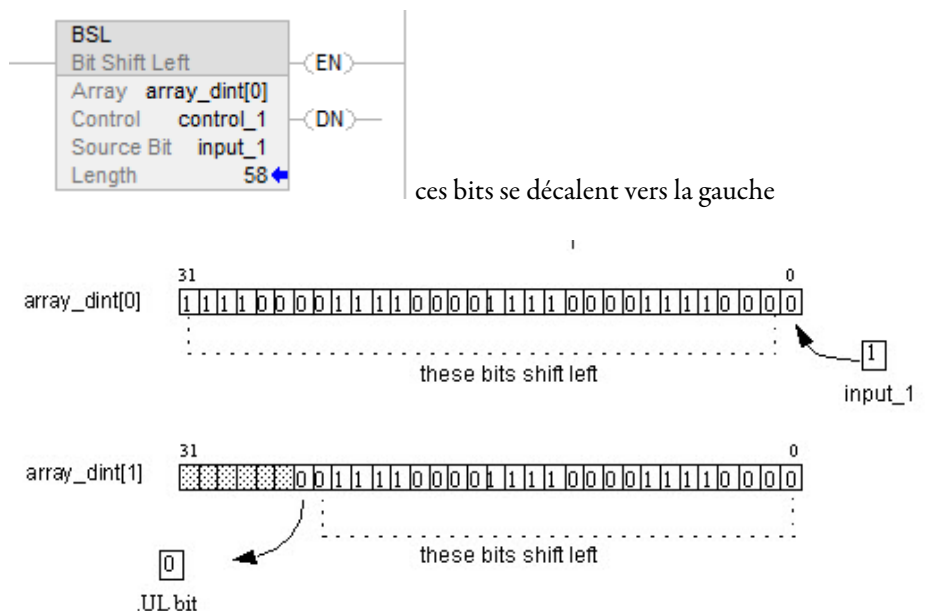
En cas d'activation, l'instruction BSL commence au bit 0 d'array\_dint[0]. Cette instruction décharge array\_dint[0].9 dans le bit .UL, déplace les bits restants et charge input\_1 dans array\_dint[0].0. Les bits restants (10-31) ne sont pas valides.

**Diagramme à relais**



**Exemple 2 :**

En cas d'activation, l'instruction BSL commence au bit 0 d'array\_dint[0]. Cette instruction décharge array\_dint[1].25 dans le bit .UL, déplace les bits restants et charge input\_1 dans array\_dint[0].0. Les bits restants (31-26 dans array\_dint[1]) ne sont pas valides.



**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

**Décalage binaire à droite (BSR)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction BSR déplace d'une position vers la droite les bits spécifiés dans Array. En cas d'activation, cette instruction décharge la valeur au niveau du bit 0 du Tableau sur le bit .UL, déplace les bits restants d'une position vers la droite et charge le bit provenant de l'adresse du Bit.

---

**Important :** Tester et confirmer que cette instruction a modifié les données correctes.

L'instruction BSR agit sur une mémoire continue. Si Array est un tableau à membres, l'instruction peut dépasser la limite du tableau et pénétrer dans d'autres membres qui y font suite. Il faut faire attention et sélectionner une longueur qui ne risque pas de provoquer ce scénario.

---

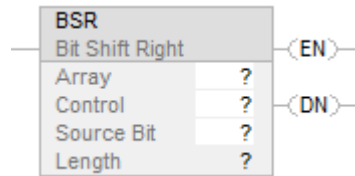
L'instruction BSR agit sur une mémoire de données contiguës. Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570 uniquement, la portée de cette instruction est limitée par l'étiquette de base. L'instruction BSL ne va pas rédiger des données en dehors de l'étiquette de base mais peut traverser des limites de membres. Lorsque vous spécifiez un tableau qui est membre d'une structure et lorsque la longueur dépasse la taille de ce tableau, vous devez tester et confirmer l'instruction BSL pour vous assurer qu'elle modifie bien les bonnes données.

Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570, les données sont limitées par le membre spécifié.

Si cette instruction essaye de lire au-delà de la fin d'un tableau (la longueur LEN est excessive), cette instruction définit le bit .ER et déclenche un défaut majeur.

**Langages disponibles**

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Diagramme à relais

Opérande	Type de données	Format	Description
Array	DINT ARRAY	étiquette	Tableau à modifier spécifier le premier élément à déplacer.
Control	CONTROL	étiquette	Structure de contrôle de l'opération
Source Bit	BOOL	étiquette	Bit à charger dans la position évacuée.
Length	DINT	immédiate	Nombre de bits dans le tableau à déplacer

### Structure CONTROL

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction BSR a été activée.
.DN	BOOL	Le bit de fin est défini pour indiquer que les bits se sont déplacés d'une position vers la droite.
.UL	BOOL	Le bit de déchargement est la sortie de cette instruction. Le bit .UL mémorise le statut du bit qui a été déplacé en dehors de la plage de bits.
.ER	BOOL	Le bit d'erreur est défini lorsque .LEN < 0.
.LEN	DINT	La longueur spécifie le nombre de bits de tableau à déplacer.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

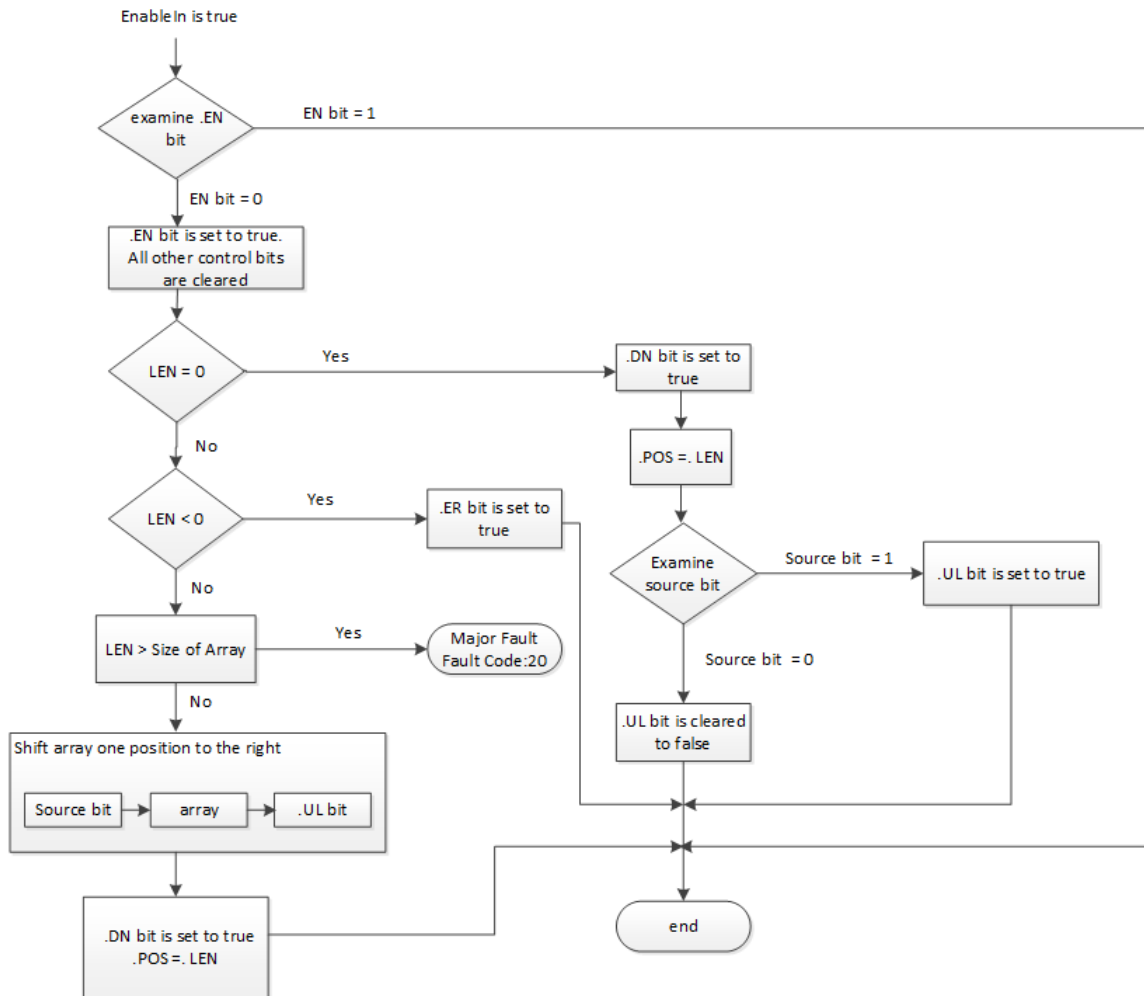
Aucun spécifique à cette instruction. Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Le bit .EN est défini sur faux. Le bit .DN est défini sur faux. Le bit .ER est défini sur faux. La valeur .POS est mise à zéro.
Condition d'entrée d'échelon est fausse	Le bit .EN est défini sur faux. Le bit .DN est défini sur faux. Le bit .ER est défini sur faux. La valeur .POS est mise à zéro.
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme BSR suivant (vrai)
Post-scrutation	N/A



**Organigramme BSR (vrai)**

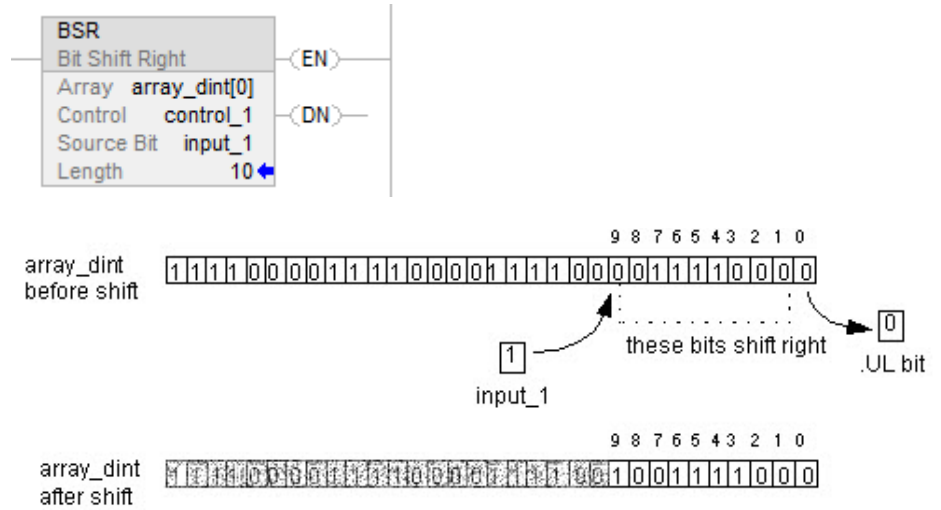


**Exemples**

**Exemple 1**

Lorsqu'elle est activée, cette instruction BSR copie array\_dint[0].0 sur le bit .UL, déplace les bits 0-9 vers la droite et charge input\_1 dans array\_dint[0].9. Les bits restants (10-31) ne sont pas valides, ce qui signifie que ces bits ne peuvent pas être modifiés.

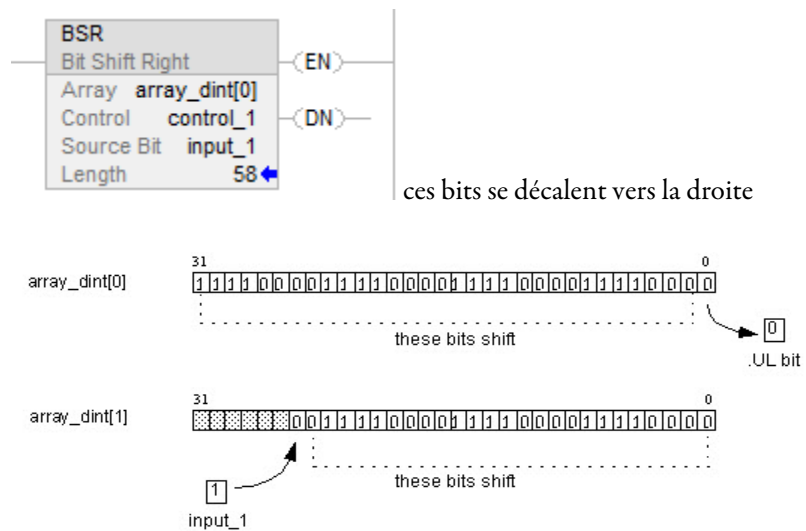
**Diagramme à relais**



**Exemple 2**

Lorsqu'elle est activée, cette instruction BSR copie array\_dint[0].0 sur le bit .UL, déplace les bits 0-9 vers la droite et charge input\_1 dans array\_dint[1].25. Les bits restants (31-26 dans dint\_array[1]) ne sont pas valides, ce qui signifie que ces bits ne peuvent pas être modifiés. Noter le déplacement d'array\_dint[1].0 qui traverse des mots puis pénètre dans array\_dint[0].31.

**Diagramme à relais**



**Voir aussi**

[Indexer via des tableaux](#) sur la [page 899](#)

[Conversions de données](#) sur la [page 889](#)

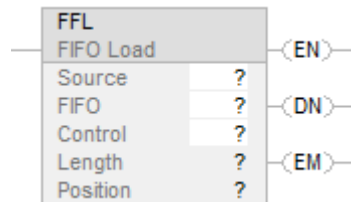
## Chargement de FIFO (FFL)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction FFL copie la valeur Source sur FIFO.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Une conversion n'a lieu que si le type d'opérande source ne correspond pas au type de FIFO.

### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	SINT INT DINT REAL Type de chaînes structure	immediate étiquette	Données à conserver dans FIFO

FIFO	SINT INT DINT REAL Type de chaînes structure	étiquette de tableau	FIFO à modifier Spécifier le premier élément de FIFO
Control	CONTROL	étiquette	Structure de contrôle de l'opération En général, utiliser le même CONTROL que FFU associée
Longueur (Length)	DINT	immediate	Nombre maximal d'éléments que FIFO peut contenir à un moment donné
Position (Position)	DINT	immediate	Emplacement suivant dans FIFO où l'instruction charge des données valeur initiale typique : 0

**Structure CONTROL**

Mnémonique	Type de donnée	Description
.EN	BOOL	Le bit d'activation indique que l'instruction FFL a été activée.
.DN	BOOL	Le bit de fin est défini pour indiquer que FIFO est plein. Le bit .DN empêche le chargement de FIFO jusqu'à ce que .POS < .LEN.
.EM	BOOL	Le bit vide indique que FIFO est vide. Si .LEN est < ou = à 0 ou si .POS < 0, le bit .EM et le bit .DN sont définis.
.LEN	DINT	Le mot de longueur spécifie le nombre maximal d'éléments dans FIFO.
.POS	DINT	Le mot de position identifie l'emplacement dans FIFO où l'instruction charge la valeur suivante.

**Description**

Utiliser l'instruction FFL avec l'instruction FFU pour mémoriser et recouvrer des données dans l'ordre premier entré/premier sorti. Lors d'une utilisation en paires, ces instructions FFL et FFU établissent un registre asynchrone de décalage.

En général, Source et FIFO sont du même type de données.

Lorsqu'elle est activée, l'instruction FFL charge la valeur Source dans la position du FIFO identifiée par la valeur .POS. Cette instruction charge une valeur lors de chaque activation de l'instruction, jusqu'à ce que FIFO soit plein.

---

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

L'instruction FFL agit sur une mémoire contiguë. L'instruction BSL agit sur une mémoire de données contiguës. Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570 uniquement, la portée de cette instruction est limitée par l'étiquette de base. L'instruction BSL ne va pas rédiger des données en dehors de l'étiquette de base mais peut traverser des limites de membres. Lorsque vous spécifiez un tableau qui est membre d'une structure et lorsque la longueur dépasse la taille de ce tableau, vous devez tester l'instruction BSL pour confirmer qu'elle ne modifie pas des données que vous ne souhaitez pas modifier.

Pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580, les données sont limitées par le membre spécifié.

Si cette instruction essaye de lire au-delà de la fin d'un tableau, cette instruction déclenche un défaut majeur.

En général, Source et FIFO sont du même type de données. Si les types de données Source et FIFO ne correspondent pas, cette instruction convertit la valeur Source au type de données de l'étiquette FIFO.

Un petit nombre entier est converti en un gros nombre entier par une extension de signe.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
L'(élément de départ + .POS) dépasse la fin du tableau FIFO	4	20

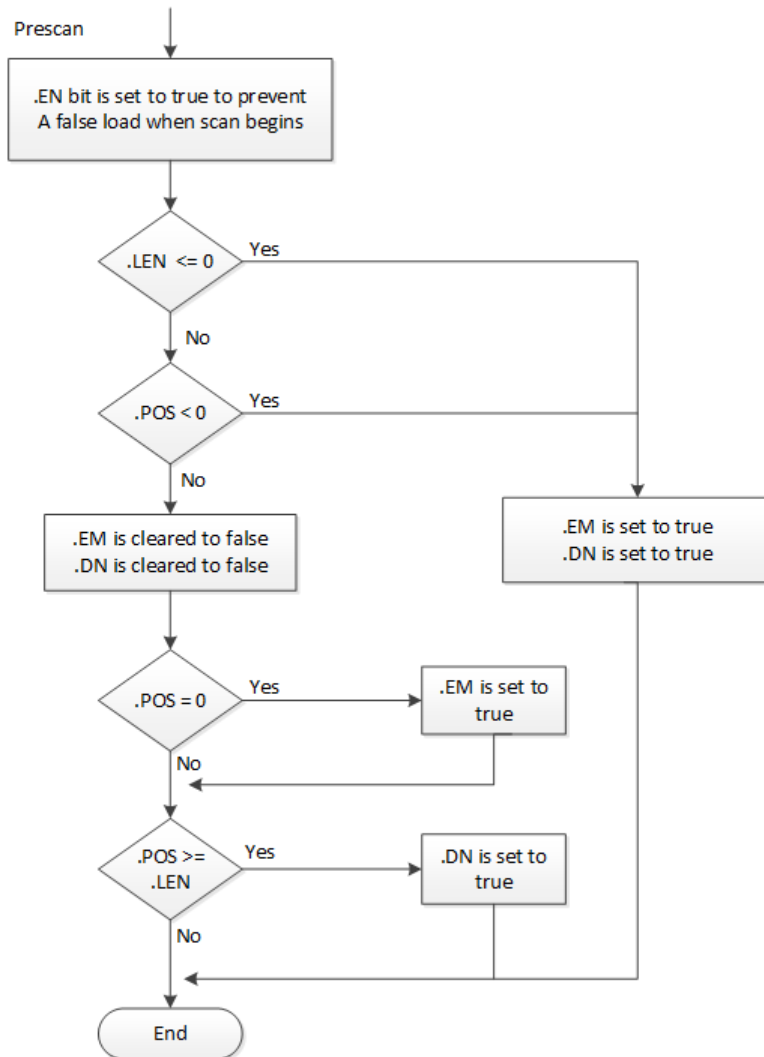
Reportez-vous à Attributs communs pour les défauts liés aux opérandes.

Exécution

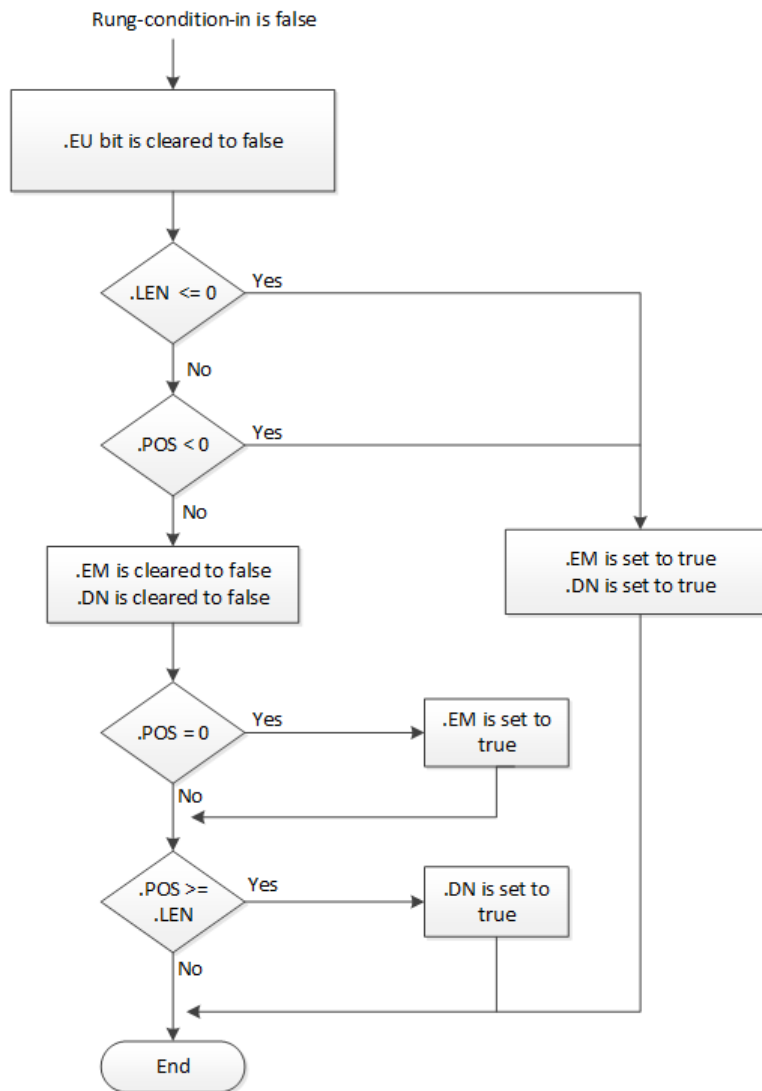
Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l'organigramme FFL (pré-scrutation)
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme FFL (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme FFL (vrai)
Post-scrutation	N/A

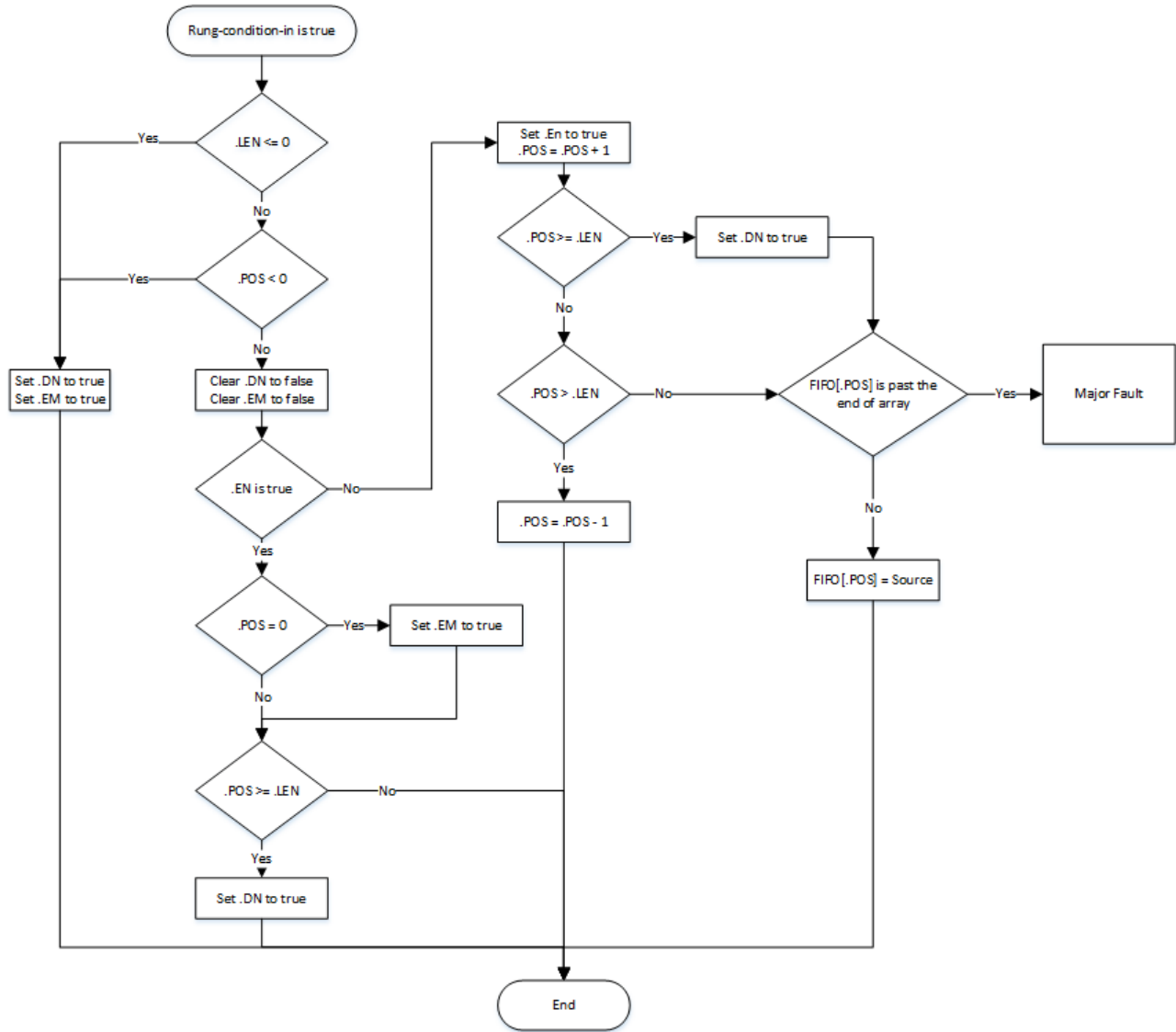
Organigramme FFL (pré-scrutation)



Organigramme FFL (faux)



Organigramme FFL (vrai)

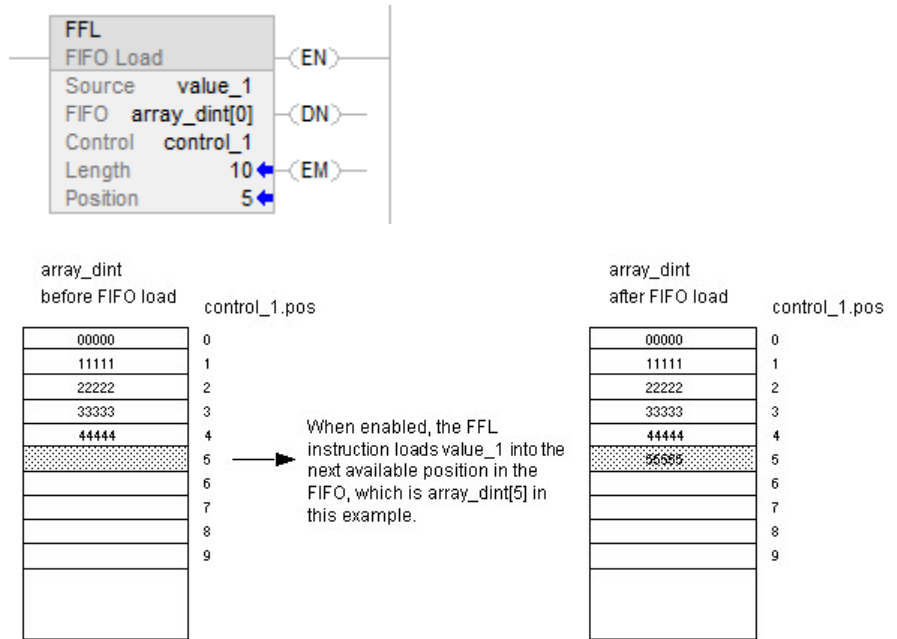




**Exemples**

**Exemple 1**

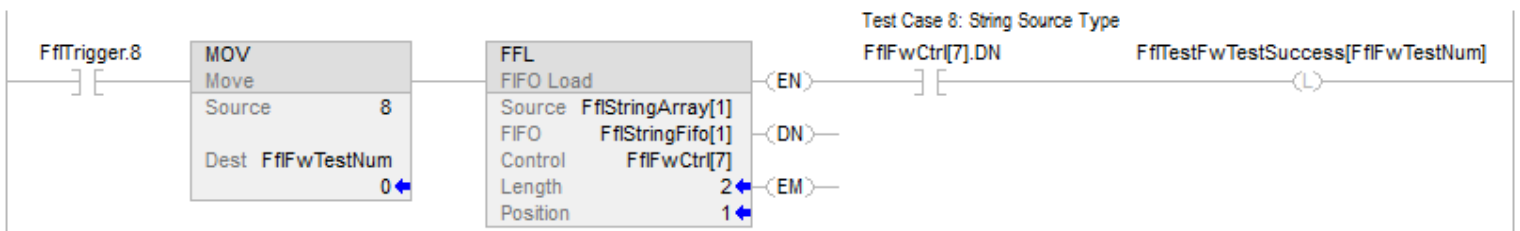
**Diagramme à relais**



**Exemple 2**

Le tableau Source est le tableau STRING ou le tableau Structure.

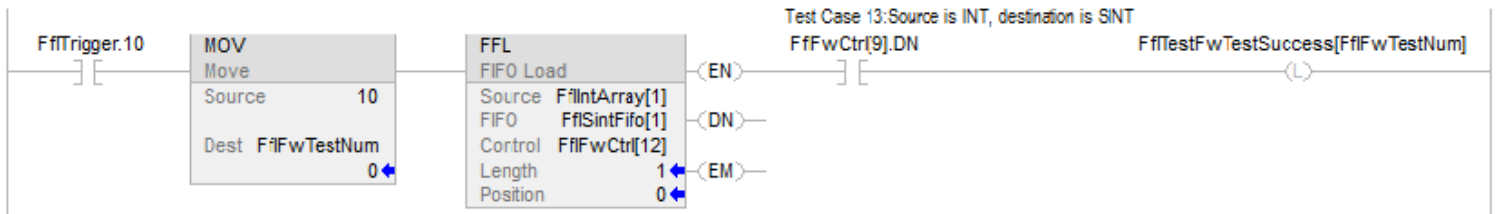
**Diagramme à relais**



**Exemple 3**

Le type de données de source ne correspond pas au type de données du tableau FIFO.

### Diagramme à relais



### Voir aussi

[Instructions de décalage/tableau \(fichier\)](#) sur la [page 569](#)

[Déchargement de FIFO \(FFU\)](#) sur la [page 586](#)

[Chargement de LIFO \(LFL\)](#) sur la [page 593](#)

[Attributs communs](#) sur la [page 885](#)

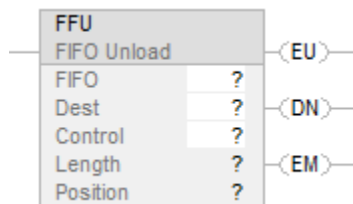
## Déchargement de FIFO (FFU)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction FFU décharge la valeur depuis la position 0 (première position) du FIFO et mémorise cette valeur dans Destination. Les données restantes du FIFO descendent d'une position.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction.

## Diagramme à relais

Opérande	Type	Format	Description
FIFO	SINT INT DINT REAL Type de chaînes structure	étiquette de tableau	FIFO à modifier Spécifier le premier élément de FIFO Ne pas utiliser CONTROL.POS dans cet indice
Destination	SINT INT DINT REAL Type de chaînes structure	étiquette	Valeur déchargée depuis FIFO.
Control	CONTROL	étiquette	Structure de contrôle de l'opération en général, utiliser le même contrôle (CONTROL) que FFL associée
Length	DINT	immédiate	Nombre maximal d'éléments que FIFO peut contenir à un moment donné
Position	DINT	immédiate	Emplacement suivant dans FIFO où l'instruction charge des données valeur initiale typique : 0

## Structure CONTROL

Mnémonique	Type de données	Description
.EU	BOOL	Le bit d'activation déchargée indique que l'instruction FFU a été activée. Le bit .EU est défini pour empêcher un déchargement faux lorsque la pré-scrutation commence.
.DN	BOOL	Le bit de fin est défini pour indiquer que FIFO est plein (.POS = .LEN).
.EM	BOOL	Le bit vide indique que FIFO est vide. Si .LEN est , ou = à 0 ou si .POS < 0, le bit .EM et le bit .DN sont définis.
.LEN	DINT	La longueur spécifie le nombre maximal d'éléments dans FIFO.
.POS	DINT	Cette position identifie la fin des données qui ont été chargées dans FIFO.

### Description

Utiliser l’instruction FFU avec l’instruction FFL pour mémoriser et recouvrer des données dans l’ordre premier entré/premier sorti.

Lorsqu’elle est activée, l’instruction FFU décharge des données depuis le premier élément de FIFO et place cette valeur dans Destination. Cette instruction décharge une valeur lors de chaque activation de l’instruction, jusqu’à ce que FIFO soit vide. Lorsque FIFO est vide, FFU renvoie un 0 à Destination.

En général, Destination et FIFO sont du même type de données. Si ces types sont différents, cette instruction convertit la valeur déchargée au même type que l’étiquette de destination.

Un petit nombre entier est converti en un gros nombre entier par une extension de signe.

### Affecte les indicateurs d’état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
Length spécifiée dépasse la fin du tableau FIFO	4	20

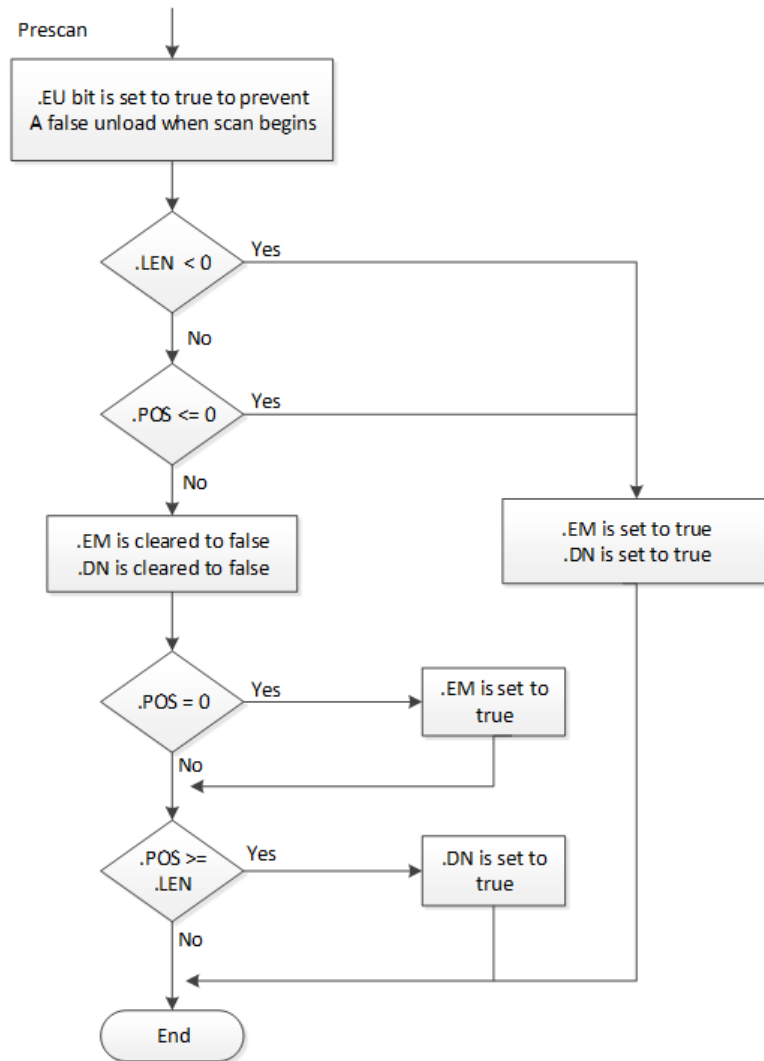
Reportez-vous à la section Attributs communs pour plus d’informations sur les défauts liés aux opérandes.

### Exécution

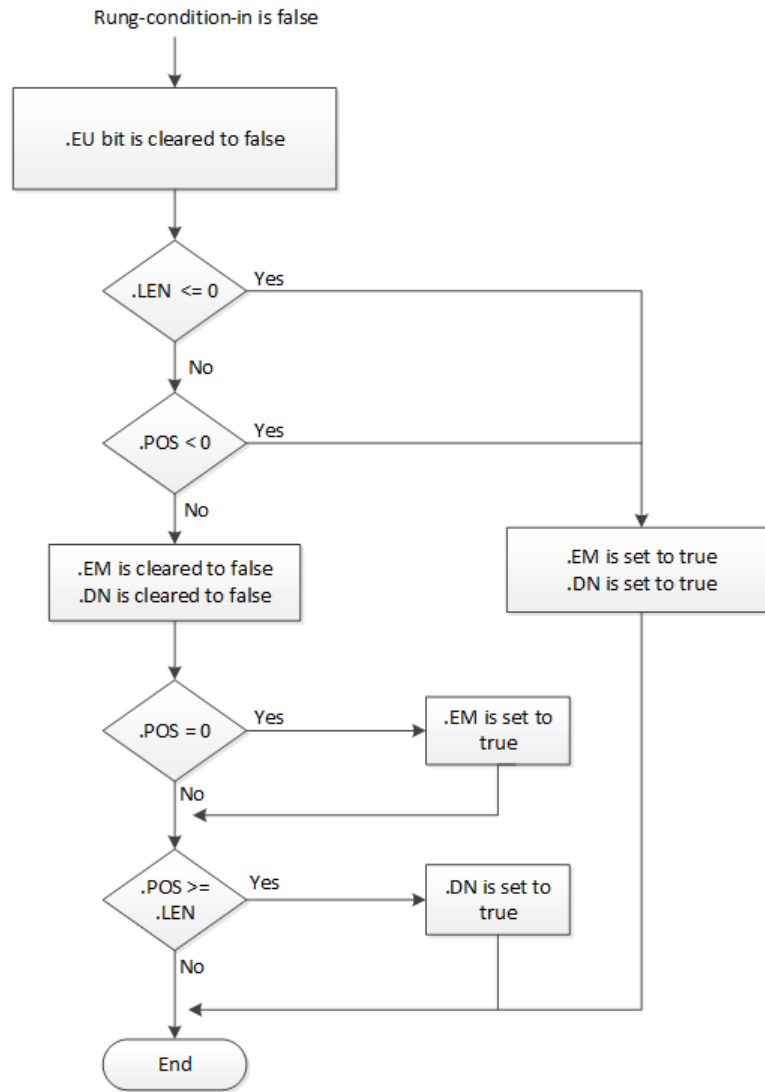
#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l’organigramme FFU (pré-scrutation)
Condition d’entrée d’échelon est fausse	Reportez-vous à l’organigramme FFL (faux)
Condition d’entrée d’échelon est vraie	Reportez-vous à l’organigramme FFU (vrai)
Post-scrutation	N/A

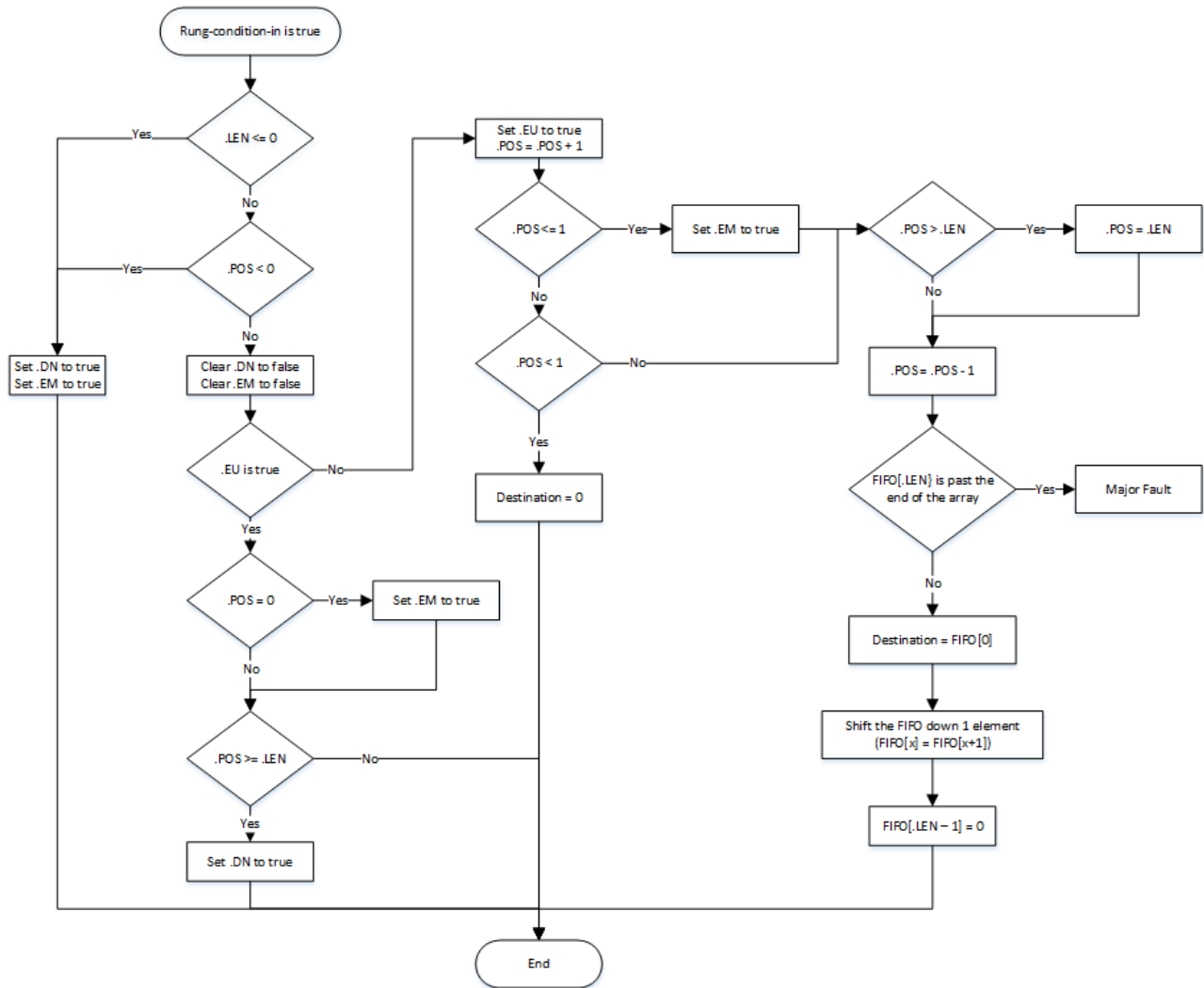
Organigramme FFU (pré-scrutation)



### Organigramme FFL (faux)



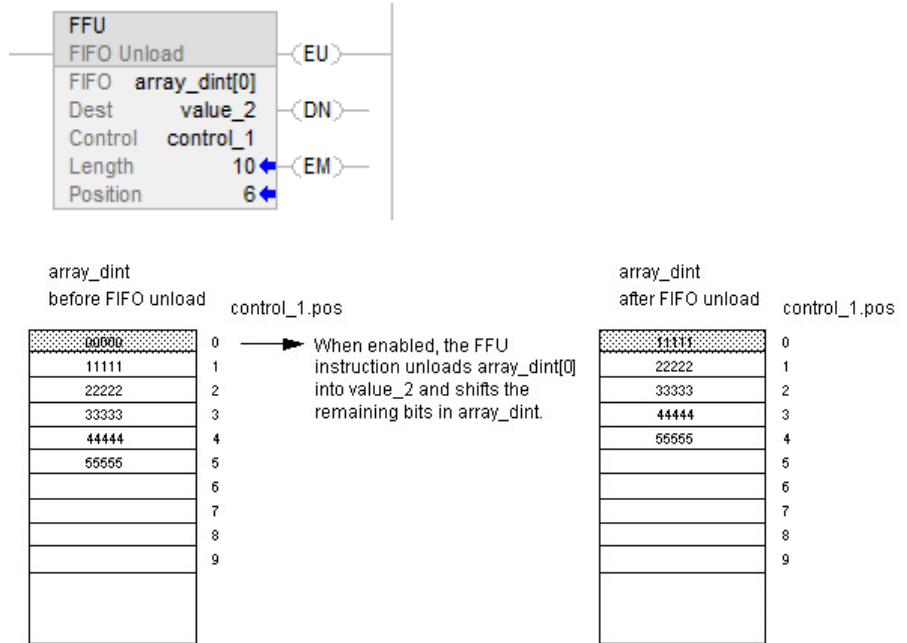
Organigramme FFU (vrai)



Exemples

Exemple 1

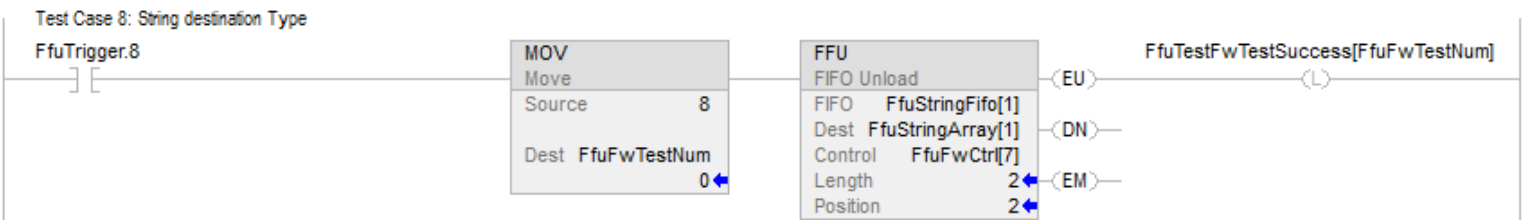
Diagramme à relais



Exemple 2

Le tableau de destination est le tableau STRING ou le tableau Structure.

Diagramme à relais

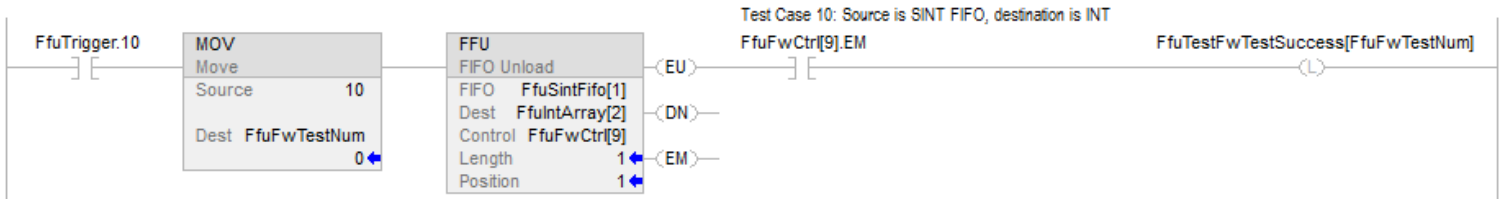


Exemple 3

Le type de données du tableau de source FIFO ne correspond pas au type de données du tableau de destination.



**Diagramme à relais**



**Voir aussi**

[Tableau \(Fichier\)/Instructions de décalage](#) sur la [page 569](#)

[Attributs communs](#) sur la [page 885](#)

[FFL](#) sur la [page 579](#)

[LFL](#) sur la [page 593](#)

[LFU](#) sur la [page 600](#)

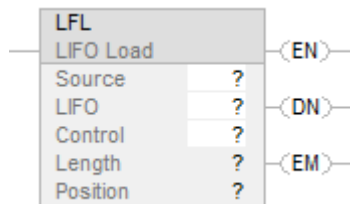
**Chargement de LIFO (LFL)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L’instruction LFL copie la valeur Source sur LIFO.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n’est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction.

### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT REAL Type de chaînes structure	immédiate étiquette	Données à conserver dans LIFO.
LIFO	SINT INT DINT REAL Type de chaînes structure	étiquette de tableau	LIFO à modifier spécifier le premier élément de LIFO
Control	CONTROL	étiquette	Structure de contrôle de l'opération en général, utiliser le même contrôle (CONTROL) que LFO associée
Length	DINT	immédiate	Nombre maximal d'éléments que LIFO peut contenir à un moment donné
Position	DINT	immédiate	Emplacement suivant dans LIFO où l'instruction charge des données valeur initiale typique : 0

### Structure CONTROL

Mnémorique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction LFL a été activée.
.DN	BOOL	Le bit de fin est défini pour indiquer que LIFO est plein (.POS = .LEN). Le bit .DN empêche le chargement de LIFO jusqu'à ce que .POS < .LEN.

.EM	BOOL	Le bit vide indique que LIFO est vide. Si .LEN est < ou = à 0 ou si .POS < 0, le bit .EM et le bit .DN sont définis.
.LEN	DINT	La longueur spécifie le nombre maximal d'éléments que LIFO peut contenir à un moment donné.
.POS	DINT	La position identifie l'emplacement dans le LIFO où l'instruction charge la valeur suivante.

### Description

Utiliser l'instruction LFL avec l'instruction LFU pour mémoriser et recouvrer des données dans l'ordre dernier entré/premier sorti. Lors d'une utilisation en paires, ces instructions LFL et LFU établissent un registre asynchrone de décalage.

En général, Source et LIFO sont du même type de données.

Lorsqu'elle est activée, l'instruction LFL charge la valeur Source dans la position de LIFO identifiée par la valeur .POS. Cette instruction charge une valeur lors de chaque activation de l'instruction, jusqu'à ce que LIFO soit plein.

---

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

L'instruction LFL agit sur une mémoire de données contiguës. Pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570, la portée de cette instruction est limitée par l'étiquette de base. En général, Source et LIFO sont du même type de données. Si les types de données Source et LIFO ne correspondent pas, cette instruction convertit la valeur Source au type de données de l'étiquette FIFO. Un petit nombre entier est converti en un gros nombre entier par une extension de signe.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
Si (élément de départ + .POS) dépasse la fin du tableau LIFO	4	20

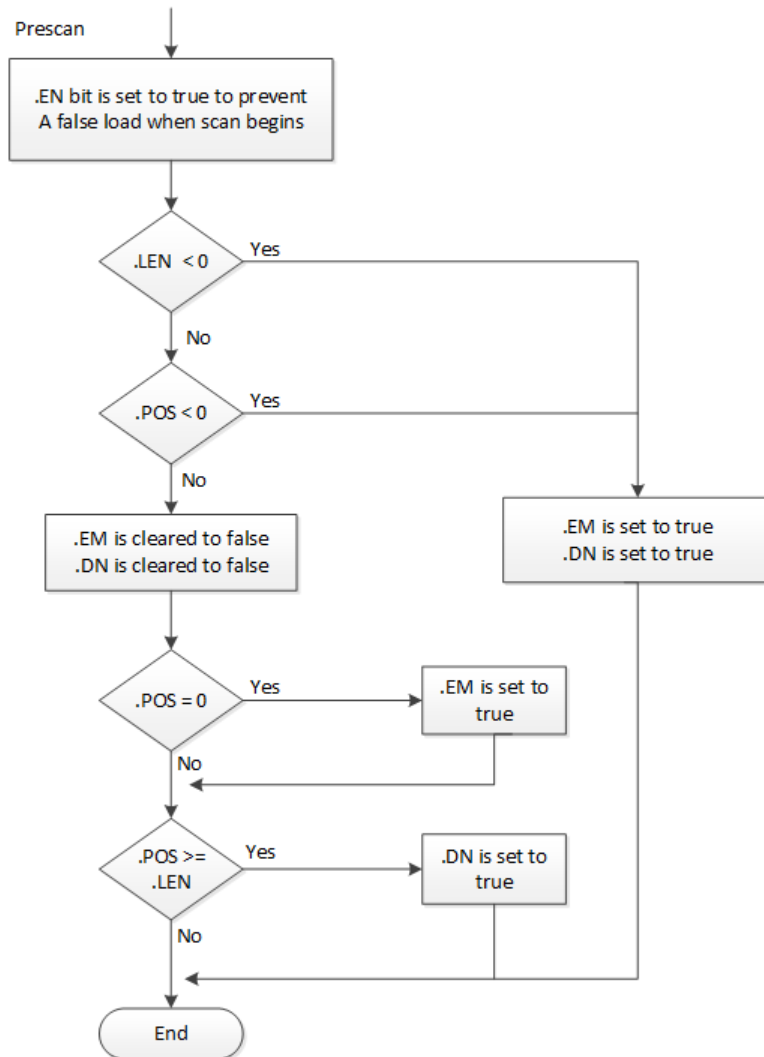
Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

Exécution

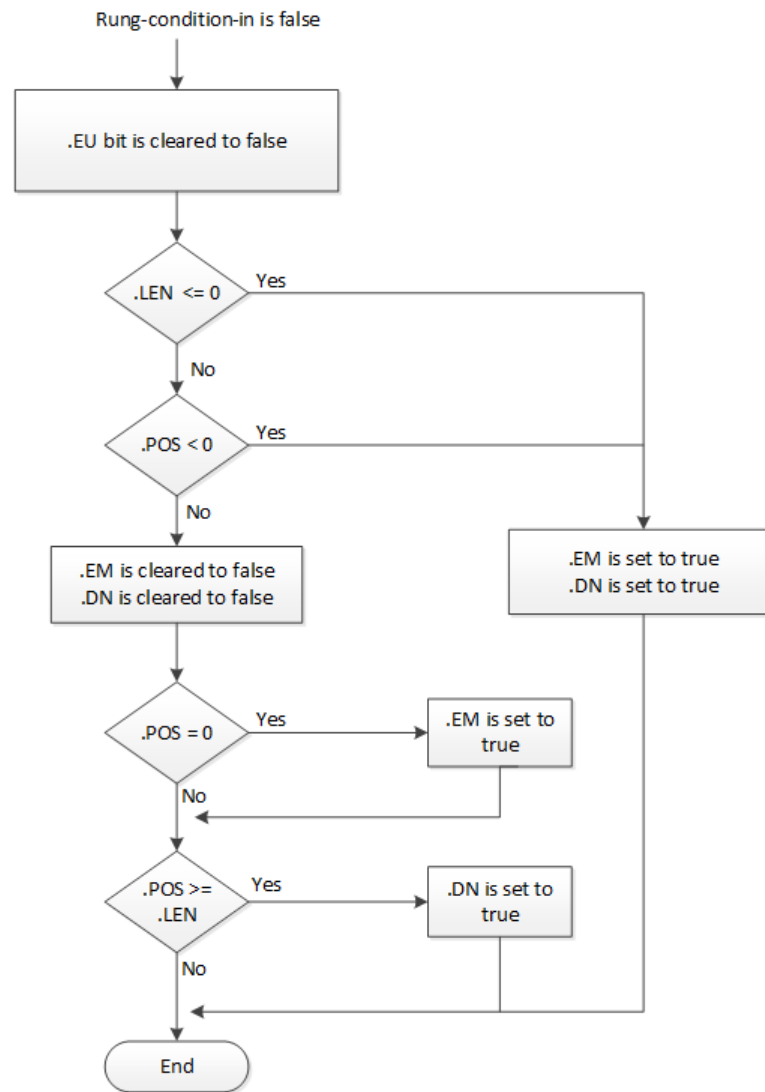
Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l'organigramme LFL (pré-scrutation)
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme LFL (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme LFL (vrai)
Post-scrutation	N/A.

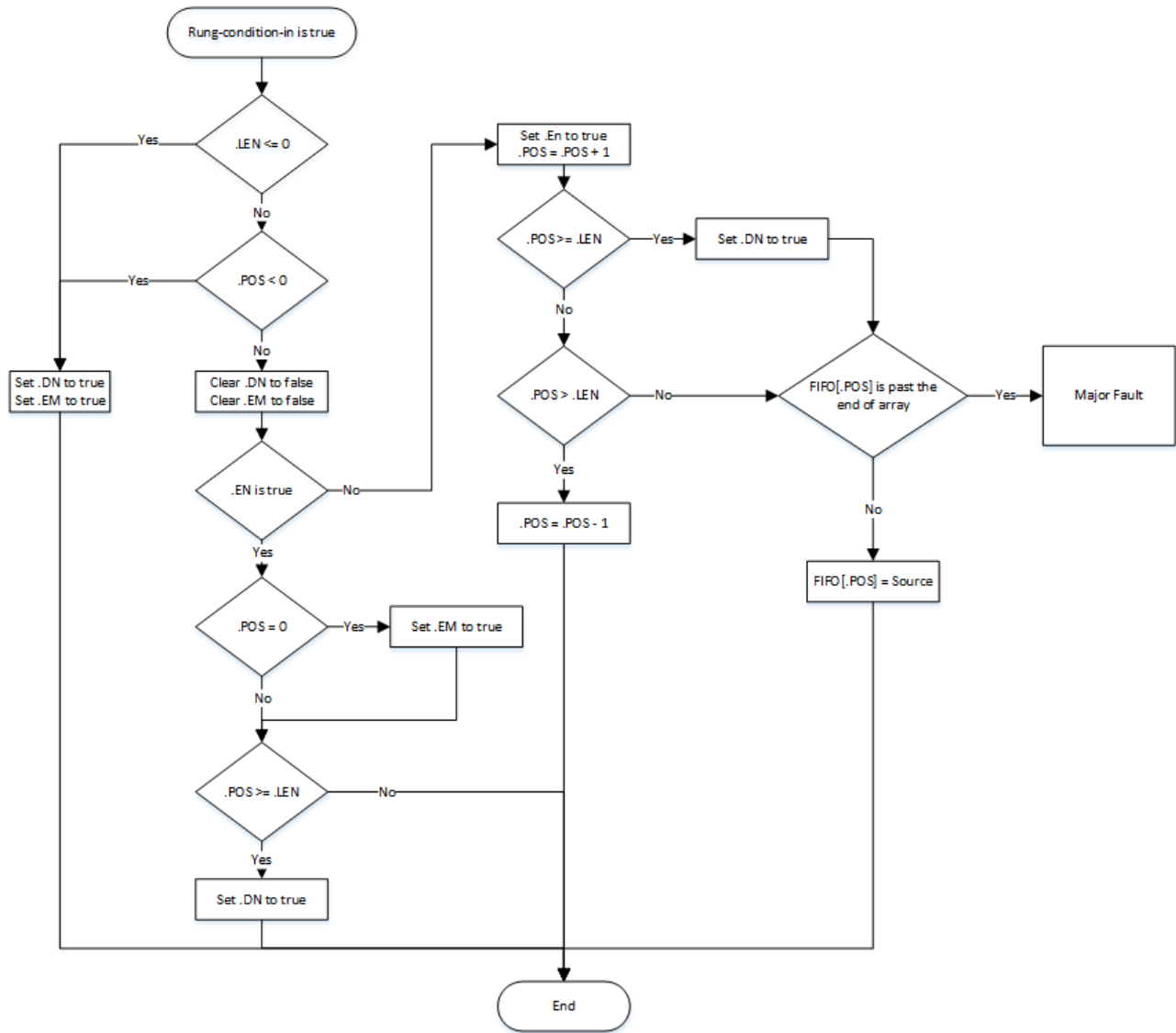
Organigramme LFL (pré-scrutation)



**Organigramme LFL (faux)**



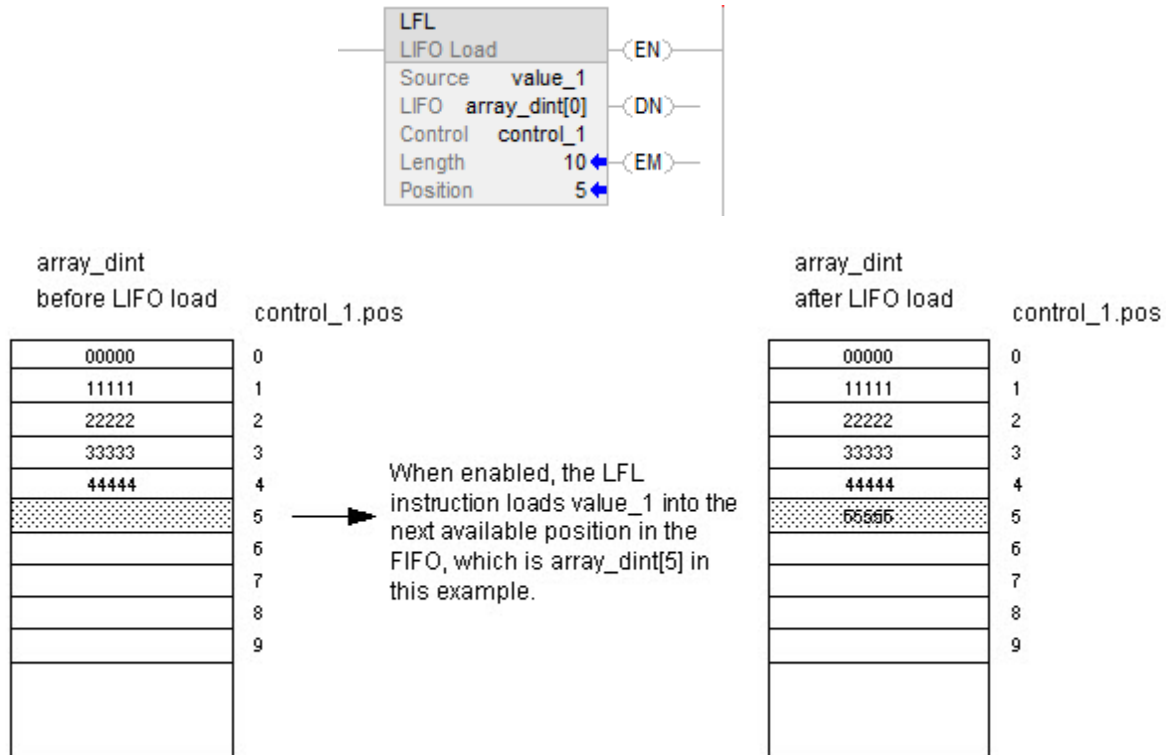
Organigramme LFL (vrai)



### Exemples

#### Exemple 1

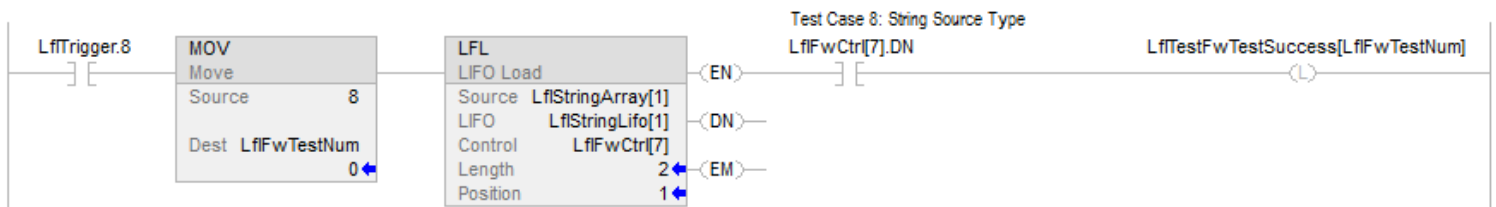
#### Diagramme à relais



#### Exemple 2

Le tableau Source est le tableau STRING ou le tableau Structure.

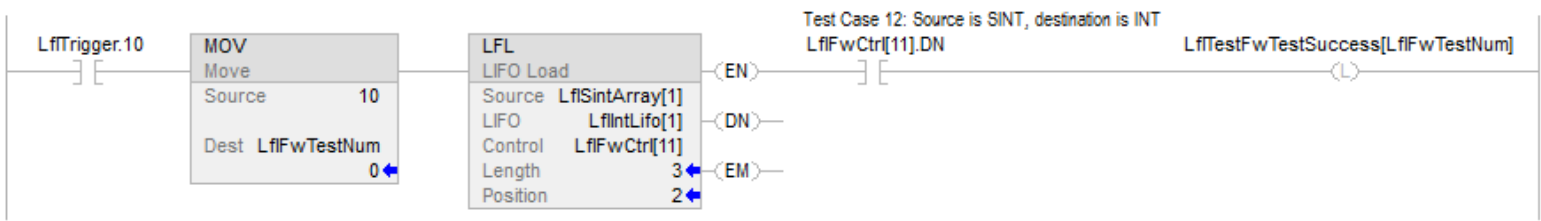
#### Diagramme à relais



#### Exemple 3

Le type de données de source ne correspond pas au type de données du tableau LIFO.

**Diagramme à relais**



**Voir aussi**

[Tableau \(Fichier\)/Instructions de décalage](#) sur la [page 569](#)

[Déchargement de LIFO \(LFU\)](#) sur la [page 600](#)

[Chargement de FIFO \(FFL\)](#) sur la [page 579](#)

[Déchargement de FIFO \(FFU\)](#) sur la [page 586](#)

[Attributs communs](#) sur la [page 885](#)

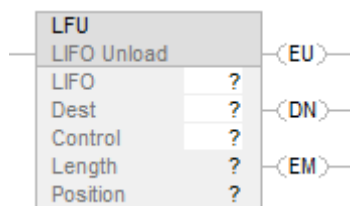
**Déchargement de LIFO (LFU)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction LFU décharge la valeur à .POS du LIFO et mémorise 0 à cet emplacement.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.



## Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction.

### Diagramme à relais

Opérande	Type	Format	Description
LIFO	SINT INT DINT REAL Type de chaînes structure	étiquette de tableau	LIFO à modifier spécifier le premier élément de LIFO Ne pas utiliser CONTROL.POS dans cet indice
Destination	SINT INT DINT REAL Type de chaînes structure	étiquette	Valeur déchargée depuis LIFO.
Control	CONTROL	étiquette	Structure de contrôle de l'opération en général, utiliser le même CONTROL que LFL associée
Length	DINT	immédiate	Nombre maximal d'éléments que LIFO peut contenir à un moment donné
Position	DINT	immédiate	Emplacement suivant dans LIFO où l'instruction décharge des données valeur initiale typique : 0

### Structure CONTROL

Mnémonique	Type de données	Description
.EU	BOOL	Le bit d'activation indique que l'instruction LFU a été activée.
.DN	BOOL	Le bit de fin est défini pour indiquer que LIFO est plein (.POS = .LEN).
.EM	BOOL	Le bit vide indique que LIFO est vide. Si .LEN est < ou = à 0 ou si .POS < 0, le bit .EM et le bit .DN sont tous deux définis.
.LEN	DINT	La longueur spécifie le nombre maximal d'éléments que LIFO peut contenir à un moment donné.

.POS	DINT	Cette position identifie la fin des données qui ont été chargées dans LIFO.
------	------	---

### Description

Utiliser l'instruction LFU avec l'instruction LFL pour mémoriser et recouvrer des données dans l'ordre dernier entré/premier sorti.

Lorsqu'elle est activée, l'instruction LFU décharge la valeur à l'emplacement .POS de LIFO et place cette valeur dans Destination. Cette instruction décharge une valeur et la remplace par un 0 lors de chaque activation de l'instruction, jusqu'à ce que LIFO soit vide. Lorsque LIFO est vide, LFU renvoie un 0 à Destination.

---

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

L'instruction LFU agit sur une mémoire contigüe. La portée de cette instruction est limitée par l'étiquette de base. L'instruction LFL ne va pas rédiger des données en dehors de l'étiquette de base mais peut traverser des limites de membres. Lorsque vous spécifiez un tableau qui est membre d'une structure et lorsque la longueur dépasse la taille de ce tableau, vous devez tester et confirmer que l'instruction LFL ne modifie pas des données que vous ne souhaitez pas modifier.

Pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580, les données sont limitées par le membre spécifié.

Si cette instruction essaye de lire au-delà de la fin d'un tableau, cette instruction définit le bit .ER et déclenche un défaut majeur.

En général, Source et LIFO sont du même type de données. Si les types de données Source et LIFO ne correspondent pas, cette instruction convertit la valeur Source au type de données de l'étiquette FIFO.

Un petit nombre entier est converti en un gros nombre entier par une extension de signe.

### Affecte les indicateurs d'état mathématique

Non

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
Lorsque Length spécifiée dépasse la fin du tableau LIFO	4	20

Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

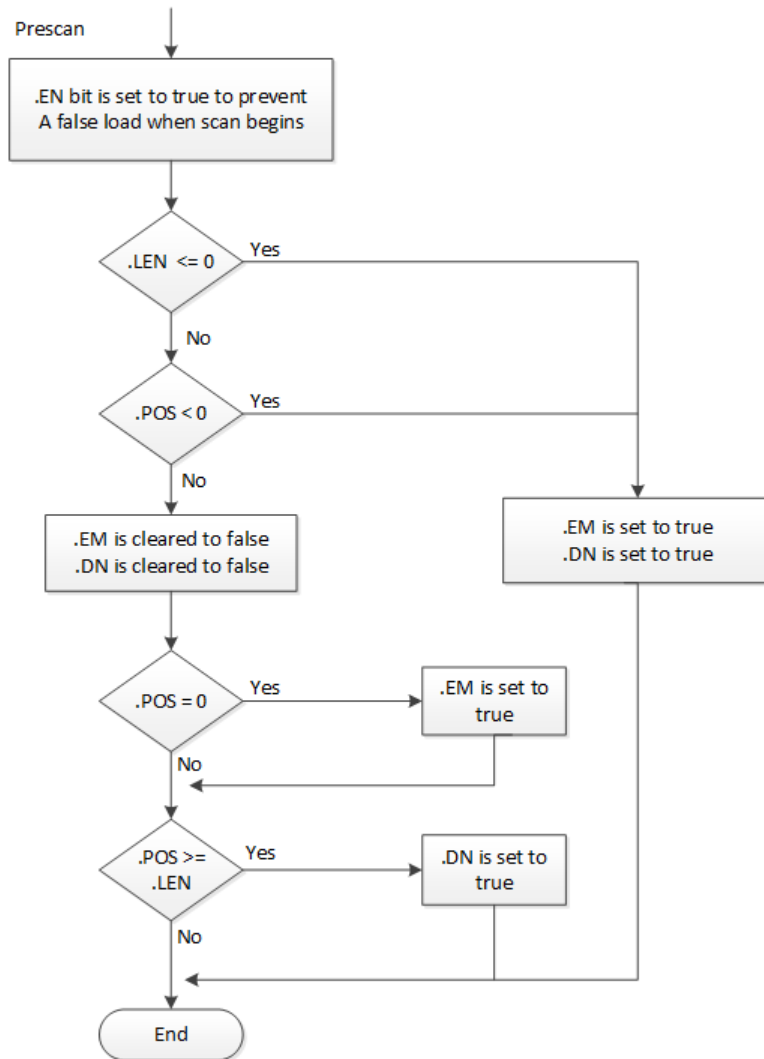
**Exécution**

Toutes ces conditions ne se produisent que dans le mode Scrutation normale

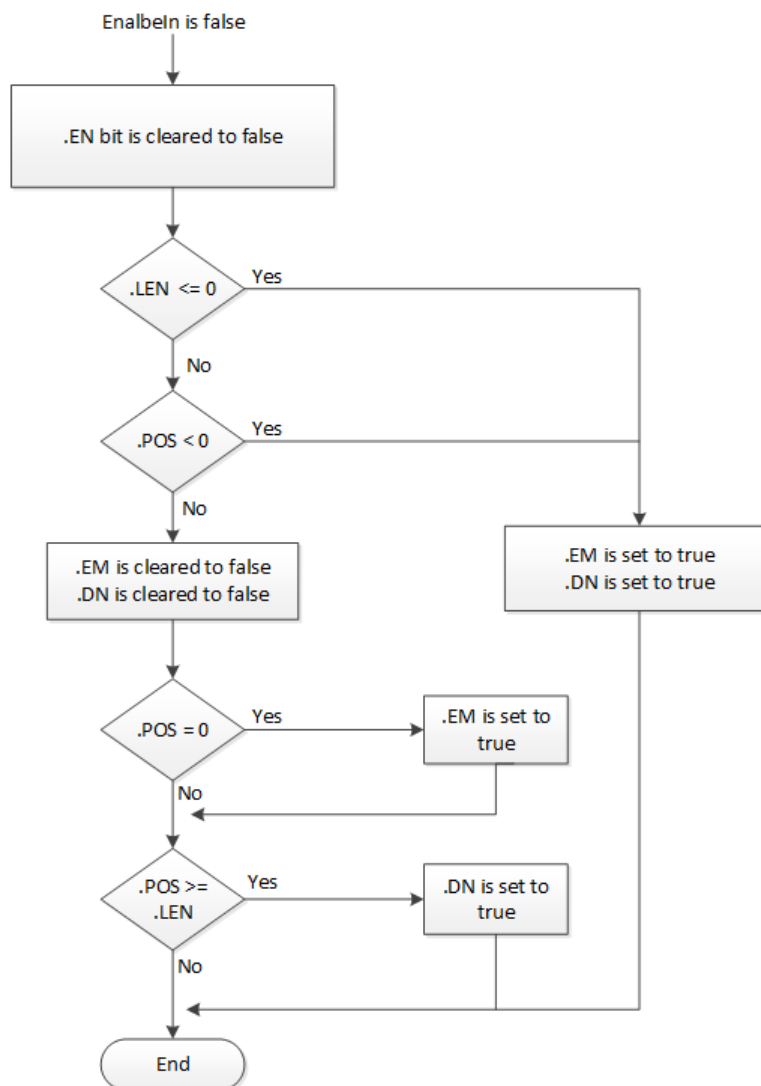
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l'organigramme LFU (pré-scrutation)
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme LFU (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme LFU (vrai)
Post-scrutation	N/A

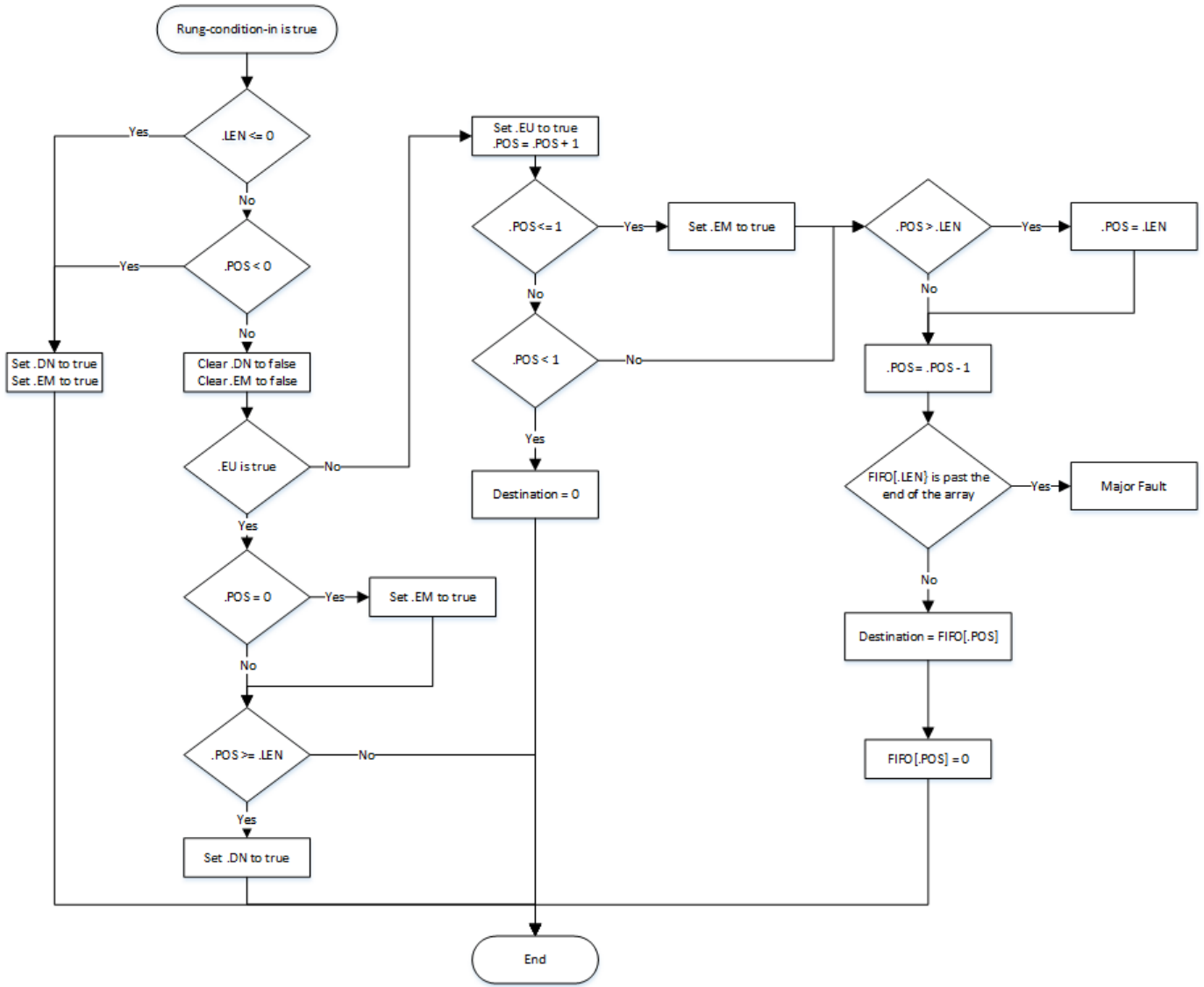
Organigramme LFU (pré-scrutation)



Organigramme LFU (faux)



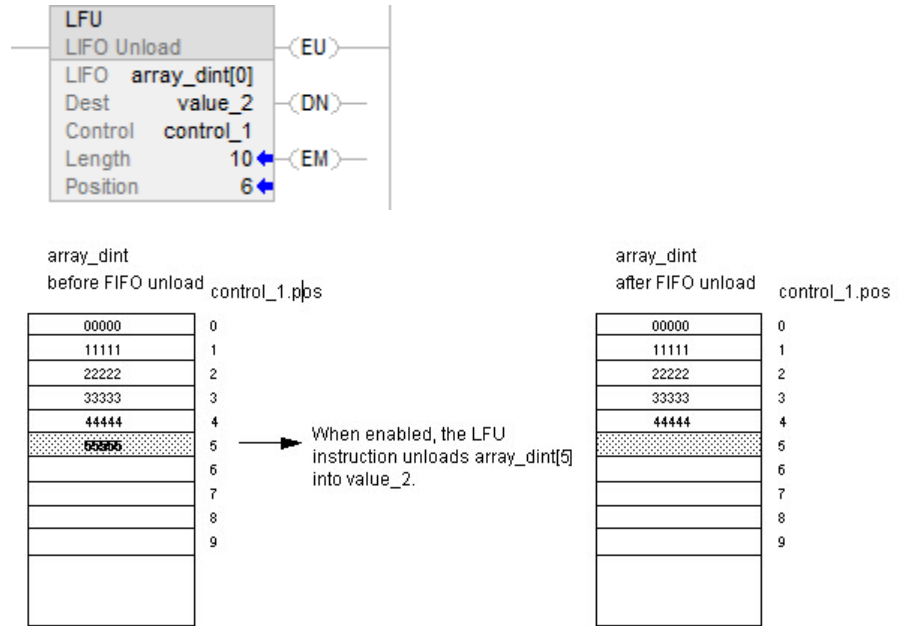
Organigramme LFU (vrai)



**Exemples**

**Exemple 1**

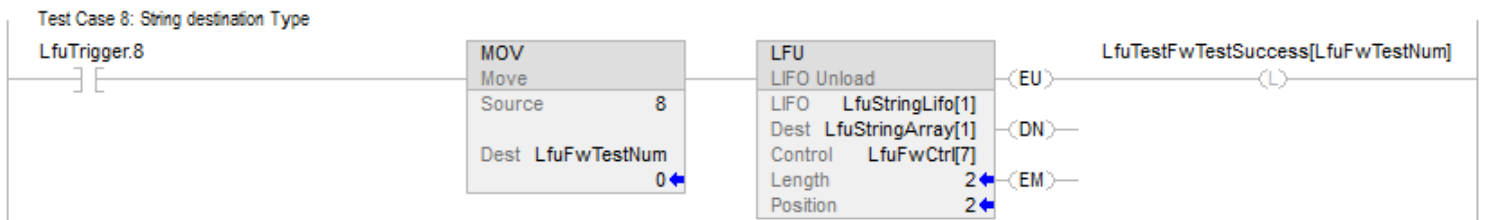
**Diagramme à relais**



**Exemple 2**

Le tableau de destination est le tableau STRING ou le tableau Structure.

