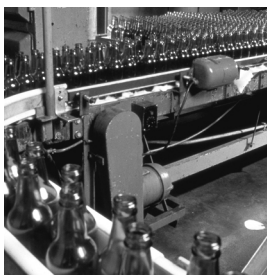


Handbuch zur Anwendungskonvertierung: Simatic S7 nach Logix5000



Anwendungslösung

Wichtige Hinweise für den Anwender

Die Betriebseigenschaften elektronischer Geräte unterscheiden sich von denen elektromechanischer Geräte. In den Sicherheitsrichtlinien für die Anwendung, Installation und Wartung von elektronischen Steuerungen (Publikation SGI-1.1, erhältlich bei Ihrem lokalen Vertriebsbüro von Rockwell Automation oder online unter <http://literature.rockwellautomation.com>) werden einige wichtige Unterschiede zwischen elektronischen Geräten und festverdrahteten elektromechanischen Geräten beschrieben. Aufgrund dieser Unterschiede und der vielfältigen Einsatzbereiche elektronischer Geräte müssen die für die Anwendung dieser Geräte verantwortlichen Personen sicherstellen, dass die Geräte zweckgemäß eingesetzt werden.





Rockwell Automation ist in keinem Fall verantwortlich oder haftbar für indirekte Schäden oder Folgeschäden, die durch den Einsatz oder die Anwendung dieses Geräts entstehen.

Die in diesem Handbuch aufgeführten Beispiele und Abbildungen dienen ausschließlich zur Veranschaulichung. Aufgrund der unterschiedlichen Anforderungen der jeweiligen Anwendung kann Rockwell Automation keine Verantwortung oder Haftung für den tatsächlichen Einsatz der Produkte auf der Grundlage dieser Beispiele und Abbildungen übernehmen.

Rockwell Automation übernimmt keine patentrechtliche Haftung in Bezug auf die Verwendung von Informationen, Schaltkreisen, Geräten oder Software, die in dieser Publikation beschrieben werden.

Die Vervielfältigung des Inhalts dieser Publikation, ganz oder auszugsweise, bedarf der schriftlichen Genehmigung von Rockwell Automation.

In dieser Publikation werden folgende Hinweise verwendet, um Sie auf bestimmte Sicherheitsaspekte aufmerksam zu machen.

WARNUNG 	Dieser Hinweis macht Sie auf Vorgehensweisen und Zustände aufmerksam, die in explosionsgefährdeten Umgebungen zu einer Explosion und damit zu Verletzungen oder Tod, Sachschäden oder wirtschaftlichen Verlusten führen können.
WICHTIG	Dieser Hinweis enthält Informationen, die für den erfolgreichen Einsatz und das Verstehen des Produkts besonders wichtig sind.
ACHTUNG 	Dieser Hinweis macht Sie auf Vorgehensweisen und Zustände aufmerksam, die zu Verletzungen oder Tod, Sachschäden oder wirtschaftlichen Verlusten führen können. Achtungshinweise helfen Ihnen, eine Gefahr zu erkennen, die Gefahr zu vermeiden und die Folgen abzuschätzen.
STROMSCHLAGGEFAHR 	An der Außenseite oder im Inneren des Geräts, z. B. einem Frequenzumrichter oder Motor, kann ein Etikett dieser Art angebracht sein, um Sie darauf hinzuweisen, dass möglicherweise eine gefährliche Spannung anliegt.
VERBRENNUNGSGEFAHR 	An der Außenseite oder im Inneren des Geräts, z. B. ein Frequenzumrichter oder Motor, kann ein Etikett dieser Art angebracht sein, um Sie darauf hinzuweisen, dass die Oberflächen möglicherweise gefährliche Temperaturen aufweisen.

Allen-Bradley, Rockwell Automation und TechConnect sind Marken von Rockwell Automation, Inc.

Marken, die nicht Rockwell Automation gehören, sind Eigentum der jeweiligen Unternehmen.