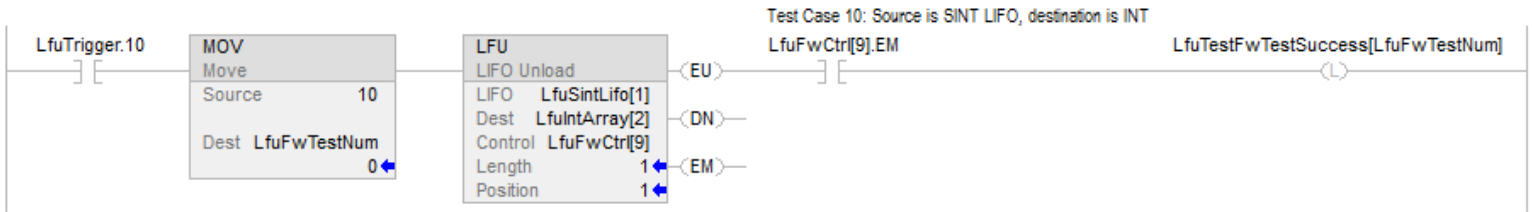
**Diagramme à relais**



**Exemple 3**

Le type de données du tableau de source LIFO ne correspond pas au type de données du tableau de destination.

### Diagramme à relais



### Voir aussi

[Tableau \(Fichier\)/Instructions de décalage](#) sur la [page 569](#)

[Chargement de LIFO \(LFL\)](#) sur la [page 593](#)

[Chargement de FIFO \(FFL\)](#) sur la [page 579](#)

[Déchargement de FIFO \(FFU\)](#) sur la [page 586](#)

[Attributs communs](#) sur la [page 885](#)



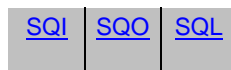
# Instructions de séquenceur

## Instructions de séquences

Les instructions de séquences surveillent les opérations homogènes et répétées.

### Instructions disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Indisponible

#### Texte structuré

Indisponible

Si vous voulez	Utiliser cette instruction
Détecter la fin de l'exécution d'une étape.	SQR
Définir les conditions de sortie pour l'étape suivante.	SQO
Charger les conditions de référence dans les tableaux de séquences	SQL

Les types de données en **caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

#### Voir aussi

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

## Entrée séquenceur (SQI)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction SQI détecte la fin de l'exécution d'une étape dans une paire d'instructions séquence SQQ/SQI.

### Langages disponibles

### Diagramme à relais

SQI	
Sequencer Input	
Array	?
Mask	?
Source	?
Control	?
Length	?
Position	?

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Les règles de conversion de données pour des types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

Opérande	Type	Format	Description
Array	DINT	étiquette de tableau	Tableau du séquenceur Spécifier le premier élément dans le tableau du séquenceur ne pas utiliser CONTROL.POS dans cet indice

Mask	SINT INT DINT	étiquette immédiate	Cet opérande permet de déterminer les bits qui doivent être bloqués (0) ou ceux qu'il faut laisser passer (1) lors de l'application à Source et l'élément de Tableau auquel .POS fait référence. Les types INT et SINT ont une extension à zéro par rapport à la taille d'un type DINT.
Source	SINT INT DINT	étiquette immédiate	Les données d'entrée utilisées pour effectuer une comparaison avec un élément de tableau auquel .POS fait référence.
Control	CONTROL	étiquette	Structure de contrôle de l'opération Il faut utiliser la même étiquette de contrôle dans les instructions SQO et SQL
Length	DINT	immédiate	Ceci représente la structure de contrôle CONTROL .LEN.
Position	DINT	immédiate	Ceci représente la structure de contrôle CONTROL .POS.

### Structure CONTROL

Mnémonique	Type de données	Description
.ER (Erreur)	BOOL	L'instruction a rencontré une erreur.
.LEN (Longueur)	DINT	La longueur précise le nombre d'étapes séquentielles dans le tableau séquentiel.
.POS (Position)	DINT	La position identifie l'élément de tableau que l'instruction est en train de comparer à Source. En général, la valeur initiale est 0

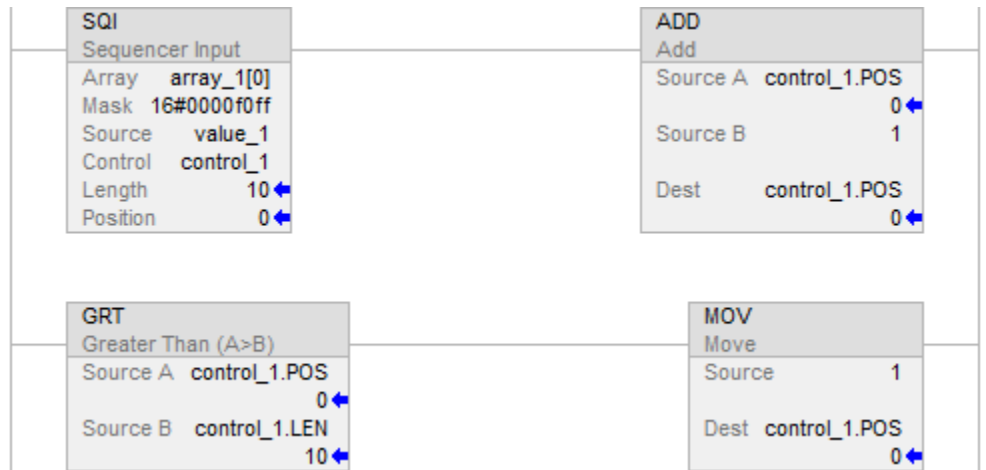
### Description

Lorsqu'elle est vraie, l'instruction SQI fait passer Source et l'élément actuel de Tableau par Mask. Les résultats de ces opérations de masquage sont comparés et, en cas d'égalité, la condition de sortie d'échelon est définie sur vrai ; sinon, la condition de sortie d'échelon est définie sur faux. En général, utiliser la même structure CONTROL que les instructions SQO et SQL.

### Utilisation de SQI sans SQO

Lorsque l'instruction SQI détermine qu'une étape est terminée, l'instruction ADD fait avancer, de manière incrémentielle, le tableau des séquences. Le GRT détermine si une autre valeur est disponible en vue de procéder à une vérification

dans le tableau des séquences. L’instruction MOV réinitialise la valeur de position après un passage, à une reprise, dans le tableau des séquences.



**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

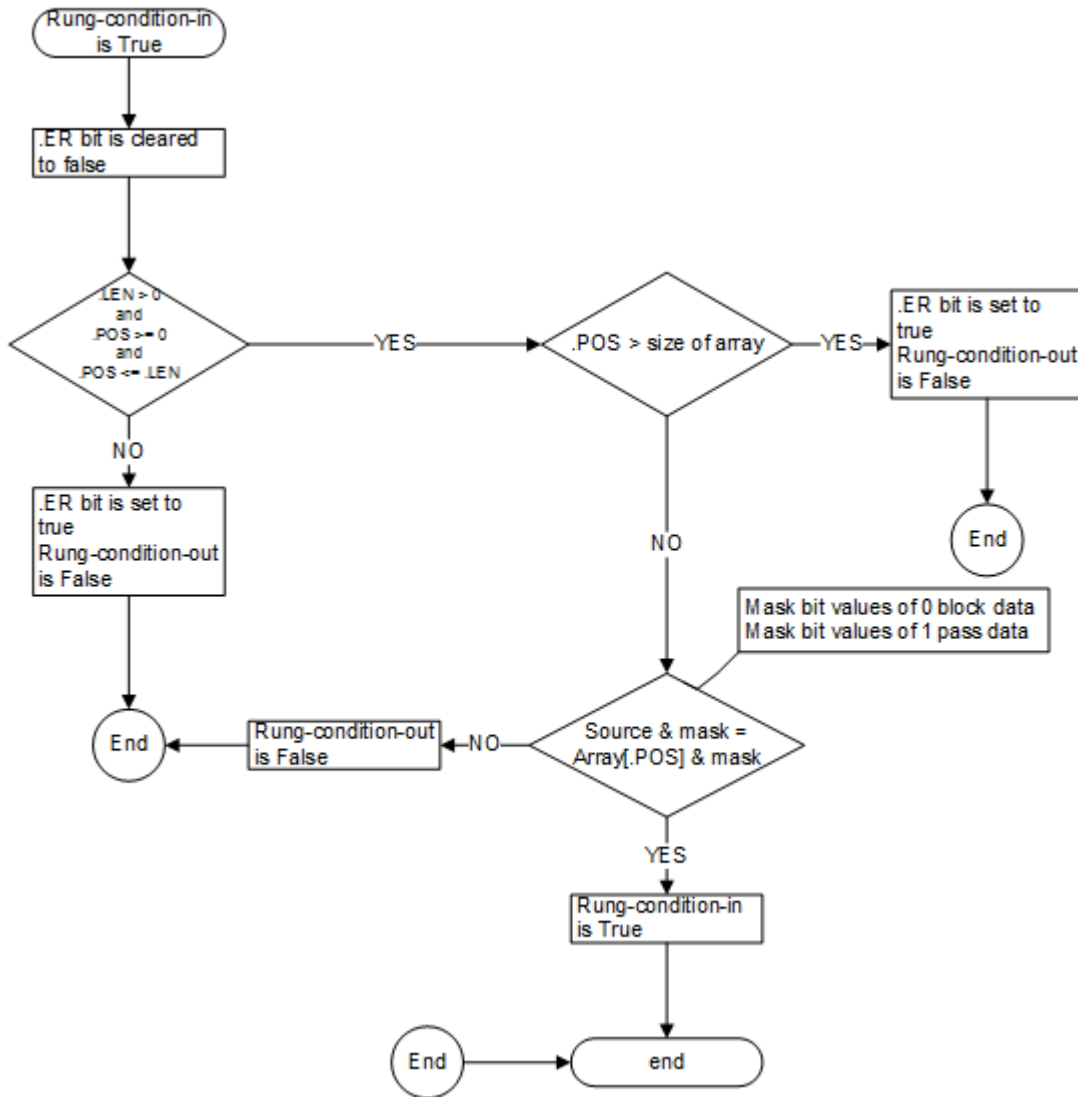
Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme (vrai)
Post-scrutation	N/A

Organigramme (vrai)



Exemple

Diagramme à relais

SQ1	
Sequencer Input	
Array	Array
Mask	Mask
Source	Source
Control	SqiControl
Length	3
Position	10

Si vous vous servez de l’instruction SQI sans une instruction appariée SQO, vous devez augmenter, depuis l’extérieur, le tableau des séquences, de manière incrémentielle.

La condition d’entrée d’échelon est définie sur vrai lorsque les instructions enableOut sont vraies lorsque le résultat de l’opération ET additionnant la valeur du tableau spécifiée par Position (par exemple, Array[Position]) à la valeur Mask est égal au résultat obtenu en utilisant l’opération ET pour la valeur Source à la valeur Mask, sinon les conditions de sortie d’échelon sont définies comme étant fausses.

**Voir aussi**

[instructions de séquences](#) sur la [page 609](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

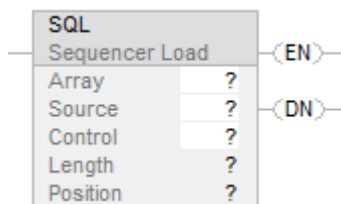
**Chargement de séquenceur (SQL)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction SQL charge la valeur de l’opérande Source dans le tableau du séquenceur.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n’est pas disponible dans le texte structuré.

## Opérandes

Les règles de conversion de données pour des types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

Opérande	Type (Type)	Format	Description (Description)
Tableau	DINT	étiquette de tableau	Tableau du séquenceur spécifier le premier élément dans le tableau du séquenceur ne pas utiliser CONTROL.POS dans cet indice
Source (Source)	SINT INT DINT	étiquette immédiate	Données à charger dans le tableau du séquenceur à la position spécifiée par .POS.
Control	CONTROL	étiquette	structure de contrôle de l'opération La même étiquette de contrôle devrait être utilisée dans les instructions SQI et SQO
Longueur (Length)	DINT	immédiate	Ceci représente la structure de contrôle CONTROL .LEN.
Position (Position)	DINT	immédiate	Ceci représente la structure de contrôle CONTROL .POS.

## Structure CONTROL

Mnémonique	Type de donnée	Description
.EN (Activation)	BOOL	Le bit d'activation indique que l'instruction SQL a été activée.
.DN (Fin)	BOOL	Le bit de fin est défini lorsque tous les éléments spécifiés ont été chargés dans le Tableau.
.ER (Erreur)	BOOL	Le bit d'erreur est défini lorsque .LEN < ou = 0, .POS < 0, or .POS > .LEN.
.LEN (Longueur)	DINT	La longueur indique le nombre d'étapes de séquenceur dans le tableau du séquenceur.
.POS (Position)	DINT	La position identifie l'endroit dans le Tableau où la valeur Source sera mémorisée.

## Description

Lorsque .EN passe de faux à vrai, .POS est incrémenté. .POS est réinitialisé à 1 lorsque .POS devient  $\geq$  à .LEN. L'instruction SQL charge la valeur Source dans le Tableau à une nouvelle position.

Lorsque .EN est vrai, l'instruction SQL charge la valeur Source dans le Tableau à la position courante.

En général, utilisez la même structure CONTROL que pour les instructions SQI et SQO.

**Important :** Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

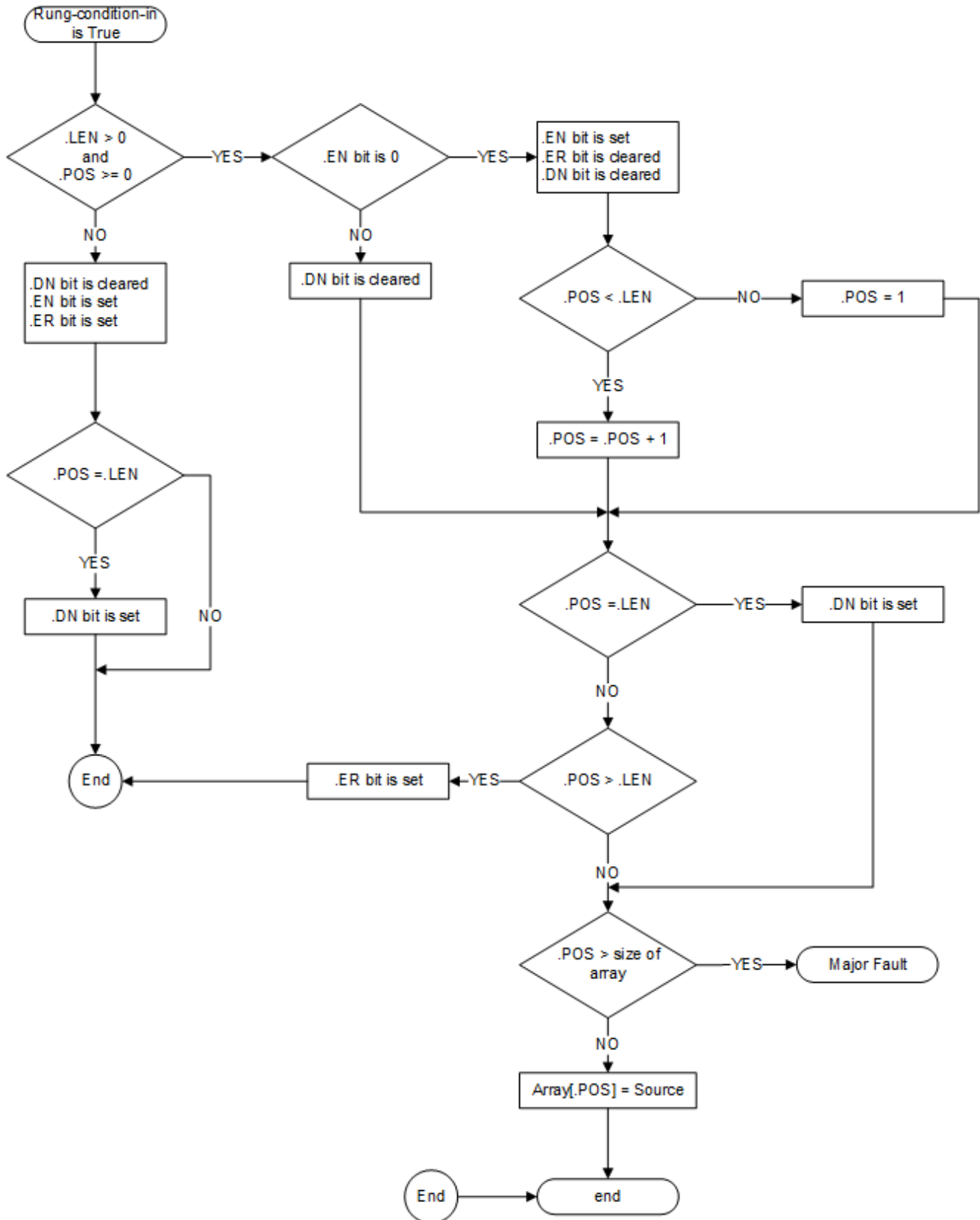
Un défaut majeur se produira si :	Type de défaut	Code de défaut
position > taille de Tableau	4	20

### Exécution

Condition/État	Action entreprise
Pré-scrutation	.EN est défini sur vrai.
Condition d'entrée d'échelon est fausse	.EN est défini sur faux
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme (vrai)
Post-scrutation	N/A

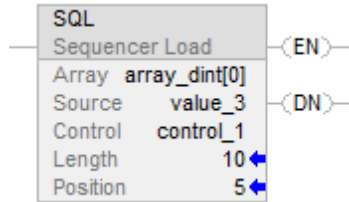


Organigramme – Vrai



**Exemple**

**Diagramme à relais**



Lorsqu'elle est activée, l'instruction SQL charge value\_3 à la prochaine position dans le tableau du séquenceur, qui est array\_dint[5] dans cet exemple.

**Voir aussi**

[Instructions de séquenceur](#) sur la [page 609](#)

[SQO](#) sur la [page 618](#)

[SQI](#) sur la [page 610](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

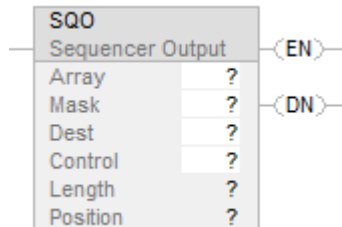
**Sortie séquenceur (SQO)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction SQO définit les conditions de sortie de la prochaine étape d'une paire en séquence d'instructions SQO/SQI.

**Langages disponibles**

**Diagramme à relais**



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Les règles de conversion de données pour des types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

Opérande	Type	Format	Description
Array	DINT	étiquette de tableau	tableau du séquenceur spécifier le premier élément dans le tableau du séquenceur ne pas utiliser CONTROL.POS dans cet indice
Mask	SINT INT DINT	étiquette immédiate	Utilisé pour déterminer quels bits bloquer (0) ou passer (1) et appliqué durant l'opération de masquage de la sortie.
Destination	DINT	étiquette	Données de sortie du tableau du séquenceur. Cette valeur est utilisée dans l'opération de masquage de la sortie.
Control	CONTROL	étiquette	structure de contrôle de l'opération La même étiquette de contrôle devrait être utilisée dans les instructions SQI et SQL
Length	DINT	immédiate	Nombre d'éléments du Tableau (tableau du séquenceur) à sortir
Position	DINT	immédiate	Position courante dans le tableau La valeur initiale est généralement 0.

### Structure CONTROL

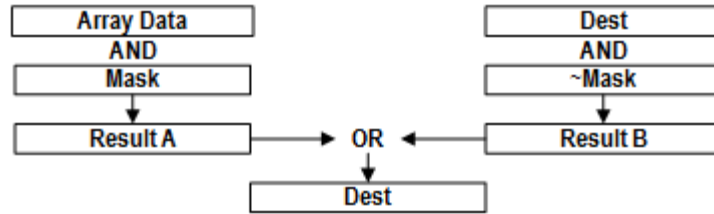
Mnémonique	Type de données	Description
.EN (Activation)	BOOL	Le bit d'activation indique que l'instruction SQO a été activée.
.DN (Fin)	BOOL	Le bit de fin est défini quand .POS = .LEN
.ER (Erreur)	BOOL	Indique que l'instruction a rencontré une erreur.
.LEN (Longueur)	DINT	La longueur indique le nombre d'étapes de séquenceur dans le tableau du séquenceur.

Mnémorique	Type de données	Description
.POS (Position)	DINT	La position identifie l'élément du Tableau que l'instruction utilise actuellement dans l'opération de masquage de la sortie.

**Description**

Lorsque .EN passe de faux à vrai, .POS est incrémenté. .POS est réinitialisé à 1 lorsque .POS devient supérieur ou égal à .LEN.

Lorsque .EN est vrai, l'instruction SQO déplace les données de tableau à la position .POS à travers Mask, puis déplace la valeur Destination courante à travers le Mask complémenté. Un OU logique est effectué entre le résultat de ces opérations, et le résultat est mémorisé dans Destination.



En général, vous devriez utiliser la même structure CONTROL que pour les instructions SQI et SQL.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

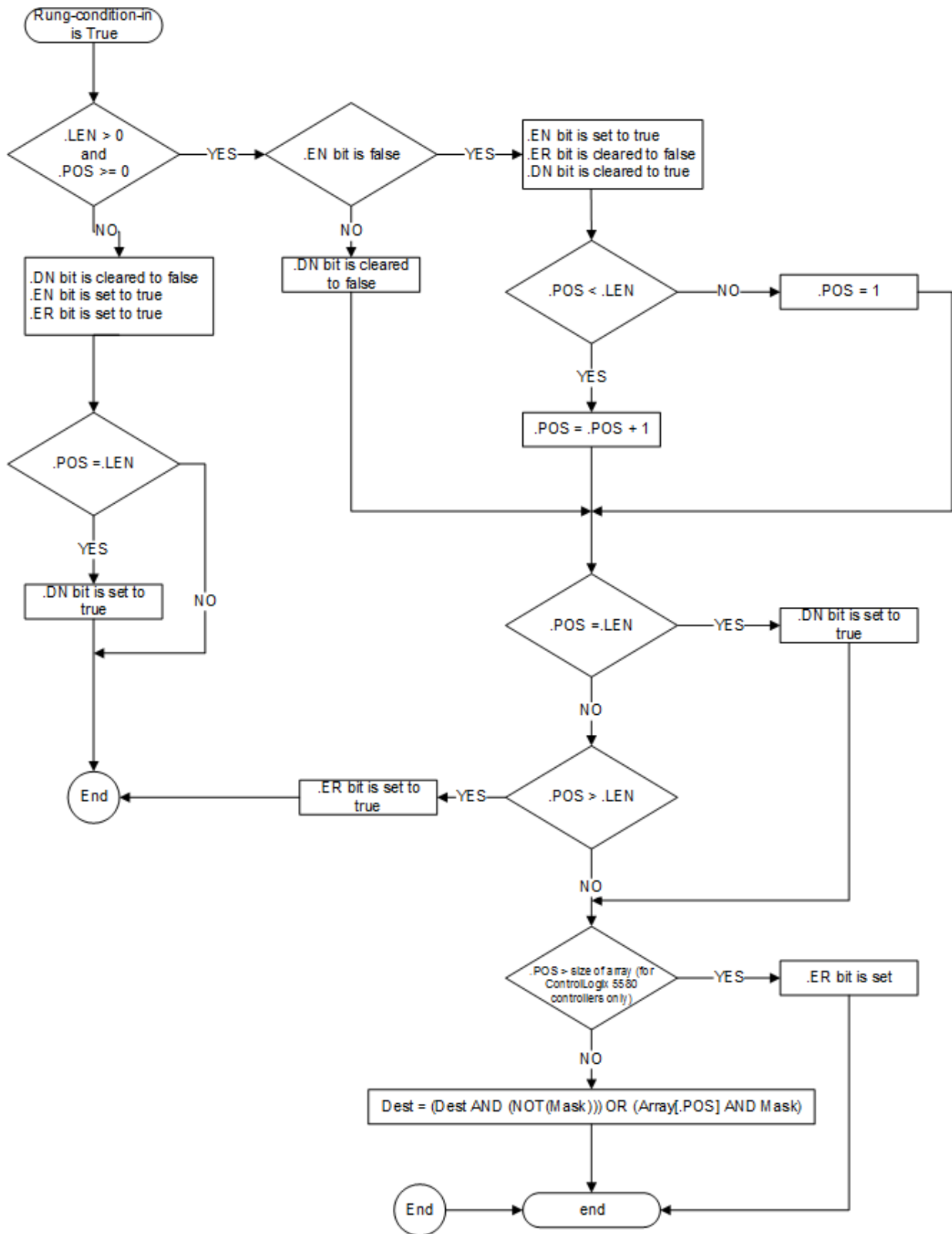
Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	.EN est défini sur vrai.
Condition d'entrée d'échelon est fausse	.EN est défini sur faux
Condition d'entrée d'échelon est vraie	Reportez-vous à l'Organigramme (vrai) suivant
Post-scrutation	N/A

Organigramme (vrai)



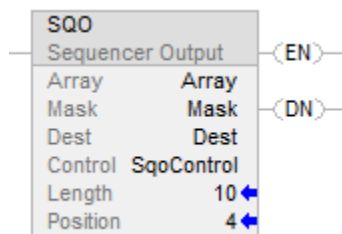
### Exemple

Un ET logique est effectué avec la valeur de Mask et la valeur de Array, par exemple Array[SqoControl.POS]. Un ET logique est effectué entre le complément de Mask et la valeur Dest courante. Un OU logique est effectué ensuite entre le résultat de ces deux opérations, et le résultat est mémorisé dans Dest.

Pour réinitialiser .POS à sa valeur initiale (.POS = 0), utiliser une instruction RES pour effacer la structure de contrôle. Cet exemple utilise l'état du bit de première scrutation pour mettre à zéro la valeur .POS.



### Diagramme à relais



### Voir aussi

[Instructions de séquences](#) sur la [page 609](#)

[SQI](#) sur la [page 610](#)

[SQL](#) sur la [page 614](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

# Instructions de contrôle du programme

Utiliser les instructions de contrôle de programme pour changer le déroulement de la logique.

## Instructions disponibles

### Diagramme à relais

<a href="#">JMP</a>	<a href="#">LBL</a>	<a href="#">JSR</a>	<a href="#">JXR</a>	<a href="#">RET</a>	<a href="#">SBR</a>	<a href="#">TND</a>	<a href="#">MCR</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

<a href="#">UID</a>	<a href="#">UIE</a>	<a href="#">SFR</a>	<a href="#">SFP</a>	<a href="#">EVENT</a>	<a href="#">AFI</a>	<a href="#">EOT</a>	<a href="#">NOP</a>
---------------------	---------------------	---------------------	---------------------	-----------------------	---------------------	---------------------	---------------------

### Bloc fonctionnel

<a href="#">JSR</a>	<a href="#">RET</a>	<a href="#">SBR</a>
---------------------	---------------------	---------------------

### Texte structuré

<a href="#">JSR</a>	<a href="#">RET</a>	<a href="#">SBR</a>	<a href="#">TND</a>	<a href="#">EVENT</a>	<a href="#">UID</a>	<a href="#">EOT</a>	<a href="#">SFR</a>
---------------------	---------------------	---------------------	---------------------	-----------------------	---------------------	---------------------	---------------------

<a href="#">UIE</a>	<a href="#">SFP</a>
---------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
Sauter une section de logique qui n'a pas toujours besoin d'être exécutée.	JMP LBL
Sauter à une routine séparée, passer des données à la routine, exécuter la routine et renvoyer les résultats.	JSR SBR RET

Sauter à une routine externe	JXR
Marquer une fin temporaire qui arrête l'exécution de la routine.	TND
Désactiver tous les échelons d'une section de logique	MCR
Désactiver toutes les tâches utilisateur.	UID
Activer toutes les tâches utilisateur.	UIE
Mettre en pause un graphe de fonctionnement séquentiel	SFP
Réinitialiser un graphe de fonctionnement séquentiel	SFR
Terminer une transition pour un graphe de fonctionnement séquentiel	EOT
Déclencher l'exécution d'une tâche d'événement	EVENT
Désactiver un échelon	AFI
Insérer un espace réservé dans la logique.	NOP

**Voir aussi**

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

## Instructions de contrôle du programme

Utiliser les instructions de contrôle de programme pour changer le déroulement de la logique.

**Instructions disponibles**

**Diagramme à relais**

<a href="#">JMP</a>	<a href="#">LBL</a>	<a href="#">JSR</a>	<a href="#">JXR</a>	<a href="#">RET</a>	<a href="#">SBR</a>	<a href="#">TND</a>	<a href="#">MCR</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

<a href="#">UID</a>	<a href="#">UIE</a>	<a href="#">SFR</a>	<a href="#">SFP</a>	<a href="#">EVENT</a>	<a href="#">AFI</a>	<a href="#">EOT</a>	<a href="#">NOP</a>
---------------------	---------------------	---------------------	---------------------	-----------------------	---------------------	---------------------	---------------------



**Bloc fonctionnel**

<a href="#">JSR</a>	<a href="#">RET</a>	<a href="#">SBR</a>
---------------------	---------------------	---------------------

**Texte structuré**

<a href="#">JSR</a>	<a href="#">RET</a>	<a href="#">SBR</a>	<a href="#">TND</a>	<a href="#">EVENT</a>	<a href="#">UID</a>	<a href="#">EOT</a>	<a href="#">SFR</a>
---------------------	---------------------	---------------------	---------------------	-----------------------	---------------------	---------------------	---------------------

<a href="#">UIE</a>	<a href="#">SFP</a>
---------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
Sauter une section de logique qui n'a pas toujours besoin d'être exécutée.	JMP LBL
Sauter à une routine séparée, passer des données à la routine, exécuter la routine et renvoyer les résultats.	JSR SBR RET
Sauter à une routine externe	JXR
Marquer une fin temporaire qui arrête l'exécution de la routine.	TND
Désactiver tous les échelons d'une section de logique	MCR
Désactiver toutes les tâches utilisateur.	UID
Activer toutes les tâches utilisateur.	UIE
Mettre en pause un graphe de fonctionnement séquentiel	SFP
Réinitialiser un graphe de fonctionnement séquentiel	SFR
Terminer une transition pour un graphe de fonctionnement séquentiel	EOT
Déclencher l'exécution d'une tâche d'événement	EVENT
Désactiver un échelon	AFI
Insérer un espace réservé dans la logique.	NOP

**Voir aussi**

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

## Toujours faux (AFI)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction AFI définit EnableOut sur faux.

### Langages disponibles

#### Diagramme à relais

— [ AFI ] —

#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

#### Diagramme à relais

Aucun

#### Description

L'instruction AFI définit sa valeur EnableOut sur faux.

#### Affecte les indicateurs d'état mathématique

Non

#### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes

## Exécution

Toutes les conditions se trouvant sous la ligne grasse continue peuvent avoir lieu uniquement pendant le mode Scrutation normale.

Condition	Action
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Définir EnableOut sur faux.
Condition d'entrée d'échelon est vraie	Définir EnableOut sur faux.
Post-scrutation	N/A

## Exemples

### Diagramme à relais

Utiliser l'instruction AFI pour désactiver temporairement un échelon lorsque vous être en train de déboguer un programme. L'instruction AFI désactive toutes les instructions de cet échelon.



### Voir aussi

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Relais de contrôle maître \(MCR\)](#) sur la [page 646](#)

[Pas d'opération \(NOP\)](#) sur la [page 650](#)

[Fin temporaire \(TND\)](#) sur la [page 657](#)

[Attributs communs](#) sur la [page 885](#)

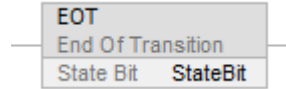
## Fin de transition (EOT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction EOT sert à définir l'état d'une transition. En général, elle se produit dans une sous-routine appelée depuis une transition (JSR). Le paramètre State Bit utilisé lors d'une instruction EOT détermine l'état de la Transition. Lorsque State Bit est défini sur vrai, la SFC passe à l'état suivant, sinon EOT agit en tant que NOP.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

EOT(StateBit);

### Opérandes

### Diagramme à relais

Opérande	Type	Format	Description
State Bit	BOOL	étiquette	état de la transition (0=exécution en cours, 1=terminé)

### Texte structuré

Opérande	Type	Format	Description
State Bit	BOOL	étiquette	état de la transition (0=exécution en cours, 1=terminé)

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description

Comme l'instruction EOT renvoie un état booléen, plusieurs routines SFC peuvent partager la même routine qui contient l'instruction EOT. Si la routine d'appel n'est pas une transition, l'instruction EOT agit comme une instruction NOP.

Dans un automate Logix, le paramètre de retour renvoie l'état de transition, puisque la condition d'échelon n'est pas disponible dans tous les langages de programmation Logix.

**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction renvoie le bit de données à la routine d'appel.
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	L'instruction renvoie le bit de données à la routine d'appel.
Post-scrutation	N/A

**Exemple**



**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

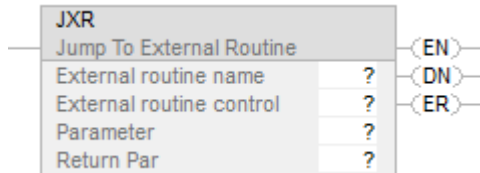
## Saut vers sous-programme externe (JXR)

Cette information s'applique à l'automate SoftLogix 5800 uniquement.

L'instruction JXR exécute une routine externe.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible pour le texte structuré.

### Opérandes

### Diagramme à relais

Opérande	Type	Format	Description
External routine name	ROUTINE	Nom	Routine externe à exécuter
External routine control	EXT_ROUTINE_CONTROL	Étiquette	Structure de contrôle
Paramètre	BOOL SINT INT DINT REAL structure	Immédiate Étiquette Étiquette de tableau	Les données de cette routine que vous voulez copier dans une variable de la routine externe Parameters est optionnel. Entrer des paramètres multiples, si nécessaire. Vous pouvez utiliser jusqu'à 10 paramètres.
Return parameter	BOOL SINT INT DINT REAL	Étiquette	L'étiquette de cette routine où vous voulez copier le résultat de la routine externe Return parameter est optionnel. Vous ne pouvez avoir qu'un seul return parameter

## Structure EXT\_ROUTINE\_CONTROL

Mnémonique	Type de données	Description	Implémentation
ErrorCode	SINT	Si une erreur se produit, cette valeur identifie l'erreur. Les valeurs valides sont comprises entre 0 et 255.	Il n'y a pas de codes d'erreur prédéfinis. Le développeur de la routine externe doit fournir les codes d'erreur.
NumParams	SINT	Cette valeur indique le nombre de paramètres associés avec cette instruction.	Uniquement pour l'affichage – cette information provient de l'entrée de l'instruction.
ParameterDefs	EXT_ROUTINE_PARAMETERS[10]	Ce tableau contient les définitions des paramètres à passer à la routine externe. L'instruction peut passer un maximum de 10 paramètres.	Uniquement pour l'affichage – cette information provient de l'entrée de l'instruction.
ReturnParamDef	EXT_ROUTINE_PARAMETERS	Cette valeur contient les définitions du paramètre de retour de la routine externe. Il n'y a qu'un seul paramètre de retour.	Uniquement pour l'affichage – cette information provient de l'entrée de l'instruction.
EN	BOOL	Lorsqu'il est défini, le bit d'activation indique que l'instruction JXR a été activée.	La routine externe définit ce bit.
ReturnsValue	BOOL	S'il est défini, ce bit indique qu'un paramètre de retour a été saisi pour l'instruction. S'il est à zéro, ce bit indique qu'aucun paramètre de retour n'a été saisi pour l'instruction.	Uniquement pour l'affichage – cette information provient de l'entrée de l'instruction.
DN	BOOL	Le bit de fin est défini lorsque la routine externe a été exécutée une fois complètement.	La routine externe définit ce bit.
ER	BOOL	Le bit d'erreur est défini lorsqu'une erreur se produit. L'exécution de l'instruction s'arrête jusqu'à ce que le programme mette à zéro le bit d'erreur.	La routine externe définit ce bit.
FirstScan	BOOL	Ce bit indique si c'est la première scrutation après le basculement de l'automate en mode Exécution. Si nécessaire, utiliser FirstScan pour initialiser la routine externe.	L'automate définit ce bit pour refléter l'état de la scrutation.
EnableOut	BOOL	Activer la sortie.	La routine externe définit ce bit.

EnableIn	BOOL	Activer l'entrée.	L'automate définit ce bit pour refléter la condition d'entrée d'échelon. L'instruction est exécutée indépendamment de la condition d'échelon. Le développeur de la routine externe devrait surveiller cet état et agir en conséquence.
User1	BOOL	L'utilisateur peut utiliser ces bits. L'automate n'initialise pas ces bits.	Soit la routine externe, soit le programme utilisateur peut définir ces bits.
User0	BOOL		
ScanType1	BOOL	Ces bits identifient le type de scrutation courant :	L'automate définit ces bits pour refléter l'état de la scrutation.
ScanType0	BOOL		
		<b>Valeurs de bit</b>	<b>Type de scrutation</b>
		00	Normal
		01	Pré-scrutation
		10	Post-scrutation (ne s'applique pas aux programmes en logique à relais)

**Description**

Utiliser l'instruction Saut vers routine externe (JXR) pour appeler la routine externe d'une routine à relais de votre projet. L'instruction JXR prend en charge des paramètres multiples, vous pouvez donc passer des valeurs entre la routine à relais et la routine externe.

L'instruction JXR est similaire à l'instruction Sauter vers sous-routine (JSR). L'instruction JXR lance l'exécution de la routine externe spécifiée :

- La routine externe n'est exécutée qu'une seule fois.
- Après l'exécution de la routine externe, l'exécution de la logique retourne à la routine qui contient l'instruction JXR.

**Affecte les indicateurs d'état mathématique**

Non



**Défauts majeurs/mineurs**

Un défaut majeur se produira si	Type de défaut	Code de défaut :
Une exception se produit dans la routine externe DLL. La DLL n'a pas pu être chargée. Le point d'entrée n'a pas été trouvé dans la DLL.	4	88

**Exécution**

L'instruction JXR peut être synchrone ou asynchrone, selon l'implémentation de la DLL. Le code de la DLL détermine aussi la façon de répondre à un état de scrutation, un état de condition d'entrée d'échelon et un état de condition de sortie d'échelon.

Pour plus d'informations sur l'utilisation de l'instruction JXR et la création de routines externes, Reportez-vous au SoftLogix5800 System User Manual, publication 1789-UM002.

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

**Saut vers une étiquette (JMP) et Etiquette (LBL)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

Les instructions JMP et LBL sautent des portions du diagramme logique à relais.

**Langages disponibles****Diagramme à relais**

—(JMP)—

—[ LBL ]—

**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

## Opérandes

### Diagramme à relais

Opérande	Type	Format	Description
Instruction JMP			
Label name		nom du label	Entrer le nom de l'instruction LBL associée
Instruction LBL			
Label name		nom du label	L'exécution saute à l'instruction LBL référencée.

### Description

Lorsqu'elle est vraie, l'instruction JMP saute à l'instruction LBL référencée et l'automate continue d'exécuter la logique à partir de là. Lorsqu'elle est fausse, l'instruction JMP n'affecte pas l'exécution du langage ladder.

L'instruction JMP et l'instruction LBL qu'elle référence doivent être dans la même routine.

L'instruction JMP peut déplacer l'exécution du langage ladder en avant ou en arrière. Un saut à un label en avant permet d'économiser du temps de scrutation de programme en omettant un segment de logique jusqu'à ce qu'il soit nécessaire. Un saut en arrière permet à l'automate d'itérer un segment de logique.

---

**Important :** Soyez prudent de ne pas sauter en arrière de trop nombreuses fois. Le temporisateur du chien de garde pourrait dépasser le délai d'attente parce que la scrutation ne s'est pas terminée à temps.

---



La logique qui est sautée n'est pas scrutée. Placer les opérations critiques à l'extérieur de la zone qui est sautée.

---

Une instruction JMP nécessite l'existence du label associé avant que vous ne puissiez :

- Télécharger lorsque vous travaillez hors ligne
- Accepter des modifications lorsque vous travaillez en ligne

L'instruction LBL doit être la première instruction sur l'échelon.

Un nom de label doit être unique à l'intérieur d'une routine. Le nom peut :

- Avoir jusqu'à 40 caractères
- Contenir des lettres, des chiffres et des soulignements ( \_ )

### Affecte les indicateurs d'état mathématique

Non.

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

### Diagramme à relais

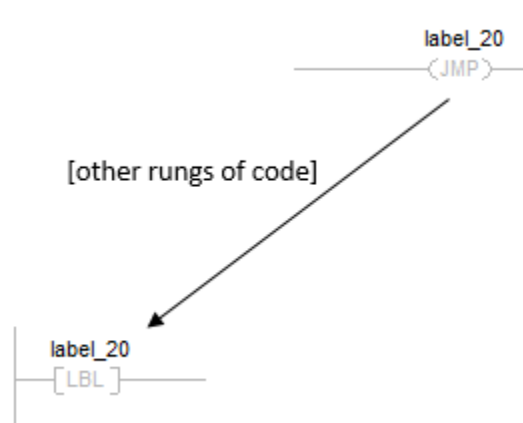
Condition	Action
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	(Pour JMP) L'exécution saute à l'échelon qui contient l'instruction LBL avec le nom de label référencé. (Pour LBL) Aucune action n'est entreprise
Post-scrutation	N/A

### Exemple

### Diagramme à relais

### JMP

Lorsque l'instruction JMP est activée, l'exécution saute par-dessus des échelons successifs de logique jusqu'à ce qu'elle atteigne l'échelon qui contient l'instruction LBL avec le nom label\_20.



**LBL**



**Voir aussi**

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Saut vers sous-routine \(JSR\), Sous-routine \(SBR\), et Retour \(RET\)](#) sur la [page 636](#)

[Pour \(FOR\)](#) sur la [page 671](#)

[Rupture \(BRK\)](#) sur la [page 669](#)

[Attributs communs](#) sur la [page 885](#)

**Saut vers sous-routine (JSR), Sous-routine (SBR), et Retour (RET)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

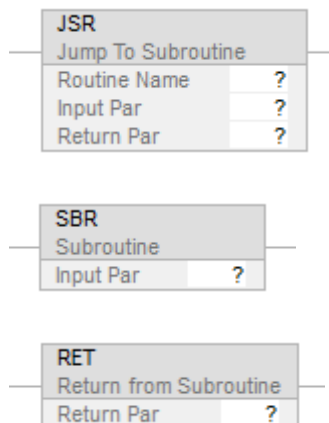
L’instruction JSR appelle une autre routine. Lorsque cette routine se termine, l’exécution retourne à l’instruction JSR.

L’instruction SBR reçoit les paramètres d’entrée passés par JSR.

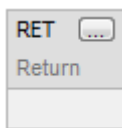
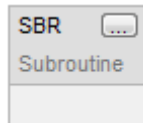
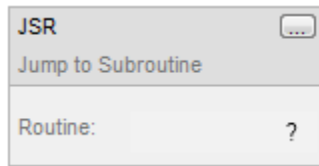
L’instruction RET renvoie les paramètres de retour à JSR et termine la scrutation de la sous-routine.

**Langages disponibles**

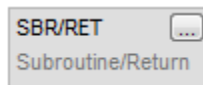
**Diagramme à relais**



## Bloc fonctionnel



## Grphe de fonctionnement séquentiel



## Texte structuré

JSR(RoutineName,InputCount,InputPar,ReturnPar);

SBR(InputPar);

RET(ReturnPar);

## Opérandes

---

**Important :** Une opération inattendue peut se produire dans les cas suivants :

- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.
- 




---

Pour chaque paramètre dans les instructions SBR ou RET, utiliser le même type de données (y compris toute dimension de tableau) que le paramètre correspondant dans l'instruction JSR. L'utilisation de différents types de données peut produire des résultats imprévus.

---

## Diagramme à relais

## Instruction JSR

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Routine Name	ROUTINE	ROUTINE	nom	Sous-routine à exécuter
Input Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	immediate étiquette étiquette de tableau	Les données de cette routine à copier dans une étiquette de la sous-routine. <ul style="list-style-type: none"> <li>Les paramètres d'entrée sont optionnels</li> <li>Entrer un nombre maximal de 40 paramètres d'entrée, si nécessaire.</li> </ul>
Return Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	étiquette étiquette de tableau	Étiquette dans cette routine pour copier le résultat de la sous-routine. <ul style="list-style-type: none"> <li>Les paramètres de retour sont optionnels</li> <li>Entrer un nombre maximal de 40 paramètres de retour, si nécessaire</li> </ul>

## Instruction SBR

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Input Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	étiquette étiquette de tableau	<ul style="list-style-type: none"> <li>Étiquette dans cette routine dans laquelle copier les paramètres d'entrée correspondant (40 au maximum) de l'instruction JSR.</li> </ul>

## Instruction RET

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Return Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	Immédiate étiquette étiquette de tableau	Données de cette routine dans laquelle copier les paramètres de retour correspondant (40 au maximum) de l'instruction JSR.

## Affecter les indicateurs d'état mathématique

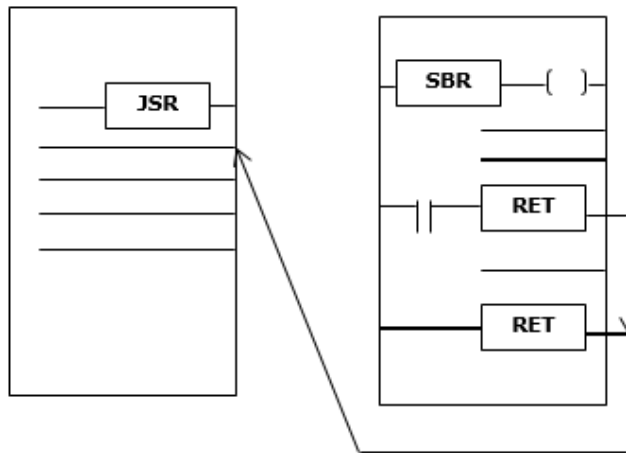
Non

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
L'instruction JSR a moins de paramètres d'entrée que l'instruction SBR	4	31
L'instruction JSR saute à une routine de défaut	4	990 ou fourni par l'utilisateur
L'instruction RET a moins de paramètres de retour que l'instruction JSR	4	31
La routine principale contient une instruction RET	4	31

**Fonctionnement**

**Important :** N'importe quelle routine peut contenir une instruction JSR mais une instruction JSR ne peut pas appeler (exécuter) une routine principale.



L'instruction JSR lance l'exécution de la routine spécifiée, qui est appelée une sous-routine :

- La sous-routine est exécutée chaque fois qu'elle est scrutée.
- Après l'exécution de la sous-routine, l'exécution de la logique retourne à la routine qui contient l'instruction JSR et continue par l'instruction après JSR.

Pour programmer un saut dans une sous-routine, observez ces lignes directrices.

**JSR**

- Pour copier des données dans une étiquette de la sous-routine, entrer un paramètre d'entrée.
- Pour copier un résultat d'une sous-routine dans une étiquette de cette routine, entrer un paramètre de retour.



- Entrer jusqu'à 40 entrées et jusqu'à 40 paramètres de retour, si nécessaire.

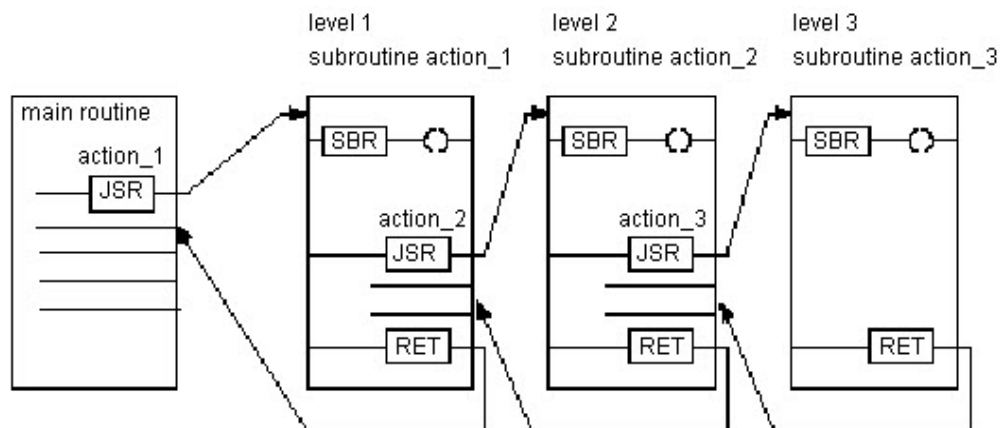
### SBR

- Si l'instruction JSR a un paramètre d'entrée, entrer une instruction SBR.
- Placer l'instruction SBR comme première instruction de la routine.
- Pour chaque paramètre d'entrée de l'instruction JSR, entrer une étiquette dans laquelle vous voulez copier les données.

### RET

- Si l'instruction JSR a un paramètre de retour, entrer une instruction RET.
- Placer l'instruction RET comme dernière instruction de la routine.
- Pour chaque paramètre de retour de l'instruction JSR, entrer un paramètre de retour à envoyer à l'instruction JSR.
- Dans une routine à relais, placer des instructions RET additionnelles pour quitter la sous-routine en fonction de conditions d'entrée différentes, si c'est nécessaire (les routines de Bloc fonctionnel ne permettent qu'une seule instruction RET).

Invoquer jusqu'à 25 sous-routines imbriquées, avec un maximum de 40 paramètres passés à une sous-routine, et un maximum de 40 paramètres renvoyés d'une sous-routine.



**Astuce :** Sélectionner le menu **Éditer > Éditer un élément à relais (Edit > Edit Ladder Element)** pour ajouter et supprimer des opérandes variables. Dans le cas des instructions JSR et SBR, ajouter un paramètre d'entrée. Dans le cas des instructions JSR et RET, ajouter un paramètre de sortie. Pour toutes les trois instructions, supprimer le paramètre d'instruction.

**Exécution****Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	L'échelon est défini sur faux. L'automate exécute toutes les sous-routines. Pour s'assurer que tous les échelons dans la sous-routine sont pré-scrutés, l'automate ignore les instructions RET (ce qui veut dire que les instructions RET ne font pas quitter la sous-routine). Les paramètres d'entrée et de retour ne sont pas passés. Si la même sous-routine est invoquée plusieurs fois, elle ne sera pré-scrutée qu'une seule fois.
Condition d'entrée d'échelon est fausse (pour l'instruction JSR)	N/A
Condition d'entrée d'échelon est vraie	Les paramètres sont passés et la sous-routine est exécutée.
Post-scrutation	Même action que pour la pré-scrutation

**Bloc fonctionnel**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
EnableIn est faux	N/A
EnableIn est vrai	Les paramètres sont transmis et la sous-routine est exécutée
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

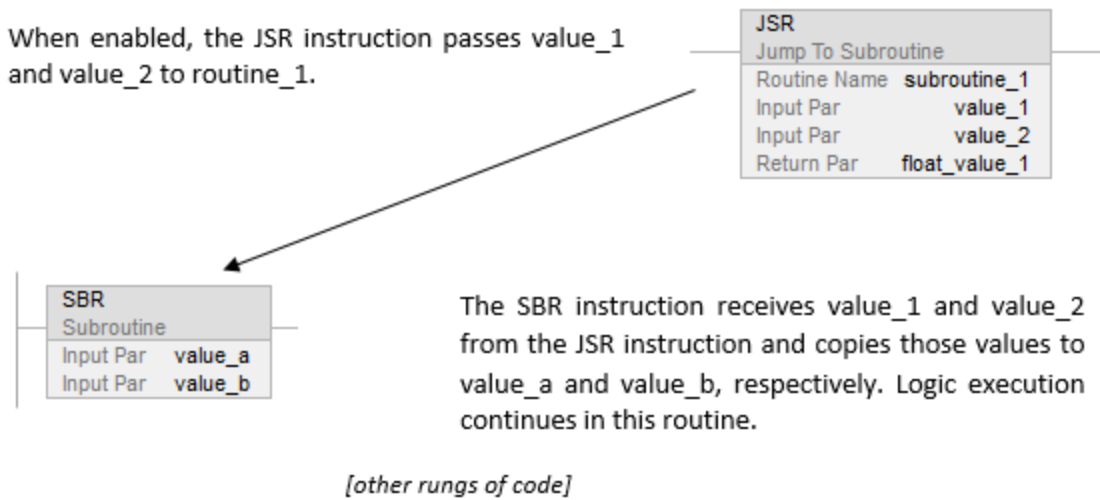
**Texte structuré**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Les paramètres sont passés et la sous-routine est exécutée.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

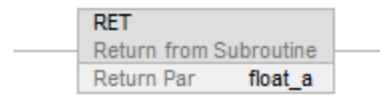
**Exemples**

**Exemple 1**

**Diagramme à relais**



When enabled, the RET instruction sends float\_a to the JSR instruction. The JSR instruction receives float\_a and copies the value to float\_value\_1. Logic execution continues with the next instruction following the JSR instruction.



**Texte structuré**

Routine	Program
Routine principale	JSR(routine_1,2,value_1,value_2,float_value_1);
Sous-routine	SBR(value_a,value_b); <statements>; RET(float_a);

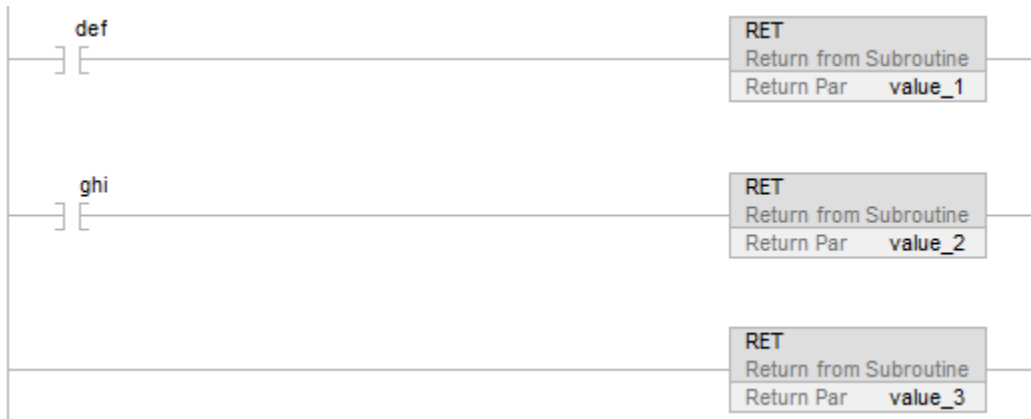
### Exemple 2

#### Diagramme à relais

#### Routine principale

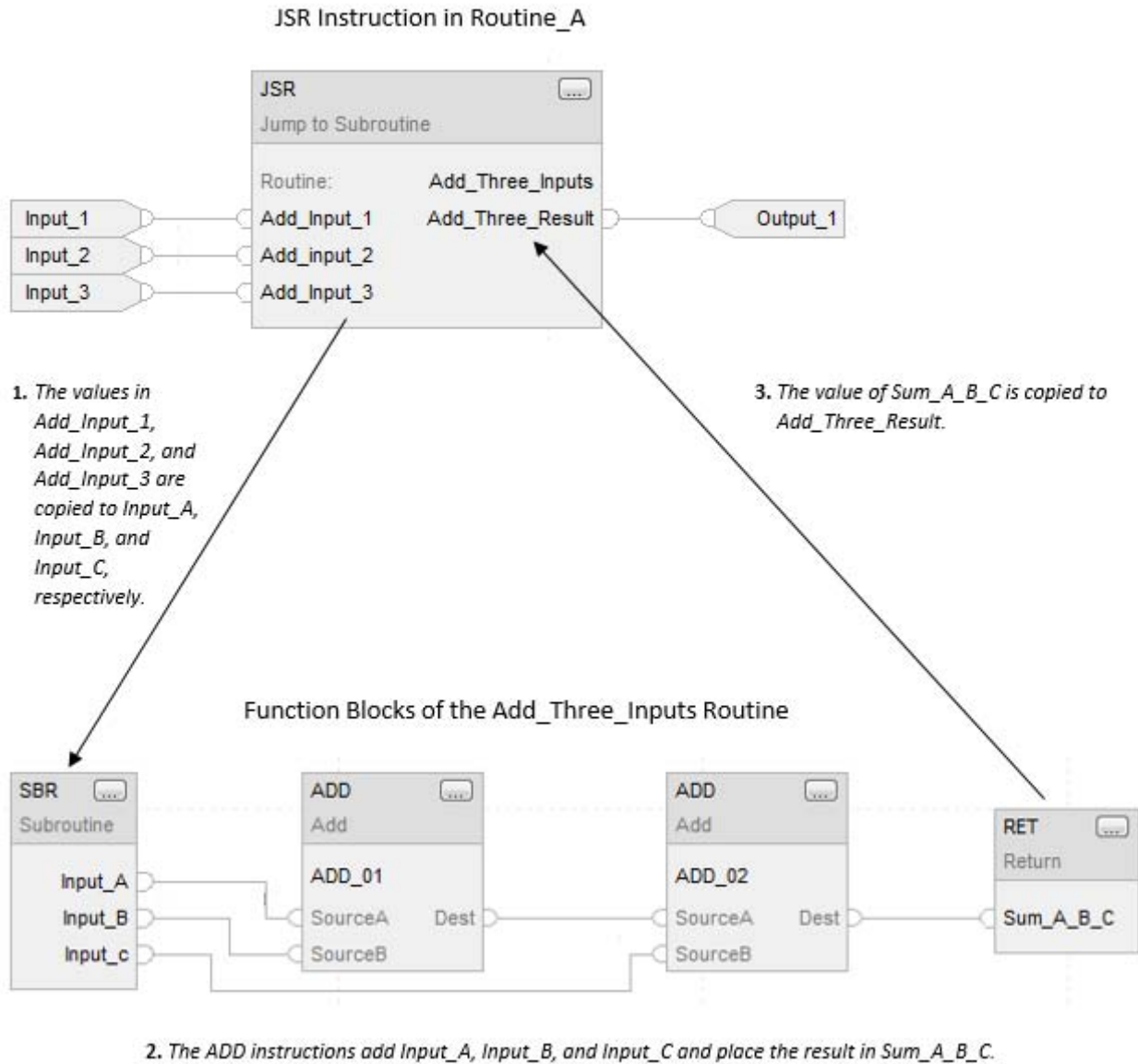


#### sous-routine\_1



### Exemple 3

#### Bloc fonctionnel



#### Voir aussi

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

## Relais de contrôle maître (MCR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction MCR simule un relais de contrôle maître (un relais câblé obligatoire qui peut être désactivé par tout interrupteur d'arrêt d'urgence connecté en série). Lorsque le relais électromécanique est désactivé, ses contacts s'ouvrent afin de désactiver tous les dispositifs d'E/S des applications. L'instruction MCR peut désactiver sélectivement une section d'échelons.

### Langages disponibles

#### Diagramme à relais

—(MCR)—

#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

#### Description

L'instruction MCR est capable d'outrepasser le comportement normal des échelons, en forçant chaque instruction à être exécutée comme si condition d'entrée d'échelon était fausse. En général, la fausse exécution d'une instruction est plus rapide que sa véritable exécution, par conséquent, le fait de désactiver sélectivement les sections de code non nécessaires peut améliorer le temps de scrutation.

Chaque fois que l'instruction MCR est exécutée et que condition d'entrée d'échelon est faux, le comportement d'outrepassement est basculé. En conséquence, deux instructions MCR sont normalement requises : une pour débiter la « zone », et une seconde pour la terminer.

L'instruction MCR ouvrante est en général mise sous conditions par une ou plusieurs instructions d'entrée. Lorsque les conditions d'entrée sont fausses, la zone est désactivée. Lorsque les conditions d'entrée sont vraies, la zone opère normalement.

L'instruction MCR fermante est généralement inconditionnelle. Si la zone est activée, l'instruction MCR fermante sera vraie, donc elle ne fera rien. Si la zone est

désactivée, alors l'instruction MCR fermante sera faussée et basculera l'outrepassement, réactivant les échelons qui suivent.

Lorsque vous programmez une zone MCR, notez que :

L'instruction MCR doit être la dernière instruction sur l'échelon.

- Vous devez terminer la zone avec une instruction MCR inconditionnelle. Si l'instruction MCR fermante est faussée, et que la zone est activée, l'instruction MCR fermante va désactiver tous les échelons qui suivent.
- Vous ne pouvez pas imbriquer une zone MCR à l'intérieur d'une autre. Il n'y a qu'un seul bit d'outrepassement dans chaque programme. Chaque instruction MCR peut faire basculer ce bit d'outrepassement. Le fait d'essayer d'imbriquer des zones MCR va en fait créer plusieurs petites zones.
- Ne faites pas de saut dans une zone MCR. Si l'instruction MCR ouvrante n'est pas exécutée, la zone ne sera pas désactivée.
- Le bit d'outrepassement est automatiquement réinitialisé à la fin de la routine. Si une zone MCR continue jusqu'à la fin de la routine, vous n'avez pas besoin de programmer une instruction MCR pour terminer la zone, cependant, pour éviter toute confusion lors de l'édition en ligne, il est recommandé de toujours utiliser l'instruction MCR fermante.

Si l'instruction MCR est désactivée dans une sous-routine ou une instruction complémentaire, le bit d'outrepassement est réinitialisé lorsque la sous-routine/l'instruction complémentaire retourne.

Les instructions complémentaires possèdent leur propre bit d'outrepassement qui est initialisé lorsque l'instruction complémentaire est appelée. Si une instruction complémentaire est appelée depuis une zone MCR désactivée, la routine de mode de scrutation faussée va être exécutée normalement. Une fois que l'instruction complémentaire retourne, l'état de la zone sera restauré à l'état qu'elle avait avant l'appel de l'instruction complémentaire.

---

**Important :** L'instruction MCR n'est pas un substitut pour un relais électromécanique de contrôle maître câblé qui permet un arrêt d'urgence. Vous devriez toujours installer un relais électromécanique de contrôle maître câblé pour permettre de couper d'urgence l'alimentation des E/S.

---

**Important :** Ne superposez pas ou n'imbriguez pas des zones MCR. Chaque zone MCR doit être distincte et complète. Si des zones se chevauchent ou s'imbriquent, le fonctionnement de la machine peut être imprévisible et entraîner des dommages matériels et des blessures.

Placer les opérations critiques à l'extérieur de la zone MCR. Si vous démarrez des instructions telles que des temporisateurs dans une zone MCR, l'exécution de ces instructions deviendra fausse lorsque la zone sera désactivée et tout temporisateur sera réinitialisé.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	Le comportement d'outrepassement est basculé, activant ou désactivant les échelons qui suivent.
Condition d'entrée d'échelon est vraie	N/A
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**

Lorsque la première instruction MCR est activée (input\_1, input\_2 et input\_3 sont définis), l'automate exécute les échelons dans la zone MCR (entre les deux instructions MCR) et définit ou met à zéro les sorties, en fonctions des conditions d'entrée.



Lorsque la première instruction MCR est désactivée (input\_1, input\_2 et input\_3 sont tous non définis), l'automate exécute les échelons dans la zone MCR (entre les deux instructions MCR) et EnableIn devient faux pour tous les échelons dans la zone MCR, indépendamment des conditions d'entrée.



Voir aussi

[Instructions de contrôle du programme](#) sur la [page 624](#)

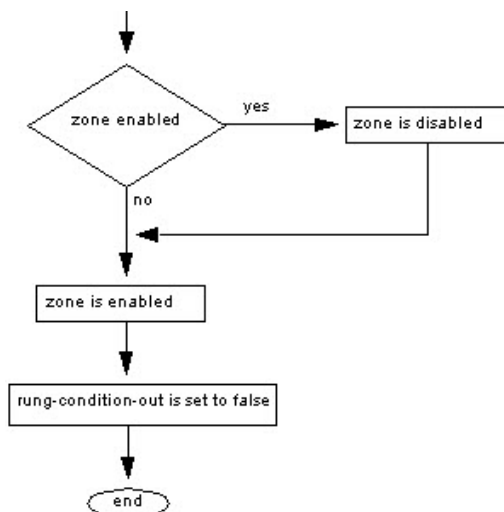
[Toujours faux \(AFI\)](#) sur la [page 626](#)

[Pas d'opération \(NOP\)](#) sur la [page 650](#)

[Fin temporaire \(TND\)](#) sur la [page 657](#)

[Attributs communs](#) sur la [page 885](#)

### Organigramme MCR (faux)



## Pas d'opération (NOP)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction NOP agit comme un espace réservé.

### Langages disponibles

### Diagramme à relais

—[NOP]—

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

### Diagramme à relais

Aucun

### Description

Vous pouvez placer l'instruction NOP n'importe où sur un échelon. Lorsqu'elle est activée, l'instruction NOP n'exécute aucune opération. Lorsqu'elle est désactivée, l'instruction NOP n'exécute aucune opération.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	N/A
Post-scrutation	N/A

**Exemples**

**Diagramme à relais**



**Voir aussi**

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Toujours Faux \(AFI\)](#) sur la [page 626](#)

[Relais de contrôle maître \(MCR\)](#) sur la [page 646](#)

[Fin temporaire \(TND\)](#) sur la [page 657](#)

[Attributs communs](#) sur la [page 885](#)

**Pause SFC (SFP)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction SFP met en pause une routine SFC.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

SFP(SFCRoutineName,TargetState);

**Opérandes**

**Diagramme à relais**

Opérande	Type	Format	Description
SFCRoutineName	ROUTINE	nom	Routine SFC à mettre en pause
TargetState	DINT	immédiate	Sélectionner-en un : <ul style="list-style-type: none"> <li>• Exécution en cours (ou saisir 0)</li> <li>• Pause en cours (ou saisir 1)</li> </ul>

**Texte structuré**

Opérande	Type	Format	Description
SFCRoutineName	ROUTINE	nom	Routine SFC à mettre en pause
TargetState	DINT	immédiate	Sélectionner-en un : <ul style="list-style-type: none"> <li>• Exécution en cours (ou saisir 0)</li> <li>• Pause en cours (ou saisir 1)</li> </ul>

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Description**

L'instruction SFP vous laisse mettre en pause une routine SFC en cours d'exécution.

**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
Le type de routine n'est pas une routine SFC	4	85

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution****Diagramme à relais**

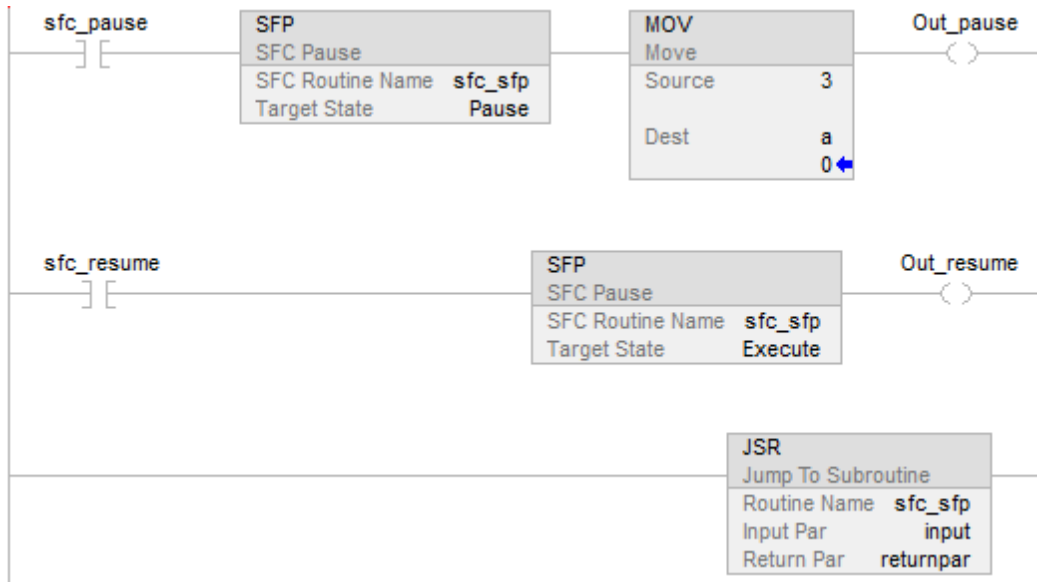
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse.	N/A
Condition d'entrée d'échelon est vraie	L'instruction met en pause ou redémarre l'exécution de la routine SFC spécifiée.
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	L'instruction met en pause ou redémarre l'exécution de la routine SFC spécifiée.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

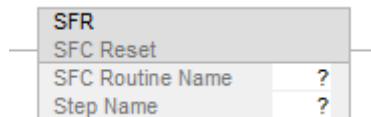
**Réinitialisation SFC (SFR)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction SFR réinitialise l’exécution d’une routine SFC à une étape spécifiée.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

SFR(SFCRoutineName,StepName);

**Opérandes****Diagramme à relais**

Opérande	Type	Format	Description
SFCRoutineName	ROUTINE	nom	Routine SFC à réinitialiser
StepName	SFC_STEP	étiquette	Étape cible à partir de laquelle reprendre l'exécution

**Texte structuré**

Opérande	Type	Format	Description
SFCRoutineName	ROUTINE	nom	Routine SFC à réinitialiser
StepName	SFC_STEP	étiquette	Étape cible à partir de laquelle reprendre l'exécution

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Description**

Lorsque l'instruction SFR est activée :

- Dans la routine SFC spécifiée, toutes les actions mémorisées cessent d'être exécutées (réinitialisation).
- La routine SFC commence à être exécutée à l'étape spécifiée.
- Si l'étape cible est 0, le graphe sera réinitialisé à l'étape initiale.

L'implémentation Logix de l'instruction SFR est différente de celle de l'automate PLC-5. Dans le cas de l'automate PLC-5, l'instruction SFR est exécutée lorsque la condition d'échelon est vraie. Après la réinitialisation, la routine SFC devrait rester en pause jusqu'à ce que l'échelon contenant l'instruction SFR devienne faux. Cela permet de temporiser l'exécution après une réinitialisation. Cette fonctionnalité de mise en pause/redémarrage de l'instruction SFR du PLC-5 a été découplée de la condition d'échelon et déplacée dans l'instruction SFP.

**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
Le type de routine n'est pas une routine SFC	4	85
L'étape cible spécifiée n'existe pas dans la routine SFC	4	89

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction réinitialise l'exécution de la routine SFC spécifiée à une étape particulière.
Post-scrutation	N/A

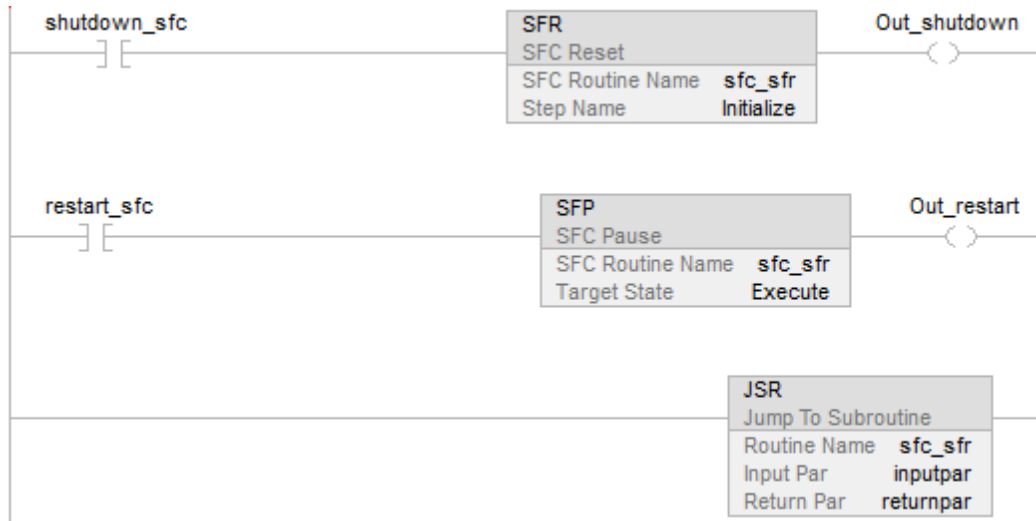
**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	L'instruction réinitialise l'exécution de la routine SFC spécifiée à une étape particulière.
Post-scrutation	N/A



## Exemple

## Diagramme à relais



## Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

**Fin temporaire (TND)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction TND termine de façon conditionnelle une routine.

**Langages disponibles****Diagramme à relais**

—(TND)—

**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

TND();

**Opérandes**

**Diagramme à relais**

Aucun

**Texte structuré**

Aucun

**Description**

Lorsqu'elle est activée, l'instruction TND agit comme la fin de la routine. Si l'instruction TND se trouve dans une sous-routine, le contrôle retourne à la routine d'appel. Si l'instruction TND se trouve dans la routine principale, le contrôle retourne au prochain programme de la tâche courante.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie.	Le routine se termine
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau du Diagramme à relais.

**Texte structuré**

```
InputA[:=] OutputB;
```

```
IF (InputA) THEN
```

```
    TND();
```

```
END_IF;
```

```
InputE [:=] OutputF;
```

**Voir aussi**

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Toujours faux \(AFI\)](#) sur la [page 626](#)

[Relais de contrôle maître \(MCR\)](#) sur la [page 646](#)

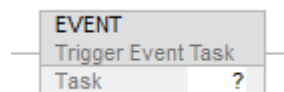
[Pas d'opération \(NOP\)](#) sur la [page 650](#)

[Attributs communs](#) sur la [page 885](#)

**Déclenchement de tâche événementielle (EVENT)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction EVENT déclenche une exécution d'une tâche d'événement.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

```
EVENT(task_name);
```

## Opérandes

### Diagramme à relais

Opérande	Type	Format	Description
Task	TASK	nom	Tâche d'événement à exécuter. Si la tâche spécifiée n'est pas la tâche d'événement, alors elle ne sera pas exécutée.

### Texte structuré

Opérande	Type	Format	Description
Task	TASK	nom	Tâche d'événement à exécuter. Si la tâche spécifiée n'est pas la tâche d'événement, alors elle ne sera pas exécutée.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description

Utiliser l'instruction EVENT pour exécuter par programmation une tâche d'événement.

Chaque fois que l'instruction est exécutée, elle déclenche la tâche d'événement spécifiée.

Assurez-vous d'accorder à la tâche d'événement suffisamment de temps pour terminer son exécution avant de la déclencher à nouveau. Sinon, un chevauchement va se produire.

Si vous exécutez une instruction EVENT alors que la tâche d'événement est en cours d'exécution, l'automate incrémente le compteur de chevauchement mais il ne déclenche pas la tâche d'événement.

L'instruction EVENT peut être utilisée pour déclencher la tâche d'événement avec tous les types de déclencheur.

### Déterminer par programmation si une instruction EVENT a déclenché une tâche

Pour déterminer si une instruction EVENT a déclenché une tâche d'événement, utilisez l'instruction Lire la valeur système (GSV) pour surveiller l'attribut Status de la tâche.

Attribut	Type de données	Instruction	Description	
Status	DINT	GSV SSV	Fournit les informations d'état relatives à la tâche. Une fois que l'automate définit un bit, vous devez le mettre à zéro manuellement pour déterminer si un autre défaut de ce type s'est produit.	
			<b>À déterminer si</b>	<b>Examiner ce bit</b>
			Une instruction EVENT a déclenché la tâche (tâche d'événement uniquement)	0
			Un délai d'expiration a déclenché la tâche (tâche d'événement uniquement)	1
			Un chevauchement s'est produit pour cette tâche	2

L'automate ne met pas à zéro les bits de l'attribut Status, une fois qu'ils sont définis. Pour utiliser un bit afin d'avoir des informations sur un nouvel état, vous devez manuellement mettre à zéro le bit. Utiliser l'instruction Définir la valeur système (SSV) pour définir l'attribut sur une valeur différente.

#### Affecte les indicateurs d'état mathématique

Non

#### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

#### Exécution

#### Diagramme à relais

Condition	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

#### Texte structuré

Condition	Action entreprise
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute.
Post-scrutation	N/A

## Exemples

### Exemple 1

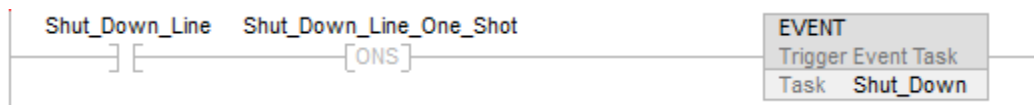
Un automate utilise plusieurs programmes, mais la procédure d'arrêt utilisée est commune. Chaque programme utilise une étiquette d'accès du programme appelée Shut\_Down\_Line qui est activée si le programme détecte une condition qui nécessite un arrêt. La logique de chaque programme est exécutée comme suit :

Si Shut\_Down\_Line = activée (conditions requises pour un arrêt) alors

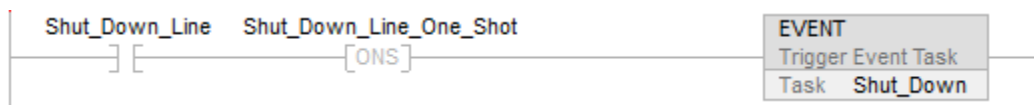
Exécute la tâche Shut\_Down une fois

### Diagramme à relais

#### Programme A



#### Programme B



### Texte structuré

#### Programme A

```
IF Shut_Down_Line AND NOT Shut_Down_Line_One_Shot THEN
```

```
EVENT (Shut_Down);
```

```
END_IF;
```

```
Shut_Down_Line_One_Shot:=Shut_Down_Line;
```

#### Programme B

```
IF Shut_Down_Line AND NOT Shut_Down_Line_One_Shot THEN
```

```
EVENT (Shut_Down);
```

```
END_IF;
```

```
Shut_Down_Line_One_Shot:=Shut_Down_Line;
```

### Exemple 2

L'exemple suivant utilise une instruction EVENT pour initialiser une tâche d'événement. Un autre type d'événement déclenche normalement la tâche d'événement.

#### Tâche continue

IF Initialize\_Task\_1 = 1 THEN

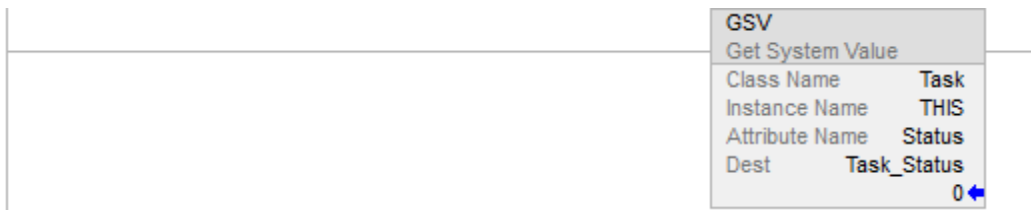
L'instruction ONS limite l'exécution de l'instruction EVENT à 1 scutation.

L'instruction EVENT déclenche une exécution de Task\_1 (tâche d'événement).



#### Task\_1 (tâche d'événement)

L'instruction GSV définit Task\_Status (étiquette DINT) = attribut Statut de la tâche d'événement. Dans l'attribut Instance Name, THIS représente l'objet TASK de la tâche dans laquelle l'instruction se trouve (par exemple, Task\_1).



Si Task\_Status.0 = 1, alors une instruction EVENT a déclenché la tâche d'événement (c.-à-d. lorsque la tâche continue exécute son instruction EVENT pour initialiser la tâche d'événement).

L'instruction RES réinitialise un compteur utilisé par la tâche d'événement.

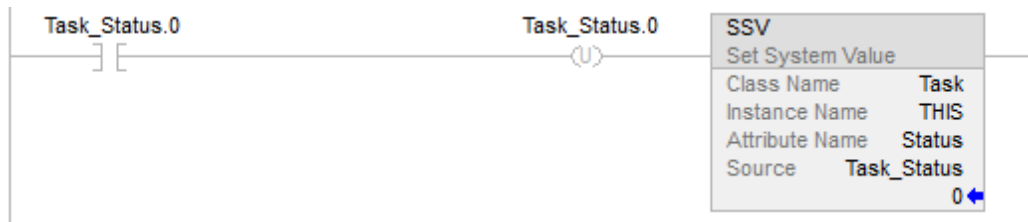


L'automate ne met pas à zéro les bits de l'attribut Status, une fois qu'ils sont définis. Pour utiliser un bit afin d'avoir des informations sur un nouvel état, vous devez manuellement mettre à zéro le bit.

Si Task\_Status.0 = 1, alors ce bit est mis à zéro.

L'instruction OTU définit Task\_Status.0 = 0.

L'instruction SSV définit l'attribut Status de la tâche THIS (Task\_1) = Task\_Status. Cela comprend le bit qui a été mis à zéro.



### Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

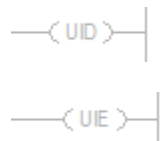
## Désactivation d'interruption par l'utilisateur (UID)/Activation d'interruption par l'utilisateur (UIE)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction UID et l'instruction UIE fonctionnent ensemble pour éviter qu'un petit nombre de critique d'échelons ne soient interrompus par d'autres tâches.

### Langages disponibles

#### Diagrammes à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

```
UID();
```

```
UIE();
```

#### Opérandes

#### Diagramme à relais

Cette instruction n'est pas disponible dans un diagramme à relais.



### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré. Vous devez saisir les parenthèses () après le mnémonique de l'instruction, même s'il n'y a pas d'opérande.

### Description (Description)

Lorsque la condition d'entrée d'échelon est vraie :

- L'instruction UID empêche les tâches de priorité plus élevée d'interrompre la tâche courante, mais elle ne désactive pas l'exécution d'une routine de défaut ou le Gestionnaire de défauts de l'automate.
- L'instruction UIE autorise d'autres tâches à interrompre la tâche courante.

Pour empêcher une série d'échelons d'être interrompus :

1. Limiter le nombre d'échelons qui ne doivent pas être interrompus autant que possible. Le fait de désactiver les interruptions pendant une longue période peut entraîner des pertes de communications.
2. Juste au-dessus du premier échelon qui ne doit pas être interrompu, entrer un échelon avec une instruction UID.
3. Juste après le dernier échelon de la série qui ne doit pas être interrompue, entrer un échelon avec une instruction UIE.
4. Si nécessaire, vous pouvez imbriquer des paires d'instruction UID/UIE.

Lorsque l'instruction UID est appelée pour la première fois, elle augmente la priorité, mémorise la priorité précédente et incrémente un compteur imbriqué. Chaque appel suivant incrémente le comptage. L'instruction UIE décrémentera le compteur imbriqué. Si la nouvelle valeur est 0, cela restaurera la priorité mémorisée.

### Affecter les indicateurs d'état mathématique

Nbre.

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

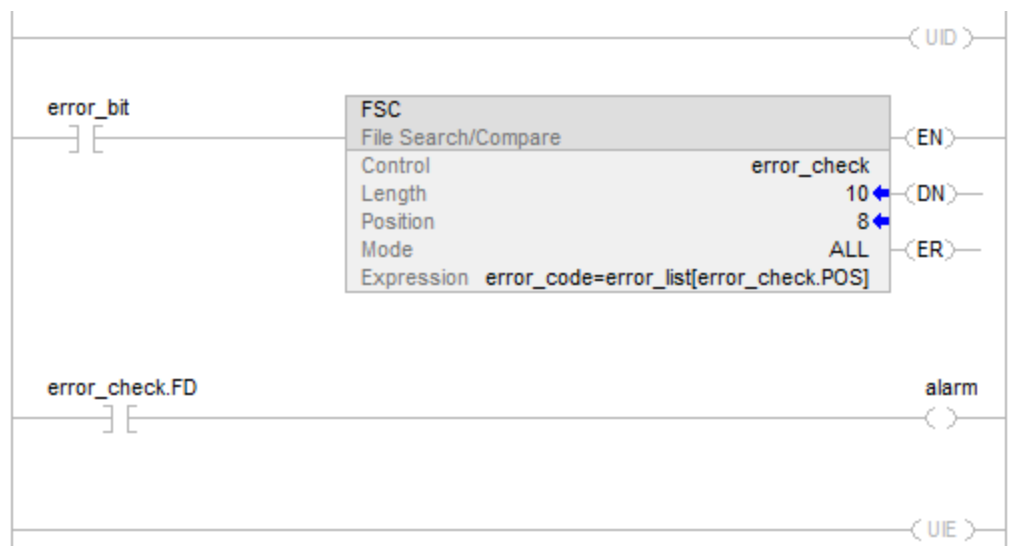
Condition/État	Action (Action)
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction UID empêche la tâche utilisateur contenante d'être interrompue. L'instruction UIE autorise l'interruption de la tâche utilisateur contenante, comme c'est normalement le cas.
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action (Action)
Pré-scrutation	N/A
Exécution normale	L'instruction UID empêche la tâche utilisateur contenante d'être interrompue. L'instruction UIE autorise l'interruption de la tâche utilisateur contenante, comme c'est normalement le cas.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Texte structuré**

UID();

<statements>

UIE());

**Voir aussi**

[Instructions de contrôle du programme](#) sur la [page 624](#)

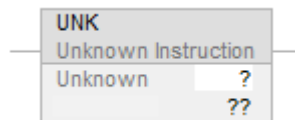
[Attributs communs](#) sur la [page 885](#)

## Instruction inconnue (UNK)

L’instruction UNK sert à indiquer que vous avez saisi un type d’instruction qui n’est pas défini à l’intérieur du jeu d’instructions de Logix Designer.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel

**Texte structuré**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Opérandes**

**Diagramme à relais**

Opérande	Type	Format	Description
Inconnu	immédiate	immédiate	

**Voir aussi**

[Instructions de contrôle du programme](#) sur la [page 624](#)



## Instructions de fin/rupture

### Instructions de fin/rupture

Utiliser l’instruction FOR pour appeler de façon répétée une sous-routine.  
Utiliser l’instruction BRK pour interrompre l’exécution d’une sous-routine.

#### Instructions disponibles

#### Diagramme à relais

<a href="#">FOR</a>	<a href="#">BRK</a>
---------------------	---------------------

Utiliser l’instruction FOR pour appeler de façon répétée une sous-routine. Utiliser l’instruction BRK pour interrompre l’exécution de la sous-routine.

Si vous voulez :	Utilisez cette instruction :
Exécute une routine de façon répétée.	Pour (FOR)
Termine l'exécution répétée d'une routine.	Rupture (BRK)
Retourne à l’instruction FOR	Retour (RET)

#### Voir aussi

[Instructions de calcul/Math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

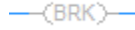
### Rupture (BRK)

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction BRK interrompt l’exécution d’une routine qui a été appelée par une instruction FOR.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Description (Description)

Lorsqu'elle est activée, l'instruction BRK fait quitter la routine et rend le contrôle à la routine contenant l'instruction FOR la plus récemment exécutée, reprenant l'exécution juste après cette instruction. Si aucune instruction FOR ne précédait cette instruction BRK dans son exécution durant cette scrutation, alors l'instruction BRK ne fait rien.

S'il y a des instructions FOR imbriquées, une instruction BRK rend le contrôle à l'instruction FOR la plus imbriquée.

### Affecter les indicateurs d'état mathématique

Non

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

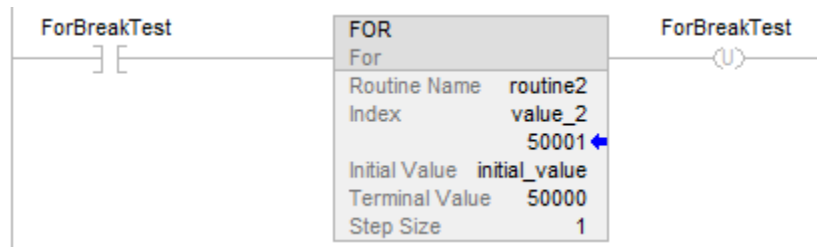
### Diagramme à relais

Condition/État	Action (Action)
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

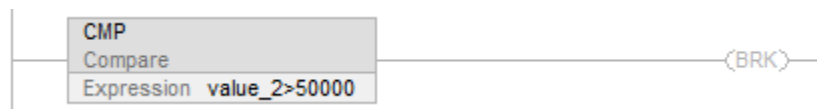
### Exemple

Lorsqu'elle est activée, l'instruction BRK arrête l'exécution de la routine courante et retourne à l'instruction qui suit l'instruction FOR d'appel.

### Diagramme à relais



Voici la routine2 :



### Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Instructions de fin/rupture](#) sur la [page 669](#)

[Pour \(FOR\)](#) sur la [page 671](#)

[Saut vers une étiquette \(JMP\) et Etiquette \(LBL\)](#) sur la [page 633](#)

[Saut vers sous-routine \(JSR\), Sous-routine \(SBR\), et Retour \(RET\)](#) sur la [page 636](#)

## Pour (FOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction FOR exécute une routine externe de façon répétée.

## Langages disponibles

### Diagramme à relais

<b>FOR</b>	
For	
Routine Name	?
Index	?
	??
Initial Value	?
Terminal Value	?
Step Size	?

### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description
Routine name	ROUTINE	étiquette	Sous-routine qui est appelée chaque fois que la boucle FOR est exécutée.
Index	DINT	étiquette	Compte combien de fois la routine a été exécutée
Initial value	SINT INT DINT	Immédiat étiquette	Valeur à laquelle démarrer l'index
Terminal value	SINT INT DINT	Immédiat étiquette	Valeur à laquelle arrêter l'exécution de la routine
Step size	SINT INT DINT	Immédiat étiquette	Quantité à ajouter à l'index chaque fois que l'instruction FOR exécute la routine

### Description

Quand elle est activée, l'instruction FOR exécute de façon répétée la routine jusqu'à ce que la valeur Index dépasse Terminal value. Cette instruction ne transmet pas de paramètres à la routine.

La valeur Step peut être positive ou négative. Si la valeur est négative, la boucle se termine lorsque Index est inférieur à Terminal value. Si la valeur est positive, la boucle se termine lorsque Index est supérieur à Terminal value.



Chaque fois que l'instruction FOR exécute la routine, elle ajoute la taille de l'incrément à Index.

Veillez à ne pas effectuer un trop grand nombre de boucles dans une seule scrutation. Un nombre excessif de répétitions peut provoquer un délai d'expiration du chien de garde de l'automate, ce qui provoque un défaut majeur.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

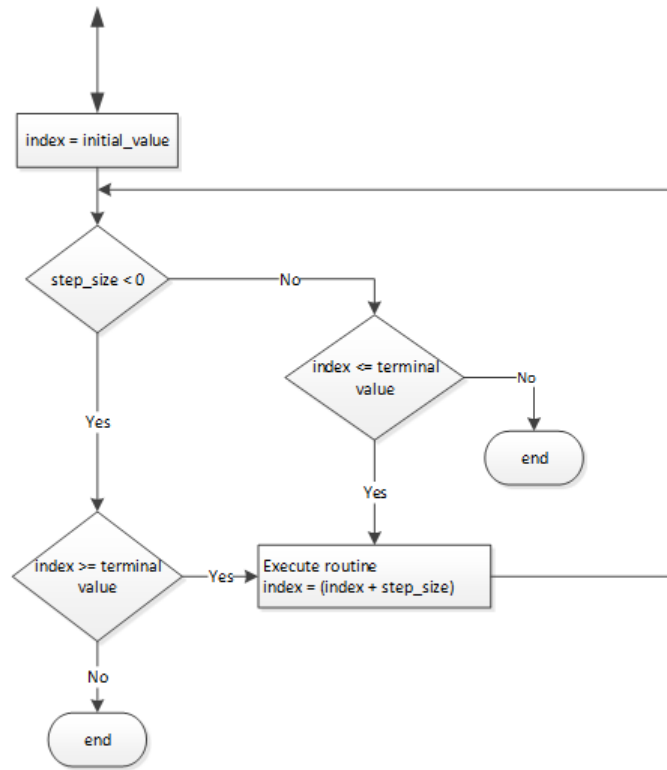
Automates	Un défaut majeur se produira si :	Type de défaut	Code de défaut
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Limite de niveaux d'imbrication > 25	4	94
	La sous-routine est un SFC qui est déjà en cours d'exécution (appel récursif)	4	82
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	N/A	N/A	N/A

Reportez-vous à *Attributs communs* pour obtenir plus d'informations sur les défauts liés aux opérandes.

### Exécution

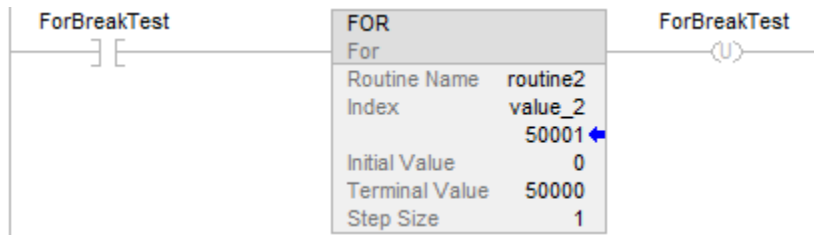
Condition/État	Action
Pré-scrutation	L'instruction va pré-scruter la sous-routine nommée si elle n'a jamais été pré-scrutée auparavant. <b>Astuce</b> : si des instructions FOR récursives existent vers la même sous-routine ou si des instructions FOR multiples existent (non récursives) vers la même sous-routine, la sous-routine n'est pré-scrutée qu'une seule fois. Cela est vrai si le subordonné a été présruté par une instruction JSR.
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Reportez-vous à l'Organigramme FOR (vrai) suivant.
Post-scrutation	L'instruction va postscruter la sous-routine nommée exactement une seule fois.

**Organigramme FOR (vrai)**



**Exemples**

Lorsqu'elle est activée, l'instruction FOR exécute de façon répétée routine\_2 et incrémente value\_2 de 1 à chaque fois. Lorsque value\_2 est > 50 000 ou qu'une instruction BRK est activée, l'instruction FOR cesse d'exécuter routine\_2.



**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

## Saut vers sous-routine (JSR), Sous-routine (SBR), et Retour (RET)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

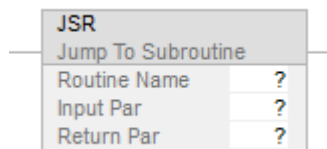
L'instruction JSR appelle une autre routine. Lorsque cette routine se termine, l'exécution retourne à l'instruction JSR.

L'instruction SBR reçoit les paramètres d'entrée passés par JSR.

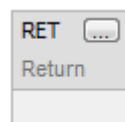
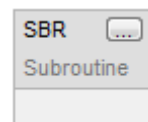
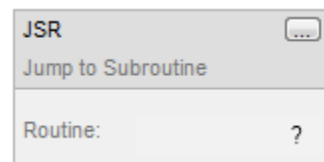
L'instruction RET renvoie les paramètres de retour à JSR et termine la scrutation de la sous-routine.

### Langages disponibles

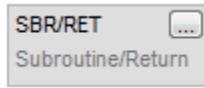
#### Diagramme à relais



#### Bloc fonctionnel



### Grphe de fonctionnement séquentiel



#### Texte structuré

JSR(RoutineName,InputCount,InputPar,ReturnPar);

SBR(InputPar);

RET(ReturnPar);

#### Opérandes

- 
- Important :** Une opération inattendue peut se produire dans les cas suivants :
- Les opérandes d'étiquette de sortie sont remplacés.
  - Les membres d'un opérande de structure sont remplacés.
  - Sauf lorsque cela est spécifié, les opérandes de structure sont partagés par plusieurs instructions.
- 



Pour chaque paramètre dans les instructions SBR ou RET, utiliser le même type de données (y compris toute dimension de tableau) que le paramètre correspondant dans l'instruction JSR. L'utilisation de différents types de données peut produire des résultats imprévus.

---

#### Diagramme à relais

##### Instruction JSR

Opérande	Type de donnée	Type de donnée	Format	Description (Description)
	Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580		
Routine Name	ROUTINE	ROUTINE	nom	Sous-routine à exécuter

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Input Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	immédiate étiquette étiquette de tableau	Les données de cette routine à copier dans une étiquette de la sous-routine. <ul style="list-style-type: none"> <li>• Les paramètres d'entrée sont optionnels</li> <li>• Entrer un nombre maximal de 40 paramètres d'entrée, si nécessaire.</li> </ul>
Return Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	étiquette étiquette de tableau	Étiquette dans cette routine pour copier le résultat de la sous-routine. <ul style="list-style-type: none"> <li>• Les paramètres de retour sont optionnels</li> <li>• Entrer un nombre maximal de 40 paramètres de retour, si nécessaire</li> </ul>

**Instruction SBR**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Input Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	étiquette étiquette de tableau	<ul style="list-style-type: none"> <li>Étiquette dans cette routine dans laquelle copier les paramètres d'entrée correspondant (40 au maximum) de l'instruction JSR.</li> </ul>

**Instruction RET**

Opérande	Type de donnée Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Type de donnée Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Format	Description (Description)
Return Par	BOOL SINT INT DINT REAL structure	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL structure	Immédiate étiquette étiquette de tableau	Données de cette routine dans laquelle copier les paramètres de retour correspondant (40 au maximum) de l'instruction JSR.

**Affecter les indicateurs d'état mathématique**

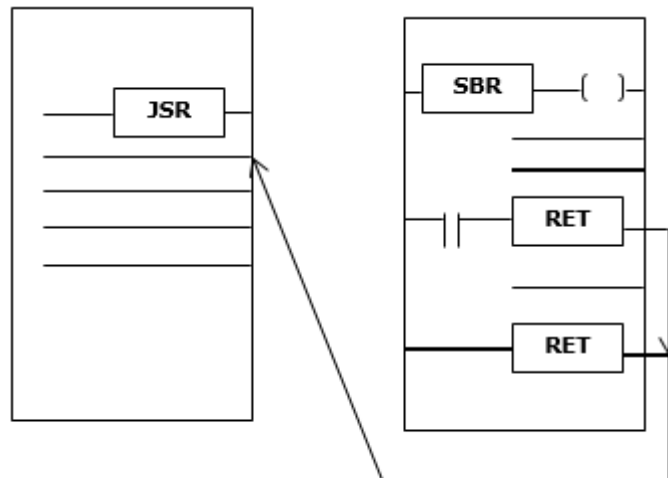
Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
L'instruction JSR a moins de paramètres d'entrée que l'instruction SBR	4	31
L'instruction JSR saute à une routine de défaut	4	990 ou fourni par l'utilisateur
L'instruction RET a moins de paramètres de retour que l'instruction JSR	4	31
La routine principale contient une instruction RET	4	31

### Fonctionnement

**Important :** N'importe quelle routine peut contenir une instruction JSR mais une instruction JSR ne peut pas appeler (exécuter) une routine principale.



L'instruction JSR lance l'exécution de la routine spécifiée, qui est appelée une sous-routine :

- La sous-routine est exécutée chaque fois qu'elle est scrutée.
- Après l'exécution de la sous-routine, l'exécution de la logique retourne à la routine qui contient l'instruction JSR et continue par l'instruction après JSR.

Pour programmer un saut dans une sous-routine, observez ces lignes directrices.

### JSR

- Pour copier des données dans une étiquette de la sous-routine, entrer un paramètre d'entrée.
- Pour copier un résultat d'une sous-routine dans une étiquette de cette routine, entrer un paramètre de retour.

- Entrer jusqu'à 40 entrées et jusqu'à 40 paramètres de retour, si nécessaire.

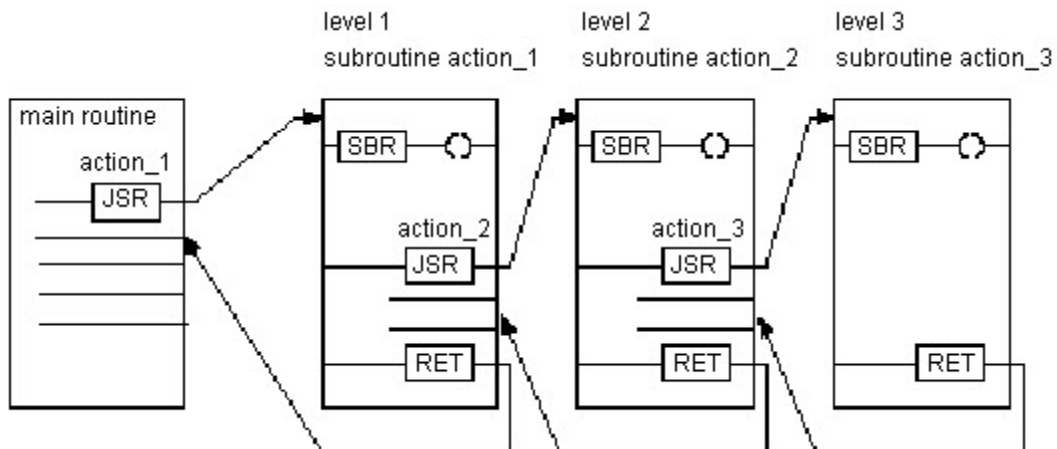
### SBR

- Si l'instruction JSR a un paramètre d'entrée, entrer une instruction SBR.
- Placer l'instruction SBR comme première instruction de la routine.
- Pour chaque paramètre d'entrée de l'instruction JSR, entrer une étiquette dans laquelle vous voulez copier les données.

### RET

- Si l'instruction JSR a un paramètre de retour, entrer une instruction RET.
- Placer l'instruction RET comme dernière instruction de la routine.
- Pour chaque paramètre de retour de l'instruction JSR, entrer un paramètre de retour à envoyer à l'instruction JSR.
- Dans une routine à relais, placer des instructions RET additionnelles pour quitter la sous-routine en fonction de conditions d'entrée différentes, si c'est nécessaire (les routines de Bloc fonctionnel ne permettent qu'une seule instruction RET).

Invoquer jusqu'à 25 sous-routines imbriquées, avec un maximum de 40 paramètres passés à une sous-routine, et un maximum de 40 paramètres renvoyés d'une sous-routine.



Astuce : Sélectionner le menu **Éditer > Éditer un élément à relais (Edit > Edit Ladder Element)** pour ajouter et supprimer des opérands variables. Dans le cas des instructions JSR et SBR, ajouter un paramètre d'entrée. Dans le cas des instructions JSR et RET, ajouter un paramètre de sortie. Pour toutes les trois instructions, supprimer le paramètre d'instruction.



**Exécution****Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	L'échelon est défini sur faux. L'automate exécute toutes les sous-routines. Pour s'assurer que tous les échelons dans la sous-routine sont pré-scruté, l'automate ignore les instructions RET (ce qui veut dire que les instructions RET ne font pas quitter la sous-routine). Les paramètres d'entrée et de retour ne sont pas passés. Si la même sous-routine est invoquée plusieurs fois, elle ne sera pré-scrutée qu'une seule fois.
Condition d'entrée d'échelon est fausse (pour l'instruction JSR)	N/A
Condition d'entrée d'échelon est vraie	Les paramètres sont passés et la sous-routine est exécutée.
Post-scrutation	Même action que pour la pré-scrutation

**Bloc fonctionnel**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
EnableIn est faux	N/A
EnableIn est vrai	Les paramètres sont transmis et la sous-routine est exécutée
Première exécution de l'instruction	N/A
Première scrutation de l'instruction	N/A
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

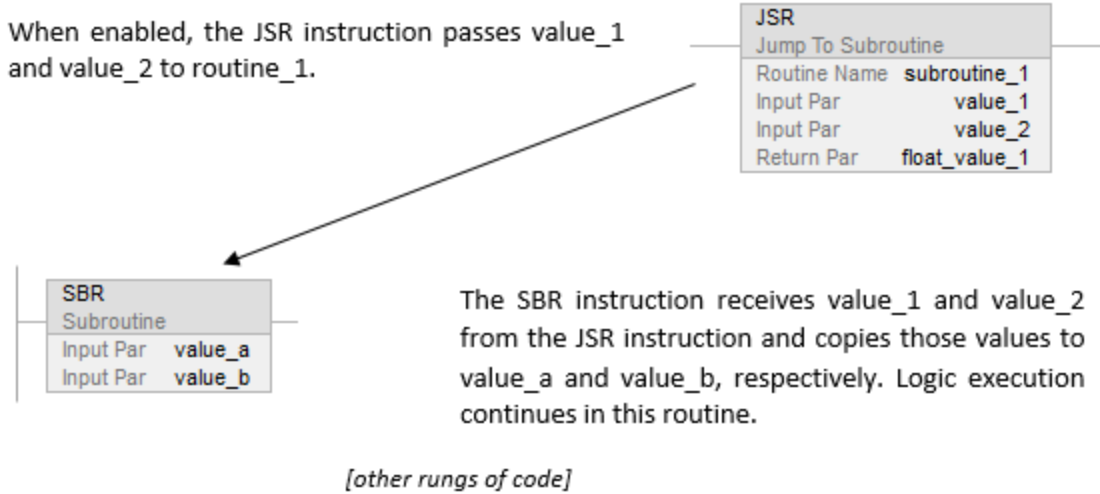
**Texte structuré**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Les paramètres sont passés et la sous-routine est exécutée.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

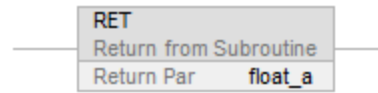
**Exemples**

**Exemple 1**

**Diagramme à relais**



When enabled, the RET instruction sends float\_a to the JSR instruction. The JSR instruction receives float\_a and copies the value to float\_value\_1. Logic execution continues with the next instruction following the JSR instruction.



**Texte structuré**

Routine	Program
Routine principale	JSR(routine_1,2,value_1,value_2,float_value_1);
Sous-routine	SBR(value_a,value_b); <statements>; RET(float_a);

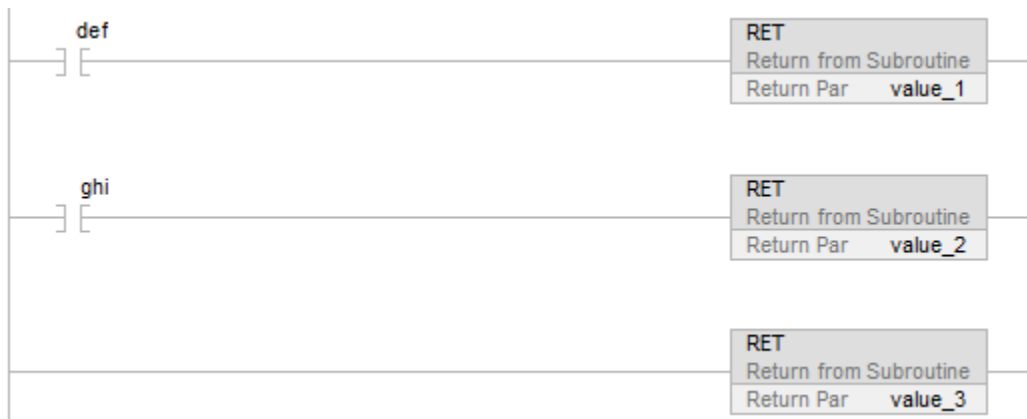
Exemple 2

Diagramme à relais

Routine principale

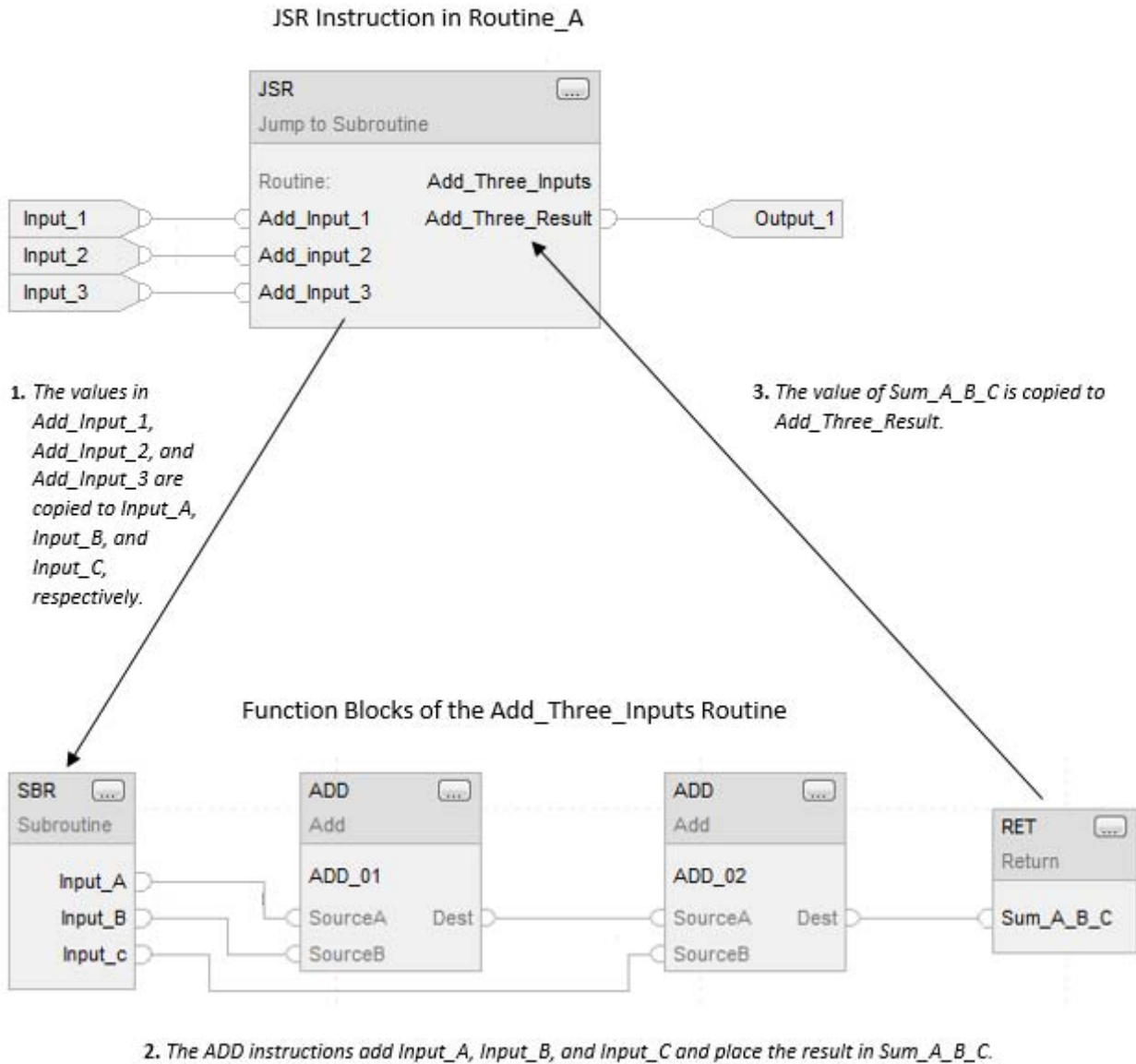


sous-routine\_1



### Exemple 3

#### Bloc fonctionnel



#### Voir aussi

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Valeurs immédiates](#) sur la [page 888](#)

## Instructions spéciales

### Instructions spéciales

Les instructions spéciales accomplissent des opérations spécifiques aux applications.

#### Instructions disponibles

##### Texte structuré

<a href="#">FBC</a>	<a href="#">DDT</a>	<a href="#">DTR</a>	<a href="#">PID</a>
---------------------	---------------------	---------------------	---------------------

##### Bloc fonctionnel

Indisponible

##### Texte structuré

Indisponible

Si vous voulez :	Utilisez cette instruction :
Comparer des données à une référence connue correcte et enregistrer toutes les différences.	FBC
Comparer des données à une référence connue correcte, enregistrer toutes les différences et actualiser la référence pour la faire correspondre à la source.	DDT
Passer les données de la source à travers un masque et comparer le résultat aux données de référence. Puis écrire la source dans la référence pour la comparaison suivante.	DTR
Contrôler une boucle PID.	PID

#### Voir aussi

[Utilisation des instructions PID](#) sur la [page 712](#)

[Anti-saturation d'intégrale et transfert sans à-coup de manuel à automatique \(PID\)](#) sur la [page 716](#)

[Temporisation de l'instruction PID](#) sur la [page 721](#)

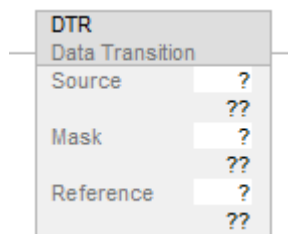
## Transition de données (DTR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction DTR passe la valeur Source à travers Mask et compare le résultat avec la valeur Reference.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	DINT	immediate étiquette	tableau à comparer à référence
Masque (Mask)	DINT	immediate étiquette	bits à bloquer ou à laisser passer
Référence	DINT	étiquette	tableau à comparer à source

### Description (Description)

L'instruction DTR passe la valeur Source à travers Mask et compare le résultat avec la valeur Reference. L'instruction DTR écrit aussi la valeur masquée de Source dans la valeur Reference pour effectuer la prochaine comparaison. Source reste inchangée.

Un « 1 » dans le masque signifie que le bit de données est passé. Un « 0 » dans le masque signifie que le bit de données est bloqué.

Lorsqu'elle est activée, Mask passe les données quand les bits de Mask sont définis, Mask bloque les données quand les bits de Mask sont à zéro.

Quand Source masquée est différente de Reference, EnableOut devient vrai pendant une scrutation. Lorsque Source masquée est identique à Reference, EnableOut est faux.

---

**Important :** Programmer en ligne en utilisant cette instruction peut être dangereux. Si la valeur Reference est différente de la valeur Source, EnableOut devient vrai. Soyez prudent lorsque vous insérez cette instruction et que le processeur est en mode Exécution ou Exécution à distance.

---

### Saisie d'une valeur de masque immédiate

Lorsque vous entrez un masque, le logiciel de programmation passe par défaut à des valeurs décimales. Si vous souhaitez entrer un masque en faisant appel à un autre format, la valeur doit être précédée du préfixe correct.

Préfixe	Description (Description)
16#	hexadécimal (par ex., 16#0F0F)
8#	octal (par ex., 8#16)
2#	binaire (par ex., 2#00110011)

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

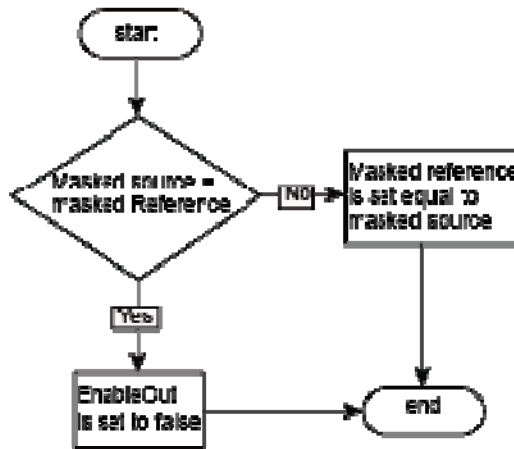
Aucun spécifique à cette instruction. Reportez-vous à Attributs communs pour les défauts liés aux opérandes.

### Exécution

### Diagramme à relais

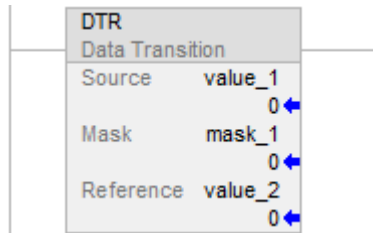
Condition	Action (Action)
Pré-scrutation	Reference = Source ET Mask.
Condition d'entrée d'échelon est fausse	Reference = Source ET Mask.
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme DTR (vrai)
Post-scrutation	N/A

Organigramme DTR (vrai)



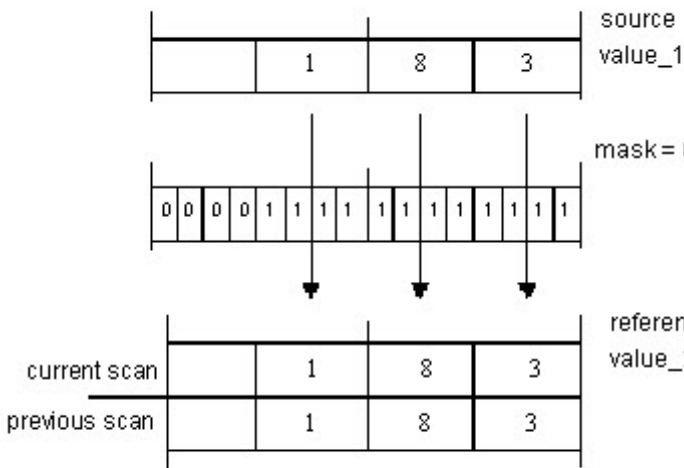
Exemple

Diagramme à relais



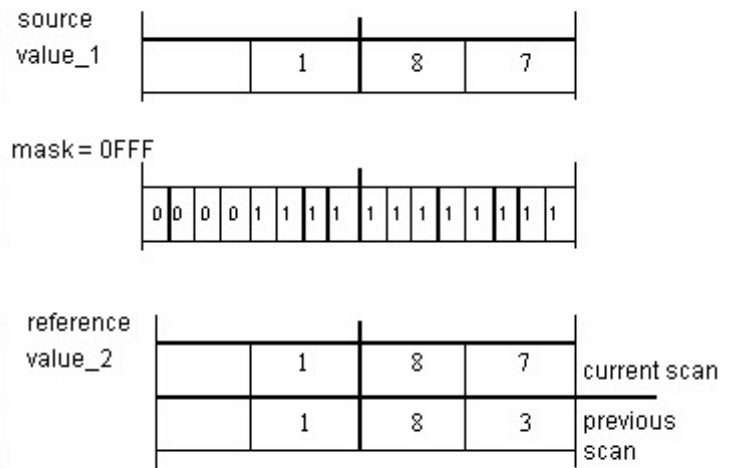
Lorsqu'elle est activée, l'instruction DTR masque value\_1. S'il existe une différence entre les deux valeurs masquées, EnableOut est défini sur vrai.

example 1



The rung remains false as long as the input value does not change.

example 2



The rung remains true for one scan when a change is detected.



Dans l'exemple 1, comme la valeur Reference est égale à sourcevalue\_1 ET Mask, EnableOut sera toujours défini sur faux. Dans l'exemple 2, pour certaines raisons, la valeur source est changée, puis reference\_value n'est plus égal à source\_value ET mask. Dans ce cas, EnableOut sera définie sur VRAI et referencevalue sera mis à jour en fonction de sourceValue et mask. C'est pourquoi vous voyez dans la scrutation précédente que la valeur Reference est 183 mais qu'elle est de 187 dans la scrutation actuelle. L'échelon reste vrai seulement pour la durée d'une scrutation lorsqu'un changement est détecté, car durant la prochaine scrutation, l'échelon restera faux car la valeur Reference sera de nouveau égale à la valeur Source ET Mask.

### Voir aussi

[Instructions spéciales](#) sur la [page 685](#)

[FBC](#) sur la [page 697](#)

[DDT](#) sur la [page 689](#)

[Attributs communs](#) sur la [page 885](#)

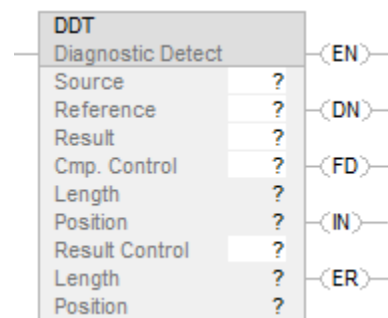
## Détection de diagnostic (DDT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction DDT compare les bits dans un tableau Source avec les bits dans un tableau Référence pour trouver les bits qui sont différents. L'emplacement de chaque bit différent est mémorisé et le bit Référence ne correspondant pas est changé afin de correspondre au bit Source.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	DINT	étiquette de tableau	Tableau à comparer à référence ne pas utiliser CONTROL.POS dans cet indice
Référence	DINT	étiquette de tableau	Tableau à comparer à source ne pas utiliser CONTROL.POS dans cet indice
Result	DINT	étiquette de tableau	Tableau où enregistrer les résultats ne pas utiliser CONTROL.POS dans cet indice
Cmp. Control	CONTROL	structure	Structure de contrôle pour la comparaison
Longueur (Length)	DINT	immediate	Nombre de bits à comparer
Position (Position)	DINT	immediate	Position courante dans la source la valeur initiale est généralement 0
Result control	CONTROL	structure	Structure de contrôle pour les résultats
Longueur (Length)	DINT	immediate	Nombre d'emplacements de mémoire dans le résultat
Position (Position)	DINT	immediate	Position courante dans le résultat la valeur initiale est généralement 0

**Important :** Utiliser des étiquettes différentes pour la structure de contrôle de comparaison et la structure de contrôle des résultats. L'utilisation de la même étiquette pour les deux structures peut entraîner des résultats imprévisibles, et donc potentiellement causer des dommages matériels ou des blessures.

### Structure COMPARE

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction DDT a été activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction DDT compare le dernier bit des tableaux Source et Référence.

.FD	BOOL	Le bit trouvé est défini chaque fois que l'instruction DDT mémorise un bit ne correspondant pas (opération une-différence-à-la-fois) ou après avoir mémorisé toutes les différences de bit (opération tous-par-scrutation)
.IN	BOOL	Le bit d'inhibition indique le mode de recherche DDT. 0 = mode tous-par-scrutation 1 = mode une-différence-à-la-fois
.ER	BOOL	Le bit d'erreur est défini quand POS ou LEN n'est pas valide.
.LEN	DINT	La valeur de la longueur identifie le nombre de bits à comparer.
.POS	DINT	La valeur de la position identifie le bit courant.

### Structure RESULT

Mnémonique	Type de donnée	Description (Description)
.DN	BOOL	Le bit de fin est défini lorsque le tableau Result est plein.
.LEN	DINT	La valeur de la longueur identifie le nombre d'emplacements de mémoire dans le tableau Result.
.POS	DINT	La valeur de la position identifie la position courante dans le tableau Result.

### Description (Description)

Lorsqu'elle est activée, l'instruction DDT compare les bits du tableau Source avec les bits du tableau Référence, elle mémorise le numéro de chaque bit ne correspondant pas dans le tableau Result, et elle change la valeur du bit Référence pour correspondre à la valeur du bit Source correspondant .

---

**Important :** L'instruction DDT fonctionne avec une mémoire contiguë. Vous devez tester et confirmer que l'instruction ne modifie pas des données dont vous ne souhaitez pas la modification.

---

La différence entre les instructions DDT et FBC est que chaque fois que l'instruction DDT trouve une différence, l'instruction DDT change le bit de Référence pour le faire correspondre au bit de Source. L'instruction FBC ne change pas le bit de Référence.

Si cette instruction essaye de lire au-delà de la fin d'un tableau, cette instruction définit le bit .ER et déclenche un défaut majeur.

### Sélection du mode de recherche

Si vous voulez détecter :	Sélectionnez ce mode :
Une différence à la fois	Définir le bit .IN dans la structure CONTROL de comparaison. Chaque fois que EnableIn passe de faux à vrai, l'instruction DDT recherche la différence suivante entre les tableaux Source et Référence. Lorsqu'elle trouve une différence, l'instruction s'arrête, définit le bit .FD, et mémorise la position de la différence.
Toutes les différences	Mettre à zéro le bit .IN dans la structure CONTROL de comparaison. Chaque fois que EnableIn passe de faux à vrai, l'instruction DDT recherche toutes les différences entre les tableaux Source et Référence.

### Affecter les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut majeur se produira si :	Type de défaut	Code de défaut
result.POS > taille du tableau résultat	4	20

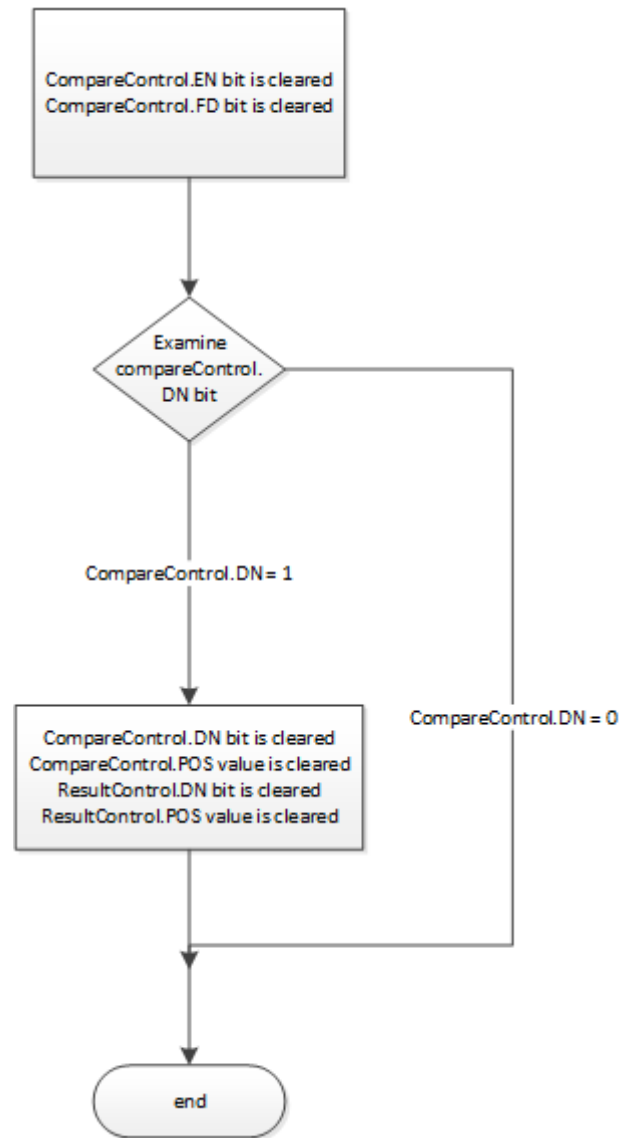
Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

### Exécution

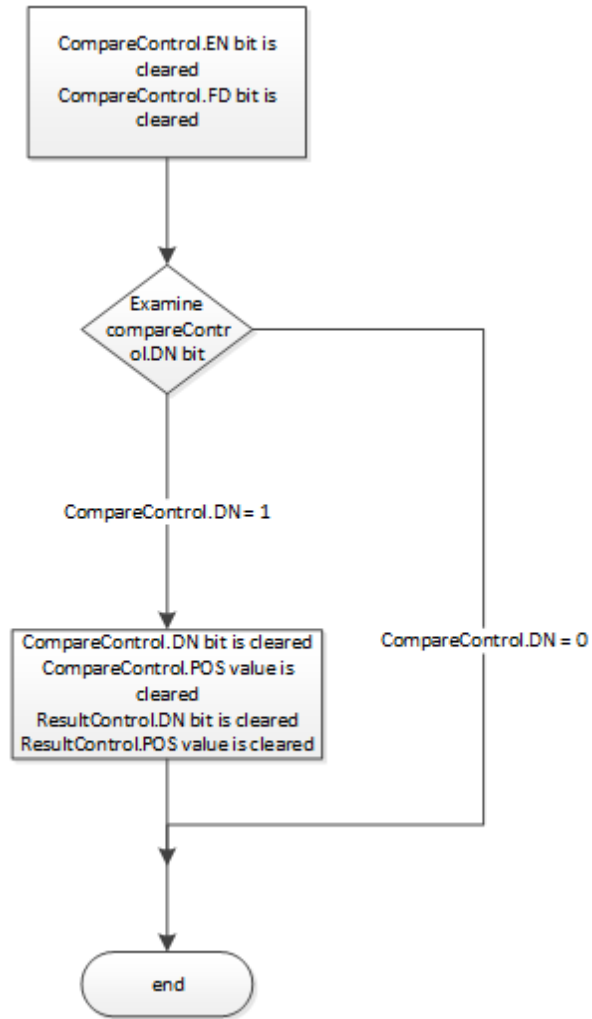
#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l'organigramme DDT (pré-scrutation)
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme DDT (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme DDT (vrai)
Post-scrutation	N/A

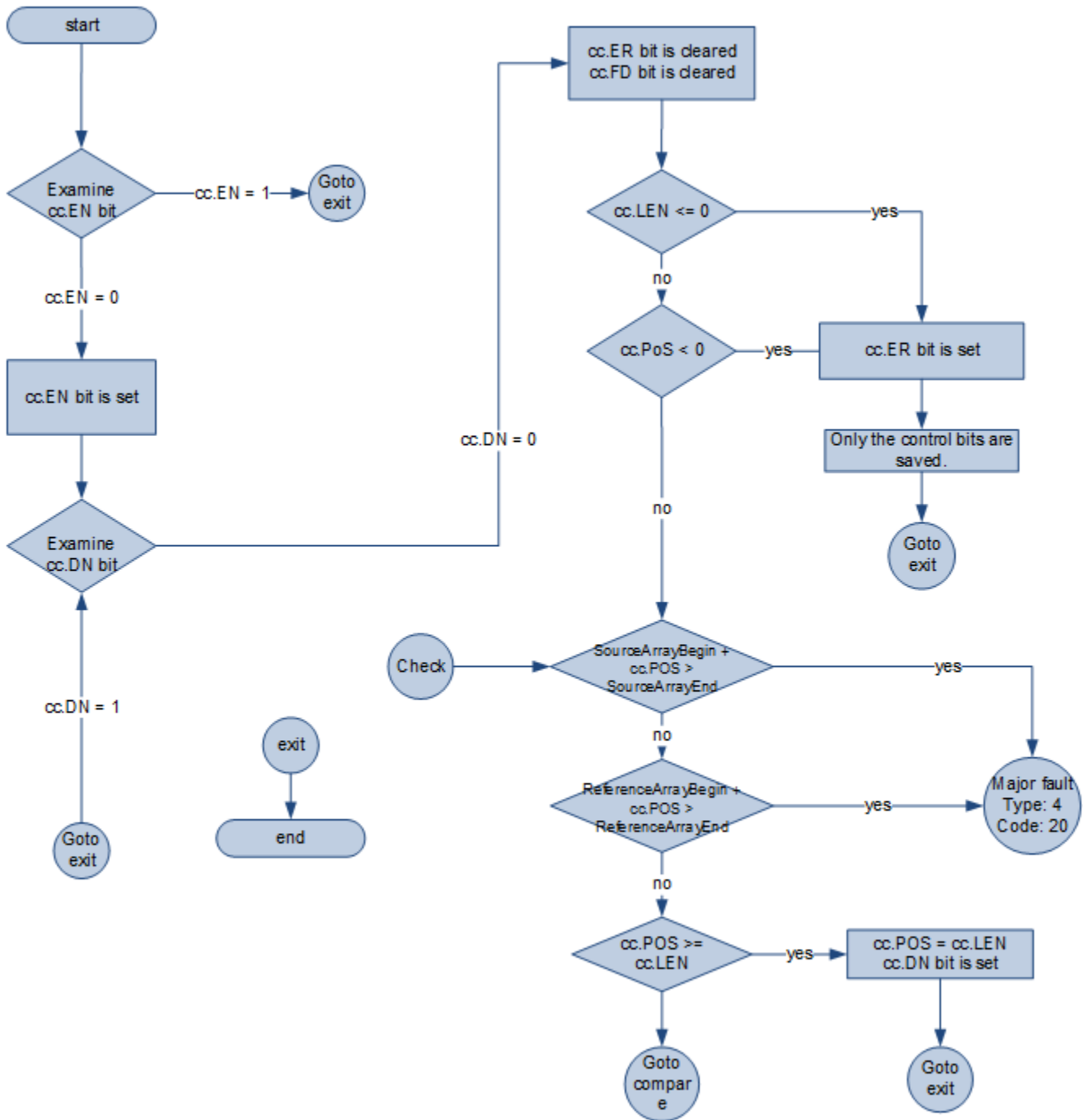
Organigramme DDT (pré-scrutation)



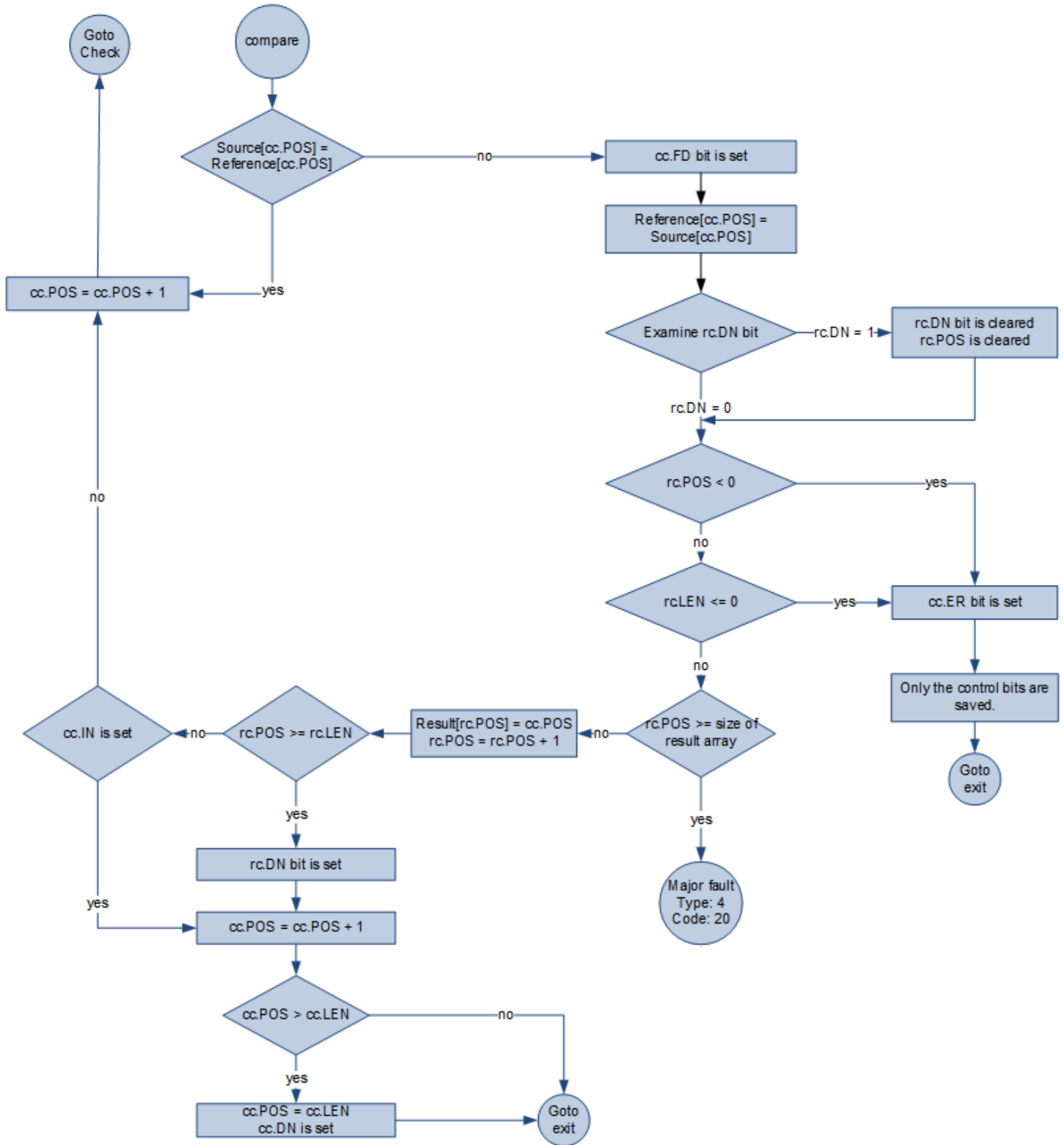
Organigramme DDT (faux)



Organigramme DDT (vrai)



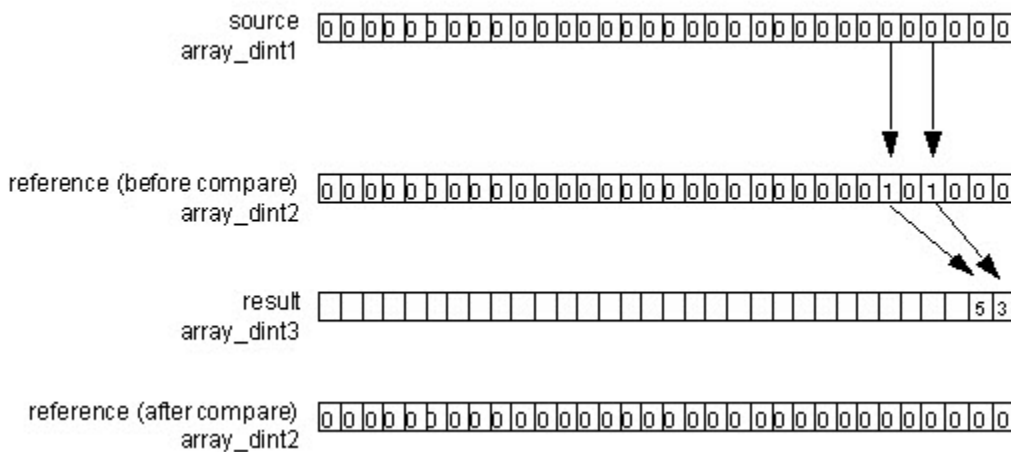
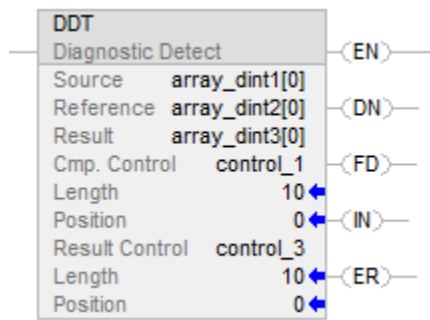
Organigramme DDT (vrai) – Suite





**Exemples**

**Diagramme à relais**



**Voir aussi**

[Instructions spéciales](#) sur la [page 685](#)

[DTR](#) sur la [page 686](#)

[FBC](#) sur la [page 697](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

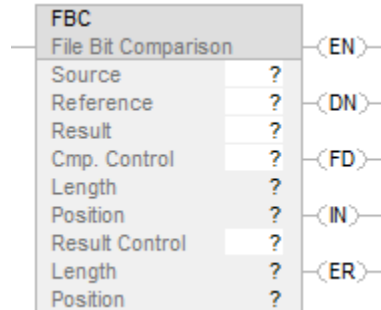
**Comparaison de fichier de bits (FBC)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction FBC compare les bits dans un tableau Source avec les bits dans un tableau Référence.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

#### Diagramme à relais

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	DINT	étiquette de tableau	Tableau à comparer à référence <b>n'utiliser pas</b> CONTROL.POS comme indice
Référence	DINT	étiquette de tableau	Tableau à comparer à source ne pas utiliser CONTROL.POS dans cet indice
Result	DINT	étiquette de tableau	Tableau où enregistrer le résultat n'utiliser pas CONTROL.POS comme indices
Cmp. Control	CONTROL	structure	Structure de contrôle pour la comparaison
Longueur (Length)	DINT	immediate	Nombre de bits à comparer
Position (Position)	DINT	immediate	Position courante dans la source valeur initiale typique : 0
Result control	CONTROL	structure	Structure de contrôle pour les résultats
Longueur (Length)	DINT	immediate	nombre d'emplacements de mémoire dans le résultat
Position (Position)	DINT	immediate	Position courante dans le résultat valeur initiale typique : 0

**Important :** Utiliser des étiquettes différentes pour la structure de contrôle de comparaison et la structure de contrôle des résultats. L'utilisation de la même étiquette pour les deux structures peut entraîner des résultats imprévisibles, et donc potentiellement causer des dommages matériels ou des blessures.

### Structure COMPARE

Mnémonique	Type de donnée	Description (Description)
.EN	BOOL	Le bit d'activation indique que l'instruction FBC a été activée.
.DN	BOOL	Le bit de fin est défini lorsque l'instruction FBC compare le dernier bit des tableaux Source et Reference.
.FD	BOOL	Le bit trouvé est défini chaque fois que l'instruction FBC mémorise un bit ne correspondant pas (opération une-différence-à-la-fois) ou après avoir mémorisé toutes les différences de bit (opération tous-par-scrutation).
.IN	BOOL	Le bit d'inhibition indique le mode de recherche FBC. 0 = mode tous-par-scrutation 1 = mode une-différence-à-la-fois
.ER	BOOL	Le bit d'erreur est défini quand POS ou LEN n'est pas valide.
.LEN	DINT	La valeur de la longueur identifie le nombre de bits à comparer.
.POS	DINT	La valeur de la position identifie le bit courant.

### Structure RESULT

Mnémonique	Type de donnée	Description (Description)
.DN	BOOL	Le bit de fin est défini lorsque le tableau Result est plein.
.LEN	DINT	La valeur de la longueur identifie le nombre d'emplacements de mémoire dans le tableau Result.
.POS	DINT	La valeur de la position identifie la position courante dans le tableau Result.

### Description (Description)

Lorsqu'elle est activée, l'instruction FBC compare les bits du tableau Source avec les bits du tableau Reference et elle mémorise le numéro de chaque bit ne correspondant pas dans le tableau Result.

**Important :** l'instruction FBC fonctionne avec une mémoire contiguë. Vous devez tester et confirmer que l'instruction ne change pas les données dont vous ne voulez pas que la valeur change.

La différence entre les instructions DDT et FBC est que chaque fois que l'instruction DDT trouve une différence, l'instruction change le bit de Référence

pour le faire correspondre au bit de Source. L'instruction FBC ne change pas le bit de Référence.

Si cette instruction essaye de lire au-delà de la fin d'un tableau, cette instruction définit le bit .ER et déclenche un défaut majeur.