Vorwort	Zweck	7
	Konvertierung im Gegensatz zu Übersetzung	7
	Terminologie	8
	Weitere Ressourcen	8
	Von Rockwell Automation angebotene SPS-Logik-Migrationsdienste.	9
	Dienstmerkmale.	9
	SPS-Programm-Migrationsdienste aus einer Hand.	9
	Vorteile der Dienste	10
	Angebotene Dienste	10
	Grundkonvertierungspaket.	10
	Paket für Konvertierung und erste Bereinigung.	10
	Zusätzliche Optionen	11
	Weitere verfügbare Programmkonvertierungen	11
	 Kapitel 1	
Hardwarekonvertierung	Einführung	13
	S7-Steuerungen	13
	E/A-Systeme.	14
	Zentrale S7-E/A.	14
	Auswahl und Konfiguration von S7-E/A-Komponenten.	14
	Zentrale Logix-E/A	16
	Auswahl und Konfiguration von Logix-E/A-Komponenten.	18
	S7 Remote I/O	20
	Konfiguration von Remote-S7-I/O mit Profibus DP	21
	Verteilte Logix-E/A.	22
	Konfiguration von dezentralen Logix-E/A	22
	Netzwerke	25
	Netzwerke in S7	25
	Netzwerke in Logix	27
	Konvertierung der Bedienerschnittstelle.	31
	Konvertierung von Systemen mit verteilten Steuerungen	32
	Hardware- und Software-Implementierung	32
	Verbinden von Geräten von Siemens und Rockwell Automation	34
	Steuerungen.	34
	Verteilte Geräte	34
	 Kapitel 2	
Merkmale von Logix, die S7-Benutzern eventuell nicht vertraut sind	Einführung	35
	S7-Bausteine (OB) im Vergleich mit Logix-Tasks	36
	Bausteine in S7	36
	Tasks in Logix.	41
	Task Monitor	46
	Tags anstelle von Adressen	47
	Datenbereiche in S7	47
	Daten in Logix	50
	E/A und Alias-Tags.	51

Programmiersprachen	53
Logix-Kontaktplan	54
Strukturierter Text in Logix	54
Logix-Funktionsblockdiagramm	55
Sequenzielles Funktionsdiagramm in Logix	55
Konvertierung von STEP 7-Code in Logix	55
Datenfelder anstelle von Zeigern	56
Add-On-Befehle	57
Zusammenfassung der Add-On-Befehle	57
Backing-Tags	58
Das Common Industrial Protocol (CIP)	58
Anzeigen des Netzwerks	59
Datenaustausch zwischen Steuerungen	60
Senden/Empfangen in STEP 7	60
Produzierte/konsumierte Tags in Logix	60
Benutzerdefinierte Datentypen	61
Asynchrone E/A-Aktualisierung	62
Der Datentyp DINT	62
Phase Manager	63
Phasenverwaltung in STEP 7	63
PhaseManager in Logix	63
Koordinierte Systemzeit (CST)	65
Eingänge mit Zeitstempel	65
Zyklische Ausgänge	65
Keine temporären Variablen	66
Keine Akkumulatoren oder besonderen Zählwerke erforderlich	66

Kapitel 3

Konvertierung von Systemsoftware und Standardfunktionen

Einführung	67
Logix-Systemfunktionen	68
Kopieren	68
Einstellen und Lesen des Datums und der Uhrzeit	69
Lesen der Systemzeit	69
Verarbeitung von Interrupts	70
Fehler	70
Status – Steuerung	71
Status – Modul	71
Status – für Bausteine und Tasks	72
Timer	72
Konvertierungsroutinen	73
Zeichenketten-Verarbeitungsroutinen	73

	Beispiele für Systemfunktionsaufrufe.....	74
	Einstellen der Uhr	74
	Deaktivieren von Interrupts	76
	Lesen der Systemzeit	78
	Abrufen von Fehlern.....	79
	Modulinformationen	80
	Abrufen der Abtastzeit.....	81
	Kapitel 4	
Konvertierung typischer Programmstrukturen	Einführung	83
	Beispiele für konvertierten Code.....	83
	Übersetzung von Kontaktplanlogik	83
	Sprünge und Entscheidungen.....	90
	Datenfelder	94
	Anwenderdatentypen	99
	Zeiger und Datenfelder	102
	Zustandsautomat	103
	STEP 7-Zustandsautomat.....	104
	Zeichenketten.....	108
	Temporäre Variablen in STEP 7	110
	Funktionen	110
	Blockkopie, COP und CPS.....	114
	Mathematische Ausdrücke	116
	Andere Themen bezüglich der Programmierung	120
	Gültigkeitsbereich von Variablen.....	120
	Bausteine, Tasks und Zeitplanung.....	120
	Ein umfangreicheres Beispiel – Feldgerät	121
	Komponenten des Feldgeräts (CM)	121
	Anwenderdatentyp „Valve“	122
	Der Add-On-Befehl	123
	Lokale Daten des Add-On-Befehls	124
	Aufruf.....	127
	Kapitel 5	
Häufige Fehler beim Konvertieren nach Logix	Einführung	129
	Auswählen nicht geeigneter Hardware	129
	Unterschätzen der Aus-wirkungen von Task-Planung	130
	Durchführen einer Übersetzung anstelle einer Konvertierung.....	130
	Verwenden ungeeigneter Logix-Sprachen.....	130
	Implementierung der falschen Datentypen – DINT gegenüber INT.....	131
	Hinzufügen von DINTs	131
	Hinzufügen von INTs	131
	Zeitmessungsergebnisse	131
	Anwendercode zur Emulation vorhandener Befehle	132
	Anwendercode	132
	COP-Befehl.....	132