**Sélection du mode de recherche**

Si vous voulez détecter :	Sélectionnez ce mode :
Une différence à la fois	Définir le bit .IN dans la structure CONTROL de comparaison. Chaque fois que EnxableIn passe de faux à vrai, l'instruction FBC recherche la différence suivante entre les tableaux Source et Reference. Lorsqu'elle trouve une différence, l'instruction définit le bit .FD, mémorise la position de la différence et son exécution s'arrête.
Toutes les différences	Mettre à zéro le bit .IN dans la structure CONTROL de comparaison. Chaque fois que EnableIn passe de faux à vrai, l'instruction FBC recherche toutes les différences entre les tableaux Source et Reference.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Un défaut majeur se produira si :	Type de défaut	Code de défaut
result.POS > taille du tableau résultat	4	20

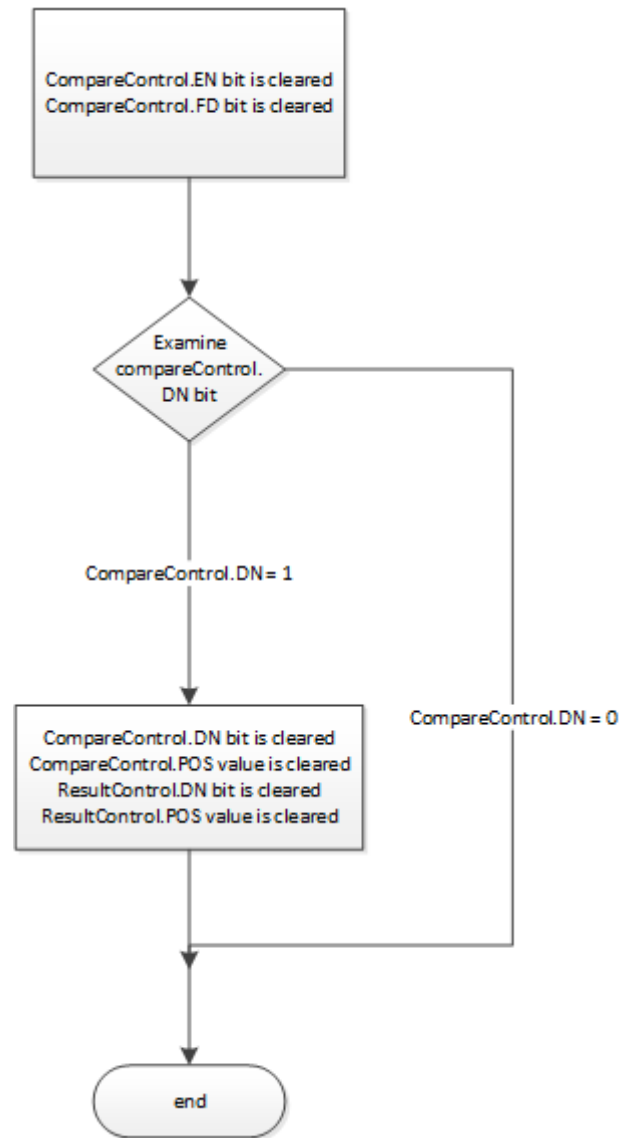
Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

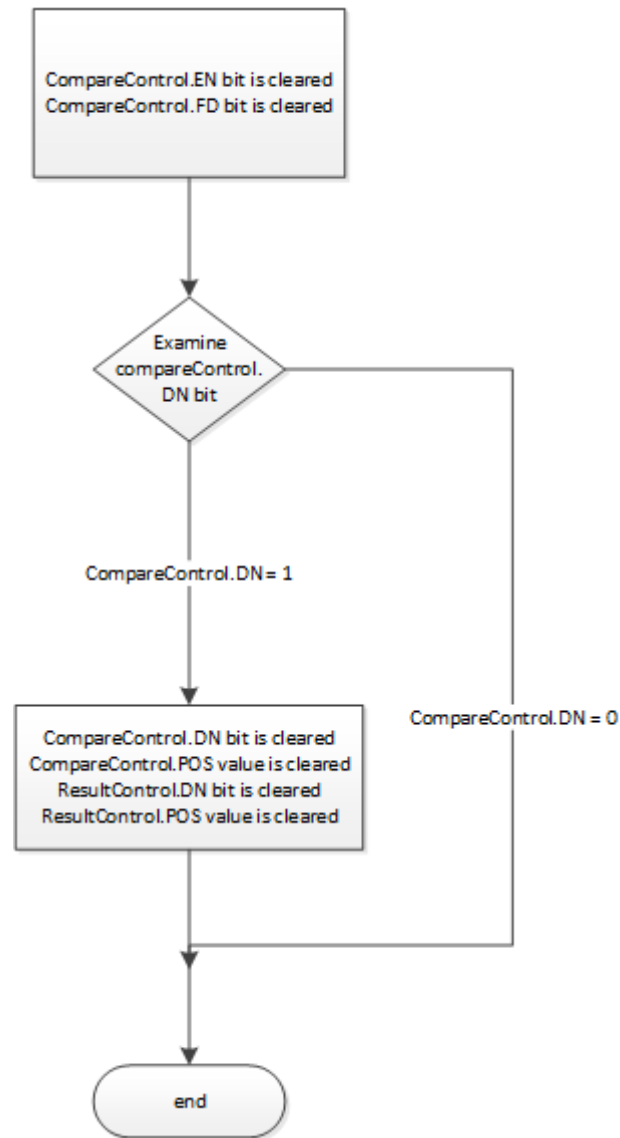
**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à l'organigramme FBC (pré-scrutation)
Condition d'entrée d'échelon est fausse	Reportez-vous à l'organigramme FBC (faux)
Condition d'entrée d'échelon est vraie	Reportez-vous à l'organigramme FBC (vrai)
Post-scrutation	N/A

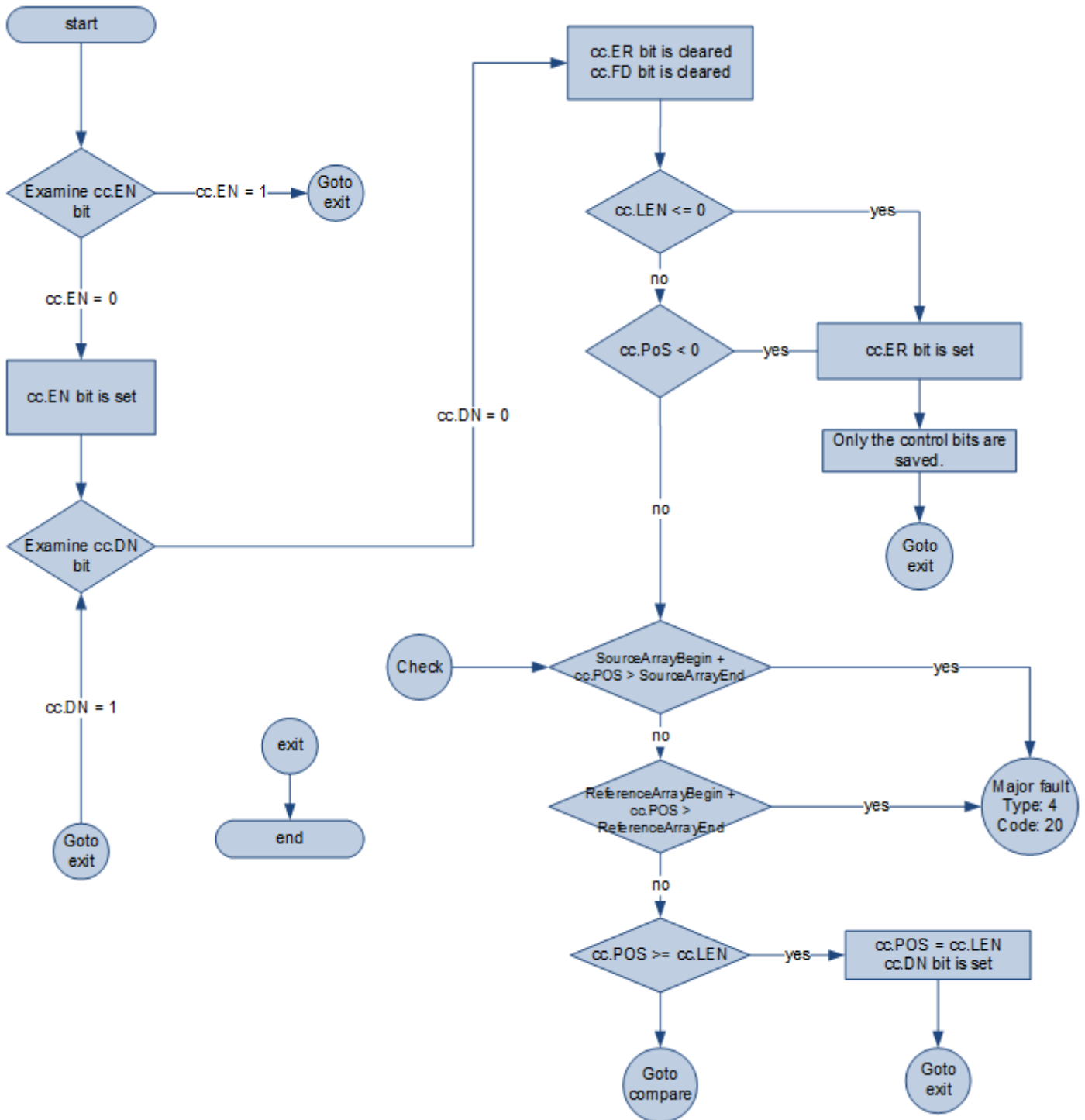
Organigramme FBC (pré-scrutation)



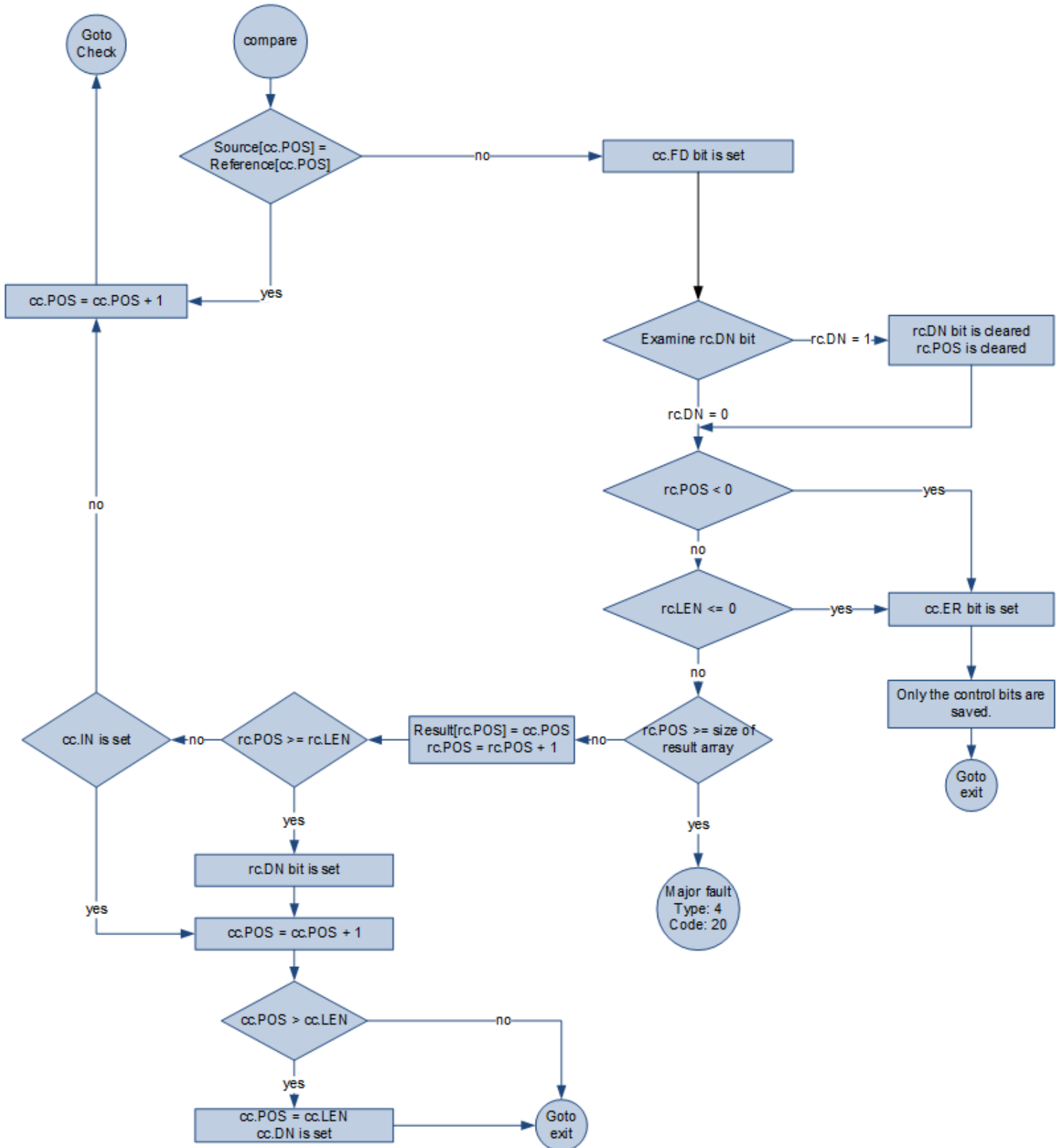
### Organigramme FBC (faux)



Organigramme FBC (vrai)



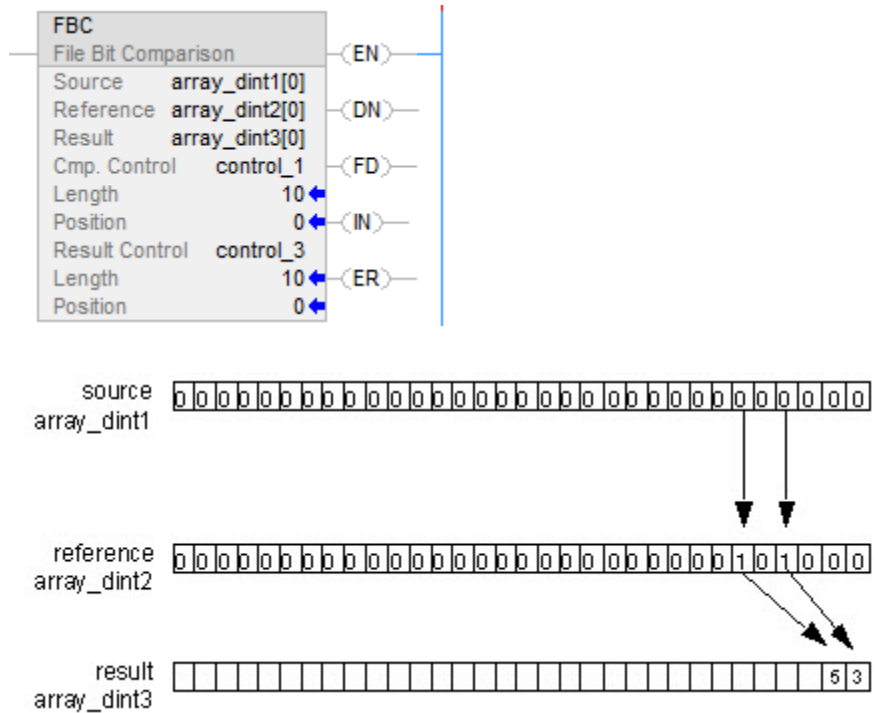
Organigramme FBC (vrai) – Suite





**Exemple**

**Diagramme à relais**



**Voir aussi**

[Instructions spéciales](#) sur la [page 685](#)

[DDT](#) sur la [page 689](#)

[DTR](#) sur la [page 686](#)

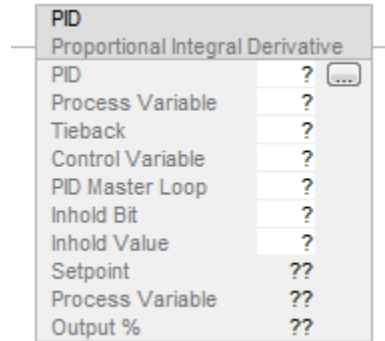
**Proportionnel, intégral et dérivé (PID)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction PID contrôle une variable de procédé comme un débit, une pression, une température ou un niveau.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

PID(PID,ProcessVariable,Tieback,ControlVariable,PIDMasterLoop,InHoldBit,InHoldValue);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
PID	PID	structure	Structure PID
Process variable	SINT	étiquette	La valeur que vous voulez contrôler
	INT		
	DINT		
	REAL		
Tieback	SINT	immediate	(facultatif)
	INT	étiquette	
	DINT		Sortie d'une station matérielle portable/auto qui contourne la sortie de l'automate. Entrer 0 si vous ne voulez pas utiliser ce paramètre
	REAL		

Variable de contrôle	SINT	étiquette	La valeur qui est envoyée au dispositif final de commande (valve, amortisseur, etc.)
	INT		
	DINT		Si vous utilisez la plage morte, la Control variable doit être du type REAL, sinon elle sera forcée à 0 lorsque l'erreur se trouve à l'intérieur de la plage morte.
	REAL		
PID master loop	PID	Structure	Facultatif
			Étiquette PID pour le PID maître
			Si vous effectuez un contrôle en cascade, et que ce PID est une boucle esclave, entrer le nom du PID maître
			Entrer 0 si vous ne voulez pas utiliser ce paramètre
Inhold bit	BOOL	étiquette	Facultatif
			État courant du bit de maintien d'un canal de sortie analogique 1756
			pour prendre en charge un redémarrage sans à-coups
Inhold value	SINT	étiquette	Facultatif
	INT		Valeur de retour de données à partir d'un sortie analogique 1756
	DINT		Canal pour prendre en charge le redémarrage sans à-coups
	REAL		Entrer 0 si vous ne voulez pas utiliser ce paramètre
Point de consigne			Uniquement pour l'affichage
			Valeur actuelle du point de consigne
Process variable			Uniquement pour l'affichage
			Valeur actuelle de Process_Variable à l'échelle
Output %			Uniquement pour l'affichage
			Valeur actuelle du pourcentage de sortie

**Texte structuré**

Opérande	Type (Type)	Format	Description (Description)
PID	PID	structure	Structure PID
Process variable	SINT	étiquette	La valeur que vous voulez contrôler
	INT		
	DINT		
	REAL		
Tieback	SINT	immediate	(facultatif)
	INT	étiquette	

	DINT		Sortie d'une station matérielle portable/auto qui contourne la sortie de l'automate. Entrer 0 si vous ne voulez pas utiliser ce paramètre
	REAL		
Variable de contrôle	SINT	étiquette	La valeur qui est envoyée au dispositif final de commande (valve, amortisseur, etc.)
	INT		
	DINT		Si vous utilisez la plage morte, la Control variable doit être du type REAL, sinon elle sera forcée à 0 lorsque l'erreur se trouve à l'intérieur de la plage morte.
	REAL		
PID master loop	PID	Structure	Facultatif
			Étiquette PID pour le PID maître
			Si vous effectuez un contrôle en cascade, et que ce PID est une boucle esclave, entrer le nom du PID maître.
			Entrer 0 si vous ne voulez pas utiliser ce paramètre
Inhold bit	BOOL	étiquette	Facultatif
			État courant du bit de maintien d'un canal de sortie analogique 1756
			pour prendre en charge un redémarrage sans à-coups
Inhold value	SINT	étiquette	Facultatif
	INT		Valeur de retour de données à partir d'un sortie analogique 1756
	DINT		Canal pour prendre en charge le redémarrage sans à-coups
	REAL		Entrer 0 si vous ne voulez pas utiliser ce paramètre
Point de consigne			Uniquement pour l'affichage
			Valeur actuelle du point de consigne
Process variable			Uniquement pour l'affichage
			Valeur actuelle de Process_Variable à l'échelle
Output %			Uniquement pour l'affichage
			Valeur actuelle du pourcentage de sortie

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Structure PID

Spécifier une structure PID unique pour chaque instruction PID.

Mnémonique	Type de donnée	Description (Description)																																																																		
.CTL	DINT	Le membre .CTL fournit un accès aux membres d'état (bits) dans un mot unique de 32 bits. L'instruction PID définit les bits 07 à 15																																																																		
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Numéro</th> <th>Description (Description)</th> </tr> </thead> <tbody> <tr> <td>.EN</td> <td>31</td> <td></td> </tr> <tr> <td>.CT</td> <td>30</td> <td>type de cascade (0=esclave ; 1=maître)</td> </tr> <tr> <td>.CL</td> <td>29</td> <td>boucle en cascade (0=non ; 1=oui)</td> </tr> <tr> <td>.PVT</td> <td>28</td> <td>suivi de la variable de procédé (0=non ; 1=oui)</td> </tr> <tr> <td>.DOE</td> <td>27</td> <td>dérivé de (0=PV ; 1=erreur)</td> </tr> <tr> <td>.SWM</td> <td>26</td> <td>mode logiciel (0=non-auto ; 1=oui-manuel logiciel)</td> </tr> <tr> <td>.CA</td> <td>25</td> <td>action de contrôle (0=inverse (SP-PV) ; 1=direct (PV-SP))</td> </tr> <tr> <td>.MO</td> <td>24</td> <td>mode de station (0=automatique ; 1=manuel)</td> </tr> <tr> <td>.PE</td> <td>23</td> <td>équation PID (0=indépendant ; 1=dépendant)</td> </tr> <tr> <td>.NDF</td> <td>22</td> <td>lissage des dérivées (0=non ; 1=oui)</td> </tr> <tr> <td>.NOBC</td> <td>21</td> <td>calcul de décalage (0=non ; 1=oui)</td> </tr> <tr> <td>.NOZC</td> <td>20</td> <td>passage par zéro (0=non ; 1=plage morte)</td> </tr> <tr> <td>.INI</td> <td>15</td> <td>PID initialisé (0=non ; 1=oui)</td> </tr> <tr> <td>.SPOR</td> <td>14</td> <td>point de consigne hors limites (0=non ; 1=oui)</td> </tr> <tr> <td>.OLL</td> <td>13</td> <td>CV est au-dessous de la valeur minimale de sortie (0=non ; 1=oui)</td> </tr> <tr> <td>.OLH</td> <td>12</td> <td>CV est au-dessus de la valeur maximale de sortie (0=non ; 1=oui)</td> </tr> <tr> <td>.EWD</td> <td>11</td> <td>l'erreur est à l'intérieur de la plage morte (0=non ; 1=oui)</td> </tr> <tr> <td>.DVNA</td> <td>10</td> <td>erreur avec alarme basse (0=non ; 1=oui)</td> </tr> <tr> <td>.DVPA</td> <td>9</td> <td>erreur avec alarme haute (0=non ; 1=oui)</td> </tr> <tr> <td>.PVLA</td> <td>8</td> <td>PV avec alarme basse (0=non ; 1=oui)</td> </tr> <tr> <td>.PVHA</td> <td>7</td> <td>PV avec alarme haute (0=non ; 1=oui)</td> </tr> </tbody> </table>	Bit	Numéro	Description (Description)	.EN	31		.CT	30	type de cascade (0=esclave ; 1=maître)	.CL	29	boucle en cascade (0=non ; 1=oui)	.PVT	28	suivi de la variable de procédé (0=non ; 1=oui)	.DOE	27	dérivé de (0=PV ; 1=erreur)	.SWM	26	mode logiciel (0=non-auto ; 1=oui-manuel logiciel)	.CA	25	action de contrôle (0=inverse (SP-PV) ; 1=direct (PV-SP))	.MO	24	mode de station (0=automatique ; 1=manuel)	.PE	23	équation PID (0=indépendant ; 1=dépendant)	.NDF	22	lissage des dérivées (0=non ; 1=oui)	.NOBC	21	calcul de décalage (0=non ; 1=oui)	.NOZC	20	passage par zéro (0=non ; 1=plage morte)	.INI	15	PID initialisé (0=non ; 1=oui)	.SPOR	14	point de consigne hors limites (0=non ; 1=oui)	.OLL	13	CV est au-dessous de la valeur minimale de sortie (0=non ; 1=oui)	.OLH	12	CV est au-dessus de la valeur maximale de sortie (0=non ; 1=oui)	.EWD	11	l'erreur est à l'intérieur de la plage morte (0=non ; 1=oui)	.DVNA	10	erreur avec alarme basse (0=non ; 1=oui)	.DVPA	9	erreur avec alarme haute (0=non ; 1=oui)	.PVLA	8	PV avec alarme basse (0=non ; 1=oui)	.PVHA	7	PV avec alarme haute (0=non ; 1=oui)
		Bit	Numéro	Description (Description)																																																																
		.EN	31																																																																	
		.CT	30	type de cascade (0=esclave ; 1=maître)																																																																
		.CL	29	boucle en cascade (0=non ; 1=oui)																																																																
		.PVT	28	suivi de la variable de procédé (0=non ; 1=oui)																																																																
		.DOE	27	dérivé de (0=PV ; 1=erreur)																																																																
		.SWM	26	mode logiciel (0=non-auto ; 1=oui-manuel logiciel)																																																																
		.CA	25	action de contrôle (0=inverse (SP-PV) ; 1=direct (PV-SP))																																																																
		.MO	24	mode de station (0=automatique ; 1=manuel)																																																																
		.PE	23	équation PID (0=indépendant ; 1=dépendant)																																																																
		.NDF	22	lissage des dérivées (0=non ; 1=oui)																																																																
		.NOBC	21	calcul de décalage (0=non ; 1=oui)																																																																
		.NOZC	20	passage par zéro (0=non ; 1=plage morte)																																																																
		.INI	15	PID initialisé (0=non ; 1=oui)																																																																
		.SPOR	14	point de consigne hors limites (0=non ; 1=oui)																																																																
		.OLL	13	CV est au-dessous de la valeur minimale de sortie (0=non ; 1=oui)																																																																
		.OLH	12	CV est au-dessus de la valeur maximale de sortie (0=non ; 1=oui)																																																																
		.EWD	11	l'erreur est à l'intérieur de la plage morte (0=non ; 1=oui)																																																																
		.DVNA	10	erreur avec alarme basse (0=non ; 1=oui)																																																																
.DVPA	9	erreur avec alarme haute (0=non ; 1=oui)																																																																		
.PVLA	8	PV avec alarme basse (0=non ; 1=oui)																																																																		
.PVHA	7	PV avec alarme haute (0=non ; 1=oui)																																																																		
.SP	REAL	point de consigne																																																																		
.KP	REAL	Indépendant – gain proportionnel (sans unité)																																																																		
		Dépendant – gain de l'automate (sans unité)																																																																		

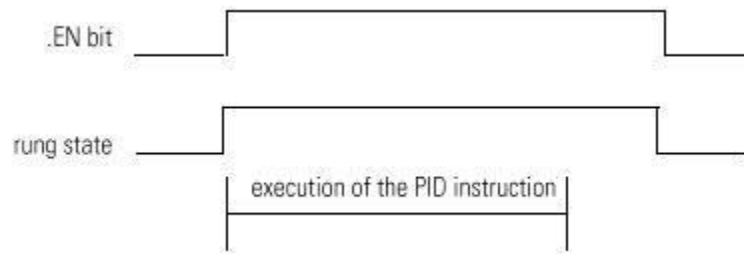
Mnémonique	Type de donnée	Description (Description)	
.KI	REAL	Indépendant – gain intégral (1/sec)	
		Dépendant – durée de réinitialisation (en minutes par répétition)	
.KD	REAL	Indépendant – gain dérivé (en secondes)	
		Dépendant – constante de temps (en minutes)	
.BIAS	REAL	% d'anticipation ou décalage	
.MAXS	REAL	valeur de mise à l'échelle d'unités de procédé maximale	
.MINS	REAL	valeur de mise à l'échelle d'unités de procédé minimale	
.DB	REAL	plage morte en unités de procédé	
.SO	REAL	définir % de sortie	
.MAXO	REAL	limite de sortie maximale (% de sortie)	
.MINO	REAL	limite de sortie minimale (% de sortie)	
.UPD	REAL	heure de mise à jour de la boucle (en secondes)	
.PV	REAL	valeur PV mise à l'échelle	
.ERR	REAL	valeur d'erreur mise à l'échelle	
.OUT	REAL	% de sortie	
.PVH	REAL	limite d'alarme haute de variable de procédé	
.PVL	REAL	limite d'alarme basse de variable de procédé	
.DVP	REAL	limite d'alarme de déviation positive	
.DVN	REAL	limite d'alarme de déviation négative	
.PVDB	REAL	plage morte d'alarme de variable de procédé	
.DVDB	REAL	plage morte d'alarme de déviation	
.MAXI	REAL	valeur PV maximale (entrée non mise à l'échelle)	
.MINI	REAL	valeur PV minimale (entrée non mise à l'échelle)	
.TIE	REAL	valeur de rétro-liaison pour contrôle manuel	
.MAXCV	REAL	valeur CV maximale (correspondant à 100 %)	
.MINCV	REAL	valeur CV minimale (correspondant à 0 %)	
.MINTIE	REAL	valeur de rétro-liaison minimum (correspondant à 100 %)	
.MAXTIE	REAL	valeur de rétro-liaison maximale (correspondant à 0 %)	
.DATA[17]	REAL	<b>Le membre .DATA mémorise :</b>	
		<b>Élément</b>	<b>Description (Description)</b>
		.DATA[0]	accumulation intégrale
		.DATA[1]	valeur temporaire de lissage des dérivées
		.DATA[2]	valeur .PV précédente
		.DATA[3]	valeur .ERR précédente
		.DATA[4]	valeur .SP valide précédente
		.DATA[5]	constante de mise à l'échelle de pourcentage
		.DATA[6]	constante de mise à l'échelle de .PV
		.DATA[7]	constante de mise à l'échelle de dérivées
.DATA[8]	valeur .KP précédente		

Mnémonique	Type de donnée	Description (Description)	
		.DATA[9]	valeur .KI précédente
		.DATA[10]	valeur .KD précédente
		.DATA[11]	gain dépendant .KP
		.DATA[12]	gain dépendant .KI
		.DATA[13]	gain dépendant .KD
		.DATA[14]	valeur .CV précédente
		.DATA[15]	constante de non-mise à l'échelle de .CV
		.DATA[16]	constante de non-mise à l'échelle de rétro-liaison

**Description (Description)**

L'instruction PID reçoit généralement la variable de procédé (PV) d'un module d'entrée analogique et module une sortie de variable de contrôle (CV) sur un module de sortie analogique pour maintenir la variable de procédé au point de consigne voulu.

Le bit .EN indique l'état de l'exécution. Le bit .EN est défini lorsque EnableIn passe de faux à vrai. Le bit .EN est mis à zéro lorsque EnableIn devient faux. L'instruction PID n'utilise pas de bit .DN. L'instruction PID est exécutée à chaque scrutation tant EnableIn est vrai.



**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
UPD ≥ 0	4	35
point de consigne hors limites	4	36

Reportez-vous à *Attributs communs* pour les défauts liés aux opérandes.

**Voir aussi**

[Instructions spéciales](#) sur la [page 685](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

**Utilisation des instructions PID**

Après avoir entré l’instruction PID et spécifié la structure PID, utiliser les onglets de configuration pour indiquer de quelle façon l’instruction PID doit fonctionner.

**Saisie des réglages**

Sélectionnez l’onglet **Réglages** (Tuning). Les changements prennent effet dès que vous cliquez sur un autre champ, sur **OK**, sur **Appliquer** (Apply) ou que vous appuyez sur la touche **Entrée** (Enter).

Dans ce champ :	Faites ce qui suit :
Point de consigne (SP) (Setpoint (SP))	Entrez une valeur pour le point de consigne (.SP).
Définir % de sortie (Set output %)	Entrez un pourcentage de décalage de la sortie (.SO). En mode Manuel du logiciel, cette valeur est utilisée pour la sortie. En mode Auto, elle affiche le pourcentage de sortie.
Décalage de la sortie (Output bias)	Entrez un pourcentage de décalage de la sortie (.BIAS).
Gain proportionnel (Kp) (Proportional gain (Kp))	Entrez le gain proportionnel (.KP). Pour des gains indépendants, il s'agit du gain proportionnel (sans unité). Pour des gains asservis, il s'agit du gain de l'automate (sans unité).
Gain intégral (Ki) (Integral gain (Ki))	Entrez le gain intégral (.KI). Pour des gains indépendants, il s'agit du gain intégral (1/sec.). Pour des gains asservis, il s'agit de la durée de réinitialisation (en minutes par répétition).
Temps dérivé (Kd) (Derivative time (Kd))	Entrez le gain dérivé (.KD) Pour des gains indépendants, il s'agit du gain dérivé (en secondes). Pour des gains asservis, il s'agit de la constante de temps (en minutes).
Mode Manuel (Manual mode)	Sélectionnez Manuel (.MO) ou Manuel du logiciel (.SWM). Le mode Manuel prévaut sur le mode Manuel du logiciel lorsqu'ils sont tous deux sélectionnés.

**Définir la configuration**

Sélectionnez l’onglet Configuration. Vous devez cliquer sur OK ou sur Appliquer (Apply) pour que les modifications prennent effet.



Dans ce champ :	Faites ce qui suit :
Equation PID (PID equation)	Sélectionnez gains indépendants ou gains asservis (.PE). Choisissez Indépendant si vous voulez que les trois gains (P, I et D) fonctionnent indépendamment. Choisissez Asservi si vous voulez un gain de l'automate global qui affecte les trois conditions (P, I et D).
Action de contrôle (Control action)	Sélectionnez E=PV-SP ou E=SP-PV pour l'action de contrôle (.CA).
Dérivé de (Derivative of)	Sélectionnez PV ou erreur (.DOE). Utilisez le dérivé de PV pour réduire le risque d'impulsions de sortie résultant des modifications du point de consigne. Utilisez le dérivé d'erreur pour obtenir des réponses rapides aux changements du point de consigne si l'algorithme accepte les dépassements.
Heure de mise à jour de la boucle (Loop update time)	Saisissez l'heure de mise à jour (.UPD) de l'instruction.
Limite haute de CV (CV high limit)	Entrez une limite haute pour la variable de contrôle (.MAXO).(1)
Limite basse de CV (CV low limit)	Entrez une limite basse pour la variable de contrôle (.MINO).(1)
Valeur de plage morte (Deadband value)	Entrez une valeur de plage morte (.DB).
Pas de lissage dérivé (No derivative smoothing)	Activez ou désactivez cette sélection (.NDF).
Pas de calcul de décalage (No bias calculation)	Activez ou désactivez cette sélection (.NOBC).
Pas de passage par zéro en plage morte (No zero crossing in deadband)	Activez ou désactivez cette sélection (.NOZC).
Suivi de PV (PV tracking)	Activez ou désactivez cette sélection (.PVT).
Boucle en cascade (Cascade loop)	Activez ou désactivez cette sélection (.CL).
Type de cascade (Cascade type)	Si l'option Boucle en cascade est activée, sélectionnez maître ou esclave (.CT).

(1) Lorsque vous utilisez l'instruction PID relais et si MAXO = MINO, l'instruction PID ramène ces valeurs aux paramètres par défaut. MAXO = 100,0 et MINO = 0,0

### Définir les alarmes

Sélectionnez l'onglet **Alarmes** (Alarms). Vous devez cliquer sur **OK** ou sur **Appliquer** (Apply) pour que les modifications prennent effet.

Dans ce champ :	Faites ce qui suit :
PV haute (PV high)	Une valeur d'alarme PV haute (.PVH)
PV basse (PV low)	Une valeur d'alarme PV basse (.PVL)
Plage morte de PV (PV deadband)	Une valeur de plage morte d'alarme PV (.PVDB)

Déviation positive (Positive deviation)	Une valeur de déviation positive (.DVP)
Déviation négative (Negative deviation)	Une valeur de déviation négative (.DVN)
Déviation plage morte (Deviation deadband)	Une valeur de plage morte d'alarme de déviation (.DVDB)

**Définir les mises à l'échelle**

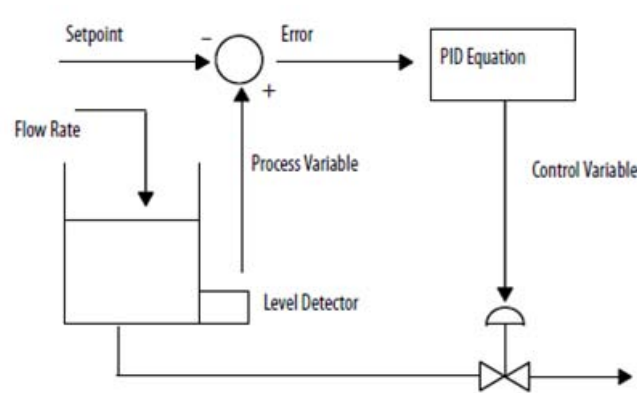
Sélectionnez l'onglet Mise à l'échelle. Vous devez cliquer sur OK ou sur Appliquer (Apply) pour que les modifications prennent effet.

Dans ce champ :	Faites ce qui suit :
PV maximale non mise à l'échelle (PV unscaled maximum)	Une valeur PV maximum (.MAXI) égale à la valeur maximum non à l'échelle transmise par le canal d'entrée analogique pour la valeur PV.
PV minimale non mise à l'échelle (PV unscaled minimum)	Une valeur PV minimum (.MINI) égale à la valeur minimum non à l'échelle transmise par le canal d'entrée analogique pour la valeur PV.
PV unité de procédé maximum (PV engineering units maximum)	Entrez les unités de procédé maximum correspondant à .MAXI (.MAXS).
PV unité de procédé minimum (PV engineering units minimum)	Entrez les unités de procédé minimum correspondant à .MINI (.MINS).
CV maximale (CV maximum)	Une valeur CV maximale correspondant à 100% (.MAXCV).
CV minimale (CV minimum)	Une valeur CV minimale correspondant à 0% (.MINCV).
Rétro-liaison maximale (Tieback maximum)	Une valeur d'élément de rétro-liaison maximale (.MAXTIE) égale à la valeur maximum non à l'échelle transmise par le canal d'entrée analogique pour la valeur d'élément de retour.
Rétro-liaison minimale (Tieback minimum)	Une valeur d'élément de rétro-liaison minimale (.MINTIE) égale à la valeur minimum non à l'échelle transmise par le canal d'entrée analogique pour la valeur d'élément de retour.
PID initialisé (PID Initialized)	Si vous changez les constantes de mise à l'échelle en mode Exécution, désactivez cette option pour réinitialiser les valeurs de mise à l'échelle internes (.INI).

**Astuce :** Lorsque vous utilisez l'instruction PID relais et si MAXO = MINO, l'instruction PID ramène ces valeurs aux paramètres par défaut. MAXO = 100,0 et MINO = 0,0

### Utiliser les instructions PID

Le contrôle du PID en boucle fermée maintient une variable de procédé au point de consigne souhaité. L'illustration montre un exemple d'un niveau de débit/fluide.



Dans l'exemple ci-dessus, le niveau dans le réservoir est comparé au point de consigne. Si ce niveau est supérieur au point de consigne, l'équation PID augmente la variable de contrôle et provoque l'ouverture du clapet de refoulement du réservoir, ce qui réduit le niveau dans le réservoir.

L'équation PID utilisée dans l'instruction PID est une équation de forme positionnelle permettant d'utiliser des gains indépendants ou des gains asservis. Lorsque vous utilisez des gains indépendants, les gains proportionnel, intégral et dérivé affectent, uniquement et respectivement, leur condition proportionnelle, intégrale ou dérivée spécifique. Lorsque vous utilisez des gains asservis, le gain proportionnel est remplacé par un gain de l'automate qui affecte l'ensemble de ces trois conditions. Vous pouvez utiliser l'une ou l'autre forme d'équation pour effectuer le même type de contrôle. Les deux types d'équation sont simplement fournies pour vous permettre d'utiliser le type d'équation qui vous est le plus familier.

Option de gains	Dérivé de
Gains asservis (norme ISA)	Erreur (E)
	Variable de procédé (PV)
Gains indépendants	Erreur (E)
	Variable de procédé (PV)

Où :

Variable	Description
KP	Gain proportionnel (sans unité) $K_p = K_c$ (sans unité)
Ki	Gain intégral (en secondes <sup>-1</sup> ) Pour effectuer une conversion entre $K_i$ (gain intégral) et $T_i$ (durée de réinitialisation), utilisez : $K_i = \frac{K_C}{60T_i}$
Kd	Gain dérivé (en secondes) Pour effectuer une conversion entre $K_d$ (gain intégral) et $T_d$ (constante de temps), utilisez la formule suivant : $K_d = K_c (T_d) 60$
KC	Gain de l'automate (sans unité)
Ti	Durée de réinitialisation (minutes/répétition)
Td	Constante de temps (minutes)
SP	Point de consigne
PV	Process variable
E	Erreur [(SP-PV) ou (PV-SP)]
BIAS	Anticipation ou écart
CV	Variable de contrôle
dt	Heure de mise à jour de la boucle

Si vous ne souhaitez pas utiliser une condition particulière de l'équation PID, il suffit de définir son gain à zéro. Par exemple, si vous ne souhaitez pas utiliser d'action dérivée, la valeur de  $K_d$  ou  $T_d$  doit être égale à zéro.

**Voir aussi**

[Redémarrage sans à-coup](#) sur la [page 717](#)

[Lissage des dérivées](#) sur la [page 720](#)

[Réglage de la plage morte](#) sur la [page 725](#)

[Montage des boucles en cascade](#) sur la [page 718](#)

[Contrôle d'un rapport](#) sur la [page 719](#)

**Anti-saturation d'intégrale et transfert sans à-coup de manuel à automatique (PID)**

L'instruction PID évite automatiquement la réinitialisation de la saturation d'intégrale en empêchant le terme intégral d'effectuer une accumulation chaque fois que la sortie CV atteint ses valeurs maximale ou minimale, telles qu'elles ont été définies par .MAXO et .MINO. Le terme intégral accumulé reste figé, jusqu'à ce que la sortie CV retombe au-dessous de sa limite maximale ou remonte au-dessus de sa limite minimale. À ce moment-là, l'accumulation intégrale normale reprend automatiquement.

Cette instruction PID prend en charge deux modes manuels de contrôle.

Mode manuel de contrôle	Description
Manuel logiciel (.SWM)	Ce mode est également connu sous la désignation de mode défini de sortie et permet à l'utilisateur de définir le % de sortie à partir du logiciel. La valeur de sortie définie (.SO) s'utilise en tant que sortie de la boucle. En général, la valeur de sortie définie part d'une entrée opérateur provenant d'un dispositif d'interface opérateur.
Manuel (.MO)	Ce mode prend la valeur de rattachement, en tant qu'entrée, et ajuste ses variables internes pour produire la même valeur que la sortie. L'entrée de rattachement à l'instruction PID est basée sur une échelle allant de 0 à 100% et dépend des valeurs de .MINTIE et .MAXTIE et sert de sortie pour la boucle. En général, l'entrée de rattachement provient de la sortie d'une station de matériel portable/auto qui contourne la sortie de l'automate. <b>Important</b> : le mode manuel a priorité sur le mode manuel logiciel lorsque les bits de ces deux modes sont réglés sur actif.

Cette instruction PID assure automatiquement des transferts sans à-coup du mode manuel logiciel au mode auto ou du mode manuel au mode auto. Cette instruction PID calcule de manière rétro-active la valeur du terme intégral d'accumulation qui est nécessaire pour permettre à la sortie CV de suivre la valeur de la sortie définie (.SO) dans le mode manuel logiciel ou l'entrée de rattachement dans le mode manuel. De cette façon, lorsque la boucle passe dans le mode auto, la sortie CV part de la sortie définie ou de la valeur de rattachement et il n'y a ainsi aucun « à-coups » au niveau de la valeur de sortie.

Cette instruction PID peut également assurer un transfert sans à-coup du mode manuel au mode auto, même lorsque le contrôle intégral n'est pas utilisé (c-à-d que  $K_i = 0$ ). Dans ce cas, cette instruction modifie le terme .BIAS afin que la sortie CV suive la sortie définie ou les valeurs de rattachement. Lorsque le contrôle automatique reprend, le terme .BIAS maintient sa dernière valeur. Désactiver le calcul rétroactif du terme .BIAS en définissant le bit .NOBC dans la structure des données PID. Lorsque vous définissez .NOBC sur vrai, l'instruction PID n'assure plus un transfert sans à-coup du mode manuel au mode auto lorsque le contrôle intégral n'est pas utilisé.

### Redémarrage sans à-coup (PID)

Cette instruction PID peut avoir des interactions avec les modules de sorties analogiques 1756 pour faciliter un redémarrage sans à-coup lorsque l'automate passe du mode Programme au mode Exécution ou lors de la montée en régime de l'automate.

Lorsqu'un module de sortie analogique 1756 perd le contact avec l'automate ou détecte que l'automate est dans le mode Programme, ce module de sortie analogique fait passer ses sorties sur les valeurs des conditions de défauts que vous avez spécifiées lors de la configuration de ce module. Lorsque l'automate revient dans le mode Exécution ou rétablit le contact avec le module de sortie analogique, vous pouvez automatiquement réinitialiser l'instruction PID sur une sortie variable de contrôle égale à la sortie analogique, en faisant appel au Inhold Bit et aux paramètres des Inhold Value de l'instruction PID.

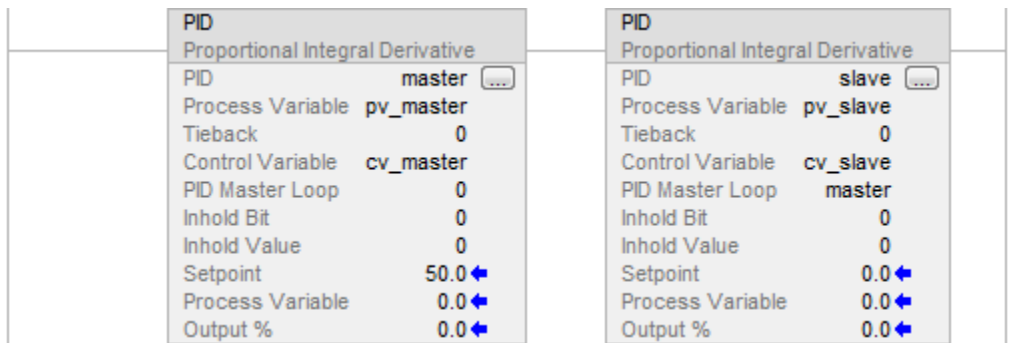
**Instructions de programmation d'un redémarrage sans à-coup**

Procéder comme suit	Détails
Configurer le canal du module de sortie analogique 1756 qui reçoit la variable de contrôle en provenance de l'instruction PID	<p>Sélectionner la case <b>Maintien pour initialisation</b> (Hold for initialization) qui se trouve sur la page des propriétés du canal spécifique du module.</p> <p>Cela ordonne au module de sortie analogique, lorsque l'automate revient dans le mode Exécution ou rétablit le contact avec ce module, de maintenir la sortie analogique à sa valeur actuelle jusqu'à ce que la valeur envoyée depuis l'automate corresponde (avec une plage de 0,1%) à la valeur actuelle qu'utilise le canal de sortie. La sortie du canal monte en régime jusqu'à la valeur de sortie actuellement maintenue, en faisant appel au terme .BIAS. Cette montée en régime est semblable au transfert auto sans à-coup.</p>
Entrer l'étiquette Inhold bit et l'étiquette Inhold Value dans l'instruction PID	<p>Le module de sortie analogique 1756 renvoie deux valeurs pour chaque canal dans sa structure de données d'entrée. Lorsque le bit d'état InHold (.Ch2InHold, par exemple) est vrai, cela indique que le canal de sortie analogique maintient sa valeur. La valeur de lecture rétroactive des données (.Ch2Data, par exemple) montre la valeur actuelle de sortie en unités techniques.</p> <p>Entrer l'étiquette du bit d'état InHold en tant que paramètre du InHold bit de l'instruction PID. Entrer l'étiquette de la valeur de lecture rétroactive des données en tant que paramètre de Inhold Value.</p> <p>Lorsque Inhold bit est vrai, l'instruction PID introduit Inhold Value de maintien dans la sortie Control variable et effectue une réinitialisation pour prendre en charge un redémarrage sans à-coup à cette valeur. Lorsque le module de sortie analogique reçoit cette valeur en provenance de l'automate, cela met hors circuit le bit d'état InHold et permet ainsi à l'instruction PID de commencer un contrôle normal.</p>

**Boucles en cascade (PID)**

L'instruction PID met deux boucles en cascade en attribuant un certain pourcentage de la sortie de la boucle maître au point de consigne de la boucle esclave. La boucle esclave convertit automatiquement la sortie de la boucle maître dans les unités de procédé correctes pour le point de consigne de la boucle esclave, en fonction des valeurs .MAXS et .MINS de la boucle esclave.

**Logique à relais**

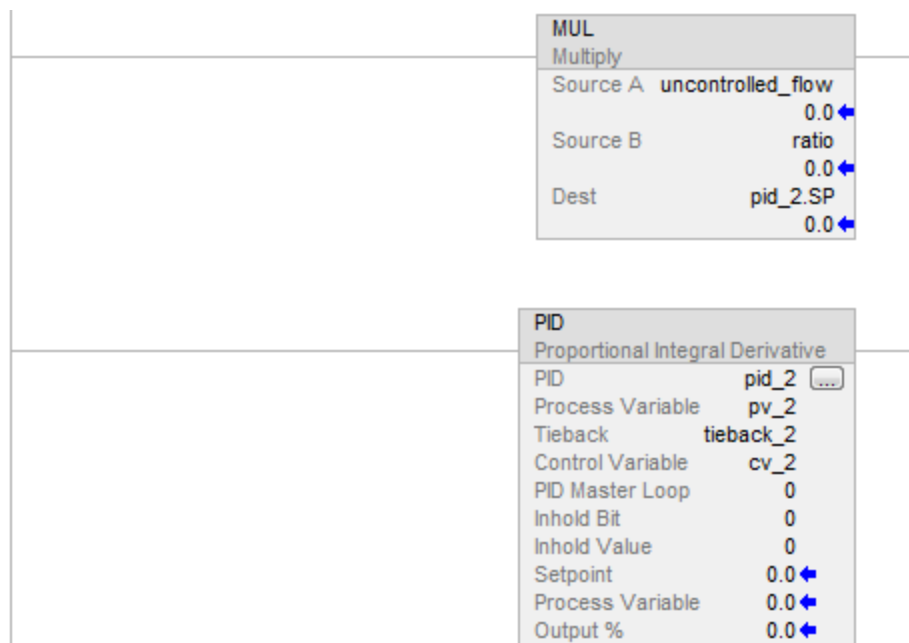


**Texte structuré**

```
PID(master,pv_master,0,cv_master,0,0,0); PID
(slave,pv_slave,0,cv_slave,0,0);
```

**Contrôle d'un rapport (PID)** Vous pouvez maintenir deux valeurs dans un rapport donné en utilisant ces paramètres :

- Valeur non asservie
- Valeur asservie (point de consigne résultant qui doit être utilisé par l'instruction PID)
- Rapport entre ces deux valeurs

**Logique à relais**

**Astuce :** Pour éviter de verrouiller l'instruction PID avec des valeurs flottantes internes non valides, assurez-vous que PV n'est pas égal à INF ou NAN avant d'appeler l'instruction comme  
XIC (PC\_timer.DN)  
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)  
XIO(S:V)  
PID(...)

**Texte structuré**

```
pid_2.sp := uncontrolled_flow * ratio
```

```
PID(pid_2,pv_2,tieback_2,cv_2,0,0,0);
```

**Astuce :** Pour éviter de verrouiller l'instruction PID avec des valeurs flottantes internes non valides, assurez-vous que PV n'est pas égal à INF ou NAN avant d'appeler l'instruction comme

```
XIC(PC_timer.DN)
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
XIO(S:V)
PID(...)
```

Pour cette multiplication	Entrer cette valeur
Destination	Valeur asservie
Source A	Valeur non asservie
Source B	Rapport

**Lissage des dérivées (PID)**

Le calcul dérivé est renforcé par un filtre de lissage dérivé. Ce filtre numérique, basse fréquence, du premier ordre minimise les impulsions élevées de la condition dérivée dues au bruit dans la PV. Ce lissage se fait plus prononcé avec des valeurs de gain dérivé élevées. Vous pouvez désactiver le lissage dérivé si votre procédé requiert des valeurs très élevées de gain dérivé (Kd > 10, par exemple).

**Pour désactiver le lissage dérivé :**

- Sélectionnez l'option **Pas de lissage dérivé** (No derivative smoothing) dans l'onglet **Configuration** (Configuration) ou définissez le bit .NDF dans la structure PID.

**Anticipation ou polarisation de la sortie (PID)**

Anticiper une perturbation en provenance du système en introduisant la valeur .BIAS dans la valeur d'anticipation/polarisation de l'instruction PID.

La valeur d'anticipation représente une perturbation qui est introduite dans l'instruction PID avant que cette perturbation ne puisse modifier la variable du procédé. L'anticipation s'utilise souvent pour contrôler des procédés qui ont un délai de transport. Par exemple, une valeur d'anticipation qui représente le « versement d'eau froide dans un mélange chaud » permet parfois d'obtenir plus rapidement la valeur de sortie, sans attendre que la variable du procédé ne change à la suite de ce mélange.

En général, une valeur de polarisation s'utilise lorsqu'aucun contrôle intégral n'est exploité. Dans ce cas, cette valeur de polarisation peut être ajustée afin de maintenir la sortie dans le créneau exigé pour conserver la PV à proximité du point de consigne.



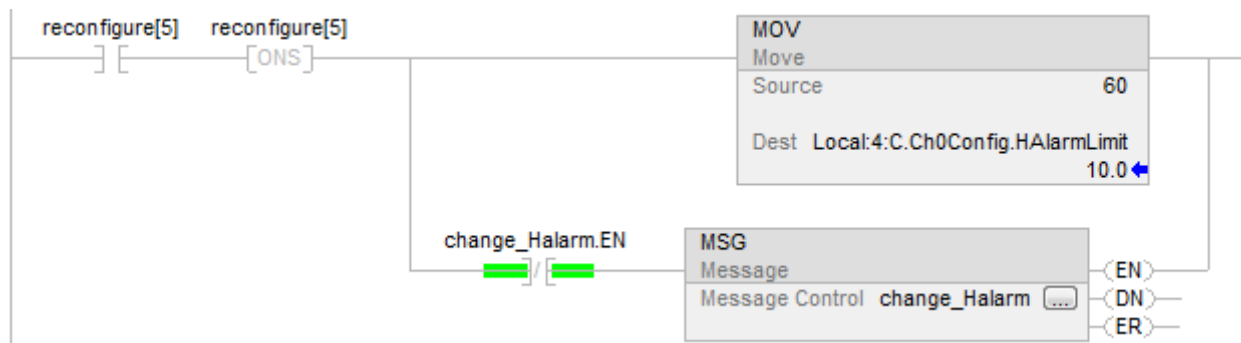
## Temporisation de l'instruction PID

L'instruction PID et l'échantillonnage de la variable de procédé doivent être mis à jour régulièrement. Cet heure de mise à jour est lié au processus physique que vous contrôlez. Pour les boucles très lentes, comme les boucles de température, une heure de mise à jour d'une seconde ou plus est généralement suffisant pour obtenir un bon contrôle. Les boucles plus rapides, comme les boucles de pression ou de débit, peuvent nécessiter une heure de mise à jour plus court de 250 ms par exemple. Ce n'est que dans des cas rares, comme pour asservir la tension d'un dévidoir, qu'une heure de mise à jour aussi court que 10 ms ou moins peut être nécessaire.

Comme l'instruction PID utilise une base de temps dans son calcul, vous devez synchroniser l'exécution de cette instruction avec l'échantillonnage de la variable de procédé (PV).

La façon la plus facile d'exécuter l'instruction PID est de mettre cette instruction dans une tâche périodique. Définir l'heure de mise à jour de boucle (.UPD) sur une valeur égale à la fréquence de la tâche périodique et assurez-vous que l'instruction PID est exécutée à chaque scrutation de la tâche périodique.

### Logique à relais



**Astuce :** Pour éviter de verrouiller l'instruction PID avec des valeurs flottantes internes non valides, assurez-vous que PV n'est pas égal à INF ou NAN avant d'appeler l'instruction comme  
 XIC (PC\_timer.DN)  
 MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)  
 XIO(S:V)  
 PID(...)

### Texte structuré

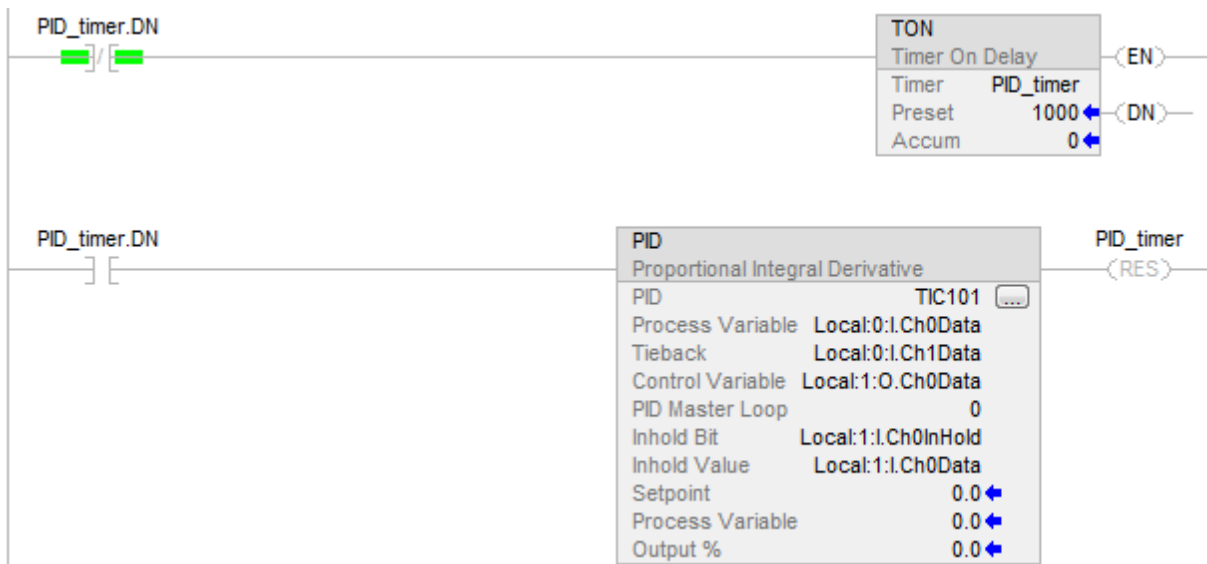
```
PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,  
Local:1:O.Ch4Data,0,Local:1:I.Ch4InHold, Local:1:I.Ch4Data);
```

Lorsque vous utilisez une tâche périodique, assurez-vous que l'entrée analogique utilisée pour la variable de procédé est mise à jour dans le processeur à une

fréquence significativement plus rapide que la fréquence de la tâche périodique. Idéalement, la variable de procédé doit être envoyée au processeur au moins cinq à 10 fois plus rapidement que la fréquence de la tâche périodique. Cela permet de minimiser la différence de temps entre les échantillons réels de la variable de procédé et l'exécution de la boucle PID. Par exemple, si la boucle PID est dans une tâche périodique de 250 ms, utiliser une heure de mise à jour de boucle de 250 ms (.UPD = 0,25) et configurer le module d'entrée analogique pour qu'il produise des données au moins toutes les 25 à 50 ms.

Une autre méthode quelque peu moins précise pour exécuter une instruction PID consiste à placer l'instruction dans une tâche continue et à utiliser un bit de fin de temporisateur pour déclencher l'exécution de l'instruction PID.

### Logique à relais



**Astuce :** Pour éviter de verrouiller l'instruction PID avec des valeurs flottantes internes non valides, assurez-vous que PV n'est pas égal à INF ou NAN avant d'appeler l'instruction comme  
XIC (PC\_timer.DN)  
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)  
XIO(S:V)  
PID(...)

### Texte structuré

```
PID_timer.pre := 1000
```

```
TONR(PID_timer);
```

```
IF PID_timer.DN THEN PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,
```

```
Local:1:O.Ch0Data,0,Local:1:I.Ch0InHold,
```

```
Local:1:I.Ch0Data);
```

```
END_IF;
```

**Astuce :** Pour éviter de verrouiller l'instruction PID avec des valeurs flottantes internes non valides, assurez-vous que PV n'est pas égal à INF ou NAN avant d'appeler l'instruction comme XIC (PC\_timer.DN)  
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)  
XIO(S:V)  
PID(...)

Dans cette méthode, l'heure de mise à jour de boucle de l'instruction PID doit être égal à la valeur prédéfinie du temporisateur. Comme dans le cas de l'utilisation d'une tâche périodique, vous devez définir le module d'entrée analogique pour qu'il produise une variable de procédé à une fréquence plus rapide que l'heure de mise à jour de boucle. Vous devriez utiliser la méthode du temporisateur pour l'exécution PID de boucles uniquement lorsque l'heure de mise à jour de boucle est au moins plusieurs fois plus long que le temps d'exécution de votre tâche continue dans le cas le plus défavorable.

La façon la plus précise d'exécuter une instruction PID est d'utiliser la fonction d'échantillonnage en temps réel (RTS) des modules d'entrée analogique 1756. Le module d'entrée analogique échantillonne ses entrées à la fréquence d'échantillonnage en temps réel que vous configurez lorsque vous configurez le module. Lorsque la période d'échantillon en temps réel du module est terminée, ce dernier met à jour ses entrées ainsi qu'un horodatage de déroulement (représenté par le membre .RollingTimestamp de la structure de données d'entrée analogique) produit par le module.

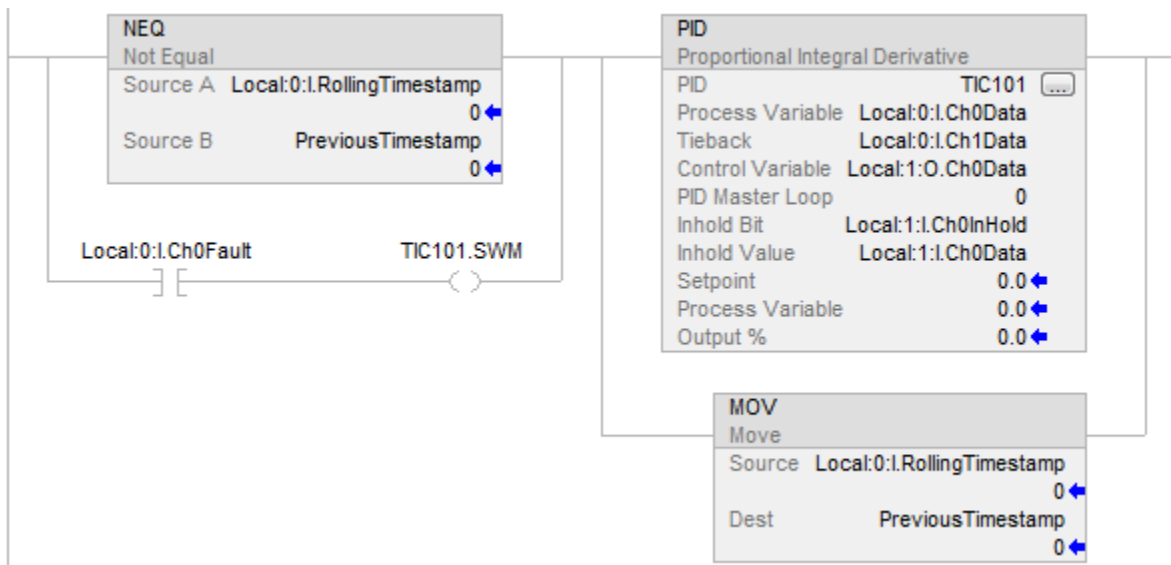
L'horodatage peut être compris entre 0 et 32 767 ms. Surveiller l'horodatage. Lorsqu'il change, un nouvel échantillon de variable de procédé a été reçu. Chaque fois qu'un horodatage change, exécuter l'instruction PID une fois. Étant donné que l'échantillon de variable de procédé est géré par le module d'entrée analogique, le temps d'échantillonnage d'entrée est très précis, et l'heure de mise à jour de boucle utilisé par l'instruction PID doit être défini comme étant égal au temps RTS du module d'entrée analogique.

Pour être sûr de ne pas rater d'échantillons de la variable de procédé, exécuter votre logique à une fréquence supérieure au temps RTS. Par exemple, si le temps RTS est de 250 ms, vous pouvez mettre la logique PID dans une tâche périodique qui est exécutée toutes les

100 ms pour être sûr de ne jamais rater un échantillon. Vous pouvez même placer la logique PID dans une tâche continue, tant que vous vous assurez que la logique sera mise à jour plus souvent que toutes les 250 ms.

Un exemple de la méthode RTS d'exécution est donné ci-dessous. L'exécution de l'instruction PID dépend de la réception de nouvelles données d'entrée analogique. Si le module d'entrée analogique échoue ou est retiré, l'automate arrête de recevoir les horodatages de déroulement et la boucle PID arrête son exécution. Vous devez surveiller le bit d'état de l'entrée analogique PV et si ce bit présente un mauvais état, forcer la boucle en mode manuel logiciel et exécuter la boucle à chaque scrutation. Cela permet à l'opérateur de changer manuellement la sortie de la boucle PID.

**Logique à relais**



**Texte structuré**

```

IF (Local:0:I.Ch0Fault) THEN TIC101.SWM [:=] 1;

ELSE TIC101.SWM := 0; END_IF;

IF (Local:0:I.RollingTimestamp<>PreviousTimestamp) OR
(Local:0:I.Ch0Fault) THEN

PreviousTimestamp := Local:0:I.RollingTimestamp;
PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,

Local:1:O.Ch0Data,0,Local:1:I.Ch0InHold,

Local:1:I.Ch0Data);

END_IF;
    
```

## Réglage de la plage morte (PID)

La plage morte ajustable vous permet de sélectionner une plage d'erreur au-dessus et en dessous du point de consigne, dans laquelle la sortie ne change pas tant que l'erreur reste dans les limites de la plage. Cette plage morte vous permet de contrôler à quel point la variable de procédé concorde avec le point de consigne sans changer la sortie. La plage morte minimise également l'usure sur le dispositif de commande final.



Le passage par zéro est le contrôle de plage morte qui permet à l'instruction d'utiliser l'erreur à des fins de calcul lorsque la variable de procédé passe par la zone morte, jusqu'à ce que la variable de procédé passe par le point de consigne. Une fois que la variable de procédé croise le point de consigne (l'erreur passe par zéro, puis change de signe) et tant que la variable de procédé reste dans la plage morte, la sortie ne change pas.

La valeur que vous indiquez définit la plage d'extension de la plage morte au-dessus et au-dessous du point de consigne. Entrez zéro pour inhiber la plage morte. La plage morte a les mêmes unités à l'échelle que le point de consigne. Pour utiliser la plage morte sans la fonction de passage par zéro, sélectionnez l'option **Pas de passage par zéro pour la plage morte** (No zero crossing for deadband) sur l'onglet **Configuration** (Configuration) ou définissez le bit .NOZC dans la structure PID.

Si vous utilisez la plage morte, la Control variable doit avoir la valeur REAL, sinon, elle est forcée à zéro lorsque l'erreur se trouve dans la plage morte.

### Pour inhiber la plage morte :

- Entrez zéro (0).

La plage morte a les mêmes unités à l'échelle que le point de consigne.

### Pour utiliser la plage morte sans la fonction de passage par zéro :

- Sélectionnez l'option **Pas de passage par zéro pour la plage morte** (No zero crossing for deadband) dans l'onglet **Configuration** (Configuration) ou définissez le bit .NOZC dans la structure PID.

Si vous utilisez la zone morte, la Control variable doit avoir la valeur REAL, sinon, elle est forcée à 0 lorsque l'erreur se trouve dans la plage morte.

**Utilisation de la limitation de sortie (PID)**

Définir une limite de sortie (en pourcentage de la sortie) sur la sortie de contrôle. Lorsque cette instruction détecte que la sortie a atteint une limite, elle définit un bit d'alarme et empêche tout dépassement de cette limite (inférieure ou supérieure) par la sortie.

## Instructions trigonométriques

Les instructions trigonométriques évaluent les opérations arithmétiques qui utilisent des opérations trigonométriques.

### Instructions disponibles

#### Diagramme à relais, bloc fonctionnel et texte structuré

<a href="#">SIN</a>	<a href="#">ATN,</a> <a href="#">ATAN</a>	<a href="#">COS</a>	<a href="#">TAN</a>	<a href="#">ASN,</a> <a href="#">ASIN</a>	<a href="#">ACS/ASO</a> <a href="#">S</a>
---------------------	--	---------------------	---------------------	--	--

Si vous voulez :	Utilisez cette instruction :
Calculer le sinus d'une valeur.	SIN
Calculer le cosinus d'une valeur.	COS
Calculer la tangente d'une valeur.	TAN
Calculer le sinus d'arc d'une valeur.	ASN
Calculer le cosinus d'arc d'une valeur.	ACS
Calculer la tangente d'arc d'une valeur.	ATN

Vous pouvez mélanger des types de données, mais cela peut entraîner une perte de précision et une erreur d'arrondi et l'exécution de l'instruction prendra plus de temps. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données **en caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

Une instruction trigonométrique est exécutée une fois, chaque fois que l'instruction est scrutée, aussi longtemps que la condition d'entrée d'échelon est vraie. Si vous voulez que l'instruction ne soit évaluée qu'une fois, utiliser une instruction ONS pour déclencher l'instruction trigonométrique.

**Voir aussi**

[Instructions de temporisateur et de compteur](#) sur la [page 101](#)

[Instructions spéciales](#) sur la [page 685](#)

[Instructions de séquenceur](#) sur la [page 609](#)

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Instructions de mouvement/logique](#) sur la [page 433](#)

## Instructions trigonométriques

Les instructions trigonométriques évaluent les opérations arithmétiques qui utilisent des opérations trigonométriques.

**Instructions disponibles**

**Diagramme à relais, bloc fonctionnel et texte structuré**

<a href="#">SIN</a>	<a href="#">ATN,</a> <a href="#">ATAN</a>	<a href="#">COS</a>	<a href="#">TAN</a>	<a href="#">ASN,</a> <a href="#">ASIN</a>	<a href="#">ACS/ASO</a> <a href="#">S</a>
---------------------	--	---------------------	---------------------	--	--

Si vous voulez :	Utilisez cette instruction :
Calculer le sinus d'une valeur.	SIN
Calculer le cosinus d'une valeur.	COS
Calculer la tangente d'une valeur.	TAN
Calculer le sinus d'arc d'une valeur.	ASN
Calculer le cosinus d'arc d'une valeur.	ACS
Calculer la tangente d'arc d'une valeur.	ATN

Vous pouvez mélanger des types de données, mais cela peut entraîner une perte de précision et une erreur d'arrondi et l'exécution de l'instruction prendra plus de temps. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données **en caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.



Une instruction trigonométrique est exécutée une fois, chaque fois que l'instruction est scrutée, aussi longtemps que la condition d'entrée d'échelon est vraie. Si vous voulez que l'instruction ne soit évaluée qu'une fois, utiliser une instruction ONS pour déclencher l'instruction trigonométrique.

### Voir aussi

[Instructions de temporisateur et de compteur](#) sur la [page 101](#)

[Instructions spéciales](#) sur la [page 685](#)

[Instructions de séquenceur](#) sur la [page 609](#)

[Instructions de contrôle du programme](#) sur la [page 624](#)

[Instructions de mouvement/logique](#) sur la [page 433](#)

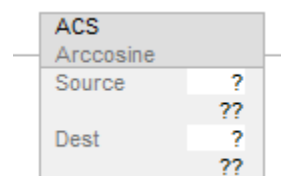
## Cosinus d'arc (ACS, ACOS)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

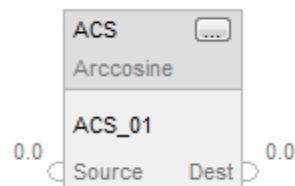
L'instruction ACS calcule le cosinus d'arc de la valeur Source et mémorise le résultat dans Destination (en radians).

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel



#### Texte structuré

```
dest := ACOS(source);
```

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	trouver le cosinus de cette valeur
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

### Texte structuré

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	immédiate étiquette	trouver le cosinus de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

Utiliser ACOS comme une fonction. Cette fonction calcule le cosinus d'arc de source et renvoie le résultat REAL.

### Bloc fonctionnel

Opérande	Type	Format	Description
ACS tag	FBD_MATH_ADVANCED	Structure	Structure ACS

### Structure FBD\_MATH\_ADVANCED

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	REAL	Entrée de l'instruction mathématique.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction mathématique.

### Description

L'instruction ACS calcule le cosinus d'arc de la valeur Source et mémorise et renvoie le résultat REAL dans Destination (en radians). Source doit être supérieur ou égal à -1 ou inférieur ou égal à 1. La valeur résultante dans Destination est supérieure ou égale à 0 ou inférieure ou égale à pi. Si Source est inférieur à -1 ou supérieur à 1 alors Destination est défini sur NAN.

Vous pouvez utiliser ACS comme un opérateur dans une expression du langage ladder et vous pouvez utiliser ACOS comme un opérateur dans des instructions de Texte structuré.

### Affecte les indicateurs d'état mathématique

Automates	Affecte l'indicateur d'état mathématique
ControlLogix 5580	Conditionnel, Reportez-vous à la section Indicateurs d'état mathématique.
CompactLogix 5370, ControlLogix 5570	Oui

### Défauts majeurs/mineurs

Si Destination est défini sur NAN, un débordement, avec ses défauts mineurs conditionnels, sera généré.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule l'arc cosinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

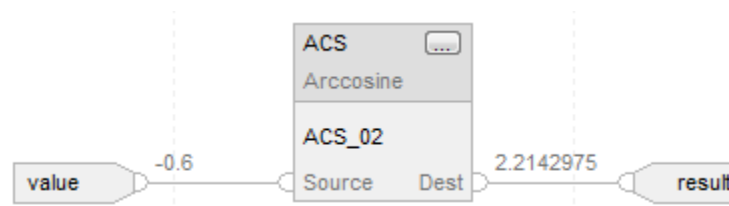
Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	L'automate calcule l'arc cosinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := ACOS(value);

**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

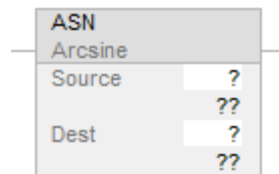
## Sinus d'arc (ASN, ASIN)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

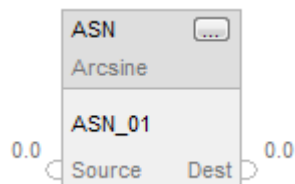
L'instruction ASN calcule le sinus d'arc de la valeur Source et mémorise le résultat dans Destination (en radians).

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel



#### Texte structuré

```
dest :=ASIN(source);
```

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT <b>REAL</b>	Immédiate étiquette	calcule le sinus d'arc de cette valeur
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	immédiate étiquette	calcule le sinus d'arc de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

Utiliser ASIN comme une fonction. Cette fonction calcule le sinus d'arc de Source et renvoie le résultat REAL.

**Bloc fonctionnel**

Opérande	Type	Format	Description
ASN tag	FBD_MATH_ADVANCED	Structure	Structure ASN

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source	REAL	Entrée de l'instruction mathématique. Valide = tout nombre flottant

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction.

### Description

L'instruction ASN calcule le sinus d'arc de la valeur Source et mémorise et renvoie le résultat REAL dans Destination (en radians). Source doit être supérieur ou égal à -1 ou inférieur ou égal à 1. La valeur résultante dans Destination est supérieure ou égale à  $-\pi/2$  et inférieure ou égale à  $\pi/2$ . Si Source est inférieur à -1 ou supérieur à 1 alors Destination est défini sur NAN.

Vous pouvez utiliser ASN comme un opérateur dans une expression du langage ladder et vous pouvez utiliser ASIN comme un opérateur dans des instructions de Texte structuré.

Cette instruction offre une meilleure précision sur les automates traditionnels, ce qui donne de meilleurs résultats.

### Affecte les indicateurs d'état mathématique

Automates	Affecte l'indicateur d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

### Défauts majeurs/mineurs

Si Destination est défini sur NAN, un débordement, avec ses défauts mineurs conditionnels, sera généré.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	L'automate calcule le sinus d'arc de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := ASIN(value);



**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Attributs communs](#) sur la [page 885](#)

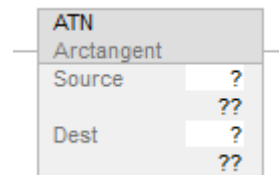
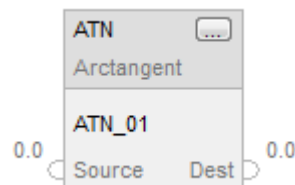
[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

**Tangente d'arc  
(ATN, ATAN)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction ATN calcule la tangente d'arc de la valeur Source et mémorise le résultat dans Destination (en radians).

**Langages disponibles****Diagramme à relais****Bloc fonctionnel****Texte structuré**

```
dest := ATAN(source);
```

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	Calcule la tangente d'arc de cette valeur
Destination	SINT INT DINT REAL	étiquette	Étiquette où enregistrer le résultat

**Texte structuré**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	Calcule la tangente d'arc de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

Utiliser ATAN comme une fonction. Cette fonction calcule la tangente d'arc de Source et renvoie le résultat REAL.

**Bloc fonctionnel**

Opérande	Type	Format	Description
ATN tag	FBD_MATH_ADVANCED	Structure	Structure ATN

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source	REAL	Entrée de l'instruction mathématique. Valide = tout nombre flottant

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction.

### Description

L'instruction ATN calcule la tangente d'arc de la valeur Source et mémorise le résultat dans Destination (en radians). La valeur résultante dans Destination est supérieure ou égale à  $-\pi/2$  et inférieure ou égale à  $\pi/2$ .

Vous pouvez utiliser ATN comme un opérateur dans une expression du langage ladder et vous pouvez utiliser ATAN comme un opérateur dans des instructions de Texte structuré.

### Affecte les indicateurs d'état mathématique

Automates	Affecte l'indicateur d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section Indicateurs d'état mathématique.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule la tangente d'arc de Source et place le résultat dans Destination.
Post-scrutation	N/A

#### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux

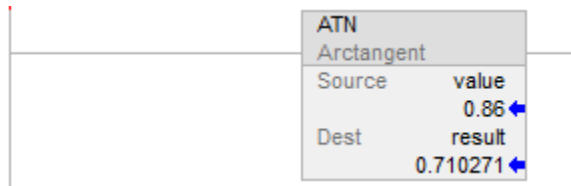
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

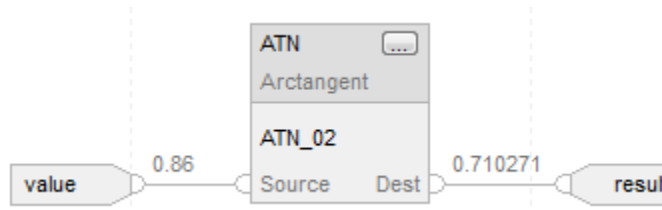
Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	L'automate calcule la tangente d'arc de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := ATAN(value);.