Falsche Verwendung von COP, MOV und CPS 133
 Falsche Verwendung von CPT 133
 Handhabung von Zeichenketten auf nicht optimale Weise..... 133
 Übermäßige Verwendung von Jump-Befehlen 133
 Keine Verwendung von Alias-Tags..... 133

Kapitel 6

Glossar: S7 nach Logix

Einführung 135
 Hardware-Terminologie 135
 Software-Terminologie 136

Anhang A

S7 300- und S7 400-Komponenten und die RA-Äquivalente

Einführung 139
 Kompakt-CPU S7 300 140
 Standard-CPU S7 300 140
 Technologie-CPU S7 300 141
 Fail-Safe-CPU S7 300 142
 S7 300-Digitaleingabebaugruppen 142
 S7 300-Digitalausgabebaugruppen 143
 S7 300-Relaisausgabebaugruppen 143
 S7 300-Digitalein/-ausgabebaugruppen 144
 S7 300-Analogeingabebaugruppen 144
 S7 300-Analogausgabebaugruppen 145
 S7 300-Analogein-/ausgabebaugruppen 145
 S7 400-Standardsteuerungen 146
 Redundante und Fail-Safe-Steuerungen 146
 Digitaleingabebaugruppen 147
 Digitalausgabebaugruppen 147
 Analogeingabebaugruppen 147
 Analogausgabebaugruppen 148

Anhang B

Referenztable für die Siemens-HMI

SIMATIC Micro-Panels und Rockwell Automation-Äquivalente ... 149
 SIMATIC-Panels, 7x-Serie, und Rockwell
 Automation-Äquivalente 151
 SIMATIC-Panels, 17x-Serie, und Rockwell
 Automation-Äquivalente 152
 SIMATIC-Panels, 27x-Serie, und Rockwell
 Automation-Äquivalente 154
 SIMATIC-Multi-Panels, 27x-Serie, und Rockwell
 Automation-Äquivalente 156
 SIMATIC-Multi-Panels, 37x-Serie, und Rockwell
 Automation-Äquivalente 158
 Kundendienst von Rockwell Automation 162
 Unterstützung bei der Installation 162
 Rückgabeverfahren bei neuen Produkten 162

Zweck

Dieses Benutzerhandbuch bietet Anwendern und Entwicklern Hilfe, die bereits Steuerungssysteme auf der Basis einer dieser beiden Plattformen verwendet haben:

- Siemens-Steuerung S7
- Rockwell Automation-PAC (programmierbare Automatisierungssteuerung) Logix

Weiterhin sollte Folgendes gelten:

- Sie haben Interesse daran, die PAC-Funktionen zu nutzen, oder befinden sich in einer frühen Phase der Migration von S7 zu Logix.
- Sie verfügen über spezifischen STEP 7-Programmcode, der in effektiven und effizienten RSLogix 5000-Code konvertiert werden soll.

Dieses Handbuch hilft Ihnen dabei, beim Konvertieren des Projekts nach Logix auf bewährte Methoden zurückzugreifen und häufige Fehler zu vermeiden.

Konvertierung im Gegensatz zu Übersetzung

Die Gegenüberstellung der Konvertierung und der Übersetzung wird in diesem Handbuch zur Anwendungskonvertierung mehrfach wiederholt. Eine einfache Übersetzung befasst sich nur mit einer Codezeile und dem Suchen eines Äquivalents in den Logix-Sprachen. Um eine Anwendung optimal zu konvertieren, müssen Sie über die Übersetzung hinausgehen. Beispielsweise kann es von Vorteil sein, eine andere Programmiersprache auszuwählen, unterschiedliche Programmiertechniken zu nutzen oder für die Lösung derselben Aufgabe auf ein anderes Planungsschema zurückzugreifen. Die Konvertierung wird daher im Zusammenhang mit dem Entwurf und den Kenntnissen der Stärken des Logix-Systems auf einer höheren Ebene durchgeführt.