**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Cosinus (COS)

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

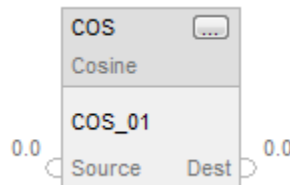
Cette instruction COS prend le cosinus de la valeur Source (en radians) et mémorise le résultat dans Destination.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

dest := COS(source);

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	trouver le cosinus de cette valeur

Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat
-------------	-----------------------------	-----------	--------------------------------------

**Texte structuré**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	immédiate étiquette	trouver le cosinus de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
COS tag	FBD_MATH_ADVANCED	Structure	structure COS

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	REAL	Entrée de l'instruction mathématique.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction mathématique.

**Description**

Cette instruction COS calcule le cosinus de la valeur Source (en radians) et mémorise le résultat dans Destination.

Cette instruction calcule le cosinus de Source et renvoie le résultat REAL. La valeur qui en résulte est toujours égale ou supérieure à -1 et inférieure ou égale à 1.

Vous pouvez vous servir de COS en tant qu'opérateur dans les expressions à logique à relais et en tant qu'opérateur dans les déclarations à texte structuré.

Cette instruction offre une meilleure précision sur les automates traditionnels, ce qui donne de meilleurs résultats.

### Affecte les indicateurs d'état mathématique

automates	Affecte l'indicateur d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section Indicateurs d'état mathématique.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

### Défauts majeurs/mineurs

Aucun. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule le cosinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

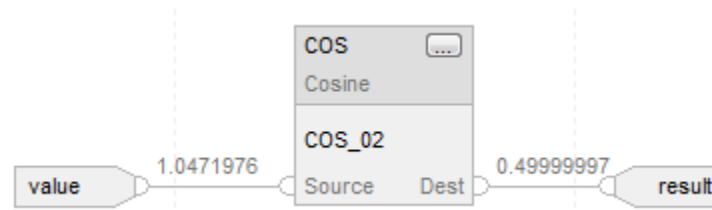
Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	L'automate calcule le cosinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := COS(value);

**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Radians \(RAD\)](#) sur la [page 779](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

**Sinus (SIN)**

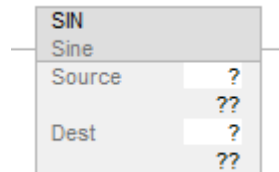
Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.



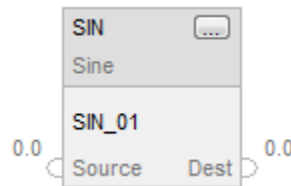
L’instruction SIN calcule le sinus de la valeur Source (en radians) et mémorise le résultat dans Destination.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

dest := SIN(source);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à *Conversion de données*.

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	Immédiate étiquette	calcule le sinus de cette valeur
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	Immédiate étiquette	calcule le sinus de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
SIN tag	FBD_MATH_ADVANCED	Structure	Structure SIN

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source (Source)	REAL	Entrée de l'instruction mathématique.

Paramètre de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction mathématique.

**Aspects d'opérateur**

L'opérateur SIN peut être utilisé dans diverses expressions. De même, la fonction SIN peut être appelée dans les instructions de Texte structuré. Les deux applications de SIN retourne un résultat REAL contenant le sinus de Source. En fonction du contexte, cette valeur peut être convertie, quand cela est approprié.

**Description (Description)**

L'instruction SIN calcule le sinus de la valeur Source (en radians) et mémorise le résultat dans Destination.

Cette instruction calcule le sinus de Source et renvoie le résultat REAL. La valeur qui en résulte est toujours égale ou supérieure à -1 et inférieure ou égale à 1.

Vous pouvez utiliser SIN comme un opérateur dans une expression du langage ladder et comme une fonction dans des instructions de Texte structuré.

### Affecter les indicateurs d'état mathématique

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes

### Exécution

### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule le sinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

### Bloc fonctionnel

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

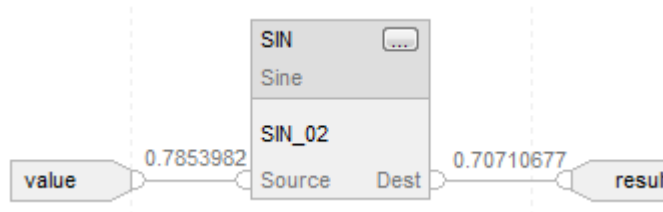
Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	L'automate calcule le sinus de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := SIN(value);

**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

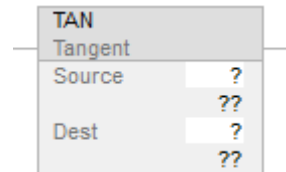
**Tangente (TAN)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L’instruction TAN calcule la tangente de la valeur Source (en radians) et mémorise le résultat dans Destination.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

dest := TAN(source);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	trouver le cosinus de cette valeur
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	immédiate étiquette	calcule la tangente de cette valeur

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
TAN tag	FBD_MATH_ADVANCED	Structure	Structure TAN

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	REAL	Entrée de l'instruction mathématique.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction mathématique.

**Description**

L'instruction TAN calcule la tangente de la valeur Source (en radians) et mémorise le résultat dans Destination.

L'instruction calcule la tangente de Source et renvoie le résultat REAL.

Vous pouvez utiliser TAN comme un opérateur dans une expression du langage ladder et comme un opérateur dans des instructions de Texte structuré.

Cette instruction offre une meilleure précision sur les automates traditionnels, ce qui donne de meilleurs résultats.

**Affecte les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecte l'indicateur d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section Indicateurs d'état mathématique
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution****Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule la tangente de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	L'automate calcule la tangente de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := TAN(value);

**Voir aussi**

[Instructions de trigonométrie](#) sur la [page 728](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)



# Mathématiques avancées

## Instructions mathématiques avancées

Les instructions mathématiques avancées comprennent ces instructions :

### Diagramme à relais et bloc fonctionnel

<a href="#">LN</a>	<a href="#">LOG</a>	<a href="#">XPY</a>
--------------------	---------------------	---------------------

### Texte structuré

<a href="#">LN</a>	<a href="#">LOG</a>	<a href="#">XPY</a>
--------------------	---------------------	---------------------

Si vous voulez :	Utilisez cette instruction :
Calcule le logarithme naturel d'une valeur	LN
Calcule le logarithme décimal d'une valeur	LOG
Met une valeur à la puissance d'une autre valeur	XPY

Le mélange de types de données peut causer des erreurs de précision et d'arrondi et allonger le temps d'exécution de l'instruction. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données en **gras** sont les types de données optimaux. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

Une instruction mathématique avancée est exécutée une fois, chaque fois que l'instruction est scrutée, aussi longtemps que la condition d'entrée d'échelon est vraie. Si vous voulez que l'instruction ne soit évaluée qu'une fois, utiliser une instruction ONS pour déclencher l'instruction mathématique.

### Voir aussi

[Instructions de diverse/tableau \(fichier\)](#) sur la [page 499](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

## Logarithme décimal (LOG)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

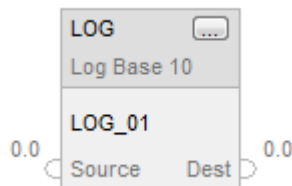
L'instruction LOG calcule le logarithme décimal de Source et mémorise le résultat dans Destination.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel



#### Texte structuré

```
dest := LOG(source);
```

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

#### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	Calcule le logarithme de cette valeur

Destination	SINT INT DINT REAL	étiquette	Étiquette où enregistrer le résultat
-------------	-----------------------------	-----------	--------------------------------------

### Texte structuré

Utiliser LOG comme une fonction. Cette fonction calcule le logarithme de Source et mémorise le résultat dans dest.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Bloc fonctionnel

Opérande	Type	Format	Description
LOG tag	FBD_MATH_ADVANCED	Structure	Structure LOG

### Structure FBD\_MATH\_ADVANCED

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	REAL	Entrée de l'instruction mathématique.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Activer la sortie.
Dest	REAL	Résultat de l'instruction mathématique. Les indicateurs d'état mathématique sont définis pour cette sortie.

### Description

L'instruction LOG calcule le logarithme décimal de Source et mémorise le résultat dans Destination. Source doit être supérieur à zéro, sinon un défaut mineur sera généré.

Source	Destination
Pas un nombre Nombre négatif Infinité négative,	Pas un nombre, le défaut mineur de débordement se produit

Zéro Nombre négatif Nombre positif	Infinité négative, le défaut mineur de débordement se produit
Nombre positif	Résultats normaux
Infinité positive	Infinité positive, le défaut mineur de débordement se produit

**Affecte les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecte l'indicateur d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section Indicateurs d'état mathématique.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	N/A.
Condition d'entrée d'échelon est vraie	L'automate calcule le logarithme naturel de Source et place le résultat dans Destination.
Post-scrutation	N/A.

**Bloc fonctionnel**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.

Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	N/A.

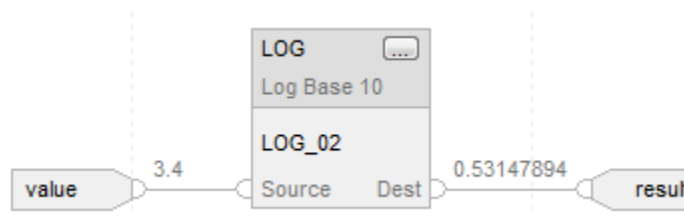
**Exemple**

Calcule le logarithme de valeur et place le résultat dans résultat.

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := LOG(value);

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Instructions mathématiques évoluées](#) sur la [page 753](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

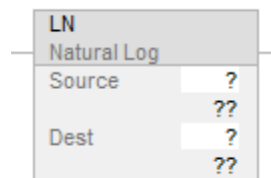
## Logarithme naturel (LN)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

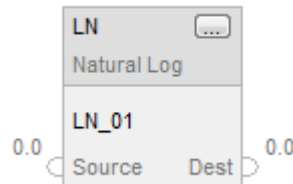
L'instruction LN calcule le logarithme naturel de Source et mémorise le résultat dans Destination.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

```
dest := LN(source);
```

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT REAL	Immédiate étiquette	Calcule le logarithme naturel de cette valeur
Destination	SINT INT DINT REAL	étiquette	Étiquette où enregistrer le résultat

**Texte structuré**

Utiliser LN comme une fonction. Cette fonction calcule le logarithme naturel de Source et mémorise le résultat dans dest.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Activer la sortie.
Dest	REAL	Résultat de l'instruction mathématique. Les indicateurs d'état mathématique sont définis pour cette sortie.

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	REAL	Entrée de l'instruction mathématique.

**Description**

L'instruction LN calcule le logarithme naturel de Source et mémorise le résultat dans Destination. Source doit être supérieur à zéro, sinon un défaut mineur sera généré.

Le tableau suivant illustre les cas spéciaux des valeurs flottantes de Source.

Source	Destination
Pas un nombre Nombre négatif Infinité négative,	Pas un nombre, le défaut mineur de débordement se produit
Zéro Nombre négatif Nombre positif	Infinité négative, le défaut mineur de débordement se produit
Infinité positive	Infinité positive, le défaut mineur de débordement se produit

**Affecte les indicateurs d'état mathématique**

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate calcule le logarithme naturel de Source et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.



Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	N/A.

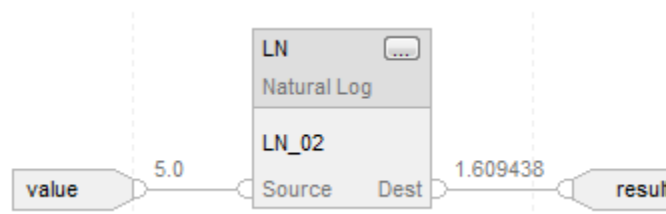
**Exemple**

Calcule le logarithme naturel de valeur et place le résultat dans résultat.

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := LN(value);

**Voir aussi**

[Instructions mathématiques évoluées](#) sur la [page 753](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

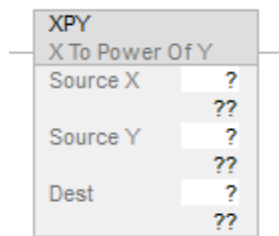
## X à la puissance Y (XPY)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

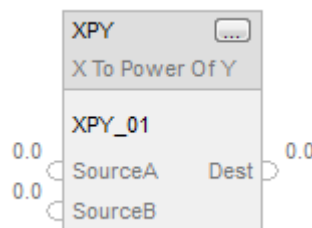
L'instruction XPY met Source A (X) à la puissance Source B (Y) et mémorise le résultat dans Destination.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

```
dest := sourceX ** sourceY;
```

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à *Conversion de données*.

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Source X	SINT INT DINT REAL	immediate étiquette	Valeur à exponentier
Source Y	SINT INT DINT REAL	immediate étiquette	exposant
Dest	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Utiliser deux signes de multiplication adjacents « \*\* » comme opérateur à l'intérieur d'une expression.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
XPY tag	FBD_MATH	Structure	Structure XPY

**Structure FBD\_MATH**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
SourceA	REAL	Valeur de référence.
SourceB	REAL	Exposant.
Paramètre de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Activer la sortie.

Dest	REAL	Résultat de l'instruction mathématique. Les indicateurs d'état mathématique sont définis pour cette sortie.
------	------	---

**Description (Description)**

L'instruction XPY met Source A (X) à la puissance Source B (Y) et mémorise le résultat dans Destination.

Si Source A (X) est négatif, Source B (Y) doit avoir une valeur non fractionnelle, sinon un défaut mineur sera généré.

Dans le cas des automates CompactLogix 5370 et ControlLogix 5570, si la valeur de référence est négative et l'exposant est un réel, la valeur absolue de la valeur de référence est utilisée.

**Affecter les indicateurs d'état mathématique**

Automates	Affecter les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Automates	Un défaut majeur se produira si :	Type de défaut	Code de défaut
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	N/A	N/A	N/A
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Source X est négatif et Source Y n'est pas une valeur entière.	4	4

Aucun spécifique à cette instruction. Reportez-vous à *Attributs communs* pour les défauts liés aux opérands.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse.	N/A.
Condition d'entrée d'échelon est vraie.	L'automate met Source X à la puissance Source Y et place le résultat dans Destination.
Post-scrutation	N/A.

**Bloc fonctionnel**

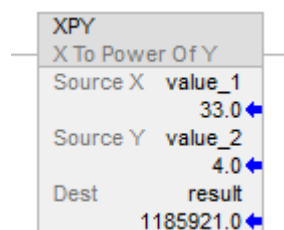
Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

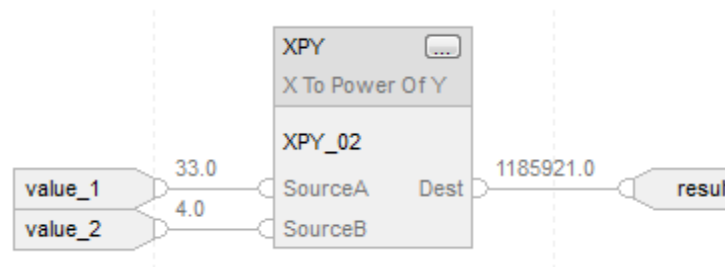
Condition/État	Action entreprise
Pré-scrutation	N/A.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie.
Post-scrutation	N/A.

**Exemple**

**Diagramme à relais**



### Bloc fonctionnel



### Texte structuré

```
result := (value_1 ** value_2);
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Instructions mathématiques avancées](#) sur la [page 753](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Attributs communs](#) sur la [page 885](#)

# Instructions de conversion mathématique

## Instructions de conversion mathématique

Les instructions conversion mathématique convertissent des valeurs.

Instructions disponibles

Diagramme à relais et bloc fonctionnel

<a href="#">DEG</a>	<a href="#">RAD</a>	<a href="#">TOD</a>	<a href="#">FRD</a>	<a href="#">TRN</a>
---------------------	---------------------	---------------------	---------------------	---------------------

Texte structuré

<a href="#">DEG</a>	<a href="#">RAD</a>	<a href="#">TRN</a>
---------------------	---------------------	---------------------

Si vous voulez	Utiliser cette instruction
Convertir des radians en degrés.	DEG
Convertir des degrés en radians.	RAD
Convertir une valeur entière en une valeur BCD.	TOD
Convertir une valeur BCD en une valeur entière.	FRD
Supprimer la partie fractionnelle d'une valeur.	TRN

Vous pouvez mélanger les types de données, mais cela pourrait entraîner une perte en précision et des erreurs d'arrondi et l'instruction prendra plus de temps à s'exécuter. Vérifier le bit S:V pour voir si les résultats sont tronqués.

Les types de données **en caractères gras** correspondent aux types optimaux de données. Une instruction s'exécute au mieux de sa vitesse et au minimum de sa mémoire requise si tous ses opérandes de l'instruction utilisent les mêmes types optimaux de données, généralement DINT ou REAL.

Une instruction conversion mathématique est exécutée une fois, chaque fois que l'instruction est scrutée, aussi longtemps que la condition d'entrée d'échelon est

vraie. Si vous voulez que l’instruction ne soit évaluée qu’une fois, utiliser une instruction ONS pour déclencher l’instruction de conversion.

**Voir aussi**

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

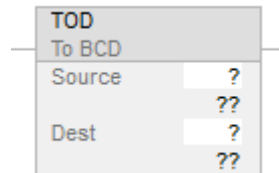
**Conversion en BCD (TOD)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

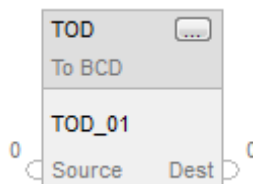
L’instruction TOD convertit une valeur décimale ( $0 \leq \text{Source} \leq 99\,999\,999$ ) en une valeur BCD et mémorise le résultat dans Destination.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

Cette instruction n’est pas disponible dans le texte structuré.



## Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

### Diagramme à relais

Opérande	Type	Format	Description
Source	SINT INT DINT	Immédiate étiquette	valeur à convertir en BCD $0 \leq \text{Source} \leq 99\,999\,999$
Destination	SINT INT DINT	étiquette	étiquette où enregistrer le résultat

### Bloc fonctionnel

Opérande	Type	Format	Description
TOD tag	FBD_CONVERT	Structure	Structure TOD

### Structure FBD\_CONVERT

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si l'option est désactivée, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est définie.
Source	DINT	Entrée de l'instruction de conversion. Valide = tout nombre entier

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Activer la sortie.
Dest	DINT	Résultat de l'instruction de conversion. Les indicateurs d'état mathématique sont définis pour cette sortie.

### Description

BCD est le système de numération décimal codé binaire qui code les chiffres décimaux individuels (de 0 à 9) en 4 bits binaires.

Source	Destination	Type de la destination
Source négative < 0	0	
Source > 99 999 999	16#9999_9999	DINT

Source > 99 999 999	16#9999	INT
Source > 99 999 999	16#99	SINT

**Affecte les indicateurs d'état mathématique**

<b>Automates</b>	<b>Affecte les indicateurs d'état mathématique</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A.
Condition d'entrée d'échelon est fausse	N/A.
Condition d'entrée d'échelon est vraie	L'automate convertit Source en BCD et place le résultat dans Destination.
Post-scrutation	N/A.

**Bloc fonctionnel**

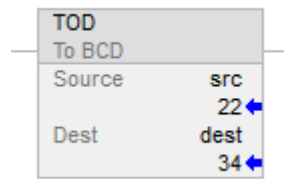
<b>Condition/État</b>	<b>Action entreprise</b>
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Exemple**

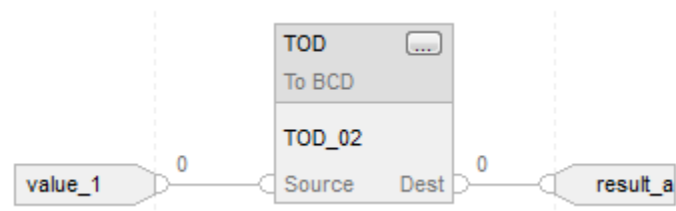
**Exemple 1**

L'instruction TOD convertit value\_1 en un valeur BCD et place le résultat dans result\_a.

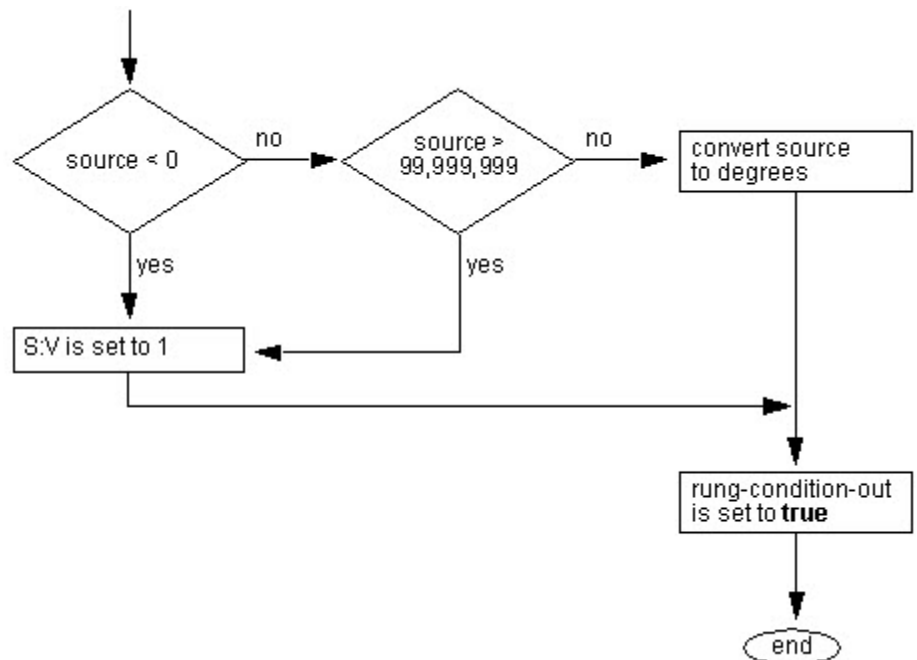
**Diagramme à relais**



**Bloc fonctionnel**



**Organigramme TOD (vrai)**



**Voir aussi**

[Instructions Calcul](#) sur la [page 373](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

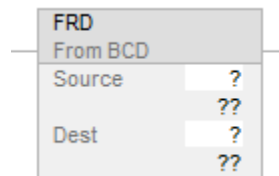
**Conversion en nombre entier (FRD)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

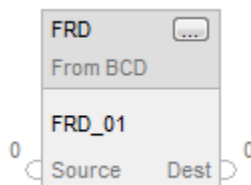
L'instruction FRD convertit une valeur BCD (Source) en une valeur décimale et mémorise le résultat dans Destination.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT	Immédiate étiquette	valeur à convertir en nombre décimal
Destination	SINT INT DINT	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
FRD tag	FBD_CONVERT	Structure	Structure FRD

**Structure FBD\_CONVERT**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source	DINT	Entrée de l'instruction de conversion. Valide = tout nombre entier

Paramètres de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	DINT	Résultat de l'instruction de conversion.

**Description**

L'instruction FRD convertit une valeur BCD (Source) en une valeur décimale et mémorise le résultat dans Destination

**Affecte les indicateurs d'état mathématique**

Automates	Affecte les indicateurs d'état mathématique
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

**Exécution****Diagramme à relais**

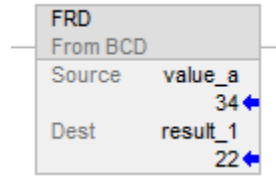
Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate convertit Source en une valeur décimale et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

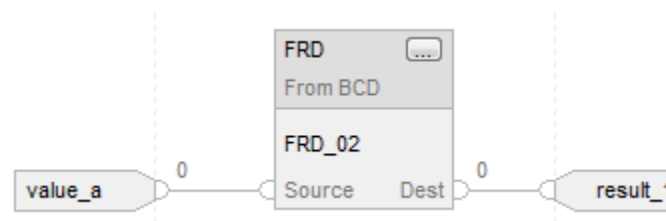
Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Exemples**

**Diagramme à relais**



**Bloc fonctionnel**



**Voir aussi**

[Instructions Calcul](#) sur la [page 373](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

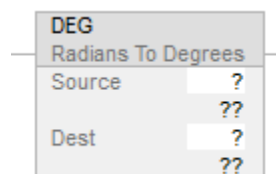
**Degrés (DEG)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

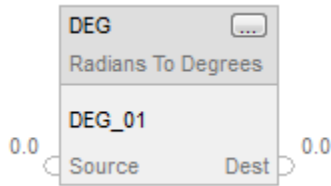
L'instruction DEG convertit Source (en radians) en degrés et mémorise le résultat dans Destination.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

dest := DEG(source);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

**Diagramme à relais**

Opérande	Type	Format	Description
Source	SINT INT DINT REA	Immédiate étiquette	valeur à convertir en degrés
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Utiliser DEG comme une fonction. Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type	Format	Description
DEG tag	FBD_MATH_ADVANCED	Structure	Structure DEG



**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de données	Description
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source	REAL	Entrée de l'instruction de conversion.

Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Indique si l'instruction est activée.
Dest	REAL	Résultat de l'instruction de conversion.

**Description**

L'instruction DEG utilise cet algorithme :

$$\text{Source} * 180 / \pi = \text{Source} * 57,29578$$

**Affecte les indicateurs d'état mathématique**

Automates	Indicateurs d'état mathématique affectés
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
Un débordement a été détecté	4	4

Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate convertit Source en radians et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

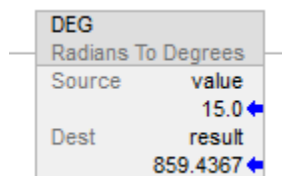
Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

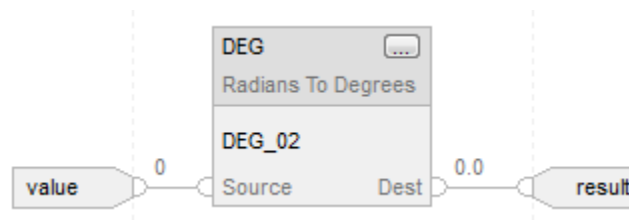
**Texte structuré**

Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel****Texte structuré**

```
result := DEG(value);
```

**Voir aussi**

[Instructions mathématiques évoluées](#) sur la [page 753](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

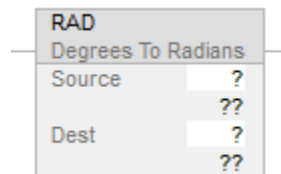
[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

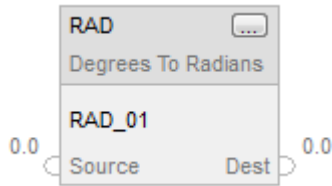
**Radians (RAD)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction RAD convertit Source (en radians) en degrés et mémorise le résultat dans Destination.

**Langages disponibles****Diagramme à relais**

**Bloc fonctionnel**



**Texte structuré**

dest := RAD(source);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à *Conversion de données*.

**Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Source (Source)	SINT INT DINT REAL	Immédiate étiquette	valeur à convertir en radians
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat

**Texte structuré**

Utiliser RAD comme une fonction. Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Bloc fonctionnel**

Opérande	Type (Type)	Format	Description (Description)
RAD tag	FBD_MATH_ADVANCED	structure	Structure FRD

**Structure FBD\_MATH\_ADVANCED**

Paramètre d'entrée	Type de donnée	Description (Description)
EnableIn	BOOL	Activer l'entrée. Si défini sur faux, l'instruction n'est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source (Source)	REAL	Entrée de l'instruction de conversion.

Paramètre de sortie	Type de donnée	Description (Description)
EnableOut	BOOL	Activer la sortie.
Dest	REAL	Résultat de l'instruction de conversion.

**Affecter les indicateurs d'état mathématique**

Automates	Indicateurs d'état mathématique affectés
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'automate convertit Source en radians et place le résultat dans Destination.
Post-scrutation	N/A

**Bloc fonctionnel**

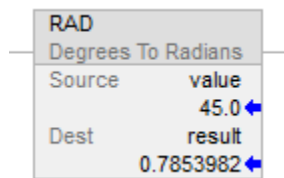
Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

**Texte structuré**

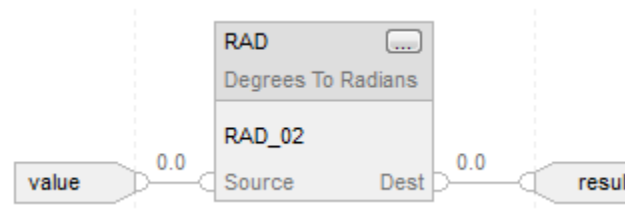
Condition/État	Action entreprise
Pré-scrutation	N/A
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Bloc fonctionnel**



**Texte structuré**

result := RAD(value);

**Voir aussi**

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

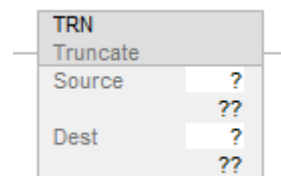
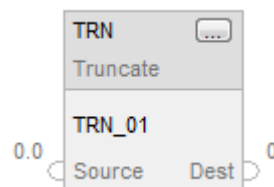
[Conversions de données](#) sur la [page 889](#)

[Instructions mathématiques avancées](#) sur la [page 753](#)

**Troncation (TRN)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.

L'instruction TRN supprime (tronque) la partie fractionnelle de Source et mémorise le résultat dans Destination.

**Langages disponibles****Diagramme à relais****Bloc fonctionnel****Texte structuré**

```
dest := TRUNC(source);
```

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section *Conversion de données*.

Le Diagramme à relais et le Bloc fonctionnel utilisent TRN comme une instruction. Lorsque l’instruction TRN est utilisée dans le Diagramme à relais, l’opérande Source n’accepte qu’une étiquette REAL ou des valeurs immédiates, Destination peut être du type REAL, DINT, SINT ou INT. Mais dans un Bloc fonctionnel, Destination peut uniquement être du type DINT.

Le Texte structuré utilise TRUNC comme un opérateur. Lorsque TRUNC est un opérateur, l’opérande Source peut accepter des valeurs REAL, SINT, INT et DINT. Mais la destination ne peut être que du type DINT.

Lorsque TRUNC est utilisé à l’intérieur d’une instruction pour expression telle que CPT, TRUNC est considéré comme un opérateur. L’opérande Source peut être de n’importe quel type d’entier comme SINT, INT, DINT ainsi que du type REAL.

**Diagramme à relais**

Opérande	Type	Format	Description
Source*	REAL	immédiate étiquette	valeur à tronquer
Destination	SINT INT DINT REAL	étiquette	étiquette où enregistrer le résultat
Conversion de données : Les étiquettes SINT et INT sont converties par extension de signe.			

**Bloc fonctionnel**

Opérande	Type	Format	Description
TRN tag	FBD_TRUNCAT E	Structure	Structure TRN

**Structure FBD\_TRUNCATE**

Paramètre d’entrée	Type de données	Description
EnableIn	BOOL	Activer l’entrée. Si défini sur faux, l’instruction n’est pas exécutée et les sorties ne sont pas mises à jour. La valeur par défaut est vraie.
Source	REAL	Entrée de l’instruction de conversion. L’entrée peut aussi être une valeur DINT, SINT et INT. Mais le type entier sera d’abord converti en REAL. Il n’y a aucune perte de précision durant la conversion de SINT ou INT en REAL. Mais il peut y avoir une perte de précision durant la conversion de DINT en REAL. Les deux types de données stockent les données en 32 bits mais le type REAL utilise certains de ces 32 bits pour stocker la valeur exposante. Si la précision est perdue, l’automate la prend de la partie moins significative des DINT.



Paramètre de sortie	Type de données	Description
EnableOut	BOOL	Activer la sortie. Définir sur faux s'il y a un débordement de Dest, sinon définir sur vrai.
Dest	DINT	Résultat de l'instruction de conversion.

### Texte structuré

Utiliser TRUNC comme une fonction. Cette fonction tronque la source et retourne un résultat entier.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

Opérande	Type	Format	Description
Source	REAL DINT SINT INT	immédiate étiquette	Entrée de l'instruction de conversion.

### Description

Tronquer n'arrondit pas la valeur, la partie non fractionnelle reste la même, quelle que soit la valeur de la partie fractionnelle.

Tronquer un grand nombre réel qui pourrait causer un débordement durant des calculs mathématiques renvoie une valeur à la place de la valeur zéro.

Vous pouvez utiliser TRN comme un opérateur dans une expression du diagramme à relais et vous pouvez utiliser TRUNC comme un opérateur dans des instructions de Texte structuré.

### Affecte les indicateurs d'état mathématique

Automates	Indicateurs d'état mathématique affectés
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Conditionnel, reportez-vous à la section <i>Indicateurs d'état mathématique</i> .
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Oui

### Défauts majeurs/mineurs

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	L'échelon est défini sur faux.
Condition d'entrée d'échelon est fausse	N/A.
Condition d'entrée d'échelon est vraie	L'automate supprime la partie fractionnelle de Source et place le résultat dans Destination. La condition d'entrée d'échelon est définie sur vrai.
Post-scrutation	L'échelon est défini sur faux.

#### Bloc fonctionnel

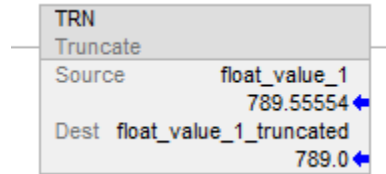
Condition/État	Action entreprise
Pré-scrutation	N/A
Tag.EnableIn est faux.	EnableOut est défini sur faux
Tag.EnableIn est vrai	EnableOut est définie sur vrai. Si le bloc produit un débordement, EnableOut est défini sur faux.
Première scrutation de l'instruction	N/A
Première exécution de l'instruction	N/A
Post-scrutation	N/A

#### Texte structuré

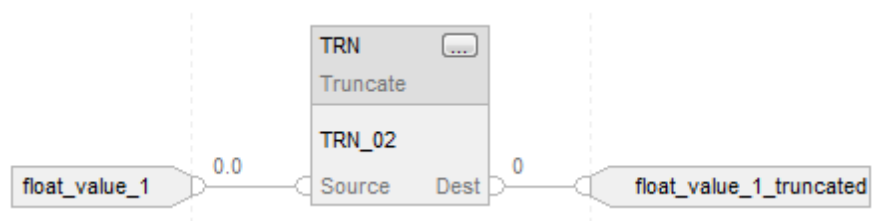
Condition/État	Action entreprise
Pré-scrutation	Reportez-vous à Pré-scrutation dans le Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est définie sur vrai dans le Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

## Exemple

### Diagramme à relais



### Bloc fonctionnel



### Texte structuré

```
float_value_1_truncated := TRUNC(float_value_1);
```

### Voir aussi

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Instructions mathématiques évoluées](#) sur la [page 753](#)

[Attributs communs](#) sur la [page 885](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)



## Instructions pour port série ASCII

### Instructions pour port série ASCII

Utiliser les instructions de port série ASCII pour lire et écrire des caractères ASCII.

---

**Important :** Pour utiliser les instructions de port série ASCII, vous devez configurer le port série de l'automate. Reportez-vous à LOGIX 5000 Controller Common Procedures manual (publication 1756-PM001) pour plus d'informations.

---

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les projets utilisant des automates qui n'ont pas de ports séries.

#### Instructions disponibles

##### Un diagramme à relais et texte structuré

<a href="#">ABL</a>	<a href="#">ACB</a>	<a href="#">ACL</a>	<a href="#">AHL</a>	<a href="#">ARD</a>	<a href="#">ARL</a>	<a href="#">AWA</a>	<a href="#">AWT</a>
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------

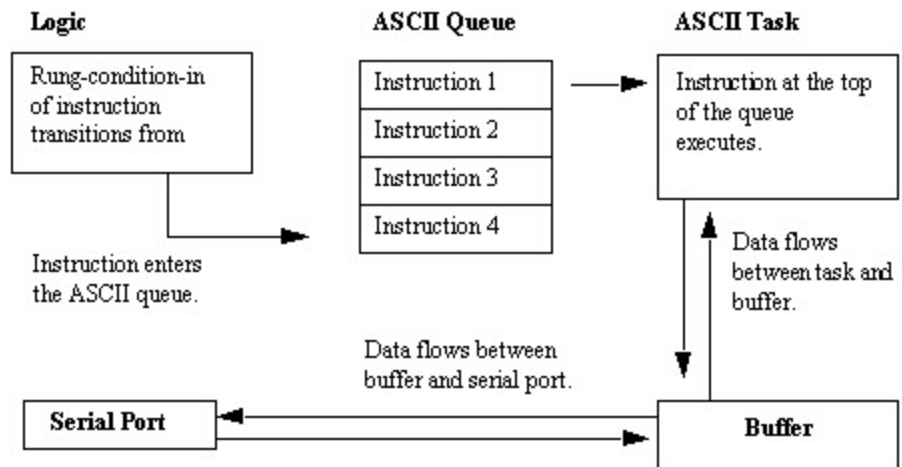
#### Bloc fonctionnel

Indisponible

Si vous voulez :	Utilisez cette instruction :
Vérifier si des données contiennent des caractères de terminaison	ABL
Vérifier si le tampon contient le nombre requis de caractères avant de les lire	ACB
Effacer le tampon. Par exemple, effacer d'anciennes données dans le tampon au démarrage, ou synchroniser le tampon avec un dispositif. Effacer les instructions de port série ASCII qui sont en cours d'exécution ou qui sont dans la file d'attente.	ACL
Obtenir l'état des lignes de commande du port série. Par exemple, faire raccrocher un modem. Activer ou désactiver le signal DTR Activer ou désactiver le signal RTS	AHL

Lire un nombre fixe de caractères. Par exemple, lire des données d'un dispositif qui envoie le même nombre de caractères à chaque transmission.	ARD
Lire un nombre variable de caractères, jusqu'à et y compris le premier ensemble de caractères de terminaison. Par exemple, lire des données d'un dispositif qui envoie un nombre variable de caractères à chaque transmission.	ARL
Envoyer des caractères et ajouter automatiquement un ou deux caractères pour marquer la fin des données. Par exemple, envoyer des messages qui utilisent toujours le ou les mêmes caractères de terminaison.	AWA
Envoyer des caractères. Par exemple, envoyer des messages qui utilisent des caractères de terminaison variés.	AWT

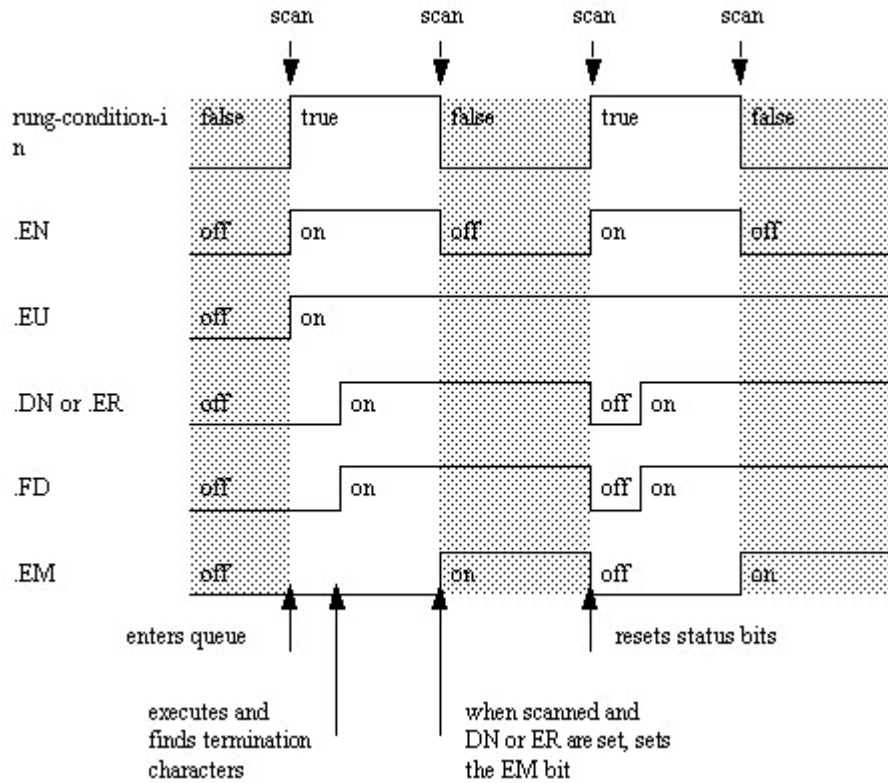
Les instructions de port série ASCII sont exécutées de façon asynchrone par rapport à la scrutation de la logique :



Chaque instruction ASCII, à l'exception de l'instruction ACL, utilise la structure SERIAL\_PORT\_CONTROL. L'opérande SerialPort Control :

- contrôle l'exécution de l'instruction
- fournit des informations d'état au sujet de l'instruction ASCII de façon asynchrone à la scrutation de la logique :

Les bits de l'opérande SerialPort Control fournissent des informations d'état :



Voir aussi

[Types de chaînes](#) sur la [page 828](#)

[Codes d'erreur ASCII](#) sur la [page 829](#)

## Caractères ASCII dans la mémoire tampon (ACB)

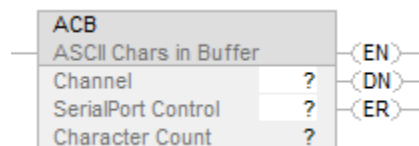
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction ACB compte les caractères dans le tampon.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

### Diagramme à relais



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

ACB(Channel,SerialPortControl);

**Opérandes**

**Diagramme à relais**

Opérande	Type	Format	Description
Canal	DINT	immédiate étiquette	0
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération
Character Count	DINT	immédiate	0 Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

**Texte structuré**

Opérande	Type	Format	Description
Canal	DINT	immédiate étiquette	0
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération
Character Count	DINT	immédiate	0 Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

Vous pouvez spécifier la valeur du Character Count en accédant au membre .POS de la structure SERIAL\_PORT\_CONTROL, plutôt que d'inclure la valeur dans la liste d'opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.



## Structure SERIAL\_PORT\_CONTROL

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le file d'attente indique l'instruction saisie dans le file d'attente ASCII.
.DN	BOOL	Le bit de fin indique quand l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé indique que l'instruction a trouvé un caractère.
.POS	DINT	La position détermine le nombre de caractères dans le tampon, jusqu'à et y compris le premier ensemble de caractères de terminaison.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

**Description**

L'instruction ACB compte les caractères dans le tampon.

Pour programmer une instruction ACB, observer ces lignes directrices :

- Configurer le port série de l'automate en mode Utilisateur.

Il s'agit d'une instruction de transition :

- Dans le diagramme à relais, basculer EnableIn de zéro à défini chaque fois que l'instruction doit être exécutée.
- Dans le texte structuré, mettre une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition

**Indicateurs d'état mathématique**

Non

**Conditions de défaut**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

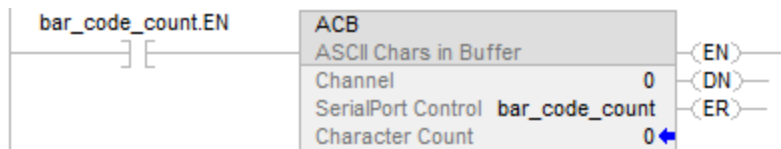
Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction est exécutée lorsque EnableIn bascule de zéro à défini.
Post-scrutation	N/A

**Texte structuré**

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction est exécutée lorsque EnableIn bascule de zéro à défini.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Texte structuré**

ACB(0,bar\_code\_count);

**Voir aussi**

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Attributs communs](#) sur la [page 885](#)

## Effacement ASCII de la mémoire tampon (ACL)

Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction ACL efface immédiatement le tampon et la file d'attente ASCII.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

#### Diagramme à relais

ACL	
ASCII Clear Buffer	
Channel	?
Clear Serial Port Read	?
Clear Serial Port Write	?

#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

ACL(Channel,ClearSerialPortRead,ClearSerialPortWrite);

#### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description
Canal	DINT	immédiate étiquette	0
Clear Serial Port Read	BOOL	immédiate étiquette	Pour vider le tampon et supprimer les instructions ARD et ARL de la file d'attente, entrer 1.
Clear Serial Port Write	BOOL	immédiate étiquette	Pour supprimer les instructions AWA et AWT de la file d'attente, entrer 1.

#### Texte structuré

Opérande	Type	Format	Description
Canal	DINT	immédiate étiquette	0
Clear Serial Port Read	BOOL	immédiate étiquette	Pour vider le tampon et supprimer les instructions ARD et ARL de la file d'attente, entrer 1.
Clear Serial Port Write	BOOL	immédiate étiquette	Pour supprimer les instructions AWA et AWT de la file d'attente, entrer 1.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Description**

L'instruction ACL exécute immédiatement une ou deux des actions suivantes :

- Elle efface le tampon ou les caractères et efface les instructions de lecture de la file d'attente ASCII
- Elle efface les instructions d'écriture de la file d'attente ASCII. Pour programmer les instructions ACL, observez ces lignes directrices :

Configurer le port série de l'automate :

Si votre application :	Alors :
Utilise l'instruction ARD ou l'instruction ARL	Sélectionner le mode Utilisateur
N'utilise ni l'instruction ARD, ni l'instruction ARL	Sélectionner soit le mode Système, soit le mode Utilisateur

Pour déterminer si une instruction a été supprimée du file d'attente ou abandonnée, examinez ce qui suit dans l'instruction appropriée :

- le bit .ER est défini
- le membre .ERROR vaut 16#E

**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

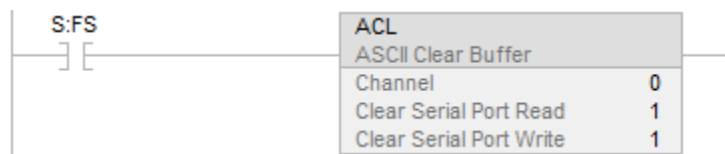
**Exécution**

**Diagramme à relais**

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction efface les instructions et le ou les tampons spécifiés
Post-scrutation	N/A

**Exemple****Diagramme à relais****Texte structuré**

```
IF (osri_1.OutputBit THEN
```

```
ACL(0,0,1);
```

```
END_IF;
```

**Voir aussi**

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Test ASCII pour ligne de mémoire tampon \(ABL\)](#) sur la [page 813](#)

[Caractères ASCII dans la mémoire tampon \(ACB\)](#) sur la [page 791](#)

[Lignes d'échange ASCII \(AHL\)](#) sur la [page 798](#)

[Lecture ASCII \(ARD\)](#) sur la [page 803](#)

[Ligne de lecture ASCII \(ARL\)](#) sur la [page 807](#)

[Ajout d'écriture ASCII \(AWA\)](#) sur la [page 823](#)

[Écriture ASCII \(AWT\)](#) sur la [page 817](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

## Lignes d'échange ASCII (AHL)

Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction AHL obtient l'état des lignes de commande et active ou désactive les signaux DTR et RTS.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

AHL(Channel,ANDMask,ORMask,SerialPortControl);

#### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description	
Canal	DINT	immédiate étiquette	0	
ANDMask	DINT	immédiate étiquette	Reportez-vous à la description	
ORMask	DINT	immédiate étiquette		
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	Étiquette qui contrôle l'opération	
Channel Status (Decimal)	DINT	immédiate	0	
			Durant l'exécution, affiche l'état des lignes de commande.	
			<b>Pour l'état de cette ligne de commande :</b>	<b>Examiner ce bit :</b>
		CTS	0	

			RTS	1
			DSR	2
			DCD	3
			DTR	4
			A reçu le caractère XOFF	5

**Texte structuré**

Opérande	Type	Format	Description	
Canal	DINT	immédiate étiquette	0	
ANDMask	DINT	immédiate étiquette	Reportez-vous à la description	
ORMask	DINT	immédiate étiquette		
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	Étiquette qui contrôle l'opération	
Channel Status (Decimal)	DINT	immédiate	0 Durant l'exécution, affiche l'état des lignes de commande.	
			<b>Pour l'état de cette ligne de commande :</b>	<b>Examiner ce bit :</b>
			CTS	0
			RTS	1
			DSR	2
			DCD	3
			DTR	4
A reçu le caractère XOFF	5			

Vous pouvez spécifier la valeur de Channel Status en accédant au membre .POS de la structure SERIAL\_PORT\_CONTROL, plutôt que d'inclure la valeur dans la liste d'opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Structure SERIAL\_PORT\_CONTROL**

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique que l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.

.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé ne s'applique pas à cette instruction.
.POS	DINT	La position détermine le nombre de caractères dans le tampon, jusqu'à et y compris le premier ensemble de caractères de terminaison.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

**Description**

L'instruction AHL peut :

- Obtenir l'état des lignes de commande du port série
- Activer ou désactiver le signal Terminal de données prêt (DTR)
- Activer ou désactiver le signal Demande pour émettre (RTS)

Pour programmer l'instruction AHL, observer ces lignes directrices :

Configurer le port série de l'automate :

<b>Si votre application :</b>	<b>Alors :</b>
Utilise l'instruction ARD ou l'instruction ARL	Sélectionner le mode Utilisateur
N'utilise ni l'instruction ARD, ni l'instruction ARL	Sélectionner soit le mode Système, soit le mode Utilisateur

Utiliser le tableau suivant pour sélectionner les valeurs correctes des opérandes ANDMask et ORMask :

Pour mettre DTR dans l'état :	Et mettre RTS dans l'état :	Entrer cette valeur ANDMask :	Et entrer cette valeur ORMask :
Off	Off	3	0
		on	1
		Inchangé	1
On	Off	2	1
		on	0
		Inchangé	0
Inchangé	Off	2	0
		on	0
		Inchangé	0

Il s'agit d'une instruction de transition :



- Dans le diagramme à relais, basculer EnableIn de zéro à défini chaque fois que l'instruction doit être exécutée.
- Dans le texte structuré, mettre une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition

### Affecte les indicateurs d'état mathématique

Non

### Conditions de défaut

Type	Code	Cause	Méthode de récupération
4	57	L'exécution de l'instruction AHL a échoué car le port série est défini sur Pas d'échange.	Changer le paramètre Ligne de commande du port série. ou Supprimer l'instruction AHL

### Exécution

#### Diagramme à relais

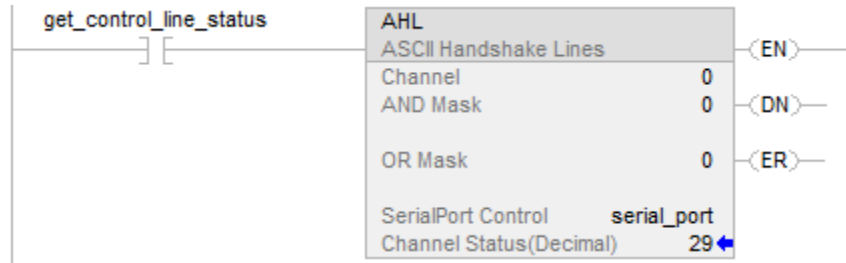
Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction est exécutée lorsque la condition d'entrée d'échelon passe de zéro à défini.
Post-scrutation	N/A

#### Texte structuré

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction est exécutée lorsque la condition d'entrée d'échelon passe de zéro à défini.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Texte structuré**

```
osri_1.InputBit := get_control_line_status;
OSRI(osri_1);
IF (osri_1.OutputBit) THEN
    AHL(0,0,0,serial_port);
END_IF;
```

**Voir aussi**

- [Instructions pour port série ASCII sur la page 789](#)
- [Test ASCII pour ligne de mémoire tampon \(ABL\) sur la page 813](#)
- [Caractères ASCII dans la mémoire tampon \(ACB\) sur la page 791](#)
- [Effacement ASCII de la mémoire tampon \(ACL\) sur la page 795](#)
- [Lecture ASCII \(ARD\) sur la page 803](#)
- [Ligne de lecture ASCII \(ARL\) sur la page 807](#)
- [Ajout d'écriture ASCII \(AWA\) sur la page 823](#)
- [Écriture ASCII \(AWT\) sur la page 817](#)
- [Attributs communs sur la page 885](#)

## Lecture ASCII (ARD)

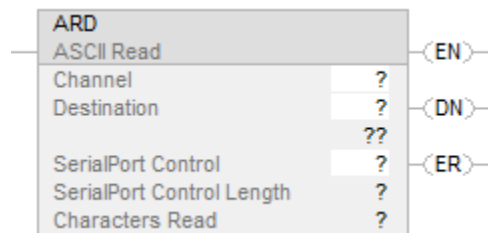
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction ARD supprime des caractères du tampon et les enregistre dans Destination.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

ARD(Channel, Destination, SerialPortControl);

### Opérandes

### Diagramme à relais

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Destination	Type de chaîne SINT INT DINT	étiquette	étiquette dans laquelle les caractères sont déplacés (c'est-à-dire, lus) : Pour un type de chaîne, entrezr le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez

Serial Port Control	SERIAL_POR T_CONTROL	étiquette	étiquette qui contrôle l'opération	
Serial Port Control Length	DINT	immédiate	nombre de caractères à déplacer vers la destination (lus)	La valeur Serial Port Control Length doit être inférieure ou égale à la taille de Destination. Si vous souhaitez définir la valeur Serial Port Control Length égale à la taille de Destination, entrer 0.
Characters Read	DINT	immédiate	0	Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

**Texte structuré**

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Destination	Type de chaîne SINT INT DINT	étiquette	étiquette dans laquelle les caractères sont déplacés (c'est-à-dire, lus) : Pour un type de chaîne, entrezr le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
Serial Port Control	SERIAL_POR T_CONTROL	étiquette	étiquette qui contrôle l'opération	
Serial Port Control Length	DINT	immédiate	nombre de caractères à déplacer vers la destination (lus)	La valeur Serial Port Control Length doit être inférieure ou égale à la taille de Destination. Si vous souhaitez définir la valeur Serial Port Control Length égale à la taille de Destination, entrer 0.
Characters Read	DINT	immédiate	0	Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

Vous pouvez indiquer les valeurs Serial Port Control Length et Characters Read en accédant aux membres .LEN et .POS de la structure SERIAL\_PORT\_CONTROL, plutôt qu'en incluant les valeurs dans la liste des opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Structure SERIAL\_PORT\_CONTROL

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique que l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé ne s'applique pas à cette instruction.
.LEN	DINT	La longueur indique le nombre de caractères à déplacer vers la destination (c'est-à-dire, lus).
.POS	DINT	La position affiche le nombre de caractères lus.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

### Description

L'instruction ARD supprime le nombre de caractères indiqué du tampon et les enregistre dans Destination.

- L'instruction ARD continue de s'exécuter jusqu'à ce qu'elle ait supprimé le nombre de caractères indiqué (opérande Serial Port Control Length).
- Aucune autre instruction de port série ASCII ne s'exécute pendant l'exécution de l'instruction ARD.

Pour programmer l'instruction ARD, observer ces lignes directrices :

1. Configurer le port série de l'automate en mode Utilisateur.
2. Utilisez le résultat d'une instruction ACB pour déclencher l'instruction ARD.

Cela empêche l'instruction ARD de retenir la file d'attente pendant qu'elle attend le nombre requis de caractères. Pour plus d'informations, reportez-vous à l'exemple d'instruction ARD ci-dessous.

3. Il s'agit d'une instruction de transition :  
 Dans le diagramme à relais, basculez EnableIn de mis à zéro à défini chaque fois que l'instruction doit s'exécuter.  
 Dans le texte structuré, mettez une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition
4. Pour déclencher une action suivante lorsque l'instruction est terminée, examiner le bit .EM.

**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

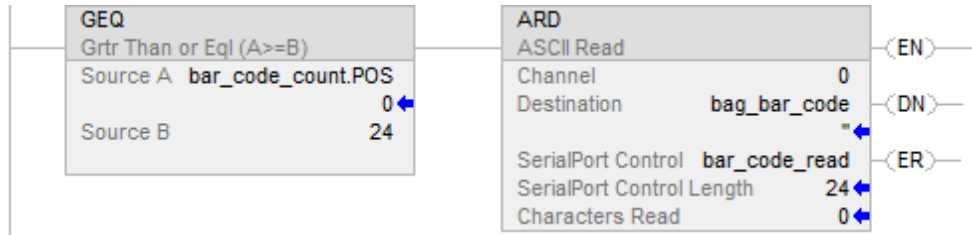
Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

**Texte structuré**

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

## Exemples

## Diagramme à relais



## Texte structuré

```
ACB(0,bar_code_count);

IF bar_code_count.POS >= 24 THEN

bar_code_read.LEN := 24;

ARD(0,bag_bar_code,bar_code_read);

END_IF;
```

## Voir aussi

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Ligne de lecture ASCII (ARL)

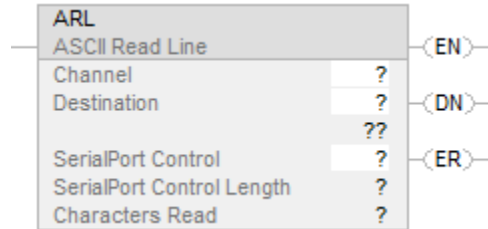
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction ARL supprime des caractères du tampon et les enregistre dans Destination.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

ARL(Channel, Destination, SerialPortControl);

#### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Destination	Type de chaîne SINT INT DINT	étiquette	étiquette dans laquelle les caractères sont déplacés (c'est-à-dire, lus) Pour un type de chaîne, entrez le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrez le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération	



Serial Port Control Length	DINT	immédiate	nombre maximum de caractères à lire si aucun caractère de terminaison n'est trouvé.	Entrer le nombre maximum de caractères contenus dans tous les messages (c'est-à-dire qui indique quand arrêter la lecture si aucun caractère de terminaison n'est trouvé). Par exemple, si les messages ont de 3 à 6 caractères, entrer 6. La valeur Serial Port Control Length doit être inférieure ou égale à la taille de Destination. Si vous souhaitez définir la valeur Serial Port Control Length égale à la taille de Destination, entrer 0.
Characters Read	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères lus

### Texte structuré

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Destination	Type de chaîne SINT INT DINT	étiquette	étiquette dans laquelle les caractères sont déplacés (c'est-à-dire, lus) Pour un type de chaîne, entrez le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération	

Serial Port Control Length	DINT	immédiate	nombre maximum de caractères à lire si aucun caractère de terminaison n'est trouvé.	Entrer le nombre maximum de caractères contenus dans tous les messages (c'est-à-dire qui indique quand arrêter la lecture si aucun caractère de terminaison n'est trouvé). Par exemple, si les messages ont de 3 à 6 caractères, entrer 6. La valeur Serial Port Control Length doit être inférieure ou égale à la taille de Destination. Si vous souhaitez définir la valeur Serial Port Control Length égale à la taille de Destination, entrer 0.
Characters Read	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères lus

Cependant, vous pouvez indiquer les valeurs Serial Port Control Length et Characters Read en accédant aux membres .LEN et .POS de la structure SERIAL\_PORT\_CONTROL, plutôt qu'en incluant les valeurs dans la liste des opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Structure SERIAL\_PORT\_CONTROL**

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique que l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé ne s'applique pas à cette instruction.
.LEN	DINT	La longueur indique le nombre maximum de caractères à déplacer vers la destination (c'est-à-dire qui indique quand arrêter la lecture si aucun caractère de terminaison n'est trouvé).
.POS	DINT	La position affiche le nombre de caractères lus.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

## Description

L'instruction ARL supprime des caractères du tampon et les enregistre dans Destination, comme suit :

- L'instruction ARL continue de s'exécuter jusqu'à ce qu'elle supprime :
  - soit le premier ensemble de caractères de terminaison
  - soit le nombre de caractères indiqués (opérande String Length)

Aucune autre instruction ASCII ne s'exécute pendant l'exécution de l'instruction ARL. Pour programmer l'instruction ARL, observer ces lignes directrices :

1. Configurer le port série de l'automate pour le mode Utilisateur et définir les caractères servant de caractères de terminaison.
2. Utilisez les résultats d'une instruction ABL pour déclencher l'instruction ARL.  
Cela empêche l'instruction ARL de retenir la file d'attente pendant qu'elle attend les caractères de terminaison. Pour plus d'informations, reportez-vous à l'exemple d'ARL ci-dessous.
3. Il s'agit d'une instruction de transition :  
Dans le diagramme à relais, basculez EnableIn de mis à zéro à défini chaque fois que l'instruction doit s'exécuter. Dans le texte structuré, mettez une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition
4. Pour déclencher une action suivante lorsque l'instruction est terminée, examiner le bit .EM.

## Affecte les indicateurs d'état mathématique

Non

## Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

## Exécution

### Diagramme à relais

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A

Condition d'entrée d'échelon est vraie	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

**Texte structuré**

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

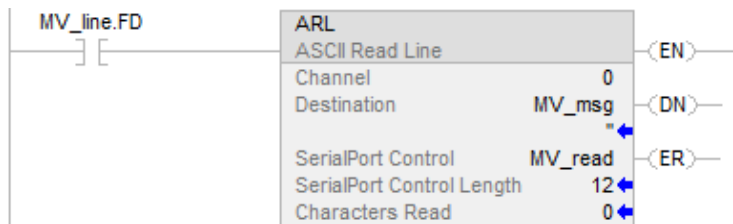
**Exemple**

Teste le tampon en continu pour obtenir un message du terminal MessageView. Étant donné que chaque message se termine sur un retour chariot (\$r), le retour chariot est configuré comme le caractère de terminaison sur l'onglet Protocole utilisateur (User Protocol) de la boîte de dialogue Propriétés de l'automate (Controller Properties).

Lorsque l'instruction ABL trouve un retour chariot, elle définit le bit .FD. Lorsque l'instruction ABL trouve le retour chariot (MV\_line.FD est défini), l'automate a reçu un message complet.

L'instruction ARL supprime les caractères du tampon, jusqu'à et y compris le retour chariot, et les places dans le membre DATA de l'étiquette MV\_msg, qui est un type de chaîne.

**Diagramme à relais**



**Texte structuré**

```

ABL(0,MV_line);

osri_1.InputBit :=MVLine.FD

OSRI(osri_1);

IF osri_1.OutputBit) THEN

mv_read.LEN := 12;
    
```

```
ARL(0,MV_msg,MV_read);
```

```
END_IF;
```

### Voir aussi

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Test ASCII pour ligne de mémoire tampon \(ABL\)](#) sur la [page 813](#)

[Caractères ASCII dans la mémoire tampon \(ACB\)](#) sur la [page 791](#)

[Effacement ASCII de la mémoire tampon \(ACL\)](#) sur la [page 795](#)

[Lignes d'échange ASCII \(AHL\)](#) sur la [page 798](#)

[Lecture ASCII \(ARD\)](#) sur la [page 803](#)

[Ajout d'écriture ASCII \(AWA\)](#) sur la [page 823](#)

[Écriture ASCII \(AWT\)](#) sur la [page 817](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Test ASCII pour ligne de mémoire tampon (ABL)

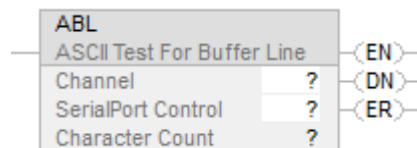
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction ABL compte les caractères présents dans le tampon jusqu'au premier caractère de terminaison inclus.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

#### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

ABL(Channel,SerialPortControl);

**Opérandes**

**Diagramme à relais**

Opérande	Type	Format	Description
Canal	DINT	immédiate	0
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération
Character Count	DINT	immédiate	0 Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

**Texte structuré**

Opérande	Type	Format	Description
Canal	DINT	immédiate	0
SerialPort Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération
Character Count	DINT	immédiate	0 Durant l'exécution, affiche le nombre de caractères dans le tampon, y compris le premier ensemble de caractères de terminaison.

Vous pouvez accéder à la valeur Character Count par le biais du membre .POS de la structure SERIAL\_PORT\_CONTROL.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Structure SERIAL\_PORT\_CONTROL**

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique quand l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).

.FD	BOOL	Le bit de découverte indique que l'instruction a trouvé les caractères de terminaison.
.POS	DINT	La position détermine le nombre de caractères dans le tampon, jusqu'à et y compris le premier ensemble de caractères de terminaison. L'instruction ne renvoie ce nombre qu'après avoir trouvé les caractères de terminaison.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

### Description

L'instruction ABL lance une recherche du premier ensemble de caractères de terminaison dans le tampon. Si l'instruction trouve les caractères de terminaison :

- elle définit le bit .FD
- elle compte les caractères présents dans le tampon, jusqu'à et y compris le premier ensemble de caractères de terminaison.

L'onglet **Protocole Utilisateur** (User Protocol) de la boîte de dialogue **Propriétés de l'automate** (Controller Properties) définit les caractères ASCII que l'instruction considère comme les caractères de terminaison.

Pour programmer l'instruction ABL, observer ces lignes directrices :

- Configurer le port série de l'automate pour le mode Utilisateur et définir les caractères servant de caractères de terminaison.

Il s'agit d'une instruction de transition :

- Dans le diagramme à relais, basculer EnableIn de zéro à défini chaque fois que l'instruction doit être exécutée.
- Dans le texte structuré, mettre une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition

### Affecte les indicateurs d'état mathématique

Non

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

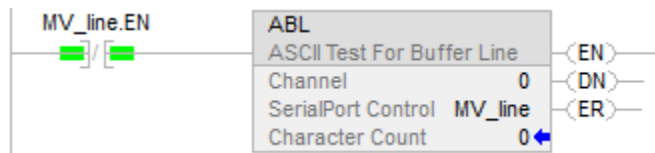
Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

**Texte structuré**

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

**Exemple**

**Diagramme à relais**



**Texte structuré**

ABL(0,MV\_line);

**Voir aussi**

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Caractères ASCII dans la mémoire tampon \(ACB\)](#) sur la [page 791](#)

[Effacement ASCII de la mémoire tampon \(ACL\)](#) sur la [page 795](#)

[Lignes d'échange ASCII \(AHL\)](#) sur la [page 798](#)

[Lecture ASCII \(ARD\)](#) sur la [page 803](#)

[Ligne de lecture ASCII \(ARL\)](#) sur la [page 807](#)

[Ajout d'écriture ASCII \(AWA\)](#) sur la [page 823](#)



[Ecriture ASCII \(AWT\)](#) sur la [page 817](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Ecriture ASCII (AWT)

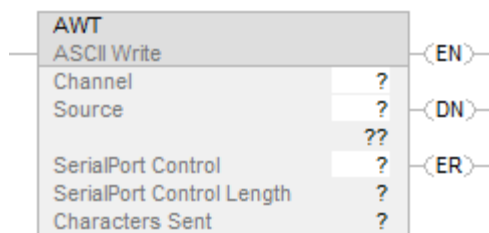
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction AWT envoie des caractères du tableau Source vers un appareil de série.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

```
AWT(Channel,Source,SerialPortControl);
```

### Opérandes

#### Diagramme à relais

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Source	Type de chaîne SINT INT DINT	étiquette	Étiquette contenant les caractères à envoyer Pour un type de chaîne, entrez le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
Serial Port Control	SERIAL_PORT_CONTROL	étiquette	Étiquette qui contrôle l'opération	
Serial Port Control Length	DINT	immédiate	Nombre de caractères à envoyer	La valeur Serial Port Control Length doit être inférieure ou égal à la taille de Source. Si vous souhaitez définir la valeur Serial Port Control Length égale au nombre de caractères dans Source, entrer 0.
Characters Sent	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères envoyés

#### Texte structuré

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Source	Type de chaîne SINT INT DINT	étiquette	Étiquette contenant les caractères à envoyer Pour un type de chaîne, entrez le nom de l'étiquette. Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
Serial Port Control	SERIAL_PORT_CONTROL	étiquette	Étiquette qui contrôle l'opération	
Serial Port Control Length	DINT	immédiate	Nombre de caractères à envoyer	La valeur Serial Port Control Length doit être inférieure ou égal à la taille de Source. Si vous souhaitez définir la valeur Serial Port Control Length égale au nombre de caractères dans Source, entrer 0.
Characters Sent	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères envoyés

Vous pouvez indiquer les valeurs Serial Port Control Length et Characters Sent en accédant aux membres .LEN et .POS de la structure SERIAL\_PORT\_CONTROL, plutôt qu'en incluant les valeurs dans la liste des opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Structure SERIAL\_PORT\_CONTROL

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique que l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.
.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé ne s'applique pas à cette instruction.
.LEN	DINT	La longueur indique le nombre de caractères à envoyer.
.POS	DINT	La position affiche le nombre de caractères envoyés.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

### Description

L'instruction AWT envoie le nombre de caractères indiqué (c'est-à-dire la longueur de contrôle du port série) de l'étiquette Source à l'appareil connecté au port série de l'automate.

Pour programmer l'instruction AWT, observer ces lignes directrices :

1. Configurer le port série de l'automate :

Si votre application :	Alors :
Utilise l'instruction ARD ou l'instruction ARL	Sélectionner le mode Utilisateur
N'utilise ni l'instruction ARD, ni l'instruction ARL	Sélectionnez le mode Système ou Utilisateur

2. Il s'agit d'une instruction de transition : Dans le diagramme à relais, basculer EnableIn de zéro à défini chaque fois que l'instruction doit être exécutée. Dans le texte structuré, mettre une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition
3. Souhaitez-vous toujours envoyer le même nombre de caractères chaque fois que l'instruction s'exécute ?

Si :	Alors :
Oui	Dans Serial Port Control Length, entrer le nombre de caractères à envoyer.
Non	Avant que l'instruction s'exécute, déplacer le membre LEN de l'étiquette Source vers le membre LEN de l'étiquette Serial Port Control. Voir exemple 2 ci-dessous.

### Affecte les indicateurs d'état mathématique

Non

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

#### Texte structuré

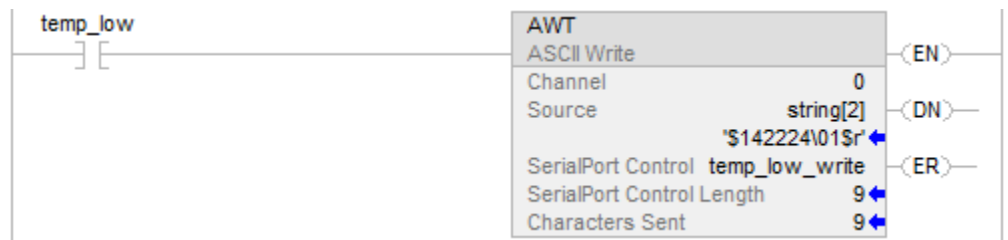
Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

## Exemples

### Exemple 1

Lorsque la température dépasse la limite basse (`temp_low` s'est défini), l'instruction AWT envoie un message au terminal MessageView connecté au port série de l'automate. Le message contient neuf caractères du membre DATA de l'étiquette `string[2]`, qui est un type de chaîne. (Le `$14` compte comme un caractère ; c'est un code hexadécimal pour le caractère Ctrl-T.) Le dernier caractère est un retour chariot (`$r`), qui marque la fin du message.

### Diagramme à relais



### Texte structuré

```
osri_1.InputBit := temp_low;

OSRI(osri_1);

IF (osri_1.OutputBit) THEN

temp_low_write.LEN := 9;

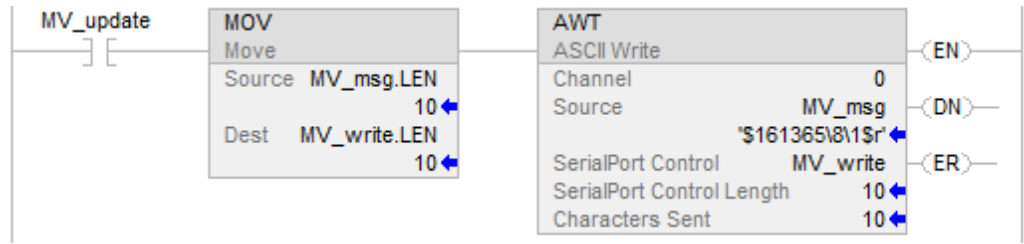
AWT(0.string[2],temp_low_write);

END_IF;
```

### Exemple 2

Lorsque `MV_update` est activé, l'instruction AWT envoie les caractères dans `MV_msg`. Étant donné que le nombre de caractères dans `MV_msg` varie, l'échelon déplace d'abord la longueur de la chaîne (`MV_write.LEN`) vers Serial Port Control Length de l'instruction AWT (`MV_write.LEN`). (Dans `MV_msg`, le `$16` compte comme un caractère ; c'est le code hexadécimal pour le caractère Ctrl-T.)

**Diagramme à relais**



**Texte structuré**

```

osri_1.InputBit := MV_update;

OSRI(osri_1);

IF (osri_1.OutputBit) THEN

MV_write.LEN := Mv_msg.LEN;

AWT(0.MV_msg,MV_write);

END_IF;

```

**Voir aussi**

- [Instructions pour port série ASCII](#) sur la [page 789](#)
- [Test ASCII pour ligne de mémoire tampon \(ABL\)](#) sur la [page 813](#)
- [Caractères ASCII dans la mémoire tampon \(ACB\)](#) sur la [page 791](#)
- [Effacement ASCII de la mémoire tampon \(ACL\)](#) sur la [page 795](#)
- [Lignes d'échange ASCII \(AHL\)](#) sur la [page 798](#)
- [Lecture ASCII \(ARD\)](#) sur la [page 803](#)
- [Ligne de lecture ASCII \(ARL\)](#) sur la [page 807](#)
- [Ajout d'écriture ASCII \(AWA\)](#) sur la [page 823](#)
- [Attributs communs](#) sur la [page 885](#)
- [Syntaxe du texte structuré](#) sur la [page 917](#)

## Ajout d'écriture ASCII (AWA)

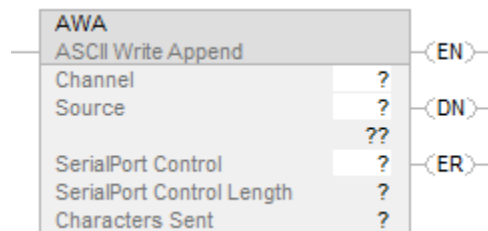
Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

L'instruction AWA envoie des caractères du tableau Source vers un appareil en série et ajoute un ou deux caractères prédéfinis.

**Astuce :** Les instructions de port série ASCII (AWT, AWA, ARD, ARL, ABL, ACB, AHL, ACL) ne sont pas disponibles pour les automates qui n'ont pas de port série.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

AWA(Channel,Source,SerialPortControl);

### Opérandes

### Diagramme à relais

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Source	Type de chaîne SINT INT DINT	étiquette	étiquette contenant les caractères à envoyer Pour un type de chaîne, entrer le nom de l'étiquette Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
Serial Port Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération	

Serial Port Control Length	DINT	immédiate	nombre de caractères à envoyer	La valeur Serial Port Control Length doit être inférieure ou égal à la taille de Source. Si vous souhaitez définir la valeur Serial Port Control Length égale au nombre de caractères dans Source, entrer 0.
Characters Sent	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères envoyés.

**Texte structuré**

Opérande	Type	Format	Description	Remarques
Canal	DINT	immédiate étiquette	0	
Source	Type de chaîne SINT INT DINT	étiquette	étiquette contenant les caractères à envoyer Pour un type de chaîne, entrer le nom de l'étiquette Pour un tableau SINT, INT ou DINT, entrer le premier élément du tableau.	Si vous voulez comparer, convertir ou manipuler les caractères, entrer une étiquette de type de chaîne. Il existe différents types de chaînes : type de données STRING par défaut tout nouveau type de chaîne que vous créez
Serial Port Control	SERIAL_PORT_CONTROL	étiquette	étiquette qui contrôle l'opération	
Serial Port Control Length	DINT	immédiate	nombre de caractères à envoyer	La valeur Serial Port Control Length doit être inférieure ou égal à la taille de Source. Si vous souhaitez définir la valeur Serial Port Control Length égale au nombre de caractères dans Source, entrer 0.
Characters Sent	DINT	immédiate	0	Pendant l'exécution, affiche le nombre de caractères envoyés.

Vous pouvez indiquer les valeurs Serial Port Control Length et Characters Sent en accédant aux membres .LEN et .POS de la structure SERIAL\_PORT\_CONTROL, plutôt qu'en incluant les valeurs dans la liste des opérandes.

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Structure SERIAL\_PORT\_CONTROL**

Mnémonique	Type de données	Description
.EN	BOOL	Le bit d'activation indique que l'instruction a été activée.
.EU	BOOL	Le bit de file d'attente indique l'instruction saisie dans la file d'attente ASCII.
.DN	BOOL	Le bit de fin indique que l'instruction est terminée, mais il est asynchrone par rapport à la scrutation de la logique.
.RN	BOOL	Le bit d'exécution indique que l'instruction est en cours d'exécution.
.EM	BOOL	Le bit vide indique quand l'instruction est terminée, mais il est synchrone par rapport à la scrutation de la logique.



.ER	BOOL	Le bit d'erreur indique quand l'instruction échoue (erreurs).
.FD	BOOL	Le bit trouvé ne s'applique pas à cette instruction.
.LEN	DINT	La longueur indique le nombre de caractères à envoyer.
.POS	DINT	La position affiche le nombre de caractères envoyés.
.ERROR	DINT	L'erreur contient une valeur hexadécimale qui identifie la cause d'une erreur.

### Description

L'instruction AWA :

- Envoie le nombre de caractères indiqué (c'est-à-dire la longueur de contrôle du port série) de l'étiquette Source à l'appareil connecté au port série de l'automate.
- Ajoute à la fin des caractères (c'est-à-dire ajoute) un ou deux caractères définis sur l'onglet Protocole utilisateur (User Protocol) de la boîte de dialogue Propriétés de l'automate (Controller Properties).

Pour programmer l'instruction AWA, observer ces lignes directrices :

1. Configurer le port série de l'automate :

Si votre application :	Alors :
Utilise l'instruction ARD ou l'instruction ARL	Sélectionner le mode Utilisateur
N'utilise ni l'instruction ARD, ni l'instruction ARL	Sélectionner soit le mode Système, soit le mode Utilisateur

2. Il s'agit d'une instruction de transition : Dans le diagramme à relais, basculer EnableIn de zéro à défini chaque fois que l'instruction doit être exécutée. Dans le texte structuré, mettre une condition sur l'instruction afin qu'elle ne soit exécutée que lors d'une transition
3. Souhaitez-vous toujours envoyer le même nombre de caractères chaque fois que l'instruction s'exécute ?

Si :	Alors :
Oui	Dans Serial Port Control Length, entrer le nombre de caractères à envoyer.
Non	Avant que l'instruction s'exécute, déplacer le membre LEN de l'étiquette Source vers le membre LEN de l'étiquette Serial Port Control. (Voir exemple 2 ci-dessous.)

### Affecte les indicateurs d'état mathématique

Non

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

#### Texte structuré

Condition	Action de texte structuré
Pré-scrutation	N/A
Exécution normale	L'instruction s'exécute. EnableIn bascule de mis à zéro à défini.
Post-scrutation	N/A

### Exemples

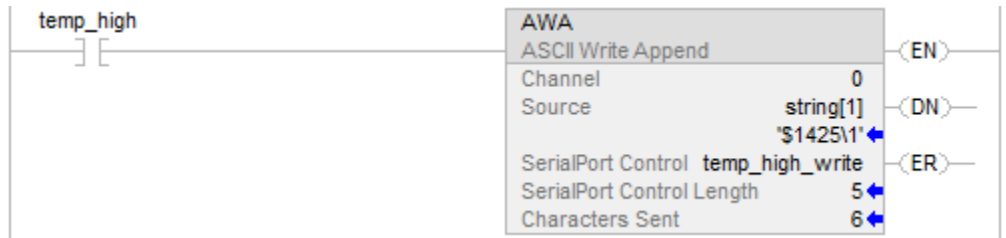
#### Exemple 1

Lorsque la température dépasse la limite haute (temp\_high défini), l'instruction AWA envoie un message au terminal MessageView connecté au port série de l'automate.

Le message contient cinq caractères du membre DATA de l'étiquette string[1], qui est un type de chaîne. (Le \$14 compte comme un caractère ; c'est un code hexadécimal pour le caractère Ctrl-T.)

L'instruction envoie également (ajoute) les caractères définis dans les propriétés de l'automate. Dans cet exemple, l'instruction AWA envoie un retour chariot (\$0D), qui marque la fin du message.

**Diagramme à relais**



**Texte structuré**

```

IF temp_high THEN
temp_high_write.LEN := 5;
AWA(o,string[1],temp_high_write);
temp_high := 0;
END_IF;

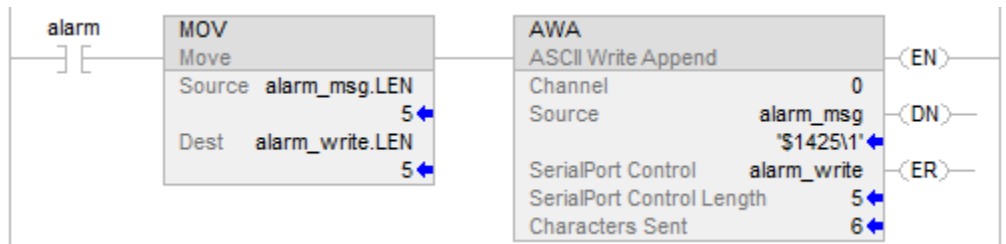
```

**Exemple 2**

Lorsque l'alarme est activée, l'instruction AWA envoie le nombre de caractères indiqué dans alarm\_msg et ajoute un ou plusieurs caractères de terminaison. Étant donné que le nombre de caractères dans alarm\_msg varie, l'échelon déplace d'abord la longueur de la chaîne (alarm\_msg.LEN)

vers Serial Port Control Length de l'instruction AWA (alarm\_write.LEN). Dans alarm\_msg, le \$14 compte comme un caractère ; c'est le code hexadécimal pour le caractère Ctrl-T.

**Diagramme à relais**



**Texte structuré**

```

osri_1.InputBit := alarm;
OSRI(osri_1);

```

```
IF(osri_1.OutputBit) THEN  
  
alarm_write.LEN := alarm_msg.LEN;  
  
AWA(0,alarm_msg,alarm_write);  
  
END_IF;
```

**Voir aussi**

[Instructions pour port série ASCII](#) sur la [page 789](#)

[Test ASCII pour ligne de mémoire tampon \(ABL\)](#) sur la [page 813](#)

[Caractères ASCII dans la mémoire tampon \(ACB\)](#) sur la [page 791](#)

[Effacement ASCII de la mémoire tampon \(ACL\)](#) sur la [page 795](#)

[Lignes d'échange ASCII \(AHL\)](#) sur la [page 798](#)

[Lecture ASCII \(ARD\)](#) sur la [page 803](#)

[Ligne de lecture ASCII \(ARL\)](#) sur la [page 807](#)

[Ecriture ASCII \(AWT\)](#) sur la [page 817](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

## Types de chaînes

Stocker les caractères ASCII dans des étiquettes qui utilisent comme type de données un type de chaînes pour :

- Utiliser le type de données STRING par défaut, qui stocke jusqu'à 82 caractères.
- Créer un nouveau type de chaînes qui stocke un nombre de caractères plus ou moins grand.

Pour créer un nouveau type de chaîne, consultez [LOGIX 5000 Controllers ASCII Strings Programming Manual](#) , [1756-PM013](#) .

Chaque type de chaînes contient les membres suivants :

Nom (Name)	Type de donnée	Description (Description)	Remarques
LEN	DINT	nombre de caractères dans la chaîne	<p>LEN est automatiquement mis à jour avec le nouveau nombre de caractères chaque fois que vous utilisez :</p> <ul style="list-style-type: none"> <li>le navigateur de chaîne pour saisir des caractères</li> <li>des instructions qui lisent, convertissent et manipulent une chaîne</li> </ul> <p>LEN indique la longueur de la chaîne actuelle. Le membre DATA peut contenir des caractères additionnels et anciens qui ne sont pas inclus dans le nombre LEN.</p>
DATA	Tableau SINT	Caractères ASCII de la chaîne	<p>Pour accéder aux caractères de la chaîne, adressez le nom de l'étiquette. Par exemple, pour accéder aux caractères de l'étiquette string_1, saisissez string_1.</p> <p>Chaque élément du tableau DATA contient un caractère.</p> <p>Créer de nouveaux types de chaînes qui stocke un nombre de caractères plus ou moins grand.</p>

**Voir aussi**

[Valeurs littérales des chaînes de caractères](#) sur la [page 930](#)

**Codes d'erreur ASCII**

Si une instruction de port série ASCII ne parvient pas à s'exécuter, le membre ERROR de sa structure SERIAL\_PORT\_CONTROL contient un des codes d'erreur hexadécimaux suivants :

Code hex	Indique :
16#2	Le modem s'est mis hors ligne.
16#3	Le signal CTS a été perdu pendant la communication.
16#4	Le port série était en mode Système.
16#5	Les instructions n'ont pas pu être envoyées ni reçus car la configuration du canal a été interrompue via le menu de configuration du canal.
16#6	Des paramètres incorrects ont été transmis au pilote ASCII.
16#7	Les instructions n'ont pas pu être envoyées ni reçus car la configuration du canal a été interrompue via le menu de configuration du canal.
16#8	Transmission déjà en cours. Cela provoquera une erreur dans l'instruction en cours.
16#9	La communication ASCII demandée n'est pas prise en charge par la configuration de canal actuelle.

16#10	Tentative d'exécution d'une instruction AHL alors que la canal était en mode Système.
16#A	Le bit UL a été défini avant l'exécution de l'instruction. Cela arrête l'exécution de l'instruction.
16#B	Le port sur lequel vous avez demandé à exécuter cette instruction n'existe pas.
16#C	L'automate est passé du mode Exécution au mode Programme. Cela interrompt l'exécution d'une instruction de port série ASCII et efface la file d'attente.
16#D	Sur l'onglet Protocole utilisateur (User Protocol) de la boîte de dialogue Propriétés de l'automate (Controller Properties), les paramètres de taille de tampon ou de mode écho ont été modifiés et appliqués. Cela interrompt l'exécution d'une instruction de port série ASCII et efface la file d'attente.
16#E	L'instruction ACL a exécuté et arrêté ou supprimé ce type d'instruction.
16#F	La configuration du port série est passée du mode Utilisateur au mode Système. Cela interrompt l'exécution d'une instruction de port série ASCII et efface la file d'attente.
16#51	La valeur LEN de l'étiquette de la chaîne peut être négative ou supérieure à la taille DATA de l'étiquette de la chaîne.
16#54	La longueur de Serial Port Control est supérieure à la taille du tampon.
16#55	La longueur de Serial Port Control est supérieure à la taille de Source ou de Destination.

## Instructions de chaîne ASCII

### Instructions de chaîne ASCII

Utiliser les instructions de chaîne ASCII pour modifier ou créer des chaînes de caractères ASCII.

#### Instructions disponibles

#### Un diagramme à relais et texte structuré

<a href="#">FIND</a>	<a href="#">INSERT</a>	<a href="#">MID</a>	<a href="#">CONCAT</a>	<a href="#">DELET</a> E
----------------------	------------------------	---------------------	------------------------	----------------------------

#### Bloc fonctionnel

Indisponible

Si vous voulez :	Utilisez cette instruction :
Ajouter des caractères de terminaison ou des délimiteurs à une chaîne	CONCAT
Supprimer des caractères d'une chaîne (ex. : supprimer des caractères d'en-tête ou de commande d'une chaîne)	DELETE
Déterminer la caractère de début d'une sous-chaîne	FIND
Insérer des caractères dans une chaîne	INSERT
Extraire des caractères d'une chaîne	MID

Vous pouvez également utiliser les instructions suivantes pour comparer ou convertir des caractères ASCII :

Si vous voulez :	Utilisez cette instruction :
Comparer une chaîne à une autre chaîne	CMP
Vérifier si les caractères sont égaux à des caractères spécifiques	EQU
Vérifier si les caractères ne sont pas égaux à des caractères spécifiques	NEQ
Vérifier si les caractères sont égaux ou supérieurs à des caractères spécifiques	GEQ
Vérifier si les caractères sont supérieurs à des caractères spécifiques	GRT
Vérifier si les caractères sont égaux ou inférieurs à des caractères spécifiques	LEQ

Vérifier si les caractères sont inférieurs à des caractères spécifiques	LES
Réorganiser les octets d'une étiquette INT, DINT ou REAL	SWPB
Trouver une chaîne dans un tableau de chaînes	FSC
Convertir des caractères en valeur SINT, INT, DINT ou REAL	STOD
Convertir des caractères en valeur REAL	STOR
Convertir une valeur SINT, INT, DINT ou REAL en chaîne de caractères ASCII	DTOS
Convertir une valeur REAL en chaîne de caractères ASCII	RTOS

**Voir aussi**

[Codes d'erreur ASCII](#) sur la [page 829](#)

[Types de chaînes](#) sur la [page 828](#)

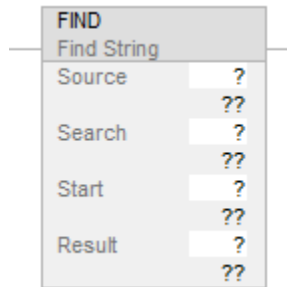
**Rechercher une chaîne (FIND)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction FIND localise la position de départ d'une chaîne précise dans une autre chaîne.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

FIND(Source,Search,Start,Result);



## Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source	ANY_STRING	Étiquette	La chaîne dans laquelle la recherche doit être faite	Il existe différents types de chaînes : Par défaut, le type de données STRING a une longueur de 82 caractères maximum pour la chaîne. Tout nouveau type de chaîne créé avec une longueur de caractères configurable pour la chaîne.
Search	ANY_STRING	Étiquette	La chaîne à rechercher	
Start	SINT INT DINT	Immédiate étiquette	La position dans Source à laquelle la recherche doit être démarrée	Entrez un numéro entre 1 et la taille DATA de Source.
Result	DINT SINT INT	Étiquette	La position dans la source à laquelle la chaîne de recherche a été trouvée	

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description

L'instruction FIND recherche la chaîne de Source pour la chaîne de Search. Si l'instruction trouve la chaîne de Search, Result montre la position de départ de la chaîne de Search dans la chaîne de Source. Sinon, Result affiche zéro.

### Affecte les indicateurs d'état mathématique

Non

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	4	51
La valeur Start est incorrecte, ou la chaîne de Source est vide.	4	56

Aucun spécifique à cette instruction. Reportez-vous à la section *Attributs communs* pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition	Action
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

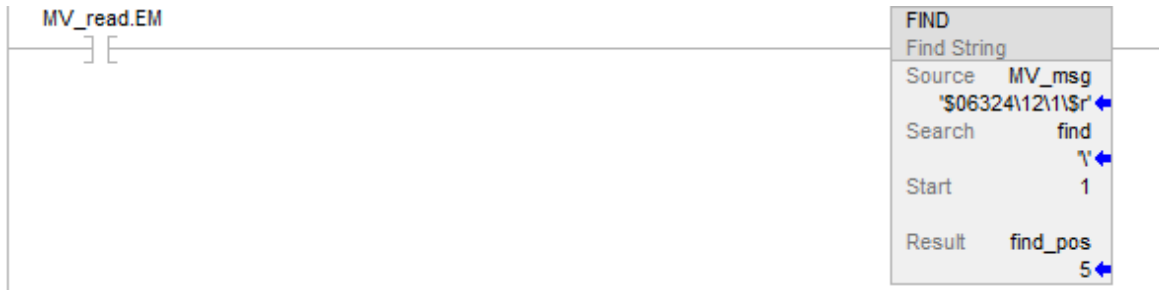
**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais

**Exemple**

Un message provenant du terminal MessageView contient plusieurs informations. La barre oblique arrière (\) sépare chaque information. Pour localiser une information, l'instruction FIND lance la recherche à partir de la barre oblique arrière et enregistre sa position dans find\_pos.

**Diagramme à relais**



**Texte structuré**

```
IF MV_read.EM THEN

    FIND(MV_msg,find,1,find_pos);

    MV_read.EM := 0;

END_IF;
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

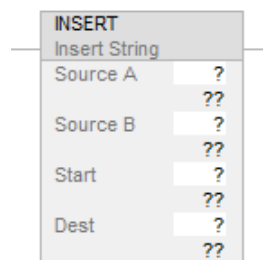
**Insertion d'une chaîne (INSERT)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

Utilisez l'instruction INSERT pour ajouter des caractères ASCII à un emplacement précis dans une chaîne.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

**Texte structuré**

INSERT (SourceA,SourceB,Start,Dest);

**Opérandes**

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données. L'instruction INSERT utilise les opérandes suivants.

**Un diagramme à relais et texte structuré**

Opérande	Type	Format	Description	Remarques
Source A	Type de chaînes	Étiquette	Chaîne à laquelle les caractères sont ajoutés	Les types de chaîne sont, par défaut, des types de données STRING ou tout nouveau type de chaîne créé par vos soins
Source B	Type de chaînes	Étiquette	Chaîne contenant les caractères à ajouter	
Start	SINT DINT	Immédiate étiquette	Position dans la Source A à laquelle les caractères sont ajoutés	Entrez un numéro entre 1 et la taille DATA de Source.
Destination	Type de chaînes	Étiquette	Chaîne dans laquelle le résultat est enregistré	

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

**Description**

L'instruction INSERT ajoute les caractères dans Source B à la position désignée dans Source A et place le résultat dans Destination.

- Start définit l'endroit auquel Source B est ajoutée dans Source A.
- A moins que Source et Destination ne soient la même étiquette, Source A reste inchangée.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	1. Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. 2. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	56	La valeur Start ou Quantity est incorrecte.	Vérifiez que la valeur Start se situe entre 1 et la taille DATA de la Source.

**Exécution****Diagramme à relais**

Condition	Action de diagramme à relais
Pré-scrutation	La condition de sortie d'échelon est définie sur faux.
Condition d'entrée d'échelon est fausse	La condition de sortie d'échelon est définie sur faux.
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. La sortie d'échelon est vraie.
Post-scrutation	La condition de sortie d'échelon est définie sur faux.

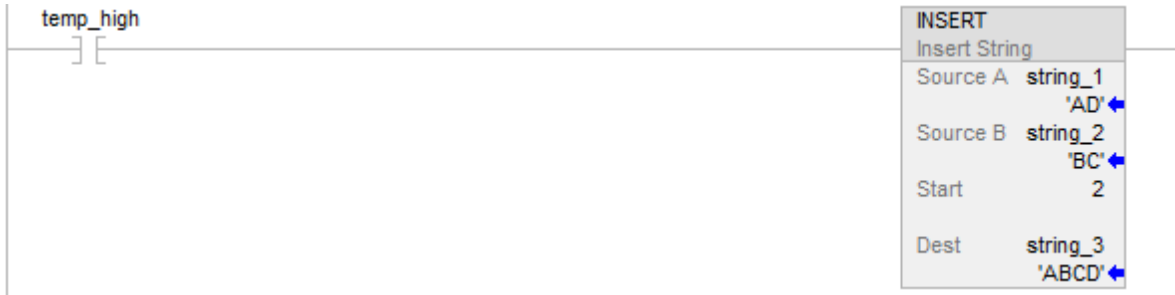
**Exécution****Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais

**Exemple**

Lorsque *temp\_high* est défini, l'instruction INSERT ajoute les caractères dans *string\_2* à la position 2 dans *string\_1* et place le résultat dans *string\_3*.

### Diagramme à relais



### Texte structuré

```
IF temp_high THEN
    INSERT(string_1,string_2,2,string_3);
    temp_high := 0;
END_IF;
```

### Voir aussi

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

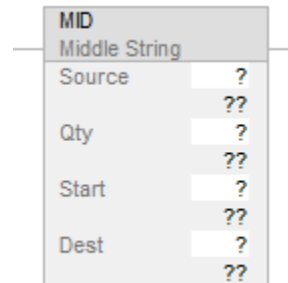
## Chaîne médiane (MID)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction MID copie un nombre précis de caractères ASCII d'une chaîne et les enregistre dans une autre chaîne.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

MID(Source,Qty,Start,Dest);

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source	ANY_STRING	Étiquette	La chaîne à partir de laquelle les caractères doivent être copiés	Il existe différents types de chaînes : Par défaut, le type de données STRING a une longueur de 82 caractères maximum pour la chaîne. Tout nouveau type de chaîne créé avec une longueur de caractères configurable pour la chaîne.
Quantity	SINT INT DINT	Immédiate étiquette	Le nombre de caractères à copier	La somme de Start et Quantity doit être inférieure ou égal à la taille de la longueur de Source plus 1.
Start	SINT INT DINT	Immédiate étiquette	La position du premier caractère à copier	Entrez un numéro entre 1 et la taille DATA de Source.

Destination	ANY_STRING	Étiquette	La chaîne dans laquelle les caractères doivent être copiés	
-------------	------------	-----------	--	--

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Syntaxe du texte structuré.

**Description**

L'instruction MID copie un groupe de caractères de Source et place le résultat dans Destination.

- La position Start et Quantity définissent les caractères à copier.
- A moins que Source et Destination ne soient la même étiquette, Source reste inchangée.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La valeur LEN de l'étiquette de la chaîne de Source est supérieure à la taille DATA de l'étiquette de la chaîne de Source.	4	51
La longueur de l'étiquette de la chaîne de sortie est supérieure à la taille DATA de l'étiquette de la chaîne de destination.	4	52
La valeur Start ou Quantity est incorrecte.	4	56

**Exécution**

**Diagramme à relais**

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

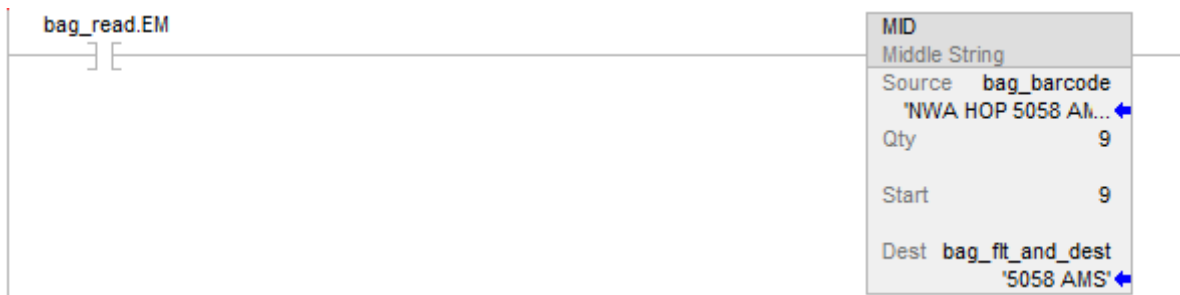


**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais

**Exemple**

Sur le tapis roulant à bagages d'un aéroport, un code barre est attribué à chaque valise. Les caractères 9 à 17 du code barre représentent le numéro de vol et l'aéroport de destination de la valise. Une fois le code barre lu (bag\_read.EM défini), l'instruction MID copie le numéro de vol et l'aéroport de destination dans la chaîne bag\_fit\_and\_dest. Les échelons suivants utilisent bag\_fit\_and\_dest pour déterminer où la valise doit être acheminée.

**Diagramme à relais****Texte structuré**

```
IF bag_read.EM THEN
    MID(bag_barcode,9,9,bag_fit_and_dest);
    bag_read.EM := 0;
END_IF;
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

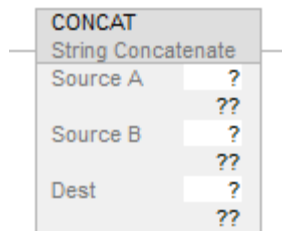
## Concaténation de chaîne (CONCAT)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction CONCAT ajoute des caractères ASCII à la fin d'une chaîne.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

CONCAT(SourceA,SourceB,Dest);

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Pour plus d'informations sur la Conversion de données, reportez-vous à la section Attributs communs.

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source A	ANY_STRING	étiquette	Étiquette contenant les caractères initiaux	Il existe différents types de chaînes : <ul style="list-style-type: none"> <li>Par défaut, le type de données STRING a une longueur de 82 caractères maximum pour la chaîne.</li> <li>Tout nouveau type de chaîne créé avec une longueur de caractères configurable pour la chaîne.</li> </ul>
Source B	ANY_STRING	étiquette	Étiquette contenant les caractères finaux	
Destination	ANY_STRING	étiquette	Étiquette où enregistrer le résultat	

Pour plus d'informations sur la syntaxe des expressions dans le texte structuré, reportez-vous à la section Attribut du texte structuré.

### Description

L'instruction CONCAT combine les caractères dans Source A aux caractères dans Source B et place le résultat dans Destination.

Les caractères de Source A apparaissent en premier, suivis des caractères de Source B.

A moins que Source et Destination ne soient la même étiquette, Source A reste inchangée.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	4	51
La somme de la longueur de Source A et de Source B est supérieure à la taille DATA de l'étiquette de la chaîne.	4	51

Reportez-vous à Indexer via des tableaux pour consulter les défauts d'indexation par tableau.

### Exécution

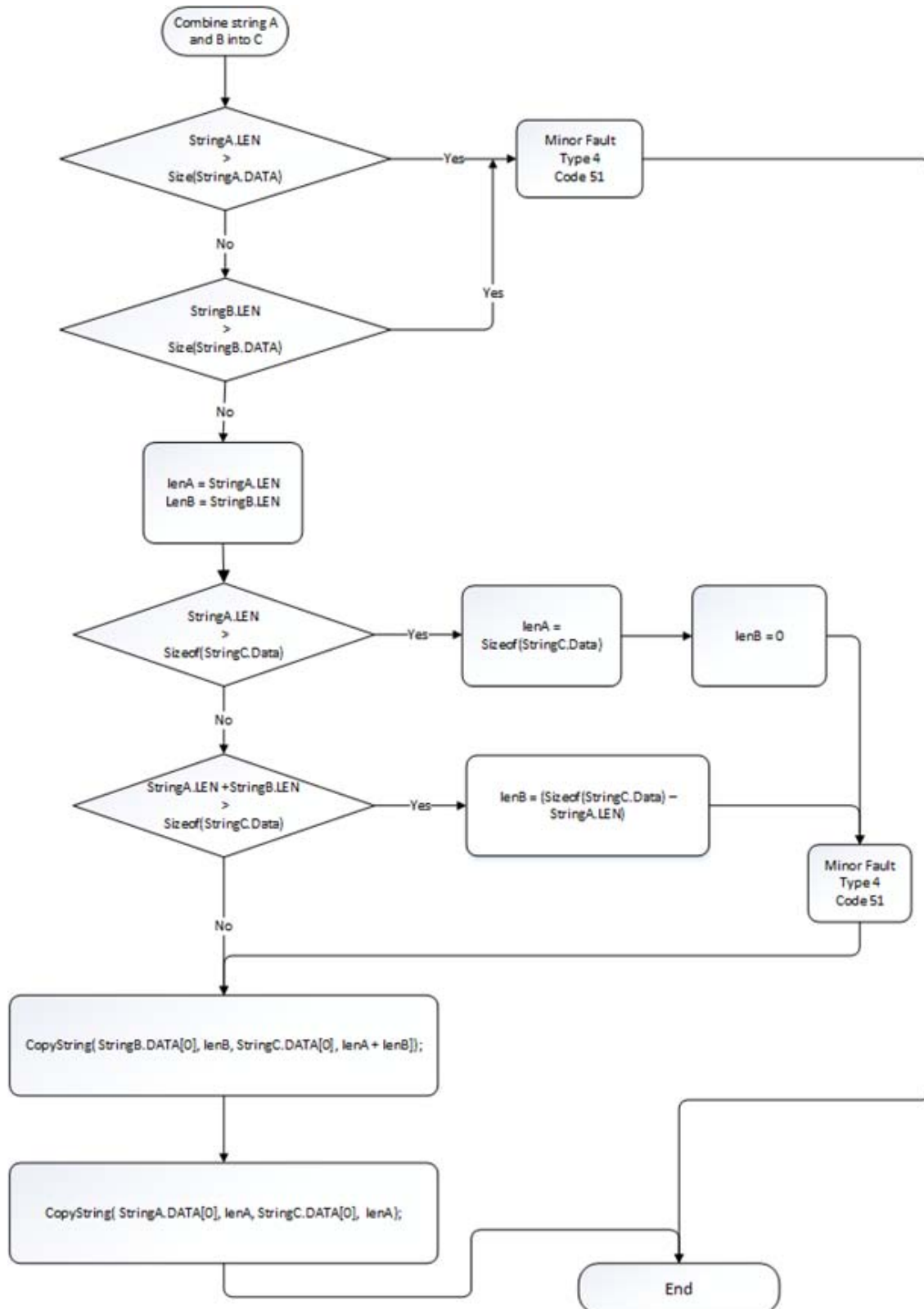
#### Diagramme à relais

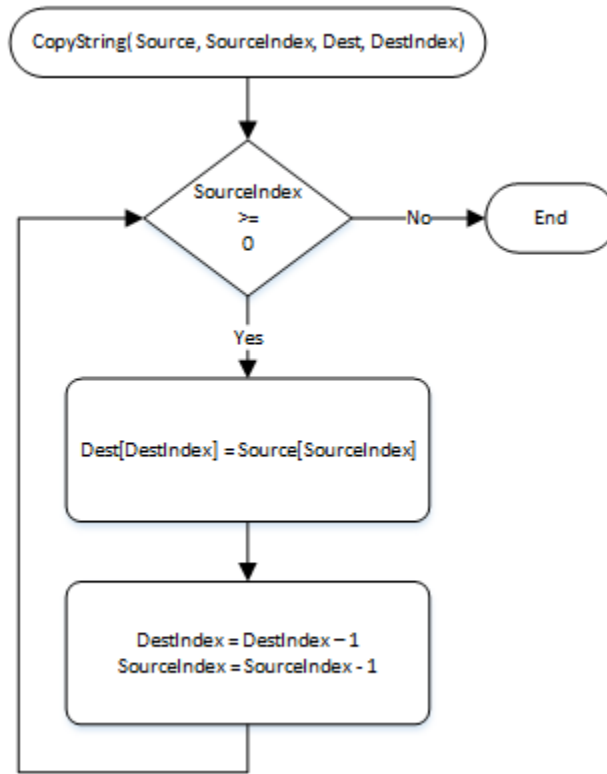
Condition	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

<b>Condition</b>	<b>Action entreprise</b>
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

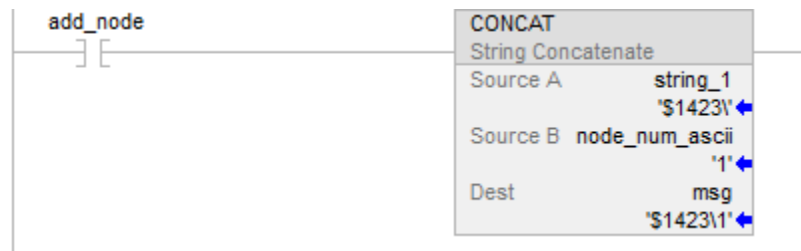
Organigramme Concat String





**Exemple**

**Diagramme à relais**



**Texte structuré**

CONCAT(string\_1,string\_2,msg);

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Attributs de texte structuré](#) sur la [page 946](#)

[Conversions de données](#) sur la [page 889](#)

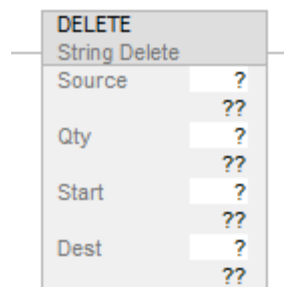
## Effacement de chaîne (DELETE)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction DELETE supprime des caractères ASCII d'une chaîne.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

DELETE(Source,Qty,Start,Dest);

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à Conversion de données.

### Un diagramme à relais et texte structuré

Opérande	Type (Type)	Format	Description (Description)	Remarques
Source (Source)	ANY_STRING	étiquette	L'étiquette contenant la chaîne dont vous souhaitez supprimer des caractères	Il existe différents types de chaînes : Par défaut, le type de données STRING a une longueur de 82 caractères maximum pour la chaîne. Tout nouveau type de chaîne créé avec une longueur de caractères configurable pour la chaîne.

Quantity	SINT INT DINT	immédiate étiquette	Le nombre de caractères à supprimer	La somme de Start et Quantity doit être inférieure ou égal à la longueur de Source plus 1.
Début (Start)	SINT INT DINT	immédiate étiquette	La position du premier caractère à supprimer	Entrez un numéro entre 1 et la taille DATA de Source.
Destination	Type de chaînes	étiquette	L'étiquette dans lequel le résultat est enregistré	

Reportez-vous à Syntaxe du texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Description (Description)**

L'instruction DELETE supprime (retire) un ou plusieurs caractères de Source et place les caractères restants dans Destination.

- La position Start et Quantity définissent les caractères à supprimer.
- A moins que Source et Destination ne soient la même étiquette, Source A reste inchangée.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Un défaut mineur surviendra si :	Type de défaut	Code de défaut
La valeur LEN de l'étiquette de la chaîne de Source est supérieure à la taille DATA de l'étiquette de la chaîne de Source.	4	51
La longueur de l'étiquette de la chaîne de sortie est supérieure à la taille DATA de l'étiquette de la chaîne de destination.	4	52
La valeur Démarrage ou Quantité est non valide.	4	56

Reportez-vous à Attributs communs pour les défauts liés aux opérandes.



**Exécution**

**Diagramme à relais**

Condition/État	Action (Action)
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

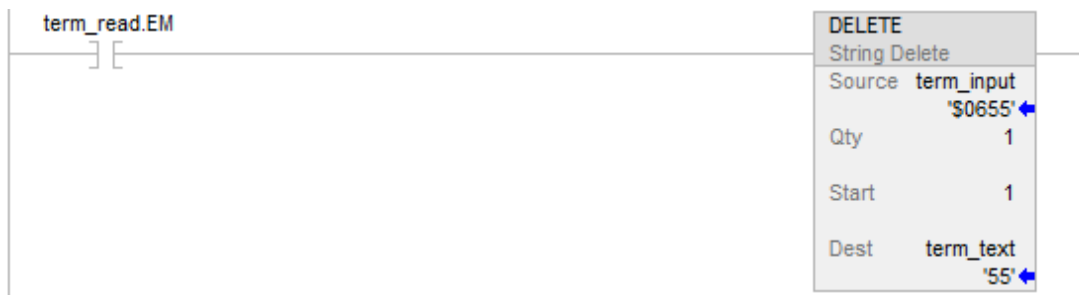
**Texte structuré**

Condition/État	Action (Action)
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais.
Exécution normale	Reportez-vous à condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais.

**Exemples**

Informations ASCII provenant d'un terminal et contenant un caractère d'en-tête. Une fois les données lues par l'automate (term\_read.EM s'affiche), l'instruction DELETE supprime le caractère d'en-tête. L'automate peut alors utiliser le texte du message ou le transmettre à un autre appareil.

**Diagramme à relais**



**Texte structuré**

```
IF term_read.EM THEN

    DELETE(term_input,1,1,term_text);

    term_read.EM := 0;
```

END\_IF;

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

## Instructions de conversion ASCII

### Instructions de conversion ASCII

Utiliser les instructions de conversion ASCII pour convertir des données en direction ou en provenance de chaînes de caractères ASCII.

#### Instructions disponibles

#### Un diagramme à relais et texte structuré

<a href="#">STOD</a>	<a href="#">STOR</a>	<a href="#">RTO</a>	<a href="#">DTOS</a>	<a href="#">LOWER</a>	<a href="#">UPPER</a>
		<a href="#">S</a>			<a href="#">R</a>

#### Bloc fonctionnel

Indisponible

Si vous voulez convertir :	Utilisez cette instruction :
des représentations ASCII de valeurs entières en valeurs SINT, INT, DINT ou REAL (par exemple, en faisant une conversion d'une échelle de poids ou d'un autre dispositif ASCII vers un entier de manière à pouvoir l'utiliser dans votre logique).	STOD
des représentations ASCII d'une valeur à virgule flottante en valeur REAL (par exemple, en faisant une conversion d'une échelle de poids ou d'un autre dispositif ASCII vers une valeur REAL de manière à pouvoir l'utiliser dans votre logique).	STOR
des valeurs SINT, INT, DINT ou REAL en chaîne de caractères ASCII (par exemple en convertissant une variable en chaîne ASCII de manière à pouvoir l'envoyer vers un terminal MessageView™).	DTOS
des valeurs REAL en chaîne de caractères ASCII (par exemple en convertissant une variable en chaîne ASCII de manière à pouvoir l'envoyer vers un terminal MessageView).	RTOS
les lettres d'une chaîne de caractères ASCII en majuscules (par exemple, en convertissant une entrée effectuée par un opérateur en majuscules de manière à pouvoir la chercher dans un tableau).	UPPER

les lettres d'une chaîne de caractères ASCII en minuscules (par exemple, en convertissant une entrée effectuée par un opérateur en minuscules de manière à pouvoir la chercher dans un tableau).	LOWER
--	-------

Vous pouvez également utiliser les instructions suivantes pour comparer ou manipuler des caractères ASCII.

Si vous voulez :	Utilisez cette instruction :
Ajouter des caractères à la fin d'une chaîne	CONCAT
Supprimer des caractères d'une chaîne	DELETE
Déterminer la caractère de début d'une sous-chaîne	FIND
Insérer des caractères dans une chaîne	INSERT
Extraire des caractères d'une chaîne	MID
Réorganiser les octets d'une étiquette INT, DINT ou REAL	SWPB
Comparer une chaîne à une autre chaîne	CMP
Vérifier si les caractères sont égaux à des caractères spécifiques	EQU
Vérifier si les caractères ne sont pas égaux à des caractères spécifiques	NEQ
Vérifier si les caractères sont égaux ou supérieurs à des caractères spécifiques	GEQ
Vérifier si les caractères sont supérieurs à des caractères spécifiques	GRT
Vérifier si les caractères sont égaux ou inférieurs à des caractères spécifiques	LEQ
Vérifier si les caractères sont inférieurs à des caractères spécifiques	LES
Trouver une chaîne dans un tableau de chaînes	FSC

**Voir aussi**

[Codes d'erreur ASCII](#) sur la [page 829](#)

[Types de chaînes](#) sur la [page 828](#)

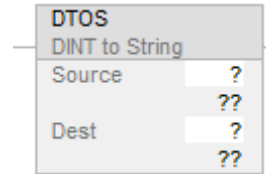
**DINT vers chaîne (DTOS)**

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction DTOS produit la représentation ASCII d'une valeur.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

DTOS(Source, Dest);

### Opérandes

#### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source	SINT INT DINT REAL	Étiquette	L'étiquette contenant la valeur	Si Source a une valeur REAL, l'instruction la convertit en valeur DINT.
Destination	Type de chaînes	Étiquette	L'étiquette dans laquelle la valeur entière est enregistrée	Il existe différents types de chaînes : <ul style="list-style-type: none"> <li>• type de données STRING par défaut</li> <li>• tout nouveau type de chaîne que vous créez</li> </ul>

### Description

L'instruction DTOS convertit Source en chaîne de caractères ASCII et place le résultat dans Destination.

### Affecte les indicateurs d'état mathématique

Non

**Défauts majeurs/mineurs**

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	52	La chaîne de sortie est plus grande que la destination.	Créer un nouveau type de chaîne suffisamment grand pour la chaîne de sortie. Utiliser le nouveau type de chaîne comme type de données pour la destination.

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

**Exemple**

Lorsque temp\_high est défini, l'instruction DTOS convertit la valeur dans msg\_num en une chaîne de caractères ASCII et place le résultat dans msg\_num\_ascii. Les échelons suivants insèrent ou concatènent msg\_num\_ascii avec d'autres chaînes pour produire un message complet pour un terminal d'affichage.

**Diagramme à relais**



**Texte structuré**

```
IF temp_high THEN
    DTOS(msg_num,msg_num_ascii);
    temp_high := 0;
END_IF;
```

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

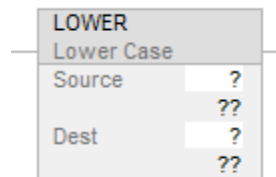
**Minuscule (LOWER)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction LOWER convertit les caractères alphabétiques en chaîne de caractères en minuscules.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

LOWER(Source, Dest);

**Opérandes**

**Un diagramme à relais et texte structuré**

Opérande	Type	Format	Description
Source	String	Étiquette	Étiquette contenant les caractères que vous souhaitez convertir en minuscules
Destination	String	Étiquette	Étiquette dans laquelle les caractères en minuscules sont enregistrés

Reportez-vous à *Texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

**Description**

L'instruction LOWER convertit toutes les lettres de Source en minuscules et place le résultat dans Destination.

- Les caractères ASCII sont sensibles à la casse. La majuscule A (\$41) est différente de la minuscule a (\$61).
- Si les opérateurs saisissent directement des caractères ASCII, convertir tous les caractères en majuscules ou en minuscules avant de les comparer.

Les caractères de la chaîne Source qui ne sont pas des lettres restent inchangés.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	52	La chaîne de sortie est plus grande que la destination	Créer un nouveau type de chaîne suffisamment grand pour la chaîne de sortie. Utiliser le nouveau type de chaîne comme type de données pour la destination.



**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

**Exemples**

Pour trouver des informations à propos d'un article particulier, un opérateur saisit le code article dans un terminal ASCII. Une fois que l'automate a lu l'entrée d'un terminal (terminal\_read est défini), l'instruction LOWER convertit les caractères présents dans item\_number en majuscules et enregistre le résultat dans item\_number\_lower\_case. Un échelon suivant effectue ensuite, dans un tableau, une recherche de caractères correspondants à ceux présents dans item\_number\_lower\_case.

**Diagramme à relais**



**Texte structuré**

```
IF terminal_read THEN

    LOWER(item_number,item_number_lower_case);

    terminal_read := 0;
```

END\_IF;

**Voir aussi**

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

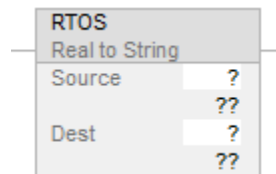
**REAL vers chaîne (RTOS)**

Ces informations s’appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L’instruction RTOS produit la représentation ASCII d’une valeur REAL.

**Langages disponibles**

**Diagramme à relais**



**Bloc fonctionnel**

Cette instruction n’est pas disponible dans le bloc fonctionnel.

**Texte structuré**

RTOS(Source, Dest);

**Opérandes**

**Un diagramme à relais et texte structuré**

Opérande	Type	Format	Description	Remarques
Source	REAL	Étiquette	L’étiquette contenant la valeur REAL	
Destination	Type de chaînes	Étiquette	L’étiquette dans laquelle la valeur ASCII est enregistrée	Il existe différents types de chaînes : <ul style="list-style-type: none"> <li>• Type de données STRING par défaut</li> <li>• Tout nouveau type de chaîne que vous créez</li> </ul>

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions.

### Description

L'instruction RTOS convertit Source en chaîne de caractères ASCII et place le résultat dans Destination.

### Affecte les indicateurs d'état mathématique

Non

### Défauts majeurs/mineurs

Type	Code	Cause	Méthode de récupération
4	52	La chaîne de sortie est plus grande que la destination	Créer un nouveau type de chaîne suffisamment grand pour la chaîne de sortie. Utiliser le nouveau type de chaîne comme type de données pour la destination.

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

#### Texte structuré

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

### Exemples

Lorsque send\_data est défini, l'instruction RTOS convertit la valeur dans data\_1 en une chaîne de caractères ASCII et place le résultat dans data\_1\_ascii. Les échelons suivants insèrent ou concatènent data\_1\_ascii avec d'autres chaînes pour produire un message complet pour un terminal d'affichage.

### Diagramme à relais



### Texte structuré

```
IF send_data THEN
    RTOS(data_1,data_1_ascii);
    send_data:= 0;
END_IF;
```

### Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

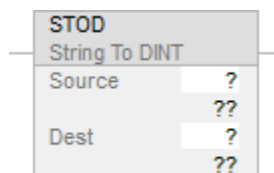
## Chaîne vers DINT (STOD)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction STOD convertit la représentation ASCII d'un entier en entier ou en valeur REAL.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

STOD(Source, Dest);

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données dans une instruction. Reportez-vous à la section *Conversion de données*.

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source	Type de chaînes	Étiquette	L'étiquette contenant la valeur en ASCII	Il existe différents types de chaînes : <ul style="list-style-type: none"> <li>• Type de données STRING par défaut</li> <li>• Tout nouveau type de chaîne que vous créez</li> </ul>
Destination	SINT INT DINT	Étiquette	L'étiquette dans laquelle la valeur entière est enregistrée	Si la valeur Source est un nombre à virgule flottante, l'instruction ne convertit que la partie non fractionnelle du nombre (quel que soit le type de données de destination).

Reportez-vous à *Syntaxe du texte structuré* pour plus d'informations concernant la syntaxe des expressions.

### Description

L'instruction STOD convertit Source en entier et place le résultat dans Destination.

- L'instruction convertit des nombres positifs et négatifs.
- Si la chaîne Source contient des caractères non-numériques, l'instruction STOD convertit le premier ensemble de nombres contigus :

L'instruction ignore tous les caractères de commande initiaux ou non numériques, sauf le signe moins précédant un nombre.

Si la chaîne contient plusieurs groupes de nombres séparés par des délimiteurs (ex. : /), l'instruction ne convertit que le premier groupe de nombres.

**Affecte les indicateurs d'état mathématique**

En diagrammes à relais uniquement. Reportez-vous à la section *Indicateurs d'état mathématique*.

**Défauts majeurs/mineurs**

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	53	Le numéro de sortie se situe au-delà des limites du type de données de destination.	<ul style="list-style-type: none"> <li>• Réduire la taille de la valeur ASCII ou</li> <li>• Utiliser un type de données de destination plus grand</li> </ul>

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

**Exécution**

**Diagramme à relais**

Condition	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute. Destination est effacée L'instruction convertit Source.
Post-scrutation	N/A

**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

### Exemple

Lorsque MV\_read.EM est défini, l'instruction STOD convertit le premier ensemble de caractères numériques présent dans MV\_msg en valeur entière. L'instruction ignore le caractère de commande initiale (\$06) et s'arrête au délimiteur (\).

### Diagramme à relais



### Texte structuré

```
IF MV_read.EM THEN
    STOD(MV_msg,MV_msg_nbr);
    MV_read.EM := 0;
END_IF;
```

### Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Conversions de données](#) sur la [page 889](#)

[Indicateurs d'état mathématique](#) sur la [page 885](#)

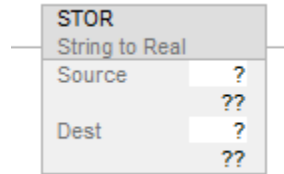
## Chaîne vers REAL (STOR)

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction STOR convertit la représentation ASCII d'une valeur à virgule flottante en valeur REAL.

## Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

STOR(Source, Dest);

### Opérandes

Il existe des règles de conversion de données pour combiner des types de données dans une instruction. Reportez-vous à la section *Conversion de données*.

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description	Remarques
Source	Type de chaînes	étiquette	L'étiquette contenant la valeur en ASCII	Il existe différents types de chaînes : <ul style="list-style-type: none"> <li>• Type de données STRING par défaut</li> <li>• Tout nouveau type de chaîne que vous créez</li> </ul>
Destination	REAL	étiquette	L'étiquette dans laquelle la valeur REAL est enregistrée	

Reportez-vous à Texte structuré pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description

L'instruction STOR convertit Source en valeur REAL et place le résultat dans Destination.

- L'instruction convertit des nombres positifs et négatifs.



- Si la chaîne Source contient des caractères non-numériques, l'instruction STOR convertit le premier ensemble de nombres contigus, y compris la virgule [,].

L'instruction ignore tous les caractères de commande initiaux ou non numériques (sauf le signe moins précédant un nombre).

Si la chaîne contient plusieurs groupes de nombres séparés par des délimiteurs (ex. : /), l'instruction ne convertit que le premier groupe de nombres.

### Affecte les indicateurs d'état mathématique

Conditionnel, en fonction de la langage de programmation. Reportez-vous à la section *Indicateurs d'état mathématique*.

### Défauts majeurs/mineurs

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	53	Le numéro de sortie se situe au-delà des limites du type de données de destination.	<ul style="list-style-type: none"> <li>• Réduire la taille de la valeur ASCII ou</li> <li>• Utiliser un type de données de destination plus grand</li> </ul>

Reportez-vous à la section *Attributs communs* pour les défauts liés aux opérandes.

### Exécution

#### Diagramme à relais

Condition	Action de diagramme à relais
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

**Exemple**

Après avoir lu le poids sur une balance (`weight_read` est défini), l'instruction `STOR` convertit les caractères numériques présents dans `weight_ascii` en valeur `REAL`.

Il est possible de constater une légère différence entre les parties fractionnelles de Source et de Destination.

**Diagramme à relais**



**Texte structuré**

```
IF weight_read THEN
    STOR(weight_ascii,weight);
END_IF;
```

**Voir aussi**

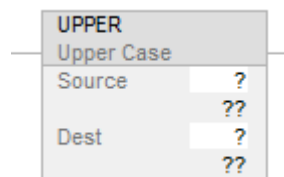
- [Attributs communs](#) sur la [page 885](#)
- [Syntaxe du texte structuré](#) sur la [page 917](#)
- [Conversions de données](#) sur la [page 889](#)
- [Indicateurs d'état mathématique](#) sur la [page 885](#)

**Haut de casse (UPPER)** Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction UPPER convertit les caractères alphabétiques en chaîne de caractères en majuscules.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

```
UPPER(Source, Dest);
```

### Opérandes

### Un diagramme à relais et texte structuré

Opérande	Type	Format	Description
Source	String	étiquette	Étiquette contenant les caractères que vous souhaitez convertir en majuscules
Destination	String	étiquette	Étiquette dans laquelle les caractères en majuscules sont enregistrés

Reportez-vous à *Texte structuré* pour plus d'informations concernant la syntaxe des expressions dans le texte structuré.

### Description

L'instruction UPPER convertit toutes les lettres de Source en majuscules et place le résultat dans Destination.

- Les caractères ASCII sont sensibles à la casse. La majuscule A (\$41) est différente de la minuscule a (\$61).

- Si les opérateurs saisissent directement des caractères ASCII, convertir tous les caractères en majuscules ou en minuscules avant de les comparer.

Les caractères de la chaîne Source qui ne sont pas des lettres restent inchangés.

**Affecte les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Type	Code	Cause	Méthode de récupération
4	51	La valeur LEN de l'étiquette de la chaîne est supérieure à la taille DATA de l'étiquette de la chaîne.	Vérifier qu'aucune instruction n'est écrite dans le membre LEN d'étiquette du type de chaîne. Dans la valeur LEN, entrez le nombre de caractères que la chaîne contient.
4	52	La chaîne de sortie est plus grande que la destination	Créer un nouveau type de chaîne suffisamment grand pour la chaîne de sortie. Utiliser le nouveau type de chaîne comme type de données pour la destination.

**Exécution**

**Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	L'instruction s'exécute.
Post-scrutation	N/A

**Texte structuré**

Condition	Action
Pré-scrutation	Reportez-vous à Pré-scrutation dans le tableau Diagramme à relais précédent
Exécution normale	Reportez-vous à Condition d'entrée d'échelon est vraie dans le tableau Diagramme à relais précédent.
Post-scrutation	Reportez-vous à Post-scrutation dans le tableau Diagramme à relais précédent

**Exemple**

Pour trouver des informations à propos d'un article particulier, un opérateur saisit le code article catalogue dans un terminal ASCII. Une fois que l'automate a lu

l'entrée d'un terminal (terminal\_read est défini), l'instruction UPPER convertit les caractères présents dans catalog\_number en majuscules et enregistre le résultat dans catalog\_number\_upper\_case. Un échelon suivant effectue ensuite, dans un tableau, une recherche de caractères correspondants à ceux présents dans catalog\_number\_upper\_case.

### Diagramme à relais



### Texte structuré

```
IF terminal_read THEN

    UPPER(catalog_number,catalog_number_upper_case);

    terminal_read := 0;