Wenn Sie Anwendungscode konvertieren möchten, müssen Sie das STEP 7-Programm verstehen, bevor Sie mit der Konvertierung beginnen – entweder indem Sie an der Entwicklung beteiligt waren, oder indem Sie die Dokumentation des Programms und des gesteuerten Prozesses gelesen haben. Wenn das Programm oder der Prozess nicht vertraut oder unzureichend dokumentiert ist, gestaltet sich eine ordnungsgemäße Konvertierung schwierig – es handelt sich dann um eine einfache Übersetzung, die wahrscheinlich nicht zufrieden stellend ausfällt. Beispielsweise gibt es in Logix einen globalen Namensraum, während in der Siemens-Umgebung Datenbausteine durch die Anwendersoftware geladen und entladen werden können. Ein Verständnis dieses Unterschieds hilft Ihnen dabei, eine Strategie für die Konvertierung auszuarbeiten.

In einigen Fällen ist die Dokumentation des Prozesses und des Programms so unbefriedigend, dass es in Bezug auf die allgemeinen Kosten und die Dauer des Projekts effizienter sein kann, eine neue Spezifikation aufzusetzen und das Logix-Programm mit minimaler Übersetzung aus dem alten Programm zu beginnen.

Terminologie

STEP 7 ist die Programmiersoftwareumgebung für Siemens SIMATIC S7-Steuerungen. Die Software RSLogix 5000 wird mit den programmierbaren Logix-Automatisierungssteuerungen von Rockwell Automation verwendet. Wir bezeichnen Logix als programmierbare Automatisierungssteuerung, da sie wesentlich mehr Möglichkeiten bietet als eine herkömmliche Mehrzweck-SPS. Sie bietet eine hervorragende Steuerungsplattform für die Steuerung in mehreren Disziplinen mit gemeinsamem Namensraum, koordinierter Systemzeit für wirklich skalierbare Architekturen mit mehreren Prozessoren, benutzerdefinierten Datentypen und vollständiger NetLinx-Konnektivität.

Mit dem Begriff „Logix“ wird auf alle Steuerungen der Serien ControlLogix, CompactLogix, GuardLogix, FlexLogix, DriveLogix oder SoftLogix und die Programmierumgebung RSLogix 5000 Bezug genommen, wobei aus dem Kontext deutlich wird, was gemeint ist.

Weitere Ressourcen

In allen Abschnitten dieses Handbuch zur Anwendungskonvertierung wird auf andere Benutzerhandbücher, Auswahlanleitungen und Dokumente von Rockwell Automation verwiesen, in denen weitere Informationen zu finden sind.

Publikationsnummer	Publikationstitel
1756-SG001	ControlLogix-Auswahlanleitung
1769-SG001	CompactLogix-Auswahlanleitung
1768-UM001	1768 CompactLogix Controllers User Manual
1769-SG002	Compact I/O-Auswahlanleitung
1756-RM094	Erläuterungen zum Aufbau von Logix5000-Steuerungssystemen Referenzhandbuch
1756-PM001	Logix5000 Controllers Common Procedures Programming Manual
1756-RM003	Logix5000 Controllers General Instructions Reference Manual
1734-SG001	POINT I/O Selection Guide
1738-SG001	ArmorPoint I/O Selection Guide
1792-SG001	ArmorBlock MaXum I/O and ArmorBlock I/O Selection Guide
1794-SG002	FLEX I/O and FLEX Ex Selection Guide
NETS-SG001	NetLinx Selection Guide
VIEW-SG001	Visualization Platforms Selection Guide
IA-RM001	Integrated Architecture: Foundations of Modular Programming
6873-SG004	Encompass Program Product Directory
1756-PM010	Logix5000 Controllers Add-On Instructions Programming Manual
1756-RM087	Logix5000 Controllers Execution Time and Memory Use Reference Manual
IASIMP-RM001	IA Recommended Literature Reference Manual

Von Rockwell Automation angebotene SPS-Logik- Migrationsdienste

Rockwell Automation bietet zusätzliche Dienste für die Konvertierung (Migration) von SPS-Logik an.

- Dienstmerkmale
- SPS-Programm-Migrationsdienste aus einer Hand
- Vorteile der Dienste
- Angebotene Dienste
- Grundkonvertierungspaket
- Paket für Konvertierung und erste Bereinigung
- Weitere verfügbare Programmkonvertierungen

Dienstmerkmale

Im Rahmen der Programm-Migrationsdienste werden die Programme für Ihre bisherigen Allen-Bradley-SPS-Systeme oder speicherprogrammierbare Steuerungen anderer Hersteller konvertiert, sodass sie auf speicherprogrammierbaren Logix-Steuerungssystemen oder auf den speicherprogrammierbaren Steuerungen der Serien SLC 500/MicroLogix oder PLC-5 ausgeführt werden können.

Veraltete Produkte sind häufig kostenaufwändig in der Unterstützung und schwierig zu reparieren. Dadurch können längere Ausfallzeiten entstehen, die eine Verringerung der Produktion zur Folge haben. Aus diesem Grund bietet der Kundensupport von Rockwell Automation jetzt Programm-Migrationsdienste an. Durch diese Dienste werden Kosten und Zeitaufwand für die Migration eines veralteten SPS-Systems auf eine unserer aktuellen PAC- oder SPS-Steuerungsplattformfamilien verringert.

SPS-Programm-Migrationsdienste aus einer Hand

Die Migration von veralteten Produkten auf aktuelle Steuerungsplattformen von Allen-Bradley führt zu einer Verbesserung Ihrer Fertigungsprozesse sowie der Systemzuverlässigkeit und Flexibilität, zu einem verbesserten Zugriff auf die Anwendungsverarbeitungsleistung und zu geringeren Anlagenreparaturkosten sowie reduziertem Ersatzteillagerbestand. Mit den Programm-Migrationsdiensten des Kundensupports von Rockwell Automation werden Ihre bestehenden Programme für speicherprogrammierbare Steuerungen fachmännisch und schnell auf neue Steuerungsfamilien konvertiert. Die Techniker des Kundensupports von Rockwell Automation helfen bei der Migration veralteter Allen-Bradley-Anlagen oder bei der Konvertierung Ihrer SPS-Systeme auf Produkte von Rockwell Automation, wobei Ausfallzeiten minimiert und der Betriebserfolg maximiert werden.

Vorteile der Dienste

Am Programm-Konvertierungsverfahren sind Spezialisten für jede Produktplattform beteiligt. Es entstehen keine schwierig auffindbaren Logikprobleme aufgrund von Tippfehlern. In den meisten Fällen wird die gesamte Datentafel reproduziert, ohne dass Daten verloren gehen, und die gesamte Originaldokumentation wird ohne erneutes Eingeben von Kommentaren und Symbolen erhalten. Originalprogramme der Marke Allen-Bradley können in den Formaten der 6200-, APS- oder AI-Serie vorliegen. Die neuen Programme weisen das entsprechende RSLogix-Format auf.

Angebotene Dienste

Es stehen zwei Programmkonvertierungs-Pakete bereit, und projektspezifisch angepasste Dienste können auf Einzelfallbasis zusammengestellt werden.

Grundkonvertierungspaket

- Das Originalprogramm der speicherprogrammierbaren Steuerung wird in das entsprechende Format für ControlLogix, CompactLogix, PLC-5 oder SLC 500/MicroLogix konvertiert.
- Das Paket enthält eine bei der Konvertierung generierte Fehlerliste, die nicht direkt konvertierbare Befehle sowie möglicherweise nicht konvertierte Adressen aufweist, z. B. Zeiger und indirekte Adressierung.
- Das Programm und die Fehlerliste werden dem Kunden übergeben, sodass Debugging und Korrektur manuell erfolgen können.

Paket für Konvertierung und erste Bereinigung

- Das Originalprogramm der speicherprogrammierbaren Steuerung wird in das entsprechende Format für ControlLogix, PLC-5 oder SLC 500/MicroLogix konvertiert.
- Befehls- und Adressierungsfehler werden gegebenenfalls von uns korrigiert und auf die neue Prozessorfamilie konvertiert.
- Das abgeschlossene Programm wird dann dem Kunden übergeben, sodass die endgültige Inbetriebnahme und das Debugging manuell erfolgen können.