END_IF;
```

### Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Syntaxe du texte structuré](#) sur la [page 917](#)



# Instructions de débogage

## Instructions de débogage

Ces instructions sont compatibles uniquement avec le logiciel Studio 5000 Logix Emulate, qui permet l'émulation d'un automate LOGIX 5000 sur un ordinateur.

Utilisez les instructions de débogage pour surveiller l'état de la logique lorsqu'elle est dans les conditions que vous avez déterminées.

### Instructions disponibles

<a href="#">BPT</a>	<a href="#">TPT</a>
---------------------	---------------------

### Bloc fonctionnel

Indisponible

### Texte structuré

Indisponible

Si vous voulez :	Utilisez cette instruction :
Arrêter le programme lorsqu'un échelon est vrai	BPT
Enregistrer les données sélectionnées lorsqu'un échelon est vrai	TPT

### Voir aussi

[Instructions de calcul/math](#) sur la [page 373](#)

[Instructions de comparaison](#) sur la [page 297](#)

[Instructions de bit](#) sur la [page 73](#)

[Instructions de chaîne ASCII](#) sur la [page 831](#)

[Instructions de conversion ASCII](#) sur la [page 851](#)

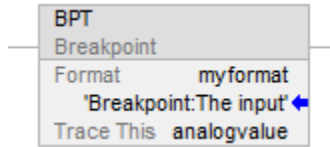
## Points d'interruption (BPT)

Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

Utilisez les instructions de débogage pour surveiller l'état de la logique lorsqu'elle est dans les conditions que vous avez déterminées.

### Langages disponibles

#### Diagramme à relais



#### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

#### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

#### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

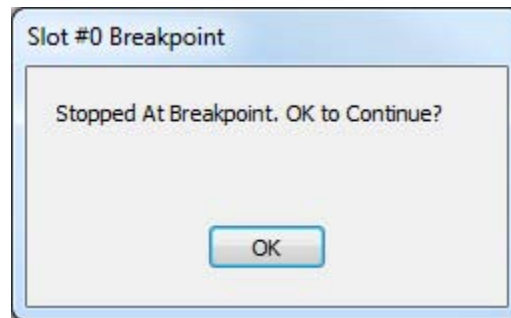
#### Diagramme à relais

Opérande	Type	Format	Description
Format	String	Étiquette	Une chaîne définissant le formatage du texte apparaissant dans la fenêtre de suivi du point d'arrêt.
Trace This	BOOL, SINT, INT, DINT, REAL	Étiquette	L'étiquette possédant une valeur que vous souhaitez afficher dans le fenêtre de suivi.



## Description

Les points d'arrêt sont programmés avec l'instruction de sortie de points d'arrêt (BPT). Lorsque les entrées d'un échelon contenant une instruction BPT sont vraies, l'instruction BPT interrompt l'exécution du programme. Le logiciel affiche une fenêtre indiquant que le point d'arrêt s'est déclenché, ainsi que les valeurs qui l'ont déclenché.



Lorsqu'un point d'arrêt se déclenche, l'émulateur affiche une fenêtre vous informant qu'un point d'arrêt s'est produit. La barre de titres de la fenêtre montre le logement contenant l'émulateur ayant rencontré le point d'arrêt.

Lorsque vous cliquez sur OK, l'émulateur reprend l'exécution du programme. Si les conditions qui ont déclenché le point d'arrêt persiste, le point d'arrêt se reproduira.

En outre, l'émulateur ouvre une fenêtre de suivi pour le point d'arrêt. La fenêtre de suivi affiche des informations à propos du point d'arrêt et de ses valeurs.

---

**Important :** Lorsqu'un point d'arrêt se déclenche, vous ne pouvez pas modifier votre projet tant que vous n'autorisez pas la poursuite de l'exécution. Vous pouvez aller en ligne avec l'émulateur pour observer l'état de votre projet, mais vous ne pouvez pas le modifier. Si vous essayez d'accepter une modification d'échelon alors qu'un point d'arrêt est déclenché, une boîte de dialogue indiquant que l'automate n'est pas dans le bon mode s'affiche.

---

## Format de chaîne

La chaîne Format présente dans les instructions de point de suivi et d'arrêt vous permet de contrôler la façon dont les étiquettes suivies apparaissent dans les fenêtres de suivis ou fenêtres de point d'arrêt. Le format de la chaîne est le suivant :

- heading:(text)%(type)

dans lequel heading est une chaîne de texte identifiant le point de suivi ou d'arrêt, text est une chaîne décrivant l'étiquette (ou tout autre texte de votre choix) et %(type) indique le format de l'étiquette. Un indicateur type est nécessaire pour chaque étiquette que vous suivrez avec l'instruction de point de suivi ou d'arrêt.

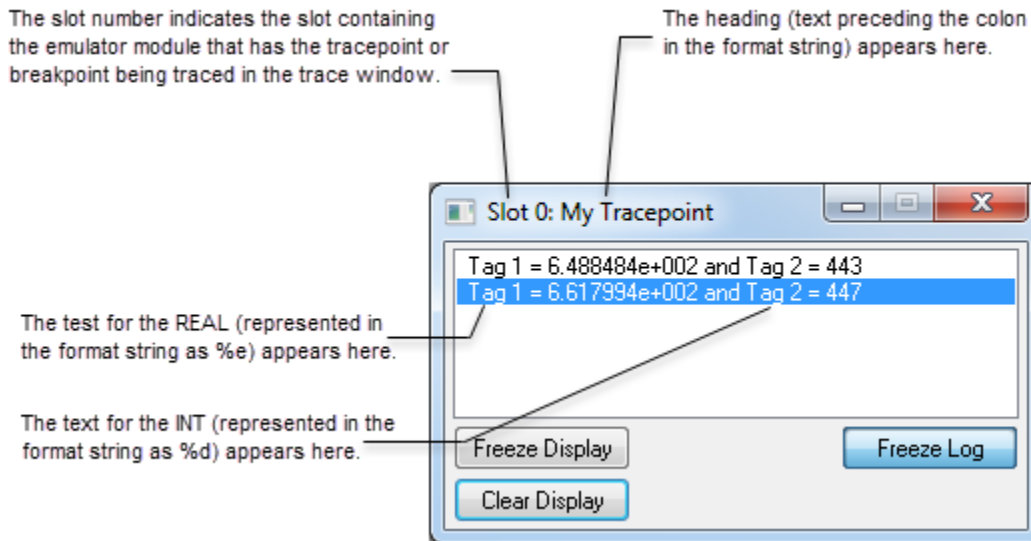
Par exemple, vous pouvez formater une chaîne de point de suivi comme indiqué ci-dessous.

- My tracepoint:Tag 1 = %e and Tag 2 = %d

Le %e formate la première étiquette suivie comme un flottant à double précision avec un exposant, et %d formate la deuxième étiquette suivie comme un entier décimal signé.

Dans ce cas, vous obtenez une instruction de point de suivi comportant deux opérandes Trace This (un pour une valeur REAL et un pour INT, bien que la valeur d'une étiquette puisse être formatée avec n'importe quel indicateur).

La fenêtre de point de suivi obtenue qui apparaît lorsque le point de suivi est déclenché à l'apparence de l'exemple ci-dessous.



### Affecte les indicateurs d'état mathématique

Non

### Conditions de défaut

Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

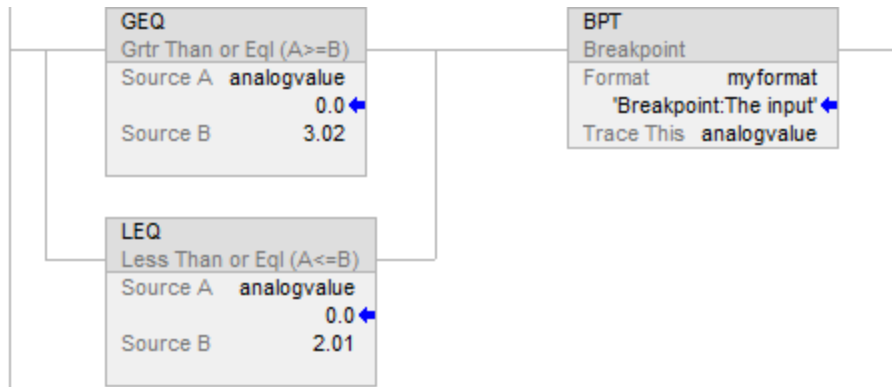
**Exécution**

Condition	Action entreprise
Pré-scrutation	L'échelon devient faux.
Condition d'entrée d'échelon est fausse	L'échelon devient faux.
Condition d'entrée d'échelon est vraie	L'échelon devient vrai. L'exécution saute à l'échelon qui contient l'instruction LBL avec le nom de label référencé.
Post-scrutation	L'échelon devient faux.

**Exemples**

Vous pouvez afficher de nombreuses valeurs d'étiquette à l'aide de l'instruction BPT. Cependant, la chaîne de formatage ne peut contenir que 82 caractères. Étant donné que la chaîne de formatage requière deux caractères pour chaque étiquette que vous souhaitez avoir dans le point d'arrêt, vous ne pouvez pas suivre plus de 41 étiquettes avec une seule instruction BPT. Cependant, pour séparer les données d'étiquette dans vos suivis, vous devez inclure des espaces et d'autres marques de formatage, afin de réduire le nombre de valeurs d'étiquette qu'une instruction BPT peut réellement afficher à une valeur bien inférieure à 41.

Cet échelon montre un point d'arrêt qui arrête l'exécution du programme lorsqu'une valeur analogique est supérieure à 3,02 ou inférieure à 2,01.

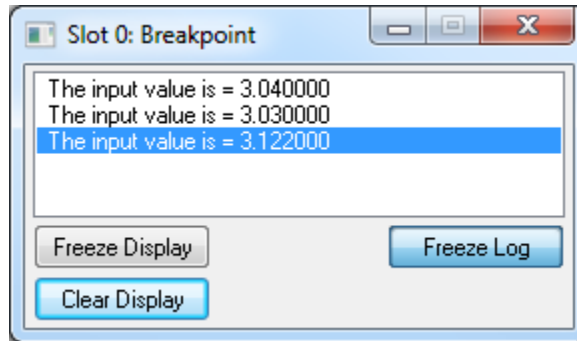


Afficher les informations de point d'arrêt dans la chaîne Format (myformat). Dans ce cas, la chaîne de format contient le texte suivant :

- Breakpoint:The input value is %f

Lorsque le point d'arrêt se déclenche, la fenêtre de suivi du point d'arrêt affiche les caractères avant les deux-points (« Breakpoint ») dans la barre de titre de la fenêtre de suivi. Les autres caractères composent les suivis. Dans cet exemple, %f représente la première (et dans ce cas, la seule) étiquette à être suivie (« analogvalue »).

Les suivis obtenus apparaissent comment suit.



Voir aussi

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)

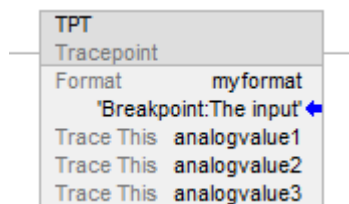
## Points de suivi (TPT)

Cette instruction est compatible uniquement avec les automates Studio 5000 Logix Emulate.

Points de suivi enregistrent les données sélectionnées lorsqu'un échelon est vrai.

### Langages disponibles

#### Diagramme à relais



### Bloc fonctionnel

Cette instruction n'est pas disponible dans le bloc fonctionnel.

### Texte structuré

Cette instruction n'est pas disponible dans le texte structuré.

### Opérandes

Il existe des règles de conversion de données pour les types de données mixtes dans une instruction. Reportez-vous à la section Conversion de données.

### Diagramme à relais

Opérande	Type	Format	Description
Format	String	Étiquette	Chaîne définissant le formatage des rapports de suivi (à la fois à l'écran et enregistrés sur le disque).
Trace This	BOOL SINT INT DINT REAL	Étiquette	L'étiquette que vous souhaitez suivre.

### Description

Les points de suivi sont programmés avec l'instruction de sortie de points de suivi (TPT). Lorsque les entrées d'un échelon contenant une instruction TPT est vrai, l'instruction TPT écrit une entrée de suivi sur un affichage de suivi ou dans un fichier journal.

L'instruction TPT vous permet de suivre de nombreuses étiquettes. Cependant, la chaîne de formatage ne peut contenir que 82 caractères. Étant donné que la chaîne de formatage requiert deux caractères pour chaque étiquette que vous souhaitez avoir dans le suivi, vous ne pouvez pas suivre plus de 41 étiquettes avec une seule instruction TPT. Cependant, pour séparer les données d'étiquette dans vos suivis, vous devez inclure des espaces et d'autres marques de formatage, afin de réduire le nombre d'étiquettes qu'une instruction TPT peut réellement suivre à une valeur bien inférieure à 41.

### Format de chaîne

La chaîne Format présente dans les instructions de point de suivi et d'arrêt vous permet de contrôler la façon dont les étiquettes suivies apparaissent dans les fenêtres de suivis ou fenêtres de point d'arrêt. Le format de la chaîne est le suivant :

- heading:(text)%(type)

dans lequel heading est une chaîne de texte identifiant le point de suivi ou d'arrêt, text est une chaîne décrivant l'étiquette (ou tout autre texte de votre choix) et %(type) indique le format de l'étiquette. Un indicateur type est nécessaire pour chaque étiquette que vous suivrez avec l'instruction de point de suivi ou d'arrêt.

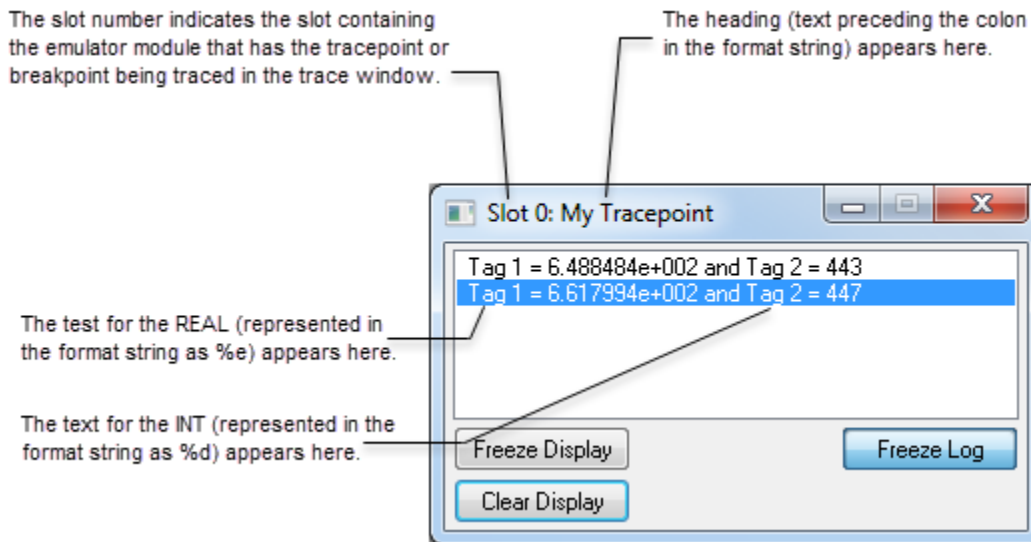
Par exemple, vous pouvez formater une chaîne de point de suivi comme indiqué ci-dessous :

- My tracepoint:Tag 1 = %e and Tag 2 = %d

Le %e formate la première étiquette suivie comme un flottant à double précision avec un exposant, et %d formate la deuxième étiquette suivie comme un entier décimal signé.

Dans ce cas, vous obtenez une instruction de point de suivi comportant deux opérandes Trace This (un pour une valeur REAL et un pour INT, bien que la valeur d'une étiquette puisse être formatée avec n'importe quel indicateur).

La fenêtre de point de suivi obtenue qui apparaît lorsque le point de suivi est déclenché a l'apparence de l'exemple ci-dessous.



**Affecte les indicateurs d'état mathématique**

Non

**Conditions de défaut**

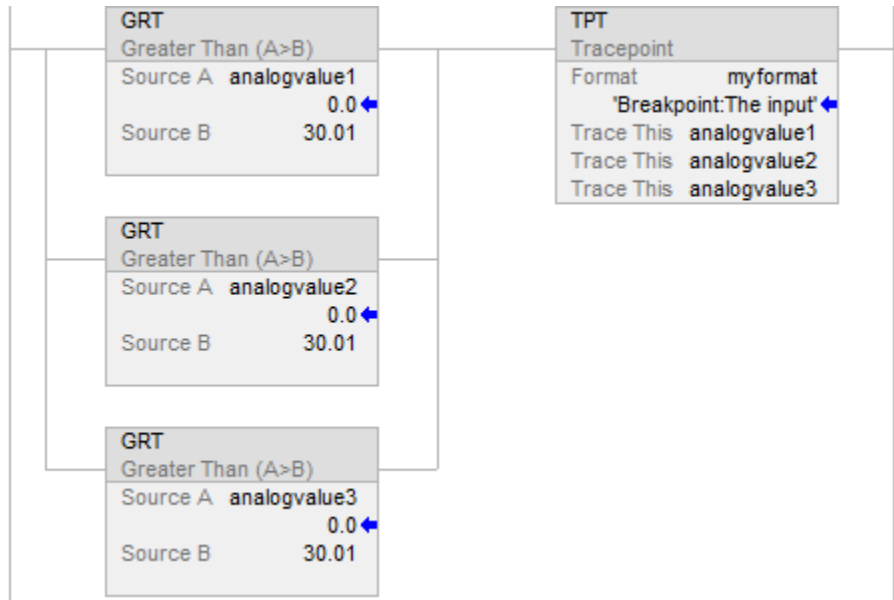
Aucun spécifique à cette instruction. Reportez-vous à la section Attributs communs pour plus d'informations sur les défauts liés aux opérandes.

**Exécution**

Condition	Action de logique à relais
Pré-scrutation	L'échelon devient faux.
Condition d'entrée d'échelon est fausse	L'échelon devient faux.
Condition d'entrée d'échelon est vraie	L'échelon devient vrai. L'exécution saute à l'échelon qui contient l'instruction LBL avec le nom de label référencé.
Post-scrutation	L'échelon devient faux.

**Exemple**

Cet échelon déclenche le suivi de trois valeurs analogiques lorsque l'une d'entre elles dépasse une valeur donnée (30,01).



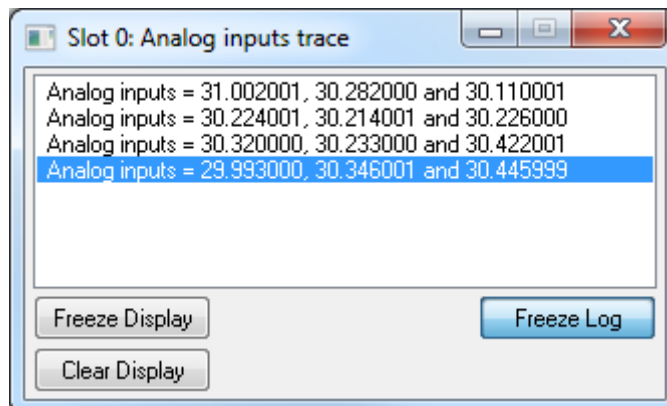
Afficher les informations de point de suivi dans la chaîne Format (myformat).

Dans ce cas, la chaîne de format contient le texte suivant :

- Analog inputs trace:Analog inputs = %f, %f, and %f

Lorsque le point de suivi se déclenche, les caractères avant les deux points (« Analog inputs trace ») apparaissent dans la barre de titres de la fenêtre de suivi. Les autres caractères composent les suivis. Dans cet exemple, %f représente les étiquettes à suivre (« analogvalue1 », « analogvalue2 » et « analogvalue3 »).

Les suivis obtenus apparaissent comment suit.



Lorsque ce suivi est enregistré sur le disque, les caractères présents avec le double-point apparaissent dans les suivis.

Cela indique le point de suivi qui a provoqué l'entrée de suivi. Voici un exemple d'entrée de suivi. « Analog inputs trace: » est le texte d'en-tête provenant de la chaîne de format du point de suivi.

Analog inputs trace: Analog inputs = 31.00201, 30.282000, and 30.110001.

**Voir aussi**

[Instructions de débogage](#) sur la [page 871](#)

[Point d'arrêt \(BPT\)](#) sur la [page 872](#)

[Attributs communs](#) sur la [page 885](#)

[Conversions de données](#) sur la [page 889](#)



# Instructions Licence

Les instructions Licence permettent de vérifier les licences utilisées dans un projet.

### Langages disponibles

### Diagramme à relais



### Bloc fonctionnel

Indisponible

### Texte structuré

Indisponible

### Voir aussi

[Instructions de conversion mathématique](#) sur la [page 767](#)

## Validation de licence (LV)

Ces informations s'appliquent aux automates Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580.

L'instruction LV (validation de licence) vérifie l'existence d'une licence associée qui n'a pas expiré avec une routine ou des instructions complémentaires dans l'automate.

### Langages disponibles

### Diagramme à relais

LV	
License Validation	
Vendor Code	?
Product Code	?

**Bloc fonctionnel**

Cette instruction n'est pas disponible dans le bloc fonctionnel.

**Texte structuré**

Cette instruction n'est pas disponible dans le texte structuré.

**Opérandes****Diagramme à relais**

Opérande	Type (Type)	Format	Description (Description)
Code vendeur (Vendor Code)	DINT	immediate	Nombre unique identifiant le vendeur de la licence associée à une routine ou à une instruction complémentaire. Accepte une valeur entière immédiate comprise entre 0 et 2 147 483 647.
Code du produit (Product Code)	DINT	immediate	Nombre unique identifiant le code de produit de la licence associée à une routine ou une instruction complémentaire. Accepte une valeur entière immédiate comprise entre 0 et 2 147 483 647.

**Affecter les indicateurs d'état mathématique**

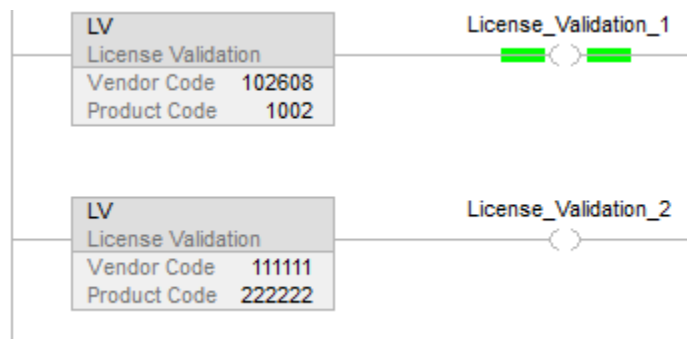
Non

**Défauts majeurs/mineurs**

Aucun spécifique à cette instruction.

**Exécution****Diagramme à relais**

Condition/État	Action entreprise
Pré-scrutation	N/A
Condition d'entrée d'échelon est fausse	N/A
Condition d'entrée d'échelon est vraie	Comparaison numérique Si la licence est valide et utilisée dans le projet Définir la condition de sortie d'échelon sur vrai sinon Définir la condition de sortie d'échelon sur faux
Post-scrutation	N/A

**Exemple****Voir aussi**

[Instructions Licence](#) sur la [page 881](#)



## Attributs courants des instructions générales

Suivez les lignes directrices de ce chapitre pour les attributs courants des instructions générales.

### Attributs communs

Pour obtenir de plus amples informations sur les attributs qui sont communs aux instructions LOGIX 5000™, cliquez sur un des rubriques ci-dessous.

[Indicateurs d'état mathématique](#) sur la [page 885](#)

[Valeurs immédiates](#) sur la [page 888](#)

[Conversions de données](#) sur la [page 889](#)

[Types de données élémentaires](#) sur la [page 893](#)

[Types de données LINT](#) sur la [page 896](#)

[Valeurs de virgules flottantes](#) sur la [page 896](#)

[Indexer via des tableaux](#) sur la [page 899](#)

[Adressage de bits](#) sur la [page 900](#)

### Indicateurs d'état mathématique

Suivez les recommandations de cette rubrique pour les Indicateurs d'état mathématiques.

#### Description (Description)

Automates	Description (Description)
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Une série d'Indicateurs d'état mathématique permettant un accès direct, avec des instructions. Ces indicateurs ne peuvent être actualisés que dans les routines de diagramme à relais et ne sont pas des étiquettes et les alias d'indicateurs ne s'appliquent pas.
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Une série d'Indicateurs d'état mathématique permettant un accès direct, avec des instructions. Ces indicateurs sont actualisés dans le cadre de tous les types de routines, mais ne sont pas des étiquettes, et les alias d'indicateurs ne s'appliquent pas.

Indicateurs d'état

Indicateur d'état	Description (Description) (pour Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580)	Description (Description) (pour Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570)
S:FS Indicateur première scrutation	<p>L'indicateur première scrutation est configuré par l'automate :</p> <ul style="list-style-type: none"> <li>• La première fois qu'un programme est scruté après que l'automate passe en mode Exécution</li> <li>• La première fois qu'un programme est scruté après que le programme est débloqué</li> <li>• Lorsqu'une routine est sollicitée par une action SFC et que l'étape qui comprend cette action est d'abord scruté.</li> </ul> <p>Utilisez l'indicateur de première scrutation pour initialiser des données à utiliser au cours de scrutations ultérieures. Il fait également référence au premier bit passé.</p>	<p>L'indicateur première scrutation est configuré par l'automate :</p> <ul style="list-style-type: none"> <li>• La première fois qu'un programme est scruté après que l'automate passe en mode Exécution</li> <li>• La première fois qu'un programme est scruté après que le programme est débloqué</li> <li>• Lorsqu'une routine est sollicitée par une action SFC et que l'étape qui comprend cette action est d'abord scruté.</li> </ul> <p>Utilisez cet indicateur pour initialiser des données à utiliser au cours de scrutations ultérieures. Il fait également référence au premier bit passé.</p>
S:N Indicateur négatif	<p>L'automate configure l'indicateur négatif lorsque le résultat d'une opération mathématique ou logique est une valeur négative. Utilisez cet indicateur pour tester rapidement une valeur négative.</p>	<p>L'automate configure l'indicateur négatif lorsque le résultat d'une opération mathématique ou logique est une valeur négative. Utilisez cet indicateur pour tester rapidement une valeur négative.</p> <p>L'utilisation de S:N est plus efficace que celle de l'instruction CMP.</p>
S:Z Indicateur zéro	<p>L'indicateur zéro est défini par l'automate lorsque le résultat d'une opération mathématique ou logique est zéro. Utilisez cet indicateur pour tester rapidement une valeur nulle.</p> <p>L'indicateur zéro s'efface au début de l'exécution d'une instruction en mesure de définir cet indicateur.</p>	<p>L'automate définit l'indicateur zéro lorsque le résultat d'une opération mathématique ou logique est zéro. Utilisez cet indicateur pour tester rapidement une valeur nulle.</p>

<p>S:V Indicateur de débordement</p>	<p>L'automate configure l'indicateur de débordement lorsque :</p> <ul style="list-style-type: none"> <li>Le résultat d'une opération mathématique entraîne un débordement. Par exemple, le fait d'ajouter 1 à une valeur SINT génère un débordement lorsque cette valeur passe de 127 à -128.</li> <li>L'étiquette de destination est trop petite pour garder la valeur. Par exemple, si vous tentez de mémoriser la valeur 123456 dans une étiquette SINT ou INT.</li> </ul> <p>Utilisez l'indicateur de débordement pour vous assurer que le résultat d'une opération est situé dans la plage autorisée.</p> <p>Si les données en cours de mémorisation sont du type chaîne, S:V est défini si la chaîne est trop importante et ne peut pas s'inscrire dans l'étiquette de destination.</p> <p><b>Astuce</b> : le cas échéant, définissez S:V avec une instruction OTE ou OTL.</p> <p>Cliquez sur <b>Propriétés de l'automate &gt; Onglet avancé &gt; Signaler les défauts de dépassement</b> (Controller Properties &gt; Advanced tab &gt; Report Overflow Faults) pour activer ou désactiver l'option de signalisation des défauts de débordement.</p> <p>Si un débordement se produit pendant l'évaluation d'un indice de tableau, un défaut mineur est généré et un défaut majeur est généré pour signaler que l'indice est hors plage.</p>	<p>L'automate configure l'indicateur de débordement lorsque :</p> <ul style="list-style-type: none"> <li>Le résultat d'une opération mathématique entraîne un débordement. Par exemple, le fait d'ajouter 1 à une valeur SINT génère un débordement lorsque cette valeur passe de 127...à 128.</li> <li>L'étiquette de destination est trop petite pour garder la valeur. Par exemple, si vous tentez de mémoriser la valeur 123456 dans une étiquette SINT ou INT.</li> </ul> <p>Utilisez l'indicateur de débordement pour vérifier que le résultat d'une opération est situé dans la plage autorisée.</p> <p>Un défaut mineur est généré à tout moment où l'indicateur de déb ordement est configuré.</p> <p><b>Astuce</b> : le cas échéant, définissez S:V avec une instruction OTE ou OTL.</p>
<p>S:C Indicateur de retenue</p>	<p>L'automate configure l'indicateur de retenue lorsque le résultat de l'opération mathématique a provoqué la création d'une réalisation du bit le plus significatif.</p> <p>Seules les instructions ADD et SUB et non pas les opérateurs + et -, avec des valeurs entières, affectent cet indicateur.</p>	<p>L'automate configure l'indicateur de retenue lorsque le résultat de l'opération mathématique a provoqué la création d'une réalisation du bit le plus significatif.</p>
<p>S:MINOR Indicateur défaut mineur</p>	<p>L'automate définit l'indicateur défaut mineur lorsqu'il existe au moins un défaut mineur dans le Program.</p> <p>Utilisez l'étiquette de défaut mineur pour vérifier si un défaut mineur s'est produit. Ce bit ne se déclenche qu'en cas de programmation de défauts comme, par exemple, le débordement. Il ne se déclenche pas en cas de défaut de batterie. Il s'efface au début de chaque scrutation.</p> <p><b>Astuce</b> : le cas échéant, définissez, de manière explicite, S:MINOR avec une instruction OTE ou OTL.</p>	<p>L'automate définit l'indicateur défaut mineur lorsqu'il existe au moins un défaut mineur dans le Program.</p> <p>Utilisez l'indicateur de défaut mineur pour vérifier si un défaut mineur s'est produit et pour agir en conséquence. Ce bit ne se déclenche qu'en cas de programmation de défauts comme, par exemple, le débordement. Il ne se déclenche pas en cas de défaut de batterie. Il s'efface au début de chaque scrutation.</p> <p><b>Astuce</b> : le cas échéant, définissez, de manière explicite, S:MINOR avec une instruction OTE ou OTL.</p>
<p><b>Important :</b></p>	<p>Les indicateurs d'état mathématique sont définis en fonction de la valeur mémorisée. Des instructions qui, normalement, n'affectent pas les indicateurs d'état mathématique risquent de donner l'impression d'affecter des indicateurs d'état mathématique si une conversion de type se produit, par rapport aux types mixtes de données, pour les paramètres d'instructions. Le processus de conversion de type règle les indicateurs d'état mathématique.</p>	

**Expressions dans les indices de tableau**

<b>Automates</b>	<b>Description (Description)</b>
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	Les expressions ne définissent pas les indicateurs d'état en fonction des résultats des opérations mathématiques. Si les expressions débordent : <ul style="list-style-type: none"> <li>• Un défaut mineur se produit si l'automate est configuré pour générer des défauts mineurs.</li> <li>• Un défaut majeur (type 4, code 20) est généré si la valeur obtenue est hors plage.</li> </ul>
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	Les expressions définissent les indicateurs d'état en fonction des résultats des opérations mathématiques. Si un indice de tableau est une expression, cette expression et l'instruction risquent de générer des défauts mineurs.

**Astuce :** Si un indice de tableau est trop important (hors plage), cela génère un défaut majeur (type 4, code 20).

**Valeurs immédiates**

Lorsque vous entrez une valeur immédiate (constante) au format décimal (par exemple, -2, 3), l'automate stocke cette valeur en utilisant 32 bits. Si vous entrez une valeur dans une base numérique autre que décimale, telle que binaire ou hexadécimale, et que vous ne spécifiez pas l'ensemble des 32 bits, l'automate place un zéro dans les bits que vous n'avez pas spécifiés (remplissage avec des zéros).

**Important :** Remplissage avec des zéros des valeurs immédiates binaires, octales ou hexadécimales inférieures à 32 bits.

<b>Si vous entrez</b>	<b>L'automate stocke</b>
-1	16#ffff ffff (-1)
16#ffff (-1)	16#0000 ffff (65535)
8#1234 (668)	16#0000 029c (668)
2#1010 (10)	16#0000 000a (10)

**Valeurs immédiates entières**

<b>Si vous entrez</b>	<b>L'automate stocke</b>
Sans aucun suffixe	DINT
« U »	UDINT
« L »	LINT
« UL »	ULINT



### Valeurs immédiates à virgule flottante

Si vous entrez	L'automate stocke
Sans aucun suffixe	REAL
« L »	LREAL

## Conversions de données

La conversion de données s'effectue lors du mélange des types de données dans la programmation.

Lors de la programmation :	Des conversions peuvent se produire lorsque vous :
Diagramme à relais Texte structuré	Mélangez des types de données de paramètres entre eux Instruction ou expression
Bloc fonctionnel	Câblez deux paramètres possédant des types de données différents

Les instructions s'exécutent plus rapidement et nécessitent moins de mémoire si tous les opérandes de l'instruction utilisent :

- Le même type de données.
- Un type de données intermédiaire :
  - Toutes les instructions du bloc fonctionnel prennent en charge seulement un opérande de types de données.
  - Si vous mélangez des types de données ou utilisez des étiquettes différentes du type de données optimal, le contrôleur convertit les données selon les règles suivantes :
    - Les opérandes sont convertis selon le classement des types de données entre SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL et LREAL, avec un classement allant de 1 (le plus faible) à 10 (le plus élevé).

**Astuce :** Pour réduire le temps et la mémoire lors de la conversion de données, utilisez le même type de données pour tous les opérandes d'une instruction.

### Convertissez SINT ou INT vers DINT ou DINT vers LINT

Une étiquette de source d'entrée SINT ou INT est promue en valeur DINT par une extension de signe de l'étiquette source. Les instructions qui convertissent des valeurs SINT ou INT en valeurs DINT utilisent l'une des méthodes de conversion suivantes.

Cette méthode de conversion	Convertit les données en plaçant
Une extension de signe	La valeur du bit le plus à gauche (signe de la valeur) dans chaque position de bits à la gauche des bits existants jusqu'à atteindre 32 ou 64 bits.
Remplissage avec des zéros	Mise à zéro à la gauche des bits existants jusqu'à atteindre 32 ou 64 bits.

Les instructions logiques utilisent le remplissage avec des zéros. Toutes les autres instructions utilisent l'extension de signe

L'exemple suivant montre les résultats de la conversion d'une valeur en utilisant l'extension de signe et le remplissage avec des zéros.

Cette valeur	2#1111_1111_1111_1111	(-1)
Convertit cette valeur avec l'extension de signe	2#1111_1111_1111_1111_1111_1111_1111_1111	(-1)
Convertit cette valeur avec le remplissage avec zéro	2#0000_0000_0000_0000_1111_1111_1111_1111	(65535)

Si vous utilisez une étiquette SINT ou INT et une valeur immédiate dans une instruction qui convertit les données par une extension de signe, utilisez l'une de ces méthodes pour traiter les valeurs immédiates.

Spécifiez une valeur immédiate dans la base de numérotation décimale.

Si vous entrez la valeur dans une base numérique autre que décimale, spécifiez tous les 32 bits de la valeur immédiate. Pour cela, entrez la valeur du bit le plus à gauche dans chaque position de bit sa gauche jusqu'à atteindre 32 bits.

Créez une étiquette pour chaque opérande et utilisez le même type de données dans toute l'instruction. Pour attribuer une valeur constante, vous pouvez utiliser l'une des méthodes suivantes :

Entrez-la dans une des étiquettes.

Ajoutez une instruction MOV qui déplace la valeur dans l'une des étiquettes.

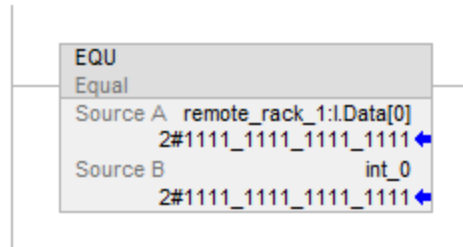
Utilisez une instruction MEQ pour vérifier uniquement les bits requis.

Les exemples suivants montrent deux manières de mélanger une valeur immédiate avec étiquette INT. Ces deux exemples, vérifient les bits d'un module d'E/S 1771 pour déterminer si tous les bits sont activés. Comme l'entrée d'un mot de donnée d'un module d'E/S 1771 est une étiquette INT, il est plus facile d'utiliser une valeur constante de 16 bits.

---

**Important :** Mélange d'une étiquette INT avec une valeur immédiate  
 Comme `remote_rack_1:l.Data[0]` est une étiquette INT, la valeur à vérifier par rapport elle est également entrée sous la forme d'une étiquette INT.

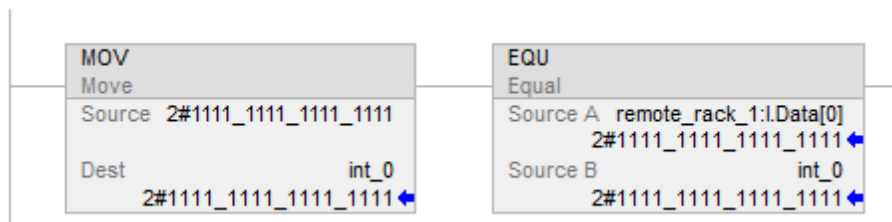
---




---

**Important :** Mélange d'une étiquette INT avec une valeur immédiate  
 Comme `remote_rack_1:l.Data[0]` est une étiquette INT, la valeur à vérifier par rapport elle est d'abord déplacée dans `int_0`, qui est également une étiquette INT. L'instruction EQU compare alors les deux étiquettes.

---



### Convertir un entier vers à REAL

L'automate mémorise des valeurs REAL en simple précision IEEE, au format de nombre à virgule flottante. Il utilise un bit pour le signe de la valeur, 23 bits pour la valeur de référence, et huit bits pour l'exposant (32 bits au total). Si vous mélangez une étiquette entière (SINT, INT ou DINT) et une étiquette REAL comme entrées dans la même instruction, l'automate convertit la valeur entière en valeur REAL avant que l'instruction ne s'exécute.

- Une valeur SINT ou INT se convertit en même valeur REAL.
- Une valeur DINT ne peut pas se convertir dans la même valeur REAL.
- Une valeur REAL utilise jusqu'à 24 bits pour la valeur de référence (23 bits stockés plus un bit « masqué »).
- Une valeur DINT utilise jusqu'à 32 bits pour la valeur (un pour le signe et 31 pour la valeur).

Si la valeur DINT nécessite plus de 24 bits de poids significatif, elle peut ne pas se convertir dans la même valeur REAL. Si elle ne le fait pas, l'automate stocke les 24 bits de poids le plus élevé arrondis à la valeur paire la plus proche.

**Convertir DINT vers SINT ou INT**

Pour convertir une valeur DINT en valeur SINT ou INT, l'automate tronque la partie supérieure de la valeur DINT et stocke les bits de poids inférieur qui correspondent au type de données. Si la valeur est trop importante, la conversion génère un débordement.

		<b>Convertir une valeur DINT vers une valeur INT et une valeur SINT</b>	
Cette valeur DINT		Se convertit en cette valeur plus petite	
16#0001_0081 (65 665)	INT :	16#0081 (129)	
	SINT	16#81 (-127)	

**Convertir REAL vers SINT, INT ou DINT**

Pour convertir une valeur REAL en une valeur entière, l'automate arrondit la partie fractionnaire et stocke les bits qui correspondent au type de données du résultat. Si la valeur est trop importante, la conversion génère un débordement.

Les nombres sont arrondis comme dans les exemples suivants.

Les fractions < 0,5 s'arrondissent au nombre entier inférieur le plus proche.

Les fractions > 0,5 s'arrondissent au nombre entier supérieur le plus proche.

Les fractions = 0,5 s'arrondissent au nombre entier inférieur ou supérieur le plus proche.

<b>Important :</b> Conversion de valeurs REAL en valeurs DINT	
<b>Cette valeur REAL</b>	<b>Se convertit en cette valeur DINT</b>
-2,5	-2
-3,5	-4
-1,6	-2
-1,5	-2
-1,4	-1
1,4	1
1,5	2
1,6	2
2,5	2
3,5	4

## Types de données élémentaires

L'automate prend en charge les types de données élémentaires définis dans les types de données définis par IEC 1131-3. Les types de données élémentaires sont :

Type de donnée	Description	Plage
BOOL	1 bit booléen	0 = mis à zéro 1 = définir
SINT	1 octet entier	de -128 à 127
INT	2 octet entier	de -32 768 à 32 767
DINT	4 octet entier	de -2 147 483 648 à 2 147 483 647
REAL	4 octet nombre à virgule flottante	-3,402823E <sup>38</sup> à -1,1754944E <sup>-38</sup> (valeurs négatives) et 0 et 1,1754944E <sup>-38</sup> à 3,402823E <sup>38</sup> (valeurs positives)
LINT	8 octet entier	de 0 à 32 535 129 599 999 999
USINT	1 octet entier non signé	de 0 à 255
UINT	2 octet entier non signé	de 0 à 65 535
UDINT	4 octet entier non signé	de 0 à 4 294 967 295
ULINT	8 octet entier non signé	de 0 à 18 446 744 073 709 551 615
REAL	4 octet nombre à virgule flottante	-3,4028235E38 à -1,1754944E-38 (valeurs négatives) et 0,0 et 1,1754944E-38 à 3,4028235E38 (valeurs positives)
LREAL	8 octet nombre à virgule flottante	-1,7976931348623157E308 à -2,2250738585072014E-308 (valeurs négatives) et 0,0 et 2,2250738585072014E-308 à 1,7976931348623157E308 (valeurs positives)

Ces automates prennent en charge les types de données élémentaires suivantes :

Automates	Type de donnée
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	SINT, INT, DINT, LINT, REAL USINT, UINT, UDINT, ULINT, LREAL
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	SINT, INT, DINT, LINT, REAL.

L'automate traite toutes les valeurs immédiates comme des types de données DINT.

En outre, le type de données REAL mémorise  $\pm$  l'infinité et  $\pm$  NAN, mais l'affichage du logiciel varie en fonction du format d'affichage.

### Conversion du type de données

Lorsque les types de données sont mélangés pour les opérandes dans une instruction, certaines instructions convertiront automatiquement les données en type de données optimales pour cette instruction. Dans certains cas, l'automate convertit les données afin qu'elles s'adaptent à un nouveau type de données ; dans d'autres cas l'automate adapte simplement les données du mieux qu'il le peut.

Conversion	Result		
de grand entier à petit entier	L'automate tronque la partie supérieure du plus grand entier et génère un débordement. Par exemple :		
	<b>Décimal</b>		<b>Binaire</b>
	DINT	65 665	0000_0000_0000_0001_0000_0000_1000_0001
	INT	129	0000_0000_1000_0001
	SINT	-127	1000_0001
SINT ou INT à REAL	Aucune précision de données perdue		
DINT à REAL	Risque de perte de la précision de données. Les deux types de données stockent les données en 32 bits mais le type REAL utilise certains de ces 32 bits pour stocker la valeur exposante. Si la précision est perdue, l'automate la prend de la partie moins significative des DINT.		
LREAL vers LREAL	Aucune précision de données n'est perdue.		
LREAL vers REAL	Risque de perte de la précision de données.		
LREAL/REAL vers entier non signé	Risque de perte de la précision de données. Si la valeur source est trop importante pour rentrer dans la destination, l'automate stocke ce qu'il peut et cela risque de produire un débordement.		
Entier signé/Entier non signé vers LREAL/REAL	Si la valeur entière possède plus de bits significatifs que ce que peut stocker la destination, les bits de poids faible sont tronqués.		
Entier signé vers entier non signé	Si la valeur source est trop importante pour rentrer dans la destination, l'automate stocke ce qu'il peut et cela risque de produire un débordement.		
Entier non signé vers entier signé	Si la valeur source est trop importante pour rentrer dans la destination, l'automate stocke ce qu'il peut et cela risque de produire un débordement.		

REAL à entier	L'automate arrondit la partie fractionnaire et tronque la partie supérieure de la partie non fractionnaire. Si les données sont perdues, l'automate règle l'étiquette d'état de débordement. Arrondi au nombre entier le plus proche : Moins de 0.5, arrondissement inférieur ; égal à 0.5, arrondissement à l'entier pair le plus proche ; plus grand que 0.5, arrondissement supérieur Par exemple :	
	<b>REAL (source)</b>	<b>DINT (résultat)</b>
	1,6	2
	-1,6	-2
	1,5	2
	-1,5	-2
	1,4	1
	-1,4	-1
	2,5	2
	-2,5	-2

Ne convertissez pas des données vers ou depuis un type de données BOOL.

**Important :** Les indicateurs d'état mathématique sont réglés en fonction des valeurs qui ont été stockées. Les instructions qui normalement n'affectent pas les mots-clés d'état de math peuvent apparaître pour le faire si une conversion de type se produit à cause des paramètres d'instruction des types de données mélangées. Le processus de conversion de type définit les mots-clés d'état de math.

### Types de données de sécurité

Les applications Logix Designer évitent les modifications de type Définition de l'utilisateur d'une Instruction complémentaire qui engendreraient un type de données invalides référencé direct ou indirectement par une étiquette de Sécurité. (Inclut les structures imbriquées.)

Les étiquettes de données peuvent être composées des types de données suivants :

- Tous les types de données élémentaires
- Types prédéfinis qui sont utilisés pour les instructions d'application de la sécurité.
- Types de données définis par l'utilisateur ou tableaux composés des deux types précédents.

### Édition en ligne de noms de membres UDT dans les étiquettes de sécurité

L'édition en ligne est autorisée pour les noms de membres de types de données définis par l'utilisateur sur les automates CompactLogix 5380, Compact GuardLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Toutefois, l'édition en ligne est désactivée lorsqu'un type de données défini par

l'utilisateur est utilisé sur une étiquette de sécurité et que l'automate est dans l'état Mode de sécurité.

**Voir aussi**

[Indicateurs d'état mathématique](#) sur la [page 885](#)

## Types de données LINT

Le type de données LINT est un entier de 64 bits.

Le type de données LINT peut être utilisé dans de nombreuses instructions sur Automate Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 ou GuardLogix 5580, toutefois, le type de données LINT ne peut pas être utilisé dans la plupart des instructions sur Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570.

Tenez compte des éléments suivants lorsque vous utilisez le type de données LINT sur Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570.

**Astuce :** Les LINT ne peuvent s'utiliser qu'avec les instructions de copie (COP, CPS). Elles s'utilisent avec l'attribut Heure CST/WallClock, la synchronisation de l'heure et les instructions complémentaires. Les étiquettes de ce type ne peuvent faire l'objet d'aucune opération d'addition, de soustraction, de multiplication ou de division.

Lors de l'utilisation de données de type LINT, il faut tenir compte des descriptions suivantes lorsque ces problèmes se produisent.

Comment	Description (Description)
Transférer/copier deux valeurs DINT à entiers doubles dans une LINT	Créer un tableau à double entier à deux éléments, total de 64 bits (à savoir DINT[2], qui peuvent ensuite être copiés dans un entier long.
Corriger l'erreur d'affichage Date/Heure	Lorsqu'une étiquette a une valeur négative, elle ne peut pas s'afficher en tant que Date/Heure. Dans l'éditeur de points, vérifiez si la valeur est négative en changeant de style d'étiquette : passez de Date/Heure à Binaire. Si le bit le plus significatif (le plus à gauche) est 1, la valeur est négative et ne peut donc pas s'afficher en tant que Date ou Heure.

## Valeurs de virgules flottantes

Ces informations s'appliquent aux automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 et GuardLogix 5580. Les différences entre les automates sont signalées, le cas échéant.



Les automates Logix prennent en charge les valeurs, flottantes conformément à la norme IEEE 754 sur l'arithmétique à virgule flottante. Cette norme définit comment les nombres à virgule flottante sont stockés et calculés. La norme IEEE 754 pour les mathématiques à virgule flottante a été conçue pour permettre la vitesse et la possibilité de prendre en charge de très grands nombres dans une quantité d'espace de stockage raisonnable.

Une étiquette REAL stocke un nombre simple précision à virgule flottante normalisé.

Une étiquette LREAL stocke un nombre double précision à virgule flottante normalisé.

Les automates prennent en charge les types de données élémentaires suivants :

Automates	Type de donnée
Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580	REAL, LREAL
Automates CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 et GuardLogix 5570	REAL

Les nombres des normalisés et -0,0 sont traités en tant que 0,0.

Si un calcul donne une valeur NAN, le bit du signe peut être positif ou négatif. Dans cette situation, le logiciel affiche 1#.NAN sans signe.

Les valeurs décimales ne peuvent pas toutes être représentées exactement dans ce format de norme, qui entraîne une perte de précision. Par exemple, si vous soustrayez 10 de 10,1, vous vous attendez à ce que le résultat soit 0,1. Dans un automate Logix, le résultat pourrait très bien être 0,10000038. Dans cet exemple, la différence entre 0,1 et 0,10000038 est de 0,000038%, soit presque zéro. Pour la plupart des opérations, cette petite imprécision est minimale. Pour replacer les choses dans leur contexte, si vous envoyiez une valeur, flottante vers un module de sortie analogique, il n'y aurait aucune différence dans la tension de sortie pour une valeur envoyée au module qui présente une différence de 0,000038 %.

### **Recommandation générale pour les opérations mathématiques sur les nombres à virgule flottante**

Respectez les recommandations suivantes :

Lorsque vous effectuez certaines opérations mathématiques avec des nombres à virgule flottante, il peut y avoir une perte de précision en raison de l'erreur d'arrondi. Les processeurs à virgule flottante possèdent leur propre précision interne qui peut avoir une incidence sur les valeurs résultantes.

N'utilisez pas les mathématiques à virgule flottante pour des valeurs concernant de l'argent ou pour des fonctions de totalisation. Utilisez des valeurs INT ou DINT, mettez les valeurs à l'échelle, et conservez le suivi de la position décimale (ou utilisez une valeur INT ou DINT pour les dollars, et une deuxième valeur INT ou DINT pour les centimes).

Ne comparez pas des nombres à virgule flottante. Vérifiez plutôt des valeurs dans une plage. L'instruction LIM est fournie spécifiquement à cet effet.

### Exemples de totalisateur

La précision du type de données REAL affecte les applications de totalisation, en produisant par exemple des erreurs lors de l'ajout de très grands nombres à de très petits nombres.

Par exemple, ajouter 1 un nombre pendant un certain temps. À un certain point, l'ajout n'affectera plus le résultat, car la somme en cours est beaucoup plus grande que 1 et il n'y a pas suffisamment de bits pour stocker la totalité du résultat. L'ajout stocke autant de bits de poids supérieur que possible et ignore les bits de poids inférieur restants.

Pour contourner cela, effectuez des opérations mathématiques sur de petits nombres jusqu'à ce que le résultat devienne important. Ensuite, transférez-les vers un autre emplacement pour effectuer des opérations mathématiques supplémentaires sur des grands nombres. Par exemple :

- x est la variable de petit incrément.
- x est la variable de grand incrément.
- z est le nombre total actuel pouvant être utilisé partout.
- $x = x + 1$ ;
- if  $x = 100,000$ ;
- {
- $y = y + 100,000$ ;
- $x = 0$ ;
- }
- $z = y + x$ ;

Voici un autre exemple :

- $x = x + \text{some\_tiny\_number}$ ;
- if ( $x \geq 100$ )
- {

- $z = z + 100;$
- $x = x - 100; //$  there might be a tiny remainder
- }

## Indexer via des tableaux

Pour modifier de manière dynamique l'élément de tableau auquel votre logique se réfère, utilisez une étiquette ou une expression en tant que indice pour mettre en évidence cet élément. Cela correspond exactement à l'adressage indirect dans la logique PLC-5. Utilisez ces opérateurs dans une expression afin de spécifier un indice de tableau :

- Astuces :**
- Logix Designer autorise les indices qui sont des étiquettes de données étendues uniquement, et ne prend pas en charge les expressions d'indice qui comportent des types de données étendus.
  - Tous les types de données élémentaires entiers peuvent être utilisés comme index d'indice. Utilisez uniquement les étiquettes SINT, INT et DINT avec des opérateurs pour créer une expression d'indice.

Opérateur	Description (Description)
+	ajouter
-	soustraire/mettre au négatif
*	multiplier
/	diviser
AND	AND
FRD	BCD à nombre entier
NOT	complément
OU	OU
TOD	nombre entier à BCD
SQR	racine carrée
XOR	exclusif OU

Par exemple :

Définitions	Exemple	Description (Description)
my_list défini en tant que DINT[10]	my_list[5]	Cet exemple fait référence à l'élément 5 du tableau. Cette référence est statique parce que la valeur de l'indice reste constante.
my_list défini en tant que DINT[10] position défini en tant que DINT	MOV the value 5 into position my_list[position]	Cet exemple fait référence à l'élément 5 du tableau. Cette référence est dynamique parce que la logique peut modifier l'indice en changeant la valeur de la position.
my_list défini en tant que DINT[10] position défini en tant que DINT décalage défini en tant que DINT	MOV the value 2 into position MOV the value 5 into offset my_list[position+offset]	Cet exemple fait référence à l'élément 7 (2+5) du tableau. Cette référence est dynamique parce que la logique peut modifier l'indice en changeant la valeur de la position ou du décalage.

**Astuce :** Lorsque vous entrez un indice de tableau, assurez-vous qu'il se situe bien dans les limites du tableau spécifié. Les instructions qui considèrent que des tableaux constituent une collection d'éléments génèrent un défaut majeur (type 4, code 20) lorsqu'un indice dépasse la valeur qui lui correspond.

## Adressage de bits

L'adressage par bit est utilisé pour accéder à un bit particulier à l'intérieur d'un conteneur plus grand. Les conteneurs plus grands incluent tous les entiers, les structures et les tableaux BOOL. Par exemple :

des coordonnées	Exemple	Description (Description)
variable0 défini en tant que LINT a 64 bits	variable0.42	Cet exemple réfère au bit 42 de variable0.
variable1 défini en tant que DINT a 32 bits	variable1.2	Cet exemple réfère au bit 2 de variable1.
variable2 défini en tant que INT a 16 bits	variable2.15	Cet exemple réfère au bit 15 de la variable2.
variable3 défini en tant que SINT a 8 bits	variable3.[4]	Cet exemple réfère au bit 4 de variable3.
variable4 défini comme une structure COUNTER a 5 bits d'état	variable4.DN	Cet exemple réfère au bit DN de la variable4.
MyVariable défini en tant que BOOL[100] MyIndex défini en tant que SINT	MyVariable[(MyIndex AND NOT 7) / 8].[MyIndex AND 7]	Cet exemple réfère un bit à l'intérieur d'un tableau BOOL.
MyArray défini en tant que BOOL[20]	MyArray[3]	Cet exemple réfère au bit 3 de MyArray.
variable5 défini en tant que ULINT détient 64 bits	variable5.53	Cet exemple réfère au bit 53 de variable5.

Utilisez l'adressage de bits partout où une étiquette de type BOOL est autorisée.

### Voir aussi

[Indexer via des tableaux](#) sur la [page 899](#)

# Attributs du bloc fonctionnel

Cliquez sur une rubrique ci-dessous pour obtenir plus d'informations sur des problèmes qui sont uniques à la programmation du bloc fonctionnel. Revoyez cette information pour vous assurer de bien comprendre le fonctionnement des routines de blocs fonctionnels.

### Voir aussi

[Sélectionner les éléments de bloc fonctionnel](#) sur la [page 901](#)

[Verrouillage des données](#) sur la [page 902](#)

[Ordre d'exécution](#) sur la [page 903](#)

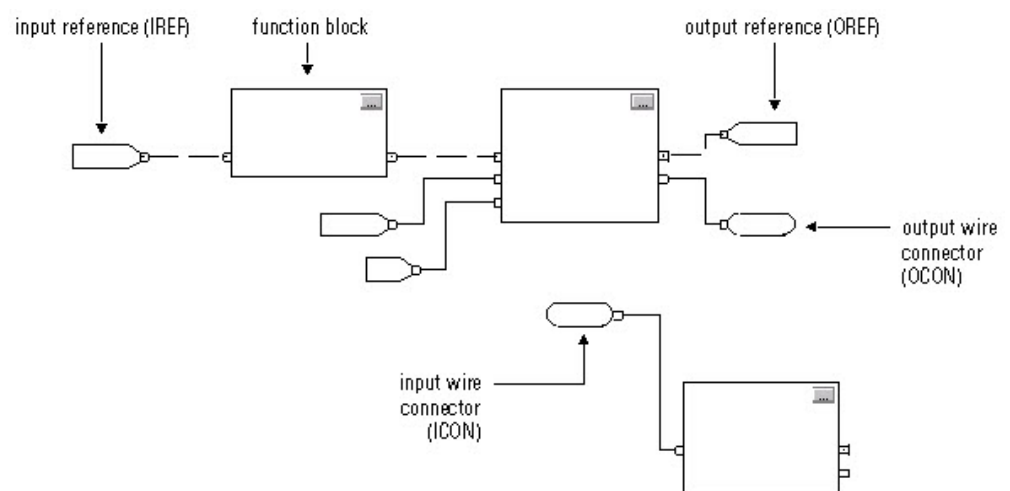
[Réponses du bloc fonctionnel aux conditions de débordement](#) sur la [page 908](#)

[Les modes de temporisation](#) sur la [page 908](#)

[Contrôle Programme/Opérateur](#) sur la [page 912](#)

## Sélectionner les éléments de bloc fonctionnel

Pour commander un dispositif, utilisez ces éléments :



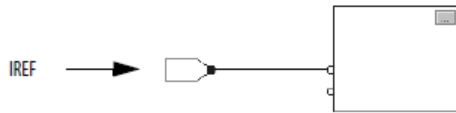
Ce tableau vous aidera à choisir les éléments pour votre bloc fonctionnel :

Si vous voulez fournir une valeur d'un dispositif d'entrée ou d'une étiquette	Alors utilisez une référence d'entrée (IREF)
Envoyer une valeur à un dispositif de sortie ou une étiquette	Référence de sortie (OREF)
Effectuer une opération sur une ou plusieurs valeurs d'entrée et produire une ou plusieurs valeurs de sortie.	Bloc fonctionnel
Transférer des données entre les blocs fonctionnels quand ils sont : <ul style="list-style-type: none"> <li>• Éloignés sur la même feuille</li> <li>• Sur différentes feuilles à l'intérieur de la même routine</li> </ul>	Connecteur de sortie (OCON) et un connecteur d'entrée (ICON)
Disperser les données entre plusieurs points de la routine	Connecteur de sortie unique (OCON) et plusieurs connecteurs d'entrée (ICON)

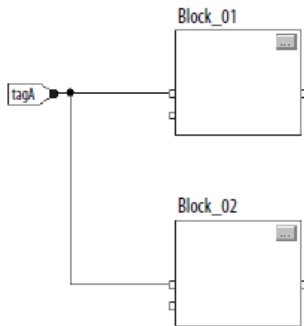
Le bloc fonctionnel déplace les références d'entrée dans la structure du bloc. Si besoin, le bloc fonctionnel convertira ces références d'entrée en valeurs REAL. Le bloc fonctionnel exécute et déplace les résultats dans les références de sortie. Encore, si nécessaire, le bloc fonctionnel convertira ces valeurs de résultat de REAL en types de données pour les référence de sortie.

## Verrouillage des données

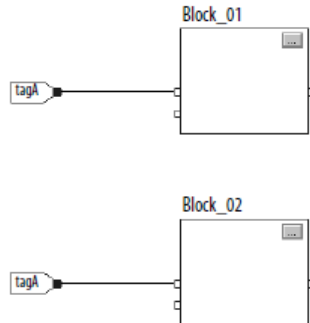
Si vous utilisez une IREF pour préciser la donnée d'entrée pour une instruction de bloc fonctionnel, les données se trouvant dans cette IREF seront verrouillées pour la scrutation de la routine de blocs fonctionnels. L'IREF verrouille les données des étiquette d'accès de l'automate et des étiquette d'accès du programme. L'automate met toutes les données IREF à jour au début de chaque scrutation.



Dans cet exemple, la valeur de tagA est stockée au début de l'exécution de la routine. La valeur stockée est utilisée quand Block\_01 s'exécute. La valeur stockée est aussi utilisée quand Block\_02 s'exécute. Si la valeur de tagA change pendant l'exécution de la routine, la valeur stockée dans l'IREF de tagA ne changera pas jusqu'à l'exécution de la prochaine routine.

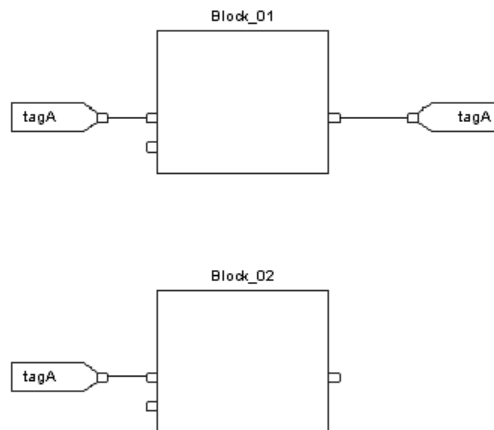


Cet exemple est identique au précédent. La valeur de tagA est stockée une seule fois au début de l'exécution de la routine. La routine utilise la valeur stockée au long de la routine.



Vous pouvez utiliser la même étiquette dans plusieurs IREFs et une OREF dans la même routine. Même si une OREF obtient une valeur d'étiquette différente pendant l'exécution de la routine, toutes les IREFs utiliseront la même valeur car les valeurs des étiquettes dans les IREFs sont verrouillées à chaque scrutation.

Dans cet exemple, quand la routine démarre cette scrutation, si la valeur de tagA est 25,4 et que Block\_01 modifie la valeur de tagA sur 50,9 alors la seconde IREF liée dans Block\_02 continuera d'utiliser la valeur 25,4 lorsque Block\_02 exécutera cette scrutation. La nouvelle valeur 50,9 de tagA ne sera utilisée par aucune IREFs dans cette routine avant le démarrage de la prochaine scrutation.



## Ordre d'exécution

L'ordre d'exécution dans une routine des blocs fonctionnels est défini par l'application de programmation de Logix Designer, lorsque vous :

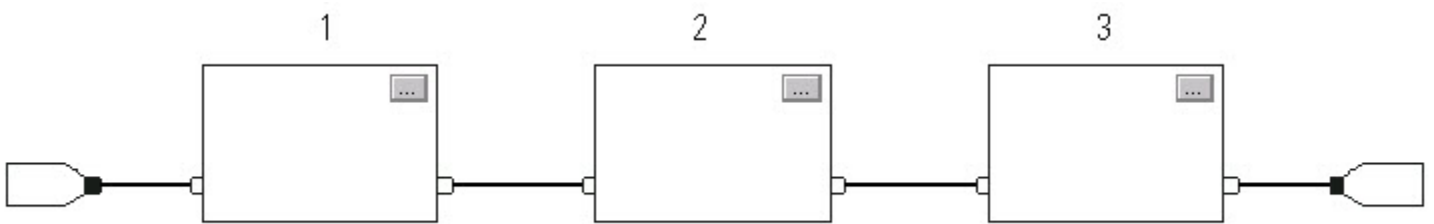
- vérifiez une routine de bloc fonctionnel
- vérifiez un projet comportant une routine de bloc fonctionnel
- téléchargez un projet comportant une routine de bloc fonctionnel

Vous définissez un ordre d'exécution en reliant des blocs fonctionnels et en indiquant le flux de données de tout câble de retour si besoin.

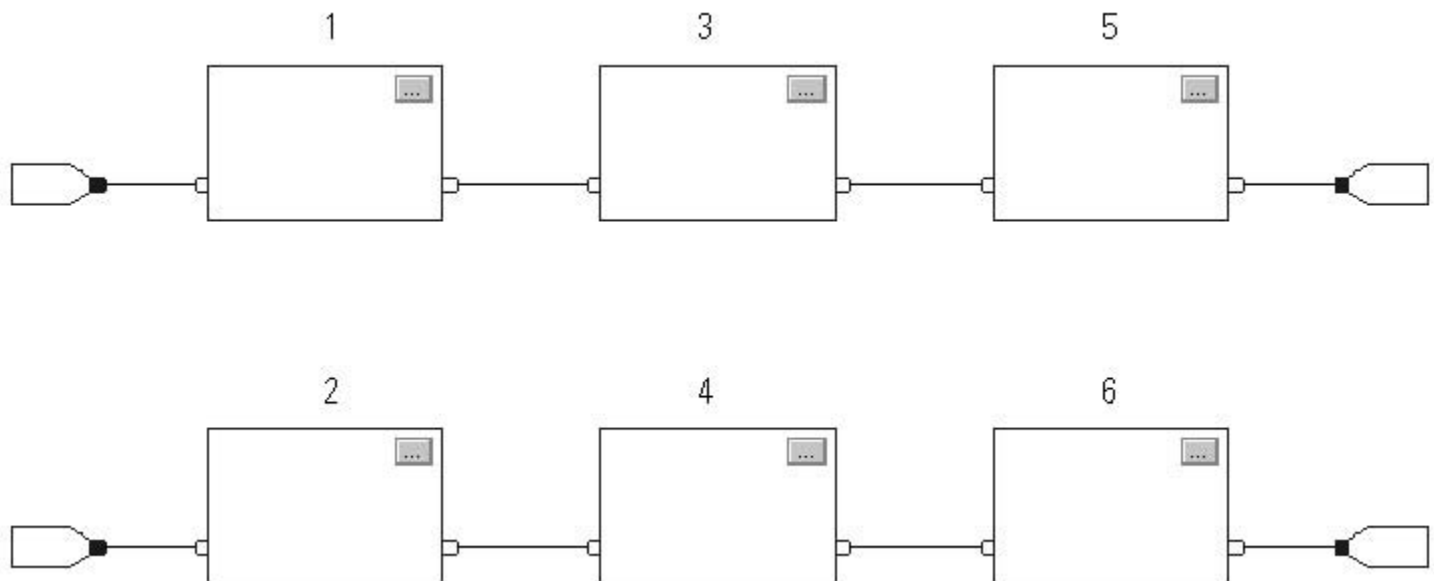
Si les blocs fonctionnels ne sont pas reliés, il n'est pas important de savoir quel bloc s'exécute en premier. Il n'y a pas de flux de données entre les blocs.



Si les blocs sont reliés de façon séquentielle, l'ordre d'exécution se déplace de l'entrée vers la sortie. Les entrées d'un bloc exigent que les données soient disponibles avant que l'automate puisse exécuter ce bloc. Par exemple, le bloc 2 doit s'exécuter avant le bloc 3 car la sortie du bloc 2 alimente les entrées du bloc 3.



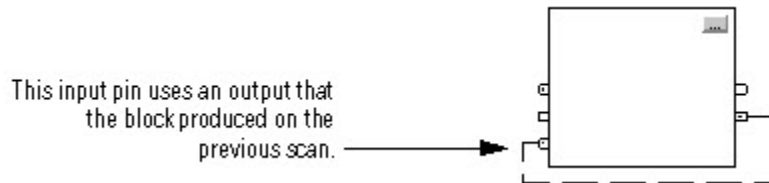
L'ordre d'exécution ne se rapporte qu'aux blocs qui sont reliés. L'exemple suivant est pertinent car les deux groupes de blocs ne sont pas reliés. Les blocs d'un groupe spécifique s'exécutent dans l'ordre approprié avec les blocs de ce groupe.



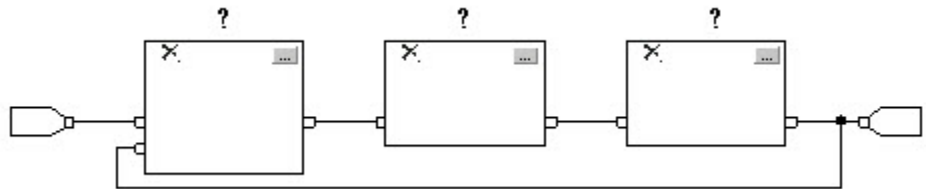


### Résoudre une boucle

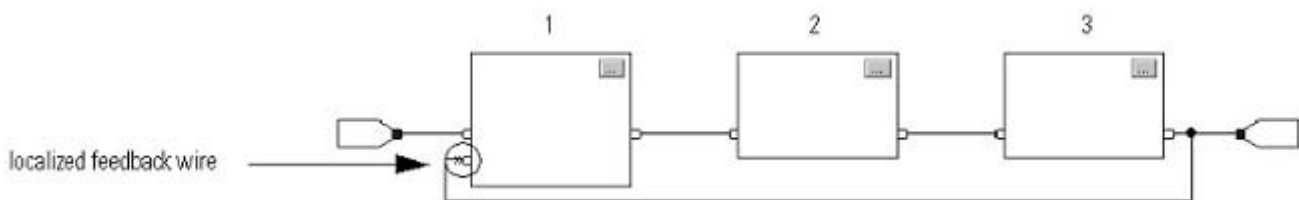
Pour créer une boucle de retour autour d'un bloc, liez une broche de sortie à une broche d'entrée du même bloc. L'exemple suivant est OK. La boucle ne contient qu'un seul bloc, l'ordre d'exécution n'a donc pas d'importance.



Si un groupe de blocs est en boucle, l'automate ne peut pas déterminer quel bloc doit être exécuté en premier. En d'autres termes, il ne peut pas résoudre la boucle.

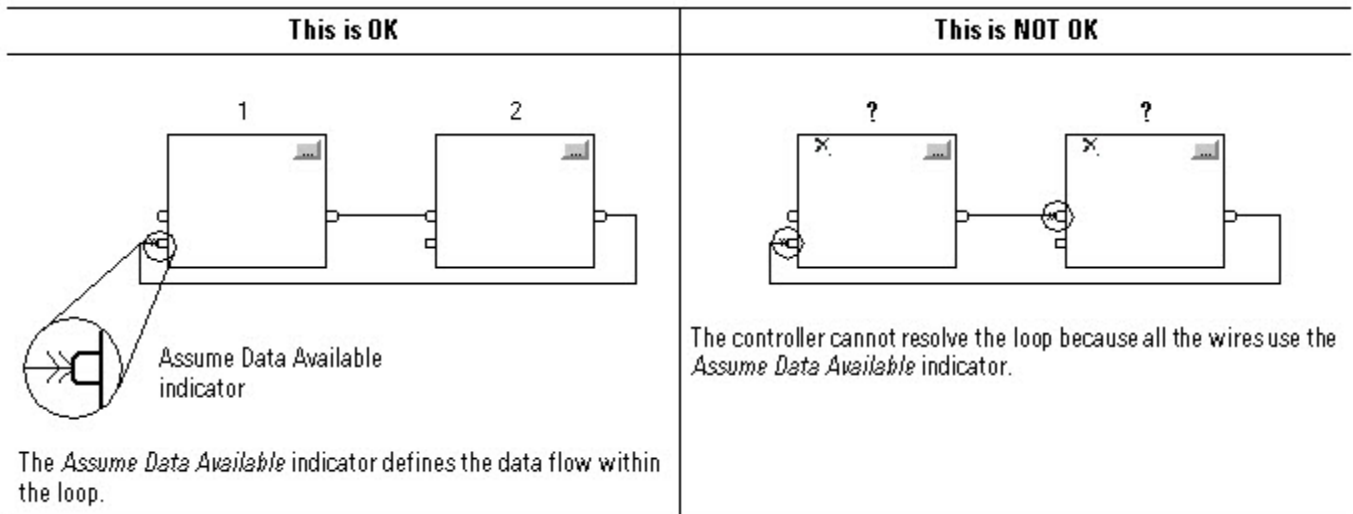


Afin d'identifier le bloc qui s'exécute en premier, marquez le câble d'entrée qui crée la boucle (le câble de retour) avec l'indicateur de *Données présumées disponibles*. Dans l'exemple suivant, le bloc 1 utilise la sortie du bloc 3 produite dans la routine d'exécution précédente.



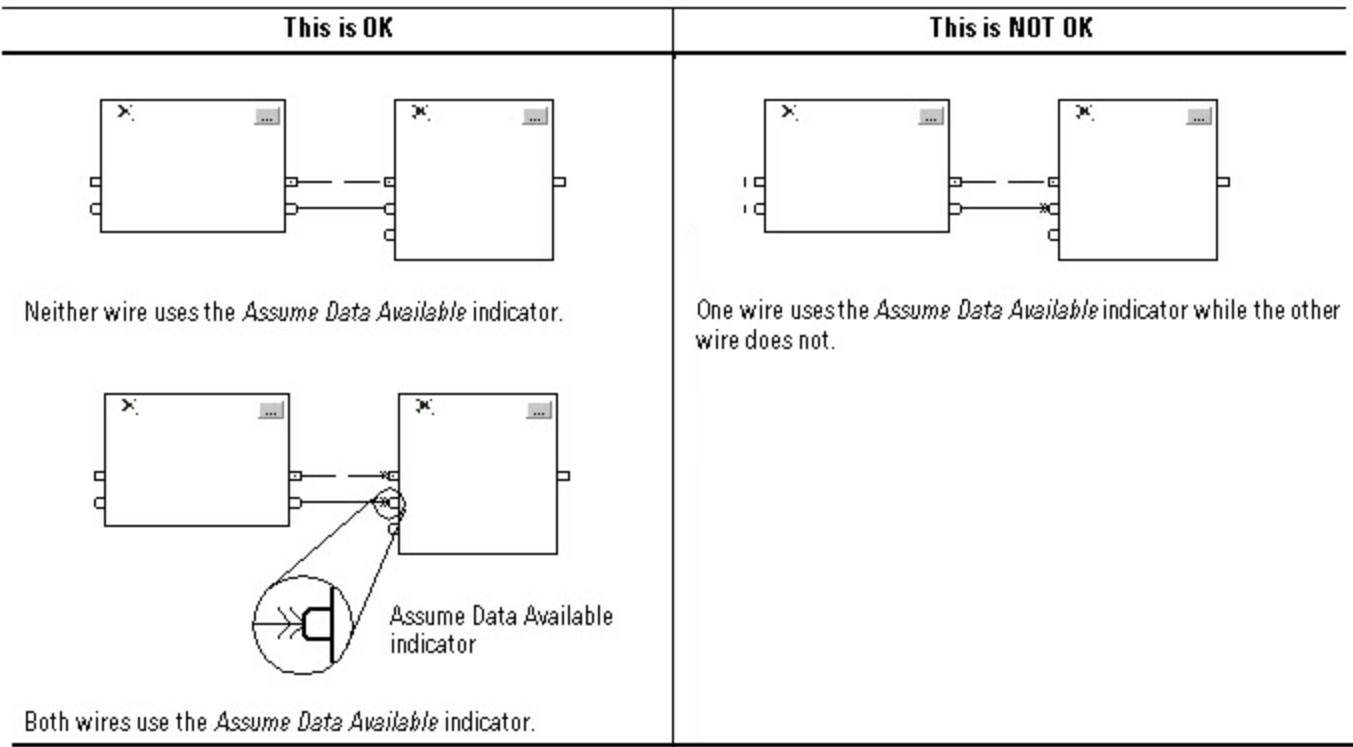
L'indicateur de *Données présumées disponibles* détermine le flux de données dans la boucle. La flèche indique que la donnée joue le rôle d'entrée du premier bloc de la boucle.

Ne marquez *pas* tous les câbles d'une boucle avec l'indicateur de *Données présumées disponibles*.



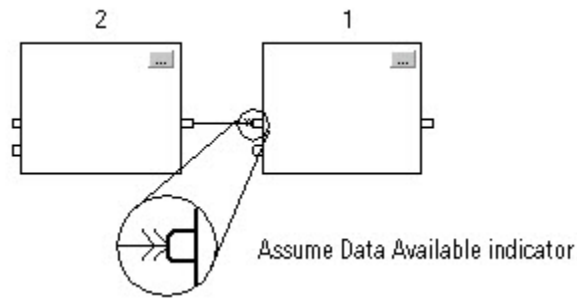
### Résoudre le flux de données entre deux blocs

Si vous utilisez deux câbles ou plus pour relier deux blocs, utilisez les mêmes indicateurs de flux de données pour tous les câbles entre deux blocs.



### Créer un report de scrutation

Pour produire un report de scrutation entre deux blocs, utilisez le même indicateur de Données supposées disponibles. Dans l'exemple suivant, le bloc 1 s'exécute en premier. Il utilise la sortie du bloc 2 produite dans la scrutation de routine précédente.



### Sommaire

Dans le sommaire, l'ordre d'exécution d'une routine de bloc fonctionnel est le suivant :

1. L'automate verrouille toutes les valeurs de données dans IREFs.
2. L'automate exécute les autres blocs fonctionnels dans l'ordre établi par leurs liaisons.
3. L'automate inscrit les sorties dans OREFs.

## Réponses du bloc fonctionnel aux conditions de débordement

En général, lors d'un débordement, les instructions du bloc fonctionnel ne mettent pas leur historique à jour pour des valeurs  $\pm$  NAN ou  $\pm$ INF. Chaque instruction a l'une de ces réponses pour une condition de débordement.

Réponse	Instruction
<p>Réponse 1</p> <p>Les blocs exécutent leur algorithme et vérifient le résultat pour <math>\pm</math>NAN ou <math>\pm</math>INF. Si <math>\pm</math>NAN ou <math>\pm</math>INF, les sorties du bloc <math>\pm</math>NAN ou <math>\pm</math>INF.</p>	<p>ALM NTCH</p> <p>DEDT PMUL</p> <p>DERV POSP</p> <p>ESEL RLIM</p> <p>FGEN RMPS</p> <p>HPF SCRIV</p> <p>LDL2 SEL</p> <p>LDLG SNEG</p> <p>LPF SRTP</p> <p>MAVE SSUM</p> <p>MAXC TOT</p> <p>MINC UPDN</p> <p>MSTD</p> <p>MUX</p>
<p>Réponse 2</p> <p>Les blocs avec limitation de sortie exécutent leur algorithme et vérifient le résultat pour <math>\pm</math>NAN ou <math>\pm</math>INF. Les limitations de sortie sont définies par les paramètres d'entrée HighLimit et LowLimit. Si <math>\pm</math>INF, le bloc produit un résultat limité. Si <math>\pm</math>NAN, les limites de sortie ne sont pas utilisées et le bloc produit <math>\pm</math>NAN.</p>	<p>HLL, INTG, PI, PIDE, SCL, SOC</p>
<p>Réponse 3</p> <p>La condition de débordement ne s'applique pas. Ces instructions ont généralement une sortie booléenne.</p>	<p>BAND, BNOT, BOR, BXOR, CUTD, D2SD, D3SD, DFF, JKFF, OSFI, OSRI, RESD, RTOR, SETD, TOFR, TONR</p>

## Les modes de temporisation

Ces instructions des variateurs et de contrôle du processus prennent en charge différents modes de temporisation.

- DEDT
- DERV
- HPF
- INTG
- LDL2
- LDLG
- LPF
- NTCH
- PI
- PIDE
- RLIM
- SCRIV
- SOC
- TOT

Il y a trois différents modes de temporisation.

Mode de temporisation	Description (Description)	
Périodique	Le mode périodique est le mode par défaut qui est adapté à la plupart des applications de contrôle. Nous vous recommandons de placer les instructions utilisant ce mode dans une routine qui est exécutée dans une tâche périodique. Le temps delta (DeltaT) de l'instruction est déterminé comme suit :	
	<b>Si l'instruction est exécutée dans un/une</b>	<b>Alors DeltaT est égale à</b>
	Tâche périodique	Période de la tâche
	Événement ou tâche continue	Temps écoulé depuis l'exécution précédente L'automate tronque le temps écoulé en millisecondes entières (ms). Par exemple, si le temps écoulé = 10,5 ms, l'automate définit DeltaT = 10 ms.
	La mise à jour de l'entrée du processus doit être synchronisée avec l'exécution de la tâche ou échantillonnée 5 à 10 fois plus rapidement que la durée d'exécution de la tâche afin de minimiser l'erreur d'échantillonnage entre l'entrée et l'instruction.	
Sur-échantillonnage	<p>Dans le mode sur-échantillonnage, le temps delta (DeltaT) utilisé par l'instruction est la valeur contenue dans le paramètre OversampleDT de l'instruction. Si l'entrée du processus dispose d'une valeur d'horodotage, utilisez le mode d'échantillonnage en temps réel à la place.</p> <p>Ajoutez à votre programme la logique dont celui-ci à besoin pour contrôler quand l'instruction est exécutée. Par exemple, vous pouvez utiliser un temporisateur défini sur la valeur OversampleDeltaT pour contrôler l'exécution en utilisant l'entrée EnableIn de l'instruction.</p> <p>L'entrée du processus doit être échantillonnée 5 à 10 fois plus rapidement que la durée d'exécution de l'instruction afin de minimiser l'erreur d'échantillonnage entre l'entrée et l'instruction.</p>	
Échantillonnage en temps réel	<p>Dans le mode d'échantillonnage en temps réel, le temps delta (DeltaT) utilisé par l'instruction est la différence entre deux horodotages qui correspondent à la mise à jour de l'entrée du processus. Utilisez ce mode quand l'entrée du processus dispose d'un horodotage associé à sa mise à jour que vous avez besoin d'une coordination précise.</p> <p>L'horodotage est lu à partir du nom d'étiquette saisi dans le paramètre RTSTimeStamp de l'instruction. Normalement, ce nom d'étiquette est un paramètre du module d'entrée associé à l'entrée du processus.</p> <p>L'instruction compare la valeur RTSTime configurée (intervalle de la mise à jour prévue) avec la valeur DeltaT calculée, pour déterminer si chaque mise à jour de l'entrée du processus est effectivement lue par l'instruction. Si DeltaT n'est pas à moins d'une milliseconde de l'intervalle configuré, l'instruction définit le bit d'état RTSMissed pour indiquer qu'il y a un problème de lecture de la valeur de mise à jour de l'entrée du module.</p>	

Dans le cas des instructions basées sur le temps, DeltaT doit avoir une valeur constante pour que l'algorithme de contrôle puisse calculer correctement la sortie du processus. Si DeltaT varie, une discontinuité se produit dans la sortie du processus. La sévérité de la discontinuité dépend de l'instruction et de la plage de variation de DeltaT.

Une discontinuité se produit en présence des facteurs suivants :

- L'instruction n'est pas exécutée durant une scrutation.
- L'instruction est exécutée de nombreuses fois durant une tâche.
- La tâche est en cours d'exécution et la fréquence de scrutation de la tâche ou le temps d'échantillon de l'entrée du processus change.
- L'utilisateur change le mode basé sur le temps alors que la tâche est en cours d'exécution.
- Le paramètre Order est changé dans un bloc de filtrage alors que la tâche est en cours d'exécution.
- Le fait de changer le paramètre Order sélectionne un algorithme de contrôle différent à l'intérieur de l'instruction.

### Paramètres d'instructions communs pour les modes de temporisation

Les instructions qui prennent en charge les modes basés sur le temps disposent de ces paramètres d'entrée et de sortie.

#### Paramètres d'entrée

Paramètre d'entrée	Type de donnée	Description (Description)
TimingMode	DINT	<p>Sélectionne le mode d'exécution de la temporisation.</p> <p>Valeur : Description :</p> <p>0 Mode périodique</p> <p>1 Mode de sur-échantillonnage</p> <p>2 Mode d'échantillonnage en temps réel</p> <p>Valide = 0 à 2</p> <p>Par défaut = 0</p> <p>Lorsque TimingMode = 0 et que la tâche est périodique, la temporisation périodique est activée et DeltaT est défini sur la fréquence de scrutation de la tâche. Lorsque TimingMode = 0 et que la tâche est événement ou continue, la temporisation périodique est activée et DeltaT est défini sur le temps écoulé depuis la dernière exécution de l'instruction.</p> <p>Lorsque TimingMode = 1, la temporisation par sur-échantillonnage est activée et DeltaT est défini sur la valeur du paramètre OversampleDT.</p> <p>Lorsque TimingMode = 2, la temporisation par échantillonnage en temps réel est activée et DeltaT est la différence entre les valeurs de l'horodotage actuel et de l'horodotage précédent obtenues du module associé à l'entrée.</p> <p>Si TimingMode n'est pas valide, l'instruction définit le bit approprié dans Status.</p>
OversampleDT	REAL	<p>Temps d'exécution pour la temporisation par sur-échantillonnage. La valeur utilisée pour DeltaT est en secondes. Si TimingMode = 1, alors OversampleDT = 0,0 désactive l'exécution de l'algorithme de contrôle. Si la valeur n'est pas valide, l'instruction définit DeltaT = 0,0 et définit le bit approprié dans Status.</p> <p>Valide = 0 à 4194.303 secondes</p> <p>Par défaut = 0,0</p>

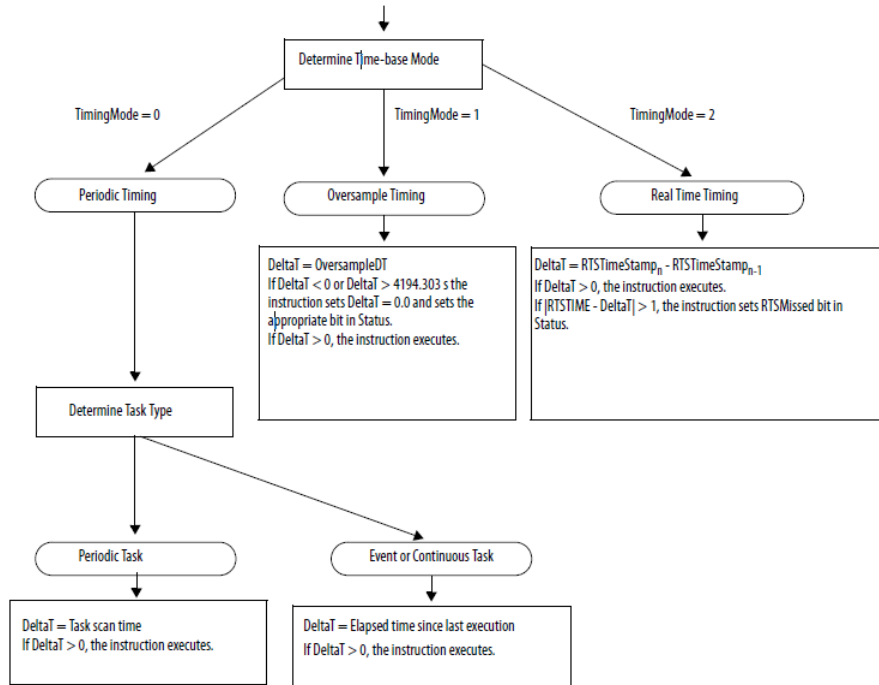
RTSTime	DINT	Période de mise à jour du module pour la temporisation par échantillonnage en temps réel. La période de temporisation DeltaT attendue est en millisecondes La période de temporisation est normalement la valeur qui a été utilisée pour configurer le temps de mise à jour du module. Si la valeur n'est pas valide, l'instruction définit le bit approprié dans Status et désactive la vérification de RTSMissed. Valide = 1...32 767 ms Par défaut = 1
RTTimeStamp	DINT	La valeur d'horodatage du module pour la temporisation par échantillonnage en temps réel. La valeur d'horodatage qui correspond à la dernière mise à jour du signal d'entrée. Cette valeur est utilisée pour calculer DeltaT. Si la valeur n'est pas valide, l'instruction définit le bit approprié dans Status, désactive l'exécution de l'algorithme de contrôle, et désactive la vérification de RTSMissed. Valide =0...32 767 ms (après 32 767, recommence à 0) 1 comptage = 1 milliseconde Par défaut = 0

### Paramètres de sortie

Paramètre de sortie	Type de donnée	Description (Description)
DeltaT	REAL	Temps écoulé entre les mises à jour. Cela correspond au temps écoulé en seconde utilisé par l'algorithme de contrôle pour calculer la sortie du processus. Périodique : DeltaT = fréquence de scrutation de la tâche si la tâche est Périodique, DeltaT = temps écoulé depuis l'exécution précédente de l'instruction si la tâche est du type Événement ou Continue Sur-échantillonnage : DeltaT = OversampleDT Échantillonnage en temps réel : DeltaT = (RTTimeStampn - RTTimeStampn-1)
État (Status)	DINT	État du bloc fonctionnel.
TimingModeInv (Status.27)	BOOL	Valeur TimingMode non valide.
RTSMissed (Status.28)	BOOL	Utilisé seulement en mode Échantillonnage en temps réel. Défini lorsque $ABS   DeltaT - RTSTime   > 1$ (,001 seconde).
RTTimeInv (Status.29)	BOOL	Valeur RTSTime non valide.
RTTimeStamp Inv (Status.30)	BOOL	Valeur RTTimeStamp non valide.
DeltaTInv (Status.31)	BOOL	Valeur DeltaT non valide.

### Présentation des modes de temporisation

Le diagramme suivant montre comment une instruction détermine le mode de temporisation approprié.



## Contrôle Programme/Opérateur

Les instructions suivantes prennent en charge le concept de contrôle Programme/Opérateur.

- Sélection améliorée (ESEL)
- Totalisateur (TOT)
- PID amélioré (PIDE)
- Rampe/Stabilisation (RMPS)
- Dispositif à 2 états discret (D2SD)
- Dispositif à 3 états discret (D3SD)

Les contrôles Programme/Opérateur vous permettent de contrôler ces instructions simultanément à la fois de votre programme utilisateur et de votre dispositif d'interface opérateur. Dans le contrôle Programme, l'instruction est contrôlée par les entrées Programme qui lui sont envoyées ; dans le contrôle Opérateur, l'instruction est contrôlée par les entrées Opérateur qui lui sont envoyées.



Le contrôle Programme ou Opérateur, est déterminé par l'utilisation de ces entrées.

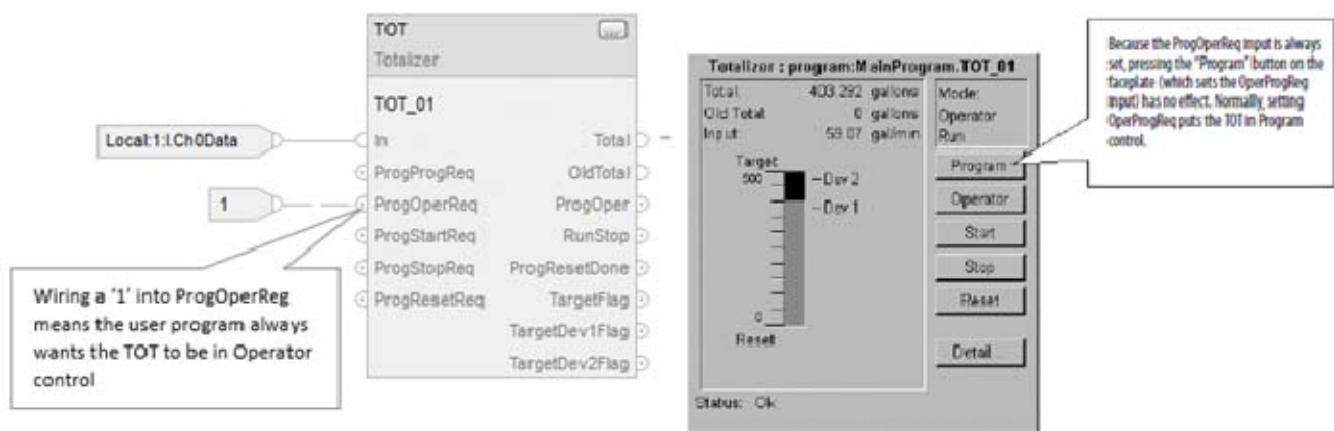
Entrée (Input)	Description (Description)
.ProgProgReq	Une requête du programme à envoyer au contrôle Programme.
.ProgOperReq	Une requête du programme à envoyer au contrôle Opérateur.
.OperProgReq	Une requête de l'opérateur à envoyer au contrôle Programme.
.OperOperReq	Une requête de l'opérateur à envoyer au contrôle Opérateur.

Pour déterminer si une instruction est sous le contrôle Programme ou Opérateur, examinez la sortie ProgOper. Si ProgOper est défini, l'instruction est sous le contrôle Programme ; si ProgOper est mis à zéro, l'instruction est sous le contrôle Opérateur.

Le contrôle Opérateur est prioritaire sur le contrôle Programme si les deux bits de requête d'entrée sont définis. Par exemple, si ProgProgReq et ProgOperReq sont tous les deux définis, l'instruction est envoyée au contrôle Opérateur.

Les entrées de requêtes de programme sont prioritaires sur les entrées de requêtes d'opérateur. Cela permet d'utiliser les entrées ProgProgReq et ProgOperReq pour « verrouiller » une instruction dans le contrôle désiré.

Par exemple, supposons qu'une instruction Totalisateur va toujours être utilisée sous le contrôle Opérateur, et que votre programme ne va jamais contrôler l'exécution ou l'arrêt du Totalisateur. Dans ce cas, vous pourriez câbler une valeur littérale de 1 dans ProgOperReq. Cela empêcherait l'opérateur de mettre le Totalisateur sous le contrôle Programme en définissant OperProgReq depuis le dispositif d'interface opérateur.

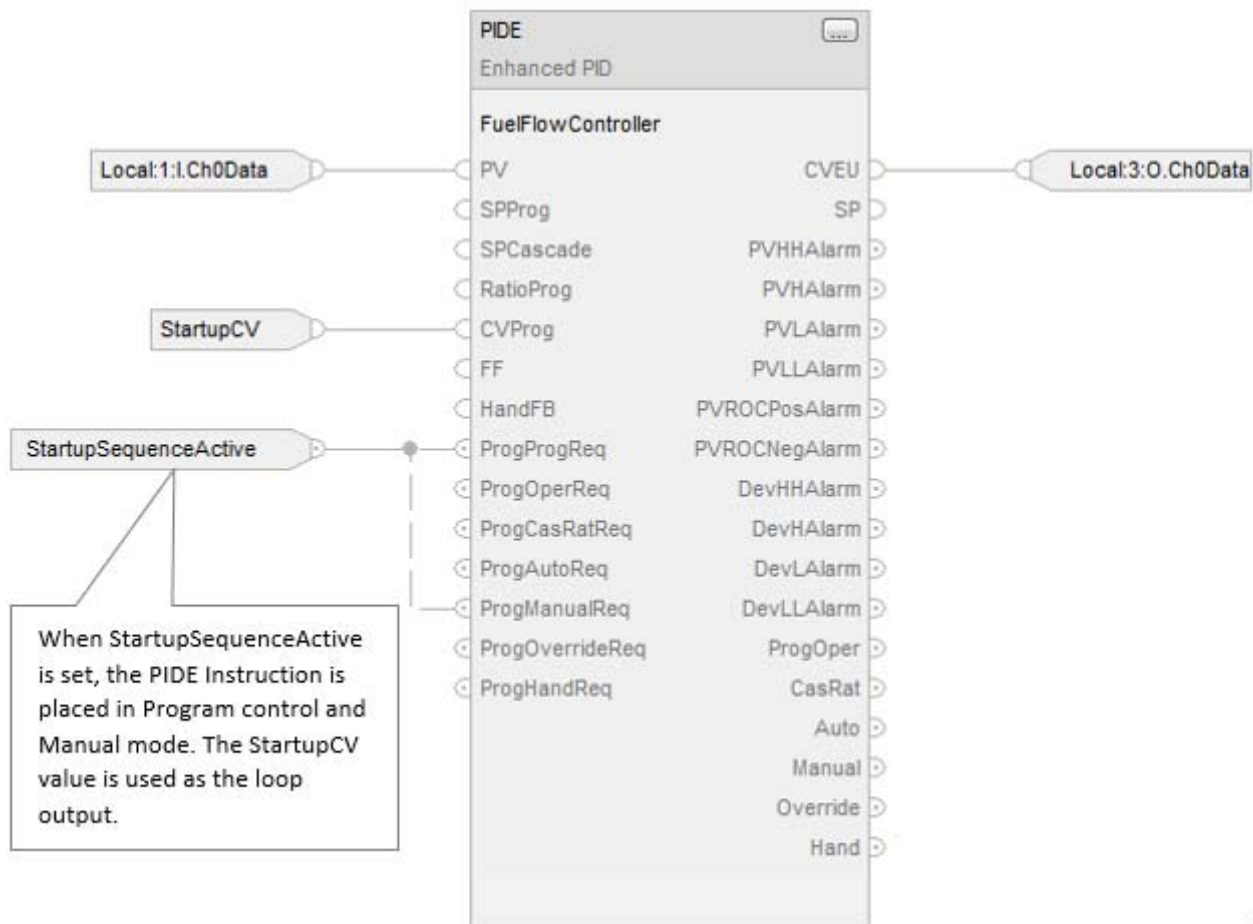


De même, le fait de définir constamment ProgProgReq peut « verrouiller » l'instruction sous le contrôle Programme. Cela est utile pour les séquences de

démarrage automatique lorsque vous désirez que le programme contrôle les actions de l'instruction sans avoir à vous soucier d'un opérateur prenant par inadvertance le contrôle de l'instruction.

Dans cet exemple, le programme définit l'entrée ProgProgReq durant le démarrage, puis il met à zéro l'entrée ProgProgReq une fois que le démarrage est terminé. Une fois que l'entrée ProgProgReq est mise à zéro, l'instruction reste sous le contrôle Programme jusqu'à ce qu'elle reçoive un requête de changement. Par exemple, l'opérateur pourrait définir l'entrée OperOperReq depuis une face avant pour prendre le contrôle de cette instruction.

L'exemple suivant montre comment verrouiller une instruction en contrôle Programme.



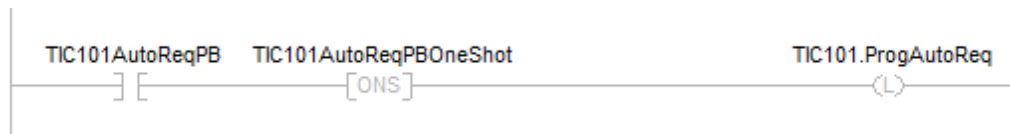
Les entrées de requêtes d'opérateur envoyées à une instruction sont toujours mises à zéro par l'instruction lorsque celle-ci est exécutée. Cela permet aux interfaces opérateur de fonctionner avec ces instructions en définissant simplement le bit de requête du mode désiré. Vous n'avez pas à programmer l'interface opérateur pour réinitialiser à zéro les bits de requête. Par exemple, si une interface opérateur définit l'entrée OperAutoReq vers une instruction PIDE, lorsque l'instruction

PIDE est exécutée, celle-ci détermine la réponse appropriée et met à zéro OperAutoReq.

Les entrées de requêtes de programme sont normalement mises à zéro par l'instruction car elles sont en général câblées comme des entrées pour l'instruction. Si l'instruction met à zéro ces entrées, l'entrée serait simplement redéfini par l'entrée câblée. Il peut exister des situations où vous voulez utiliser l'autre logique pour définir les requêtes de programme de façon à ce que les requêtes de programme soient mises à zéro par l'instruction. Dans ce cas, vous pouvez définir l'entrée ProgValueReset et l'instruction mettra toujours à zéro les entrées de requête en mode Programme, lorsqu'elle est exécutée.

Dans cet exemple, un échelon de diagramme logique à relais d'une autre routine est utilisé pour verrouiller par impulsion une entrée ProgAutoReq à une instruction PIDE lorsqu'un bouton-poussoir est enfoncé.

Lorsque le bouton-poussoir TIC101AutoReqPB est enfoncé, ProgAutoReq est verrouillé par impulsion à l'instruction PIDE TIC101. TIC101 a été configuré avec son entrée ProgValueReset définie. ProgAutoReq est réinitialisé parce que ProgValueReset est toujours défini.





# Programmation de texte structuré

Voici les problèmes qui sont uniques à la programmation de texte structuré. Revoquez les rubriques suivantes pour vous assurer de bien comprendre comment votre texte structuré va être exécuté.

[Syntaxe du texte structuré](#) sur la [page 917](#)

[Composants de texte structuré : Commentaires](#) sur la [page 919](#)

[Composants de texte structuré : Affectations](#) sur la [page 920](#)

[Composants de texte structuré : Expressions](#) sur la [page 923](#)

[Composants de texte structuré : Instructions](#) sur la [page 928](#)

[Composants de texte structuré : Construction](#) sur la [page 930](#)

[CASE...OF](#) sur la [page 932](#)

[FOR...DO](#) sur la [page 935](#)

[IF...THEN](#) sur la [page 938](#)

[REPEAT...UNTIL](#) sur la [page 941](#)

[WHILE...DO](#) sur la [page 943](#)

## Syntaxe du texte structuré

Un texte structuré est un langage de programmation textuel qui utilise des instructions pour définir les actions à exécuter.

- Le texte structuré n'est pas sensible à la casse.
- Utilisez des tabulations et des retours chariot (retours à la ligne) pour faciliter la lecture de votre texte structuré. Ceux-ci n'ont aucun effet sur l'exécution du texte structuré.

Le texte structuré n'est pas sensible à la casse. Le texte structuré peut contenir ces composants.

Terme	des coordonnées	Exemples
Affectation	Utilisez une instruction d'affectation pour affecter des valeurs aux étiquettes. L'opérateur := est l'opérateur d'affectation. Terminez l'affectation avec un point-virgule « ; ».	tag := expression;
Expression	Une expression fait partie d'une affectation complète ou d'une instruction de construction. Une expression évalue pour produire un nombre (expression numérique), une chaîne (expression au format chaîne), ou un état vrai ou faux (expression BOOL)	
Expression d'étiquette	Une zone nommée en mémoire où des données sont stockées (BOOL, SINT, INT, DINT, REAL, chaîne)	value1
Expression immédiate	Une valeur constante	4
Expression avec opérateurs	Un symbole ou un mnémonique qui spécifie une opération à l'intérieur d'une expression.	tag1 + tag2 tag1 >= value1
Expression fonctionnelle	Lorsqu'elle est exécutée, une fonction produit une valeur. Utilisez des parenthèses pour contenir les opérandes d'une fonction. Même si leurs syntaxes sont similaires, les fonctions diffèrent des instructions car les fonctions peuvent être utilisées dans les expressions. Les instructions ne peuvent pas être utilisées dans les expressions.	fonction(tag1)
Instruction	Une instruction est une commande unique. Une instruction utilise des parenthèses pour contenir ses opérandes. Une instruction peut avoir aucun, un, ou plusieurs opérandes. Lorsque l'instruction est exécutée, celle-ci produit une ou plusieurs valeurs qui font partie d'une structure de données. Terminez l'instruction avec un point-virgule « ; ». Même si leurs syntaxes sont similaires, les instructions diffèrent des fonctions car les instructions ne peuvent pas être utilisées dans les expressions. Les fonctions peuvent être utilisées dans les expressions.	instruction();  instruction(operand);  instruction(operand1, operand2,operand3);
Construction	Une instruction conditionnelle utilisée pour déclencher le code de texte structuré (c'est-à-dire d'autres instructions). Terminez la construction avec un point-virgule « ; ».	IF...THEN CASE FOR...DO WHILE...DO REPEAT...UNTIL EXIT
Commentaire	Texte qui explique ou clarifie ce que fait une section de texte structuré. Utilisez les commentaires pour faciliter la compréhension du texte structuré. Les commentaires n'ont aucun effet sur l'exécution du texte structuré. Les commentaires peuvent apparaître n'importe où dans le texte structuré.	//comment  (*start of comment . . . end of comment*)  /*start of comment . . . end of comment*/

**Voir aussi**

[Composants du texte structuré : Affectations](#) sur la [page 920](#)

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

[Composants du texte structuré : Instructions](#) sur la [page 928](#)

[Composants du texte structuré : Constructions](#) sur la [page 930](#)

[Composants du texte structuré : Commentaires](#) sur la [page 919](#)

## Composants du texte structuré : Commentaires

Pour rendre votre texte structuré plus facile à comprendre, ajoutez des commentaires.

- Les commentaires vous permettent d'utiliser un langage explicite pour décrire le fonctionnement de votre texte structuré.
- Les commentaires n'ont aucun effet sur l'exécution du texte structuré.

### Pour ajouter des commentaires à votre texte structuré :

Pour ajouter un commentaire	Utilisez l'un de ces formats
Sur une ligne unique	//comment (*comment*)
à la fin d'une ligne de texte structuré	/*comment*/
à l'intérieur d'une ligne de texte structuré	(*comment*) /*comment*/
qui est sur plus d'une ligne	(*start of comment. . .end of comment*) /*start of comment. . .end of comment*/

Par exemple :

Format	Exemple
//comment	<b>Au début d'une ligne</b> //Check conveyor belt direction IF convoyeur_direction THEN... <b>À la fin d'une ligne</b> ELSE //If conveyor isn't moving, set alarm light lumière := 1; END_IF;
(*comment*)	Sugar.Inlet[:=]1;(*open the inlet*) IF Sugar.Low (*low level LS*)& Sugar.High (*high level LS*)THEN... (*Controls the speed of the recirculation pump. The speed depends on the temperature in the tank.*) IF tank.temp > 200 THEN...
/*comment*/	Sugar.Inlet:=0;/*close the inlet*/ IF bar_code=65 /*A*/ THEN... /*Gets the number of elements in the Inventory array and stores the value in the Inventory_Items tag*/ SIZE(Inventory,0,Inventory_Items);

## Composants du texte structuré : Affectations

Utilisez une affectation pour changer la valeur stockée dans une étiquette. Une affectation a cette syntaxe :

*tag := expression;*

où :

Composant	Description (Description)	
Étiquette (Tag)	Représente l'étiquette qui reçoit la nouvelle valeur ; l'étiquette doit être un BOOL, SINT, INT, DINT, STRING ou REAL. <b>Astuce</b> : l'étiquette de type STRING est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.	
:=	Est le symbole d'affectation	
Expression	Représente la nouvelle valeur à affecter à l'étiquette	
	Si l'étiquette est de ce type de données	Utiliser ce type d'expression
	BOOL	BOOL
	SINT INT DINT REAL	Numérique
STRING (Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement).	Type de chaîne, y compris les étiquettes de chaîne et les valeurs littérales de chaîne (Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement).	
;	Termine l'affectation	

L'étiquette conserve la valeur affectée jusqu'à ce qu'une autre affectation change la valeur.

L'expression peut être simple, comme une valeur immédiate ou un autre nom d'étiquette, ou l'expression peut être complexe et comprendre plusieurs opérateurs et fonctions, ou les deux. Pour plus d'informations, reportez-vous à la section Expressions.

**Astuce** : Les données de module d'E/S sont mises à jour de façon asynchrone à l'exécution de la logique. Si vous faites référence plusieurs fois à une entrée dans votre logique, l'entrée peut changer d'état entre les différentes références à celle-ci. Si l'entrée doit avoir le même état chaque fois que vous y faites référence, mettez la valeur de l'entrée dans un tampon et faites référence à l'étiquette de ce tampon. Pour obtenir de plus amples informations, consultez [LOGIX 5000 Controllers Common Procedures](#) , publication [1756-PM001](#) .  
Vous pouvez aussi utiliser les paramètres de programme d'entrée et de sortie qui mettent automatiquement les données dans des tampons durant l'exécution d'un automate logix. Consultez [LOGIX 5000 Controllers Program Parameters Programming Manual](#) , publication [1756-PM021](#) .



**Voir aussi**

[Affecte un caractère ASCII à un membre de données chaîne](#) sur la [page 922](#)

[Spécifier une affectation non rémanente](#) sur la [page 921](#)

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

[Valeurs littérales des chaînes de caractères](#) sur la [page 930](#)

**Spécifier une affectation non rémanente**

L'affectation non rémanente est différente de l'affectation régulière décrite ci-dessus car l'étiquette dans une affectation non rémanente est réinitialisée à zéro par l'automate à chaque fois qu'il :

- Entre dans le mode Exécution
- Il quitte l'étape d'un SFC, si vous configurez le SFC pour qu'il se réinitialise automatiquement. Cela s'applique seulement si vous incluez l'affectation dans l'action de l'étape ou si vous utilisez l'action pour appeler un routine de texte structuré en utilisant une instruction JSR.

Une affectation non rémanente a cette syntaxe :

*tag* [:=] *expression* ;

où :

Composant	Description (Description)	
<i>étiquette</i>	Représente l'étiquette qui reçoit la nouvelle valeur ; l'étiquette doit être un BOOL, SINT, INT, DINT, STRING ou REAL. <b>Astuce</b> : l'étiquette de type STRING est applicable uniquement à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.	
[:=]	Est le symbole pour l'affectation non rémanente.	
<i>expression</i>	Représente la nouvelle valeur à affecter à l'étiquette.	
	<b>Si l'étiquette est de ce type de données</b>	<b>Utiliser ce type d'expression</b>
	BOOL	BOOL
	SINT	Numérique
	INT	
	DINT	
REAL		

	STRING (Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement).	Type de chaîne, y compris les étiquettes de chaîne et les valeurs littérales de chaîne Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580(uniquement)
--	--	--

**Voir aussi**

[Affecte un caractère ASCII à un membre de données chaîne](#) sur la [page 922](#)

[Composants du texte structuré : Affectations](#) sur la [page 920](#)

## Affecte un caractère ASCII à un membre de données chaîne

### Affecte un caractère ASCII à un membre de données chaîne

Utilisez l'opérateur d'affectation pour affecter un caractère ASCII à un élément du membre DATA d'une étiquette de chaîne. Pour affecter un caractère, spécifiez la valeur du caractère ou le nom de l'étiquette, le membre DATA et l'élément du caractère. Par exemple :

Ceci est correct	Ceci n'est pas correct
string1.DATA[0] := 65;	string1.DATA[0] := A;
string1.DATA[0]:= string2.DATA[0];	string1 := string2; <b>Astuce</b> : cela permet d'affecter tout le contenu de string2 à string1 plutôt qu'un seul caractère.

Pour ajouter ou insérer une chaîne de caractères à une étiquette de chaîne, utilisez l'une de ces instructions de chaînes ASCII :

à	Utiliser cette instruction
Ajouter des caractères à la fin d'une chaîne	CONCAT
Insérer des caractères dans une chaîne	INSERT

**Voir aussi**

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

[Valeurs littérales des chaînes de caractères](#) sur la [page 930](#)

## Composants du texte structuré : Expressions

Une expression est un nom d'étiquette, une équation, ou une comparaison. Pour écrire une expression, utilisez n'importe lesquels des éléments suivants :

- Les noms d'étiquettes qui stockent des valeurs (variables)
- Les nombres que vous saisissez directement dans l'expression (valeur immédiate)
- Les chaînes littérales que vous saisissez directement dans l'expression (Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580 uniquement)
- Fonctions, telles que : ABS, TRUNC
- Opérateurs, tels que : +, -, <, >, And, Or

Respectez ces recommandations pour l'écriture des expressions :

- Utilisez n'importe quelle combinaison de majuscules et de minuscules. Par exemple, vous pouvez utiliser les variations suivantes de « AND » : AND, And, and.
- Dans le cas de conditions plus complexes, utilisez des parenthèses pour grouper des expressions à l'intérieur d'autres expressions. Cela facilite la lecture de l'expression entière et vous assure que l'expression s'exécutera dans la séquence désirée.

Utilisez les expressions suivantes pour le texte structuré :

**Expression BOOL** : expression qui produit les valeurs BOOL de 1 (vrai) ou de 0 (faux).

- Une expression bool utilisent des étiquettes bool, des opérateurs relationnels et des opérateurs logiques pour comparer les valeurs ou vérifier si des conditions sont vraies ou fausses. Par exemple, tag1>65.
- Une expression bool simple peut être une simple étiquette BOOL.
- Utilisez en général les expressions bool pour rendre conditionnelle l'exécution d'autres logiques.

**Expression numérique** : expression qui calcule une valeur entière ou flottante.

- Une expression numérique utilise les opérateur arithmétique, les fonctions arithmétiques et les opérateurs de bits. Par exemple, tag1+5.
- Imbriquez une expression numérique dans une expression BOOL. Par exemple, (tag1+5)>65.

**Expression au format chaîne** : expression qui représente une chaîne.

- Une expression simple peut être une chaîne littérale ou une étiquette chaîne

Utilisez cette table pour sélectionner les opérateurs des expressions.

Si	Utilisez
Calcul d'une valeur arithmétique	Les opérateurs et les fonctions arithmétiques
Comparaison de deux valeurs ou de deux chaînes	Les opérateurs relationnels
Vérification si ses conditions sont vraies ou fausses	Les opérateurs logiques
Comparaison des bits à l'intérieur de valeurs	Les opérateurs de bits

**Voir aussi**

[Utiliser les opérateurs et les fonctions arithmétiques](#) sur la [page 924](#)

[Utiliser les opérateurs relationnels](#) sur la [page 927](#)

[Utiliser les opérateurs logiques](#) sur la [page 926](#)

[Utiliser les opérateurs de bits](#) sur la [page 925](#)

## Utiliser les opérateurs et les fonctions arithmétiques

Combiner des opérateurs et des fonctions multiples dans les expressions arithmétiques.

Les opérateurs calculent les nouvelles valeurs.

à	Utiliser cet opérateur	Type de données optimal
Ajouter	+	DINT, REAL
Soustraire/mettre au négatif	-	DINT, REAL
Multiplier	*	DINT, REAL
Exposant (x à la puissance y)	**	DINT, REAL
Diviser	/	DINT, REAL
Diviser modulo	MOD	DINT, REAL

Les fonctions effectuent des opérations mathématiques. Spécifiez une constante, une étiquette non booléenne, ou une expression pour la fonction.

Pour	Utiliser cette fonction	Type de données optimal
Valeur absolue	ABS (numeric_expression)	DINT, REAL
Cosinus d'arc	ACOS (numeric_expression)	REAL
Sinus d'arc	ASIN (numeric_expression)	REAL
Tangente d'arc	ATAN (numeric_expression)	REAL

Cosinus	COS (numeric_expression)	REAL
Radians vers degrés	DEG (numeric_expression)	DINT, REAL
Logarithme naturel	LN (numeric_expression)	REAL
Logarithme décimal	LOG (numeric_expression)	REAL
Degrés vers radians	RAD (numeric_expression)	DINT, REAL
Sinus	SIN (numeric_expression)	REAL
Racine carrée	SQRT (numeric_expression)	DINT, REAL
Tangente	TAN (numeric_expression)	REAL
Tronquer	TRUNC (numeric_expression)	DINT, REAL

Le tableau fournit des exemples d'utilisation des opérateurs et des fonctions arithmétiques.

Utilisez ce format	Exemple	
	Dans cette situation	Écrire
<i>value1 operator value2</i>	Si gain_4 et gain_4_adj sont des étiquettes DINT et votre spécification dit : « Ajouter 15 au gain_4 et stocker le résultat dans gain_4_adj »	gain_4_adj := gain_4+15;
<i>operator value1</i>	Si alarme et high_alarm sont des étiquettes DINT et votre spécification dit : « Mettre au négatif high_alarm et stocker le résultat dans alarme. »	alarm:= -high_alarm;
<i>function(numeric_expression)</i>	Si surcourse et overtravel_POS sont des étiquettes DINT et votre spécification dit : « Calculer la valeur absolue de surcourse et stocker le résultat dans overtravel_POS. »	overtravel_POS := ABS(overtravel);
<i>value1 operator (function((value2+value3)/2))</i>	Si réglage et position sont des étiquettes DINT et sensor1 et sensor2 sont des étiquettes REAL, et votre spécification dit : « Trouver la valeur absolue de la moyenne de sensor1 et sensor2, ajouter réglage, et stocker le résultat dans position. »	position := adjustment + ABS((sensor1 + sensor2)/2);

**Voir aussi**

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

**Utiliser les opérateurs de bits**

Les opérateurs de bits manipulent les bits à l'intérieur d'une valeur en fonction de deux autres valeurs.

Les éléments suivants montrent un aperçu général des opérateurs de bits.

Pour	Utiliser cet opérateur	Type de données optimal
ET au niveau du bit	&, AND	DINT
OU au niveau du bit	OU	DINT
OU exclusif au niveau du bit	XOR	DINT
complément au niveau du bit	NOT	DINT

Voici un exemple.

Utilisez ce format	Exemple	
	Dans cette situation	Utilisez
<i>value1 operator value2</i>	Si input1, input2 et result1 sont des étiquettes DINT et votre spécification dit : « Calculer le résultat de bits de input1 et input2. Stocker le résultat dans result1. »	result1 := input1 AND input2;

Voir aussi

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

## Utiliser les opérateurs logiques

Utilisez Les opérateurs logiques pour vérifier si des conditions multiples sont vraies ou fausses. Le résultat d'une opération logique est une valeur BOOL.

Si la comparaison est	Le résultat est
vrai	1
faux	0

Utilisez ces opérateurs logiques.

Pour cette comparaison	Utiliser cet opérateur	Type de données optimal
AND logique	&, AND	BOOL
OR logique	OU	BOOL
OR exclusif logique	XOR	BOOL
complément logique	NOT	BOOL

Le tableau fournit des exemples d'utilisation des opérateurs logiques.

Utilisez ce format	Exemple	
	Dans cette situation	Utilisez
BOOLtag	Si photoeye est une étiquette BOOL et votre spécification dit : « Si photoeye_1 est activé, alors... »	IF photoeye THEN...
NOT BOOLtag	Si photoeye est une étiquette BOOL et votre spécification dit : « Si photoeye est désactivé, alors... »	IF NOT photoeye THEN...
expression1 & expression2	Si photoeye est une étiquette BOOL, temp est une étiquette DINT et votre spécification dit : « Si photoeye est activé et temp est inférieur à 100, alors... »	IF photoeye & (temp<100) THEN...
expression1 OR expression2	Si photoeye est une étiquette BOOL, temp est une étiquette DINT et votre spécification dit : « Si photoeye est activé ou temp est inférieur à 100 alors... »	IF photoeye OR (temp<100) THEN...

expression1 XOR expression2	Si photoeye1 et photoeye2 sont des étiquettes BOOL et votre spécification dit : « Si : photoeye1 est on tant que photoeye2 est off ou photoeye1 est off tant que photoeye2 est on alors... »	IF photoeye1 XOR photoeye2 THEN...
BOOLtag := expression1 & expression2	Si photoeye1 et photoeye2 sont des étiquettes BOOL, open est une étiquette BOOL, et votre spécification dit : « Si photoeye1 et photoeye2 sont tous deux activés, définir open sur vrai »	open := photoeye1 & photoeye2;

**Voir aussi**

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

**Utiliser les opérateurs relationnels**

Les opérateurs relationnels comparent deux valeurs ou chaînes pour fournir un résultat vrai ou faux. Le résultat d’une opération relationnelle est une valeur BOOL.

Si la comparaison est	Le résultat est
Vrai	1
Faux	0

Utilisez ces opérateurs relationnels.

Pour cette comparaison	Utiliser cet opérateur	Type de données optimal
Égal	=	Type DINT, REAL, chaîne
Inférieur à	<	Type DINT, REAL, chaîne
Inférieur à ou égal	<=	Type DINT, REAL, chaîne
Supérieur à	>	Type DINT, REAL, chaîne
Supérieur à ou égal	>=	Type DINT, REAL, chaîne
Différent	<>	Type DINT, REAL, chaîne

Le tableau fournit des exemples d’utilisation des opérateurs relationnels

Utilisez ce format	Exemple	
	Dans cette situation	Écrire
value1 operator value2	Si temp est une étiquette DINT et votre spécification dit : « Si temp est inférieur à 100 alors... »	IF temp<100 THEN...
stringtag1 operator stringtag2	Si bar_code et dest sont des étiquettes de chaîne et votre spécification dit : « Si bar_code est égal à dest, alors... »	IF bar_code=dest THEN...
stringtag1 opérateur 'valeur littérale de chaîne de caractères'	Si bar_code est une étiquette de chaîne et votre spécification dit : « Si bar_code est égal à 'Test PASSED', alors... »	IF bar_code='Test PASSED' THEN...

char1 operator char2 Pour saisir un caractère ASCII directement dans l'expression, saisissez la valeur décimale du caractère.	Si bar_code est une étiquette de chaîne et votre spécification dit : « Si bar_code.DATA[0] est égal à 'A', alors... »	IF bar_code.DATA[0]=65 THEN...
bool_tag := bool_expressions	Si count et length sont des étiquettes DINT, done est une étiquette BOOL, et que votre spécification dit : « Si count est supérieur à ou égal à length, vous avez fini de compter. »	Done := (count >= length);

**Comment les chaînes sont évaluées**

Les valeurs hexadécimales des caractères ASCII déterminent si une chaîne est plus ou moins grande qu'une autre chaîne.

- Lorsque deux chaînes sont triées comme dans un annuaire téléphonique, l'ordre des chaînes détermine celle qui est supérieure.

ASCII Characters	Hex Codes
1ab	\$31\$61\$62
1b	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

- Les chaînes sont égales si leurs caractères correspondent.
- Les caractères sont sensibles à la casse. La lettre majuscule « A » (\$41) n'est pas égale à la lettre minuscule « a » (\$61).

**Voir aussi**

[Composants du texte structuré : Expressions](#) sur la [page 923](#)

**Composants du texte structuré : Instructions**

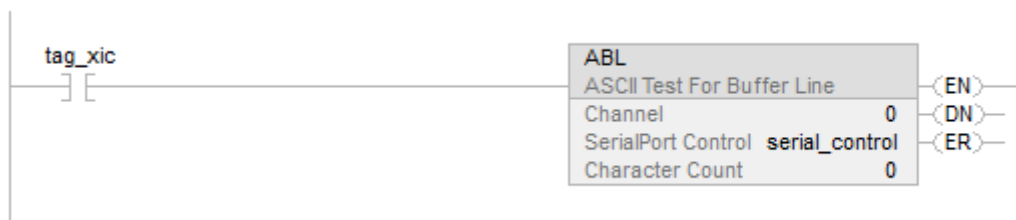
Les lignes de texte structuré peuvent aussi comprendre des instructions. Une instruction dans un texte structuré est exécutée chaque fois qu'elle est scrutée. Une instruction dans un texte structuré à l'intérieur d'une construction est exécutée chaque fois que les conditions de la construction sont vraies. Si les conditions de la construction sont fausses, les instructions à l'intérieur de la construction ne seront pas scrutées. Il n'y a aucune condition d'échelon ou transition d'état qui déclenche l'exécution.



Cela est différent des instructions de bloc fonctionnel qui utilisent EnableIn pour déclencher l'exécution. Les instructions de texte structuré sont exécutées comme si EnableIn était toujours défini.

Cela est aussi différent des instructions de diagramme à relais qui utilisent conditions d'entrée d'échelon pour déclencher l'exécution. Certaines instructions de diagramme à relais ne sont exécutées que si conditions d'entrée d'échelon bascule de faux à vrai. Ce sont les instructions de transition du diagramme à relais. Dans le texte structuré, les instructions s'exécutent chaque fois qu'elles sont scrutées à moins de rencontrer une précondition à l'exécution de l'instruction de texte structuré.

Par exemple, l'instruction ABL est une instruction de transition de diagramme à relais. Dans cet exemple, l'instruction ABL est seulement exécutée durant une scrutation lorsque tag\_xic passe de zéro à défini. L'instruction ABL n'est pas exécutée lorsque tag\_xic reste défini ou reste à zéro.



En texte structuré, si vous écrivez cet exemple ainsi :