Zusätzliche Optionen

Für beide Pakete sind zum Beispiel die folgenden zusätzlichen Optionen vorhanden:

- Telefonsupport auf Anwendungsebene während der Inbetriebnahme und der Debuggingphase im Projekt.
- Beratung zu System-Reengineering, Bedienerschnittstelle, Architektur und Kommunikationsstrategien, um die Steuerungsfähigkeiten der neuen Plattform, die kein Teil der Codeübersetzung sind, optimal ausnutzen zu können. Schulung und Inbetriebnahme vor Ort sind als Wertschöpfungsdienste von der lokalen GSS-Vertretung (Global Sales and Solutions) erhältlich.
- Vollständige, schlüsselfertige Migration oder Aufrüstung sind von der lokalen GSS-/Engineered Systems-Vertretung erhältlich.

Weitere verfügbare Programmkonvertierungen

- PLC-2-Format in das ControlLogix-, CompactLogix-, PLC-5-, SLC500/MicroLogix-Format
- PLC-3-Format in das ControlLogix-, CompactLogix- oder PLC-5-Format
- PLC-5/250-Format in das ControlLogix- oder CompactLogix-Format
- Modicon – Quantum, 984, 584, 380, 381, 480, 485, 780, 785 in das ControlLogix- oder CompactLogix-Format
- Siemens – S-5, S-7 in das ControlLogix- oder CompactLogix-Format
- TI – 520, 520C, 525, 530, 530C, 535, 560, 560/565, 565, 560/560T, 560T, 545, 555, 575 in das ControlLogix- oder CompactLogix-Format
- GE Series 6 in das ControlLogix- oder CompactLogix-Format

Programmkonvertierungen von speicherprogrammierbaren Steuerungen anderer Hersteller in Programme für Steuerungen von Allen-Bradley sind ebenfalls verfügbar. Einzelheiten erfahren Sie beim technischen Support.

Wenn Sie ein Konvertierungsprojekt ansetzen oder mehr über die Programm-Migrationsdienste erfahren möchten, wenden Sie sich an das Vertriebsbüro oder einen autorisierten Distributor von Rockwell Automation in Ihrer Nähe: Schreiben Sie uns eine E-Mail an raprogramconversion@ra.rockwell.com, oder besuchen Sie <http://support.rockwellautomation.com/>, und rufen Sie das KnowledgeBase-Dokument G19154 auf.

WICHTIG

Nutzen Sie die Beratungsdienste für das Reengineering, typischerweise zum Erweitern der Systemfunktionalität, und nicht zum Anpassen der Hardware aufgrund von Veralterung oder aus ähnlichen Gründen. Die Konvertierung und die PCE-Kommentargenerierung vom SLC- in das Logix-Format und vom PLC-5- in das Logix-Format sind in die Software RSLogix 5000 integriert.

Hardwarekonvertierung

Einführung

Dieses Kapitel bietet Richtlinien für Anwender und Techniker, die die richtige Logix-Hardware als Ersatz für vorhandene S7-Anlagen ermitteln möchten.

In dem Kapitel wird die Auswahl von Steuerungen, zentralen E/A, dezentralen E/A, Netzwerken und Bedienerschnittstellen (HMI) beschrieben. Es enthält außerdem einen Abschnitt zur verteilten Steuerungsarchitektur und Hardware-Konvertierungsbeispiele für die gebräuchlichsten S7-Module.

Thema	Seite
S7-Steuerungen	13
E/A-Systeme	14
Netzwerke	25
Konvertierung der Bedienerschnittstelle	31
Konvertierung von Systemen mit verteilten Steuerungen	32
Verbinden von Geräten von Siemens und Rockwell Automation	34

S7-Steuerungen

In dieser Tabelle ist eine Auswahl aktueller Siemens S7-Steuerungen aufgeführt, mit denen eine große Bandbreite von Anwendungen bedient wird.

Auswahl aktueller Siemens S7-Steuerungen

Steuerung	Bestellnummer	Logix-Äquivalent
313C	6ES7 313-5BF03-0AB0	L23 seriell
314C-DP	6ES7 314-6CG03-0AB0	L23 EtherNet/IP, L31
315-2 DP	6ES7 315-2AG10-0AB0	L32E, L32C
317-2 DP	6ES7 317-6TJ10-0AB0	L35CR, L35E
317T-2 DP	6ES7 317-6TJ10-0AB0	L43, L45
319-3 PN/DP	6ES7 318-3EL00-0AB0	L45, L61
414-2	6ES7 414-2XK05-0AB0	L61, L62
414-3	6ES7 414-3XM05-0AB0	L62, L63, L64, L65
414-3 PN/DP	6ES7 414-3EM05-0AB0	

Auswahl aktueller Siemens S7-Steuerungen

315F-2 PN/DP (Sicherheit)	6ES7 315-2FH13-0AB0 6ES7 317-2FK13-0AB0	GuardLogix L61S, L62S, L63S
414-H (Redundant) 417-H	6ES7 414-4HM14-0AB0 6ES7 417-4HT14-0AB0	L61-L65 mit SRM
PCS7 – Nutzt Steuerung 417-4		L3x, L4x, L6x + Software FactoryTalk View, FactoryTalk Batch

Es folgt eine Übersicht über die Eignung einiger der verbreitetsten S7-Steuerungen:

- S7 315-2DP – Kleine bis mittelgroße Maschinen.
- S7 317-2DP – Mittelgroße bis große Maschinen, kleine bis mittelgroße Prozesssteuerungsanwendungen.
- S7 414-2 – Anspruchsvolle Steuerungs- und Prozesssteuerungsanwendungen.
- S7 414-3 – Anspruchsvolle Steuerungs- und große Prozesssteuerungsanwendungen.

Das vollständige Angebot von S7-Steuerungen ist in [Anhang A](#) aufgeführt.

E/A-Systeme

In diesen Abschnitten werden Logix-E/A-Systeme beschrieben, mit denen vorhandene S7-Geräte ersetzt werden können.

Zentrale S7-E/A

Es gibt eine große Auswahl an S7-300- und S7-400-E/A-Modulen. S7-300-Module werden auf Standard-DIN-Schienen montiert und über beiliegende U-Verbinder mit angrenzenden Karten verbunden. S7-400-Module werden im S7-400-Rack montiert.

Auswahl und Konfiguration von S7-E/A-Komponenten

Die nachstehenden Bildschirmfotos stammen vom Programm STEP 7 Hardware Configuration, einem eigenen Programm in der Anwendungssuite STEP 7. In der Software RSLogix 5000 ist diese Funktionalität vollständig integriert, wie in diesem Benutzerhandbuch später ausgeführt wird.

STEP 7 Hardware-Konfigurationsprogramm

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 407 10A	6ES7 407-0KA02-0AA0					
3	CPU 414-3 DP	6ES7 414-3XJ04-0AB0	V4.1	2			
X2	DP				8151*		
X1	MPI/DP			2	8150*		
IF1						8147	
5	CP 443-1(1)	6GK7 443-1EX11-0-E0	V2.5				
7							
8							
9							
10							
11							
12							
13							
14							
15							

6ES7 421-1BL00-0AA0
Digital input module DI32, 24 VDC, grouping 32

Ziehen Sie das ausgewählte Modul in den Bildschirm für die Rack-Konfiguration.

Slot	Module	Order number	Firmware
1	PS 407 4A	6ES7 407-0DA01-0AA0	
2			
3	CPU414-2DP(1)	6ES7 414-2XG03-0AB0	V3.1
X2	DP		
X1	MPI/DP		
4	DI32xDC 24V	6ES7 421-1BL00-0AA0	
5			

Zentrale Logix-E/A

Ein breites Angebot an E/A-Modulen der Serien ControlLogix und CompactLogix ist erhältlich. 1769-E/A ist kostenoptimiert für genau das benötigte Maß an Funktionalität, wie es häufig von OEMs gefordert wird. Die E/A-Familie 1756 bietet umfassende Funktionen und Merkmale für die anspruchsvollsten Anwendungen, wie sie häufig von Endkunden angefragt werden und teilweise zum Erreichen bestimmter Leistungsstufen erforderlich sind.

CompactLogix-Module werden auf Standard-DIN-Schienen montiert, und die elektrischen und mechanischen Verbindungen zu den angrenzenden Modulen werden durch ein spezielles Kupplungssystem gesichert. Entwickler werden das mechanische Kupplungssystem begrüßen – beim S7-300-System werden die Module nur an einer speziellen Schiene und nicht aneinander befestigt (anders als beim elektrischen U-Verbinder).

ControlLogix-Module werden in den 1756-Racks montiert.