```
IF tag_xic THEN ABL(0,serial_control);
END_IF;
```

L'instruction ABL va être exécutée à chaque scrutation où tag\_xic est défini, et non pas seulement lorsque tag\_xic passe de zéro à défini.

Si vous voulez que l'instruction ABL soit exécutée uniquement lorsque tag\_xic passe de zéro à défini, vous devez rendre conditionnelle l'instruction de texte structuré. Utilisez une impulsion pour déclencher l'exécution.

```
osri_1.InputBit := tag_xic;
OSRI(osri_1);
```

```
IF (osri_1.OutputBit) THEN
    ABL(0,serial_control);
END_IF;
```

## Composants du texte structuré : Constructions

Programmez des constructions autonomes ou imbriquées dans d'autres constructions.

Si	Utilisez cette construction
Faire quelque chose si ou quand une condition spécifique se produit	IF. . . ALORS
Sélectionner l'action à effectuer en fonction d'une valeur numérique	CASE. . . OF
Faire quelque chose de spécifique un certain nombre de fois, avant de faire quoi que ce soit d'autre	FOR. . . DO
Continuer à faire quelque chose tant que certaines conditions sont vraies	WHILE. . . DO
Continuer à faire quelque chose jusqu'à ce qu'une condition soit vraie	REPEAT. . . UNTIL

### Certains mots-clés sont réservés

Ces constructions ne sont pas disponibles :

- GOTO
- REPEAT

L'application Logix Designer ne vous laissera pas les utiliser comme des noms d'étiquettes ou des constructions.

### Voir aussi

[IF THEN](#) sur la [page 938](#)

[CASE OF](#) sur la [page 932](#)

[FOR DO](#) sur la [page 935](#)

[WHILE DO](#) sur la [page 943](#)

[REPEAT UNTIL](#) sur la [page 941](#)

## Valeurs littérales des chaînes de caractères

Les valeurs littérales des chaînes de caractères comprennent des caractères encodés sur un simple octet (mono-octet) ou sur deux octets (bi-octet). Une valeur littérale de chaîne mono-octet est une séquence de zéro ou plus de caractères qui ont comme préfixe et suffixe le caractère du guillemet simple (''). Dans les chaînes de caractères mono-octet, la combinaison du trois-caractère du dollar (\$) suivi par deux chiffres hexadécimaux est interprétée comme la représentation hexadécimale du code du caractère sur huit bits, comme montré dans le tableau suivant.

- Astuces :**
- Les valeurs littérales de chaîne de caractères ne s'appliquent qu'à Automates CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 et GuardLogix 5580.
  - Le Studio 5000 ne prend en charge que les caractères mono-octets.

**Valeurs littérales des chaînes de caractères**

Nbre.	Description (Description)	Exemple
1a	Chaîne vide (longueur nulle)	"
1b	Chaîne de longueur un ou caractère CHAR contenant un seul caractère	'A'
1c	Chaîne de longueur un ou caractère CHAR contenant un caractère d'espacement « »	' '
1d	Chaîne de longueur un ou caractère CHAR contenant le caractère « guillemet unique »	'\$'
1e	Chaîne de longueur un ou caractère CHAR contenant le caractère « double guillemet »	''''
1f	Prise en charge de la combinaison de deux caractères	'R\$L'
1g	Prise en charge de la représentation de caractères avec « \$ » et deux caractères hexadécimaux	'\$0A'

**Combinaisons de deux caractères dans les chaînes de caractères**

Nbre.	Description (Description)	Exemple
1	Signe du dollar	\$\$
2	Guillemet simple	'\$'
3	Saut de ligne	\$L ou \$l
4	Retour à la ligne	\$N ou \$n
5	Saut de page	\$P ou \$p
6	Retour chariot	\$R ou \$r
7	Tabulation	\$T ou \$t

- Astuces :**
- Le caractère de retour à la ligne fournit un moyen indépendant de l'implémentation pour définir la fin d'une ligne de données, à la fois pour les E/S physiques et de fichiers. Dans le cas de l'impression, il a comme effet de terminer une ligne de données et de reprendre l'impression au début de la ligne suivante.
  - La combinaison '\$' est uniquement valide à l'intérieur des valeurs littérales de chaîne mono-octet.

**Voir aussi**

[Composants du texte structuré : Affectations](#) sur la [page 920](#)

[Types de chaînes](#) sur la [page 828](#)

## Types de chaînes

Stocker les caractères ASCII dans des étiquettes qui utilisent comme type de données un type de chaînes pour :

- Utiliser le type de données STRING par défaut, qui stocke jusqu'à 82 caractères.
- Créer un nouveau type de chaînes qui stocke un nombre de caractères plus ou moins grand.

Pour créer un nouveau type de chaîne, consultez [LOGIX 5000 Controllers ASCII Strings Programming Manual](#), 1756-PM013.

Chaque type de chaînes contient les membres suivants :

Nom (Name)	Type de donnée	Description (Description)	Remarques
LEN	DINT	nombre de caractères dans la chaîne	<p>LEN est automatiquement mis à jour avec le nouveau nombre de caractères chaque fois que vous utilisez :</p> <ul style="list-style-type: none"> <li>• le navigateur de chaîne pour saisir des caractères</li> <li>• des instructions qui lisent, convertissent et manipulent une chaîne</li> </ul> <p>LEN indique la longueur de la chaîne actuelle. Le membre DATA peut contenir des caractères additionnels et anciens qui ne sont pas inclus dans le nombre LEN.</p>
DATA	Tableau SINT	Caractères ASCII de la chaîne	<p>Pour accéder aux caractères de la chaîne, adressez le nom de l'étiquette. Par exemple, pour accéder aux caractères de l'étiquette string_1, saisissez string_1.</p> <p>Chaque élément du tableau DATA contient un caractère.</p> <p>Créer de nouveaux types de chaînes qui stocke un nombre de caractères plus ou moins grand.</p>

**Voir aussi**

[Valeurs littérales des chaînes de caractères](#) sur la [page 930](#)

## CASE\_OF

Utilisez CASE\_OF pour sélectionner quoi faire en fonction d'une valeur numérique.

### Opérandes

CASE numeric\_expression OF

selector1: statement;

selectorN: statement; ELSE

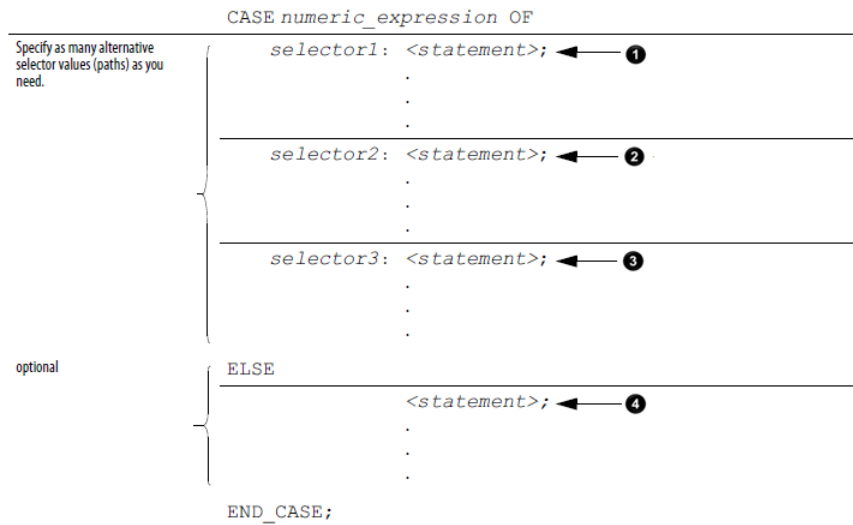
**Texte structuré**

Opérande	Type (Type)	Format	Entrer
Numeric_ expression	SINT INT DINT REAL	Expression d'étiquette	Étiquette ou expression qui est évaluée pour produire un nombre (expression numérique)
Selector	SINT INT DINT REAL	Immédiate	Même type que numeric_expression

**Important :** Si vous utilisez des valeurs REAL, utilisez une plage de valeurs pour un sélecteur car une valeur REAL aura plus de chance d'être à l'intérieur d'une plage de valeurs que de correspondre exactement à une valeur spécifique.

**Description (Description)**

La syntaxe est décrite dans le tableau.



Voilà la syntaxe pour saisir des valeurs de sélecteurs.

Quand le sélecteur est	Entrer
Une valeur	value: statement
Des valeurs multiples et distinctes	value1, value2, valueN : <statement> Utilisez une virgule (,) pour séparer chaque valeur.
Une plage de valeurs	value1..valueN : <statement> Utilisez deux points (..) pour identifier la plage.
Des valeurs distinctes plus une plage de valeurs	valuea, valueb, value1..valueN : <statement>

La construction CASE est similaire à une instruction switch dans les langages de programmation C ou C++. Avec la construction CASE, l'automate exécute seulement les instructions qui associées avec la première valeur de sélecteur correspondante. L'exécution s'arrête toujours après les instructions de ce sélecteur puis passe à l'instruction END\_CASE.

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

None

**Exemple**

Si vous voulez	Saisissez ce texte structuré
If numéro de recette = 1 alors ingrédient A sortie 1 = ouverte (1) Ingrédient B sortie 4 = ouverte (1)	CASE recipe_number OF 1 : Ingrédient_A.Outlet_1 :=1; Ingrédient_B.Outlet_4 :=1;
Si numéro de recette = 2 ou 3 alors Ingrédient A sortie 4 = ouverte (1) Ingrédient B sortie 2 = ouverte (1)	2,3 : Ingrédient_A.Outlet_4 :=1; Ingrédient_B.Outlet_2 :=1;
If numéro de recette = 4, 5, 6 ou 7 alors Ingrédient A sortie 4 = ouverte (1) Ingrédient B sortie 2 = ouverte (1)	4 à 7 : Ingrédient_A.Outlet_4 :=1; Ingrédient_B.Outlet_2 :=1;
If numéro de recette = 8, 11, 12 ou 13 alors Ingrédient A sortie 1 = ouverte (1) Ingrédient B sortie 4 = ouverte (1)	8,11...13: Ingrédient_A.Outlet_1 :=1; Ingrédient_B.Outlet_4 :=1;
Si non toutes les sorties = fermées (0)	ELSE
	Ingrédient_A.Outlet_1 [:=]0; Ingrédient_A.Outlet_4 [:=]0; Ingrédient_B.Outlet_2 [:=]0; Ingrédient_B.Outlet_4 [:=]0;
	END_CASE;

Le [:=] dit à l'automate de mettre les étiquettes de sortie à zéro chaque fois que l'automate fait ce qui suit :

Il entre dans le mode EXÉCUTION.

Il quitte l'étape d'un SFC, si le SFC est configuré pour se réinitialiser automatiquement. Cela applique seulement à l'intégration de l'affectation dans l'action de l'étape ou à l'utilisation de l'action pour appeler une routine de texte structuré via une instruction JSR.

## FOR\_DO

Utilisez la boucle FOR\_DO pour effectuer une action un certain nombre de fois avant de faire quoi que ce soit d'autre.

Quand elle est activée, l'instruction FOR exécute de façon répétée la routine jusqu'à ce que la valeur Index dépasse Terminal value. La valeur Step peut être positive ou négative. Si la valeur est négative, la boucle se termine lorsque Index est inférieur à Terminal value. Si la valeur est positive, la boucle se termine lorsque Index est supérieur à Terminal value.

Chaque fois que l'instruction FOR exécute la routine, elle ajoute Step size à Index.

Ne faites pas trop de boucles lors d'une seule scrutation. Un nombre excessif de répétitions provoque un délai d'expiration du chien de garde d'un automate et entraîne un défaut majeur.

### Opérandes

FOR count:= initial\_value TO

final\_value BY increment DO

<statement>;

END\_FOR;

Opérande	Type (Type)	Format	Description (Description)
count	SINT INT DINT	Étiquette (Tag)	Étiquette pour stocker la position du comptage durant l'exécution de FOR_DO
initial_value	SINT INT DINT	Expression immédiate d'étiquette	Son évaluation doit produire un nombre Spécifiez la valeur initiale du comptage
final_value	SINT INT DINT	Expression immédiate d'étiquette	Spécifier la valeur finale du comptage, qui détermine à quel moment sortir de la boucle
increment	SINT INT DINT	Expression immédiate d'étiquette	(Facultatif) quantité utilisée pour incrémenter le comptage après chaque fin de boucle Si vous ne spécifiez pas un incrément, le compteur est incrémenté de 1.

**Important :** N'exécutez pas la boucle un trop grand nombre de fois lors d'une seule scrutation.  
 L'automate n'exécute d'autre instruction de la routine tant qu'il n'a pas terminé la boucle.  
 Un défaut majeur survient lorsque l'achèvement de la boucle prend plus de temps que le temporisateur du chien de garde pour la tâche.  
 Envisagez d'utiliser une construction différente, telle que IF\_THEN.

**Description (Description)**

La syntaxe est décrite dans le tableau.

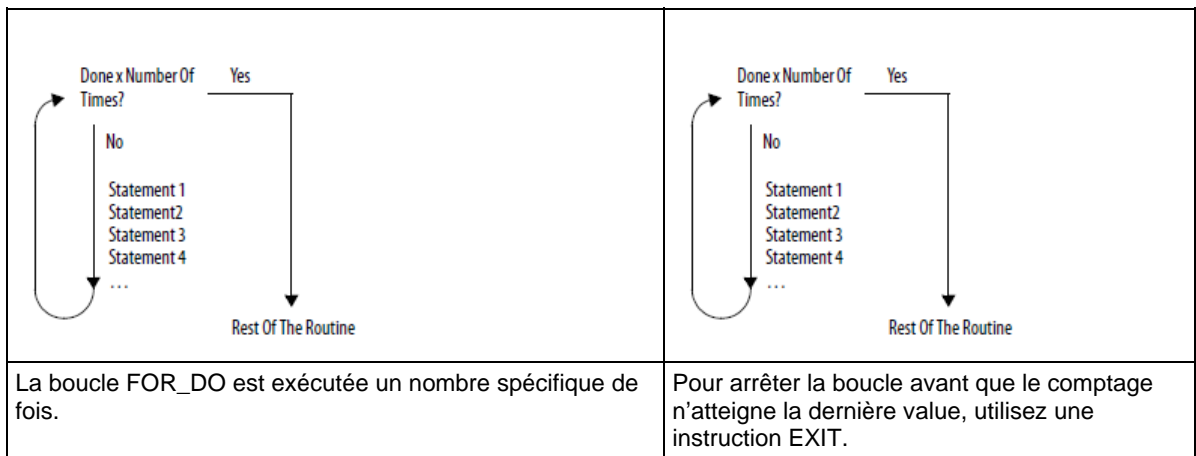
```

FOR count := initial_value
    TO final_value
    optional [ BY increment
DO
    <statement>;
    optional { IF bool_expression THEN
                EXIT;
                END_IF;
END_FOR;
    
```

If you don't specify an increment, the loop increments by 1.

If there are conditions when you want to exit the loop early, use other statements, such as an IF..THEN construct, to condition an EXIT statement.

Le diagramme explique comment s'exécute une boucle FOR\_DO et comment une instruction EXIT peut permettre de sortir plus tôt de la boucle.



**Affecter les indicateurs d'état mathématique**

Non



**Défauts majeurs/mineurs**

Un défaut majeur se produira si	Type de défaut	Code de défaut
La construction fait des boucles pendant trop longtemps.	6	1

**Exemple 1**

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
Mis à zéro les bits 0...31 d'un tableau de BOOL : Initialiser l'étiquette de l'indice à 0. Mettre i à zéro. Par exemple, lorsque l'indice = 5, mettre array[5] à zéro. Ajouter 1 à l'indice. Si l'indice est ≤ 31, répéter 2 et 3. Sinon, arrêter.	For subscript:=0 to 31 by 1 do array[subscript] := 0; End_for;

**Exemple 2**

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
Un type de données défini par l'utilisateur (structure) stocke les informations suivantes au sujet d'un article dans votre inventaire : <ul style="list-style-type: none"> <li>• ID de code-barres de l'article (type de données chaîne)</li> <li>• Quantité pour la quantité en stock de l'article (type de données DINT)</li> </ul> Un tableau de la structure ci-dessus contient un élément pour chaque article différent de votre inventaire. Vous voulez faire une recherche dans le tableau pour trouver un produit spécifique (en utilisant le code-barres) et déterminer la quantité en stock pour cet article. <ol style="list-style-type: none"> <li>1. Déterminer la dimension (nombre des articles) du tableau Inventaire et stocker le résultat dans Inventory_Items (étiquette DINT).</li> </ol> Initialiser l'étiquette de position à 0. <ol style="list-style-type: none"> <li>3. Si Code-barres correspond à l'ID de l'article dans le tableau, alors : Définir l'étiquette Quantity = Inventory[position].Qty. Cela produit la quantité en stock de l'article. Arrêter. Code-barres est une étiquette de la chaîne qui stocke le code-barres de l'article pour lequel vous recherchez. Par exemple, lorsque position = 5, compare Barcode à Inventory[5].ID.</li> <li>4. Ajouter 1 à la position.</li> <li>5. Si la position est ≤ à (Inventory_Items -1), répéter 3 et 4. Puisque le nombre d'éléments commence à 0, le numéro du dernier élément est le nombre d'éléments dans le tableau moins un.</li> </ol> Sinon, arrêter.	SIZE(Inventory,0,Inventory_Items); For position:=0 to Inventory_Items - 1 do If Barcode = Inventory[position].ID then Quantity := Inventory[position].Qty; Exit; End_if; End_for;

# IF\_THEN

Utilisez IF\_THEN pour faire quelque chose quand une condition spécifique se produit.

## Opérandes

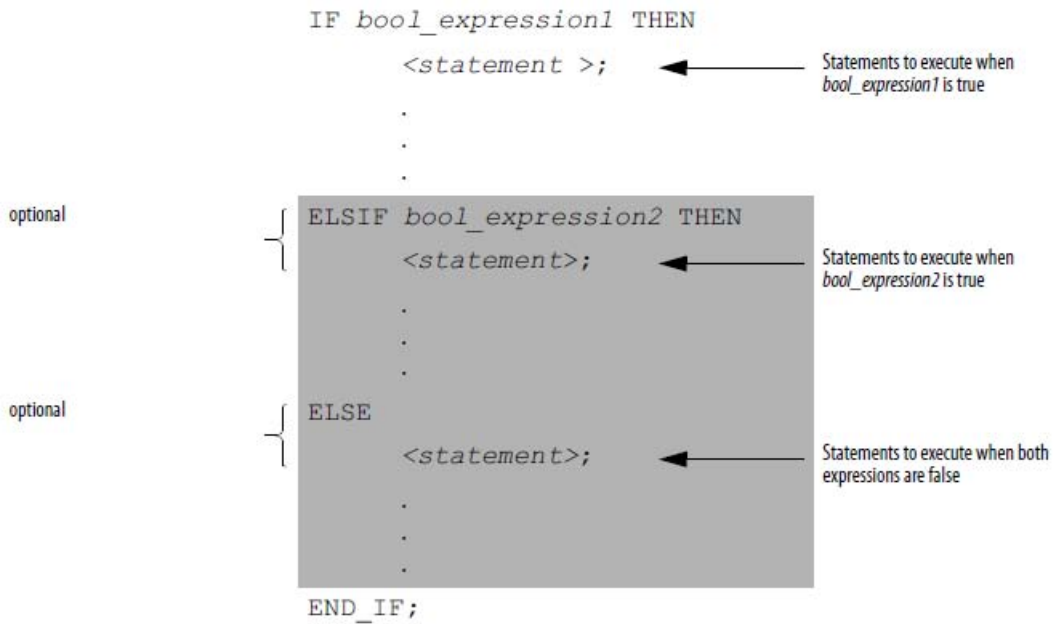
IF bool\_expression THEN

<statement>;

Opérande	Type (Type)	Format	Entrer
Bool_expression	BOOL	Expression d'étiquette	Étiquette ou expression BOOL qui est évaluée pour donner une valeur BOOL (Expression BOOL)

## Description (Description)

La syntaxe est décrite dans le tableau.



Pour utiliser ELSIF ou ELSE, observez ces recommandations.

Pour choisir à partir de plusieurs groupes d'instructions possibles, ajoutez une ou plusieurs instructions ELSIF.

Chaque ELSIF représente un chemin alternatif.

Spécifiez autant de chemins ELSIF que nécessaire.

L'automate exécute le premier IF ou ELSIF vrai et saute le reste des ELSIF et le ELSE.

Pour effectuer une action quand toutes les conditions IF ou ELSIF sont fausses, ajoutez une instruction ELSE.

Le tableau résume les différentes combinaisons de IF, THEN, ELSIF et ELSE.

Si	et	Utilisez cette construction
Faire quelque chose si ou quand les conditions sont vraies	Ne rien faire si les conditions sont fausses	IF_THEN
	Faire quelque chose d'autre si les conditions sont fausses	IF_THEN_ELSE
Choix d'instructions alternatives ou groupes d'instructions en fonction des conditions d'entrées	Ne rien faire si les conditions sont fausses	IF_THEN_ELSIF
	Affecter des instructions par défaut si toutes les conditions sont fausses	IF_THEN_ELSIF_ELSE

**Affecter les indicateurs d'état mathématique**

Non

**Défauts majeurs/mineurs**

Aucun.

**Exemples**

**Exemple 1**

IF...THEN

Pour exécuter ceci	Saisissez ce texte structuré
Si rejets > 3 alors	IF rejets > 3 THEN
convoyeur = off (0)	convoyeur := 0;
alarme = on (1)	alarm := 1;
	END_IF;

**Exemple 2**

IF\_THEN\_ELSE

Pour exécuter ceci	Saisissez ce texte structuré
Si direction du contact du convoyeur = avant (1) alors	IF convoyeur_direction THEN
lumière = off	lumière := 0;
Sinon lumière = on	ELSE
	light [:=] 1;
	END_IF;

Le [=] dit à l'automate de mettre lumière à zéro chaque fois que l'automate fait ce qui suit :

Il entre dans le mode EXÉCUTION.

Il quitte l'étape d'un SFC, si vous configurez le SFC pour qu'il se réinitialise automatiquement. (Cela s'applique seulement si vous incluez l'affectation dans l'action de l'étape ou si vous utilisez l'action pour appeler un routine de texte structuré via une instruction JSR.)

**Exemple 3**

IF...THEN...ELSIF

Pour exécuter ceci	Saisissez ce texte structuré
Si l'interrupteur de limite basse en sucre = bas (on) et interrupteur de limite haute en sucre = non haut (on) alors	IF Sugar.Low & Sugar.High THEN
vanne d'entrée = ouverte (on)	Sugar.Inlet [=] 1;
Jusqu'à ce que l'interrupteur de limite haute en sucre = haut (off)	ELSIF NOT(Sugar.High) THEN
	Sugar.Inlet := 0;
	END_IF;

Le [=] dit à l'automate de mettre Sugar.Inlet à zéro chaque fois que l'automate fait ce qui suit :

Il entre dans le mode EXÉCUTION.

Il quitte l'étape d'un SFC, si vous configurez le SFC pour qu'il se réinitialise automatiquement. (Cela s'applique seulement si vous incluez l'affectation dans l'action de l'étape ou si vous utilisez l'action pour appeler un routine de texte structuré via une instruction JSR.)

**Exemple 4**

IF...THEN...ELSIF...ELSE

Pour exécuter ceci	Saisissez ce texte structuré
Si la température du réservoir > 100	IF tank.temp > 200 THEN
alors pompe = lente	pump.fast :=1; pump.slow :=0; pump.off :=0;
Si la température du réservoir > 200	ELSIF tank.temp > 100 THEN
alors pompe = rapide	pump.fast :=0; pump.slow :=1; pump.off :=0;
Sinon pompe = off	ELSE
	pump.fast :=0; pump.slow :=0; pump.off :=1;
	END_IF;

# REPEAT\_UNTIL

Utilisez la boucle REPEAT\_UNTIL pour continuer à effectuer une action tant que certaines conditions sont vraies.

## Opérandes

REPEAT

<statement>;

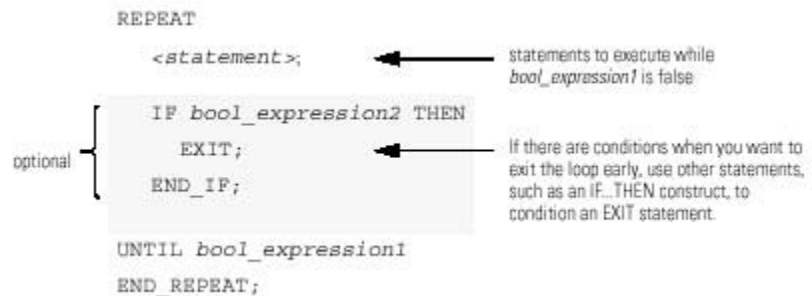
## Texte structuré

Opérande	Type (Type)	Format	Entrer
bool_ expression	BOOL	Expression d'étiquette	Étiquette ou expression BOOL qui est évaluée pour donner une valeur BOOL (Expression BOOL)

**Important :** N'exécutez pas la boucle un trop grand nombre de fois lors d'une seule scrutation.  
L'automate n'exécute d'autre instruction de la routine tant qu'il n'a pas terminé la boucle.  
Un défaut majeur survient lorsque l'achèvement de la boucle prend plus de temps que le temporisateur du chien de garde pour la tâche.  
Envisagez d'utiliser une construction différente, telle que IF\_THEN.

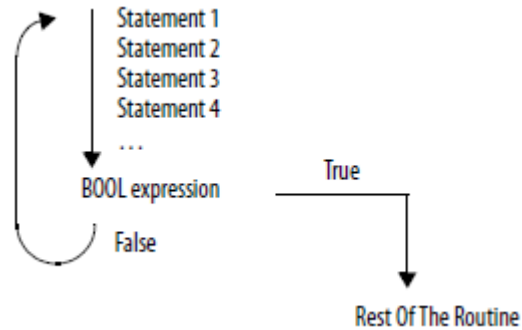
## Description (Description)

La syntaxe est :

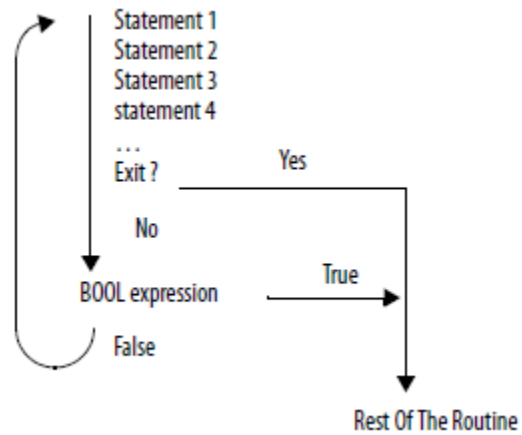


Le diagramme suivant explique comment une boucle REPEAT\_UNTIL est exécutée et comment une instruction EXIT peut permettre de sortir de la boucle plus tôt.

Tant que bool\_expression est faux, l'automate exécute uniquement les instructions à l'intérieur de la boucle REPEAT\_UNTIL.



Pour arrêter la boucle avant que les conditions soient fausse, utilisez une instruction EXIT.



**Affecter les indicateurs d'état mathématique**

Non

**Conditions de défaut**

Un défaut majeur se produira si	Type de défaut	Code de défaut
La construction fait des boucles pendant trop longtemps	6	1

**Exemple 1**

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
La boucle REPEAT_UNTIL exécute d'abord les instructions de la construction, et ce n'est qu'ensuite qu'elle détermine si les conditions sont vraies avant d'exécuter à nouveau les instructions. Cela diffère de la boucle WHILE_DO parce que la boucle WHILE_DO The WHILE_DO évalue d'abord ses conditions.	pos := -1;
Si les conditions sont vraies, l'automate exécute alors les instructions à l'intérieur de la boucle. Les instructions dans la boucle REPEAT_UNTIL sont toujours exécutées au moins une fois. Les instructions dans une boucle WHILE_DO peuvent ne jamais être exécutées.	REPEAT
	pos := pos + 2;
	UNTIL ((pos = 101) OR (structarray[pos].value = targetvalue))
	end_repeat;

**Exemple 2**

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
Déplacer les caractères ASCII d'un tableau SINT dans une étiquette de la chaîne. (Dans un tableau SINT, chaque élément contient un seul caractère.) Arrêter quand vous atteignez le retour chariot.	element_number := 0;
Initialiser Element_number à 0.	SIZE(SINT_array, 0, SINT_array_size);
Compter le nombre d'éléments dans SINT_array (tableau contenant les caractères ASCII) et stocker le résultat dans SINT_array_size (étiquette DINT).	Repeat
Définir String_tag[element_number] = le caractère stocké dans SINT_array[element_number].	String_tag.DATA[element_number] := SINT_array[element_number];
Ajouter 1 à element_number. Cela permet à l'automate de vérifier le prochain caractère dans SINT_array.	element_number := element_number + 1;
Définir le membre Longueur de String_tag = element_number. (Cela enregistre le nombre de caractères stockés dans String_tag jusqu'à présent.)	String_tag.LEN := element_number;
If element_number = SINT_array_size, alors arrêter. (Vous êtes à la fin du tableau et il ne contient pas de retour chariot.)	If element_number = SINT_array_size then
Si le caractère SINT_array[element_number] = 13 (valeur décimale du retour chariot), alors arrête.	exit;
	end_if;
	Until SINT_array[element_number] = 13
	end_repeat;

**WHILE\_DO**

Utilisez la boucle WHILE\_DO pour continuer à effectuer une action tant que certaines conditions sont vraies.

**Opérandes**

WHILE bool\_expression DO

<statement>;

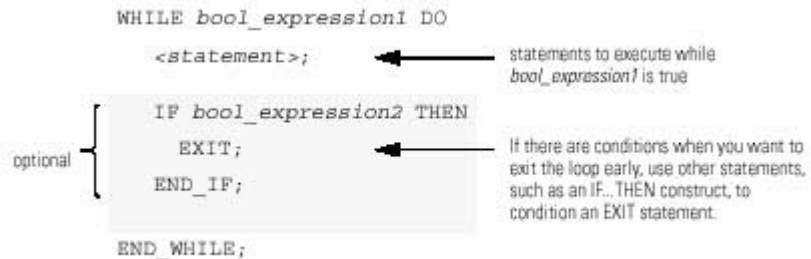
**Texte structuré**

Opérande	Type (Type)	Format	Description (Description)
<i>bool_expression</i>	BOOL	étiquette expression	Étiquette ou expression BOOL qui est évaluée pour donner une valeur BOOL

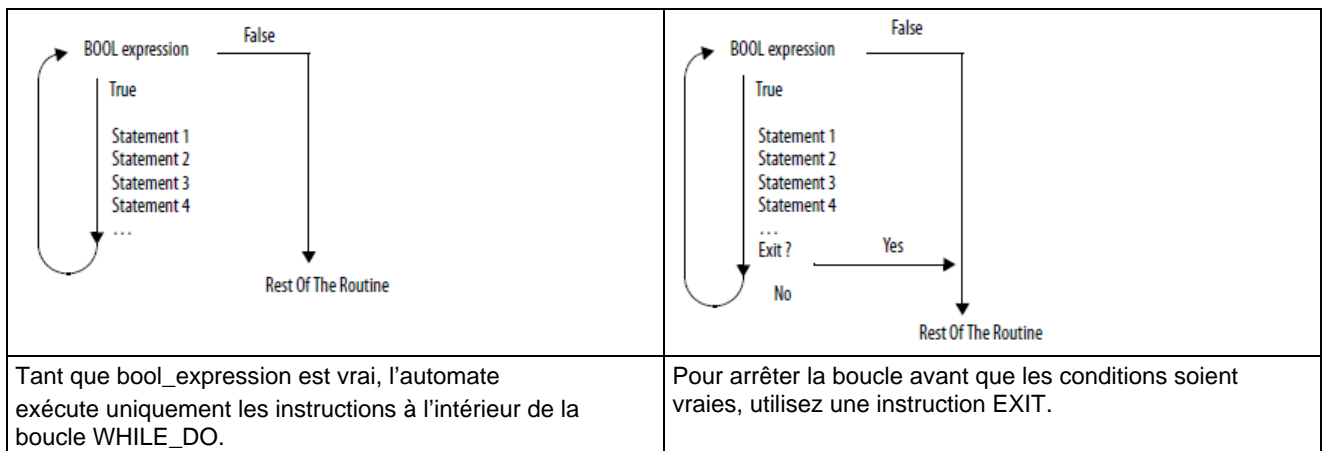
**Important :** N'exécutez pas la boucle un trop grand nombre de fois lors d'une seule scrutation.  
 L'automate n'exécute aucune autre instruction de la routine tant qu'il n'est pas sorti de la boucle.  
 Un défaut majeur survient lorsque l'achèvement de la boucle prend plus de temps que le temporisateur du chien de garde pour la tâche.  
 Envisagez d'utiliser une construction différente, telle que IF\_THEN.

**Description (Description)**

La syntaxe est :



Le diagramme suivant illustre comment s'exécute une boucle WHILE\_DO et comment une instruction EXIT peut permettre de sortir plus tôt de la boucle.





### Affecter les indicateurs d'état mathématique

Non

#### Conditions de défaut

Un défaut majeur se produira si	Type de défaut	Code de défaut
la construction fait des boucles pendant trop longtemps	6	1

#### Exemple 1

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
<p>La boucle WHILE_DO évalue d'abord ses conditions. Si les conditions sont vraies, l'automate exécute alors les instructions à l'intérieur de la boucle.</p> <p>Cela diffère de la boucle REPEAT_UNTIL car la boucle REPEAT_UNTIL exécute d'abord les instructions de la construction, et ce n'est qu'ensuite qu'elle détermine si les conditions sont vraies avant d'exécuter à nouveau les instructions. Les instructions dans la boucle REPEAT_UNTIL sont toujours exécutées au moins une fois. Les instructions dans une boucle WHILE_DO peuvent ne jamais être exécutées.</p>	pos := 0;
	While ((pos <= 100) & structarray[pos].value <> targetvalue)) do
	pos := pos + 2;
	String_tag.DATA[pos] := SINT_array[pos];
	end_while;

#### Exemple 2

Si vous exécutez les actions suivantes,	Saisissez ce texte structuré
<p>Déplacer les caractères ASCII d'un tableau SINT dans une étiquette de la chaîne. (Dans un tableau SINT, chaque élément contient un seul caractère.) Arrêter quand vous atteignez le retour chariot.</p> <p>Initialiser Element_number à 0.</p> <p>Compter le nombre d'éléments dans SINT_array (tableau contenant les caractères ASCII) et stocker le résultat dans SINT_array_size (étiquette DINT).</p> <p>Si le caractère SINT_array[element_number] = 13 (valeur décimale du retour chariot), alors arrête.</p> <p>Définir String_tag[element_number] = le caractère stocké dans SINT_array[element_number].</p> <p>Ajouter 1 à element_number. Cela permet à l'automate de vérifier le prochain caractère dans SINT_array.</p> <p>Définir le membre Longueur de String_tag = element_number. (Cela enregistre le nombre de caractères stockés dans String_tag jusqu'à présent.)</p> <p>If element_number = SINT_array_size, alors arrêter. (Vous êtes à la fin du tableau et il ne contient pas de retour chariot.)</p>	element_number := 0;
	SIZE(SINT_array, 0, SINT_array_size);
	While SINT_array[element_number] <> 13 do
	String_tag.DATA[element_number] := SINT_array[element_number];
	element_number := element_number + 1;
	String_tag.LEN := element_number;
	If element_number = SINT_array_size then
	exit;
	end_if;
	end_while;

## Attributs de texte structuré

Cliquez sur une rubrique ci-dessous pour obtenir plus d'informations sur des problèmes qui sont uniques à la programmation de texte structuré. Revoyez ces informations pour vous assurer de bien comprendre comment votre texte structuré va être exécuté.

### Voir aussi

[Composants de texte structuré : Affectations](#) sur la [page 920](#)

[Composants de texte structuré : Expressions](#) sur la [page 923](#)

[Instructions de texte structuré](#) sur la [page 928](#)

[Composants de texte structuré : Constructions](#) sur la [page 930](#)

[Composants de texte structuré : Commentaires](#) sur la [page 919](#)

## A

ABL 813  
ABS 374  
ACB 791  
ACL 795  
ACS 729  
activation de sortie (OTE) 93  
ADD 380  
addition (ADD) 380  
AFI 626  
AHL 798  
alarme analogique 28  
alarme analogique ALMA – logique à relais 28  
alarme numérique 56  
alarme numérique ALMD – logique à relais 56  
arithmétique et logique des fichiers (FAL) 509  
ASCII 789, 831, 851  
    instructions de chaîne ASCII 831, 851  
    instructions de conversion ASCII 851  
    instructions pour port série ASCII 789  
ASN 733  
AVE 526  
AWA 823  
AWT 817

## B

BAND 460  
BNOT 470  
Booléen 460, 466, 470, 474  
    ET booléen (BAND) 460  
    NON booléen (BNOT) 470  
    OU booléen (BOR) 474  
    OU exclusif booléen (BXOR) 466  
BOR 474  
BTD 435  
BTDT 438  
BXOR 466

## C

case...of 932  
chaîne médiane (MID) 838  
chargement de LIFO (LFL) 593  
CLR 479  
CMP 298

codes d'erreur 172, 176, 178, 829  
    ASCII 829  
    message 172  
comparaison de fichier de bits (FBC) 697  
comparaison masquée égal à (MEQ) 354  
comptage dégressif (CTD) 102  
comptage progressif (CTU) 107  
comptage progressif/dégressif (CTUD) 112  
COP 500  
copie de fichier (COP)\_ copie synchrone de fichier (CPS) 500  
copie de fichier synchrone – CPS 500

## D

DDT 689  
    détection de diagnostic (DDT) 689  
décalage binaire à gauche (BSL) 570  
degrés (DEG) 775  
déverrouillage de sortie (OTU) 97  
différent de (NEQ) 363  
DINT vers chaîne (DTOS) 852  
DIV 391  
division (DIV) 391

## E

Effacer (CLR) 479  
égal à (EQU) 302  
entrée de séquenceur (SQI) 610  
EQU 302  
ET 443  
étiquette (LBL) 633  
EVENT 659  
examiner si fermé (XIC) 74  
examiner si ouvert (XIO) 76

## F

FAL 509  
    organigramme FAL (faux) 509  
    organigramme FAL (vrai) 509  
FBC 697  
    comparaison de fichier de bits (FBC) 697  
FFL 579  
    organigramme FFL (faux) 579  
    organigramme FFL (pré-scrutation) 579  
    organigramme FFL (vrai) 579

FFU 586  
  organigramme FFU (faux) 586  
  organigramme FFU (pré-scrutation) 586  
  organigramme FFU (vrai) 586  
FIFO 579, 586  
  chargement de FIFO (FFL) 579  
  déchargement de FIFO (FFU) 586  
fin temporaire (TND) 657  
FLL 530  
FOR 671

## **G**

GEQ 319  
GSV 190  
GSV/SSV 206, 210, 263  
  exemple de programmation 206  
  objets 210  
  objets de sécurité 263

## **I**

if...then 938  
impulsion (ONS) 78  
impulsion sur front descendant (OSF) 80  
impulsion sur front descendant avec entrée (OSFI)  
83  
impulsion sur front montant (OSR) 86  
impulsion sur front montant avec entrée (OSRI) 90  
inférieur à (LES) 327  
inférieur ou égal à (LEQ) 336  
instruction pas d'opération (NOP) 650  
instructions d'alarme 27  
  alarme analogique 28  
  alarme numérique 56  
instructions de bit 73  
instructions de calcul/math 373  
instructions de chaîne ASCII 831, 832, 835, 838,  
842, 847  
  chaîne médiane (MID) 838  
  concaténation de chaîne (CONCAT) 842  
  effacement de chaîne (DELETE) 847  
  insertion d'une chaîne (INSERT) 835  
  rechercher une chaîne (FIND) 832  
instructions de comparaison 297  
instructions de conversion ASCII 851  
  chaîne vers DINT (STOD) 860  
  chaîne vers REAL (STOR) 863

  DINT vers chaîne (DTOS) 852  
  majuscules (UPPER) 867  
  minuscule (LOWER) 855  
  REAL vers chaîne (RTOS) 858  
instructions de fin/rupture 669  
Instructions Logix 885  
  attributs communs 885  
instructions mouvement/logique 433  
instructions pour port série ASCII 789, 828, 829  
  ajout d'écriture ASCII (AWA) 823  
  caractères ASCII dans le tampon (ACB) 791  
  codes d'erreur 829  
  écriture ASCII (AWT) 817  
  effacement ASCII de la mémoire tampon (ACL)  
  795  
  instructions pour port série ASCII 789  
  lecture ASCII (ARD) 803  
  ligne de lecture ASCII (ARL) 807  
  lignes d'échange ASCII (AHL) 798  
  test ASCII pour la ligne de tampon (ABL) 813  
  types de chaînes 828  
  types de données 828  
instructions spéciales 685

## **J**

JMP 633  
JSR 636  
JXR 630

## **L**

LBL 633  
LEQ 336  
LES 327  
LFL 593  
  organigramme LFL (faux) 593  
  organigramme LFL (pré-scrutation) 593  
  organigramme LFL (vrai) 593  
LFU 600  
  Organigramme LFU – vrai 600  
  organigramme LFU (faux) 600  
  organigramme LFU (pré-scrutation) 600  
LIM 345  
limitation de sortie (PID) 726  
lire la valeur système (GSV) 190  
LOG 754  
  logarithme décimal (LOG) 754

logarithme naturel (LN) 758  
LV 881

## M

MCR 646  
MEQ 354  
message 172  
    codes d'erreur 172  
    codes d'erreurs (.ERR) 172  
MID 838  
minuscule – LOWER 855  
MOD 397  
mode incrémentiel 566, 567  
    organigramme du mode incrémentiel (FSC) 567  
mode numérique 563  
modes de temporisation 908  
mouvement (MOV) 490  
MOV 490  
MSG 152, 162  
    exemples de configuration 162  
MUL 404  
multiplier (MUL) 404  
MVM 481  
MVMT 485

## N

NEG 411  
négation (NEG) 411  
NEQ 363  
NOP 650  
NOT 452

## O

ONS 78  
ordre d'exécution 903  
OSF 80  
OSFI 83  
OSRI 90  
OU 456  
ou au niveau du bit (OR) 456  
ou exclusif au niveau du bit (XOR) 447

## P

pause SFC – SFP 651

permutation d'octet – SWPB 493  
PID 705, 712, 717, 718, 719, 720, 721, 725, 726  
    anticipation ou écart de sortie 720  
    anti-saturation d'intégrale 716  
    boucles en cascade 718  
    contrôler un rapport 719  
    proportionnel, intégral et dérivé (PID) 705  
    redémarrage sans à-coup 717  
    réglage de la plage morte 725  
    temporisation d'instruction 721  
    transfert sans à-coup de manuel à automatique  
        716  
    utilisation de la limitation de sortie 726  
    utilisation des instructions PID 712  
pour...faire 935  
proportionnel, intégral et dérivé – PID 705

## R

racine carrée (SQR) 417  
RAD 779  
radians (RAD) 779  
REAL vers chaîne (RTOS) 858  
recherche et comparaison de fichier (FSC) 533  
rechercher une chaîne (FIND) 832  
remplissage de fichier (FLL) 530  
répartition champ de bits (BTD) 435  
répartition de champ de bits avec cible (BTDT) 438  
repeat\_until 941  
RES 117  
retour (RET) 636, 675  
RTO 120  
RTOR 125  
RTOS 858

## S

saut vers routine externe – JXR 630  
saut vers une étiquette (JMP) 633  
SBR 636  
SIN 744  
sinus (SIN) 744  
sortie de séquenceur (SQO) 618  
sortie immédiate (IOT) 194  
sous-routine (SBR) 636  
soustraction (SUB) 424  
SQI 610  
SQL 614

SQO 618  
SQR 417  
SQRT 417  
SRT 547  
SUB 424  
supérieur à (GRT) 310

## T

taille en éléments (SIZE) 557  
TAN 748  
tangente (TAN) 748  
temporisateur de rémanence activé (RTO) 120  
temporisateur de rémanence activé avec  
réinitialisation (RTOR) 125  
test de limite (LIM) 345  
texte structuré 917, 919, 920, 923, 928, 930, 946  
    affectations 920  
    attributs 946  
    commentaires 919  
    construction 930  
    expressions 923  
    Instructions 928  
    syntaxe de programmation 917  
    syntaxe du texte structuré 917  
TND 657  
TOD 768  
TOF 130  
TOFR 135  
TON 140  
TONR 145

## U

UID 664  
UIE 664

## V

Valeur absolue (ABS) 374  
Valeurs immédiates 888  
verrouillage de sortie (OTL) 95  
verrouillage des données 902

## W

while\_do 943

## X

X à la puissance Y (XPY) 762  
XIC 74  
XIO 76  
XPY 762

## Assistance de Rockwell Automation

Rockwell Automation fournit des informations techniques sur le Web pour vous aider dans l'utilisation de ses produits. Sur la page <http://www.rockwellautomation.com/support>, vous trouverez des notes techniques et d'application, des exemples de code et des liens vers des packs de service logiciels. Vous pouvez également visiter notre Centre d'assistance sur la page <https://rockwellautomation.custhelp.com> pour accéder à des mises à jour de logiciels, des chats et forums d'assistance, des informations techniques et des questions de base, ainsi que pour souscrire aux mises à jour des notifications de produits.

En outre, nous proposons divers programmes d'assistance pour l'installation, la configuration et le dépannage. Pour plus d'informations, contactez votre distributeur ou votre représentant Rockwell Automation local, ou bien rendez-vous sur <http://www.rockwellautomation.com/services/online-phone/>.

## Aide à l'installation

En cas de problème dans les 24 heures suivant l'installation du produit, consultez les informations fournies dans le présent manuel. Vous pouvez également appeler le Service client pour recevoir une aide initiale afin de mettre en service et de faire fonctionner votre produit.

États-Unis ou Canada	1.440.646.3434
À l'extérieur des États-Unis et du Canada	Utilisez le Localisateur international disponible à <a href="http://www.rockwellautomation.com/locations">http://www.rockwellautomation.com/locations</a> ou contactez votre représentant local Rockwell Automation.

## Procédure de retour de nouveau produit

Rockwell Automation teste tous ses produits pour en garantir le parfait fonctionnement à leur départ d'usine. Cependant, si votre produit ne fonctionne pas et doit faire l'objet d'un retour, observez ces procédures.

États-Unis	Contactez votre distributeur. Vous devrez lui fournir le numéro de dossier que le Service client vous aura communiqué (appelez le numéro de téléphone ci-dessus pour en obtenir un), afin de terminer la procédure de retour.
En dehors des États-Unis	Contactez votre représentant local Rockwell Automation pour savoir comment procéder.

## Commentaires sur la documentation

Vos commentaires nous aideront à vous fournir une meilleure documentation. Si vous avez des suggestions sur la façon d'améliorer ce document, remplissez le formulaire de commentaires de la publication [RA-DU002](#).

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400  
[www.rockwellautomation.com](http://www.rockwellautomation.com)

### Siège des activités « Power, Control and Information Solutions »

Amérique : Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 États-Unis, Tél: +1 414.382.2000, Fax : +1 414.382.4444

Europe / Moyen-Orient / Afrique : Rockwell Automation NV, Pegasus Park, De Kleedlaan 12a, 1831 Diegem, Belgique, Tél: +32 2 663 0600, Fax : +32 2 663 0640

Asie Pacifique : Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tél: +852 2887 4788, Fax : +852 2508 1846

Canada : Rockwell Automation, 3043 rue Joseph A. Bombardier, Laval, Québec, H7P 6C5, Tél: +1 (450) 781-5100, Fax: +1 (450) 781-5101, [www.rockwellautomation.com](http://www.rockwellautomation.com)

France : Rockwell Automation SAS – 2, rue René Caudron, Bât. A, F-78960 Voisins-le-Bretonneux, Tél: +33 1 61 08 77 00, Fax : +33 1 30 44 03 09

Suisse : Rockwell Automation AG, Av. des Baumettes 3, 1020 Renens, Tél: 021 631 32 32, Fax: 021 631 32 31, Customer Service Tél: 0848 000 278

Publication Rockwell Automation 1756-RM003T-FR-P - Novembre 2018