- Bei den Steuerungen 1769-L31, 1769-L32C, 1769-L32E und 1768-L43 können im Rack der Steuerung maximal 16 E/A-Module in bis zu 3 Bänken untergebracht werden.
- Bei den Steuerungen 1769-L35CR, 1769-L35E und 1768-L45 können maximal 30 E/A-Module, ebenfalls in 3 Bänken, im Rack der Steuerung untergebracht werden.
- Bei 1756-Steuerungen wird die maximale Anzahl der zentralen E/A-Module durch die Anzahl von Steckplätzen im Rack bestimmt. Möglich sind 4, 7, 10, 13 oder 17.

Auf beiden Plattformen können weitere E/A-Module per CIP-Netzwerk verbunden werden, wobei EtherNet/IP- und ControlNet-Netzwerke eine besonders dichte, nahtlose E/A-Integration bieten.

In dieser Tabelle sind die Logix-Äquivalente für einige verbreitete S7-E/A-Module aufgeführt.

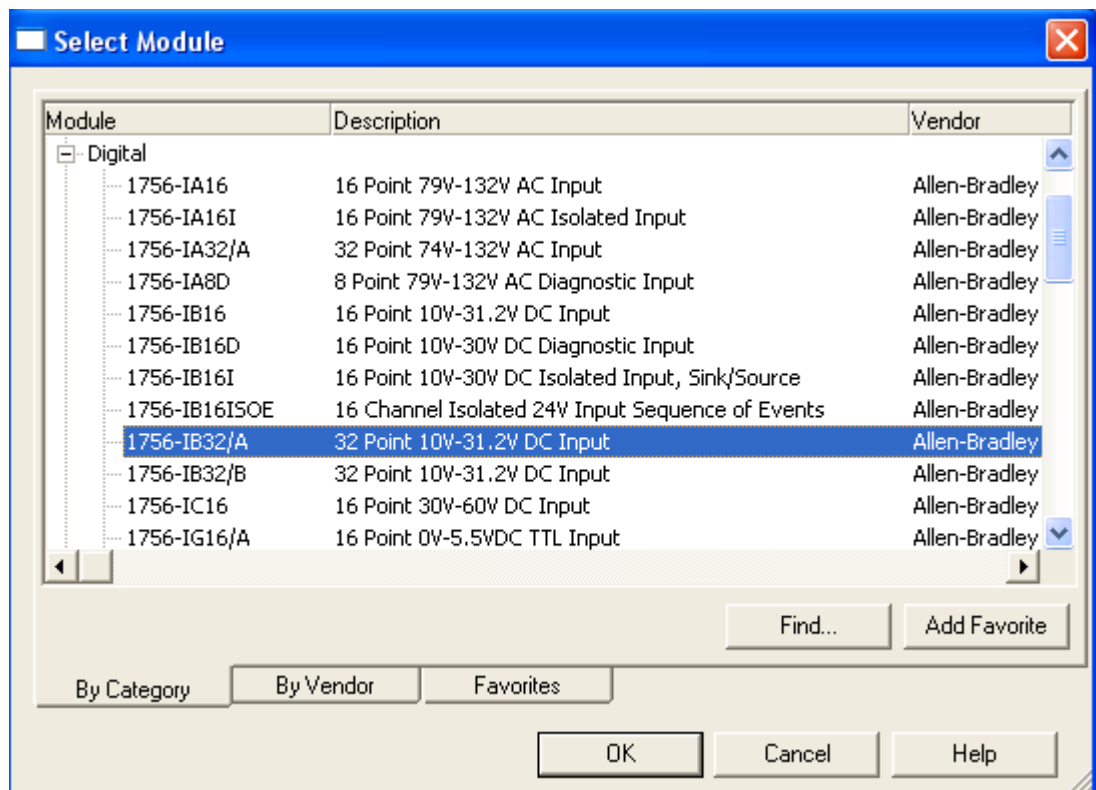
Logix-Äquivalente für S7-E/A-Baugruppen

S7-E/A-Baugruppe	Beschreibung	Logix-Äquivalent	Beschreibung
6ES7 321-1BL00-0AA0	S7-300-Digitaleingang mit 32 Kanälen	1769-IQ32	CompactLogix-Digitaleingang mit 32 Kanälen
6ES7 322 - 1BH01-0AA0	S7-300-Digitalausgang mit 16 Kanälen	1769-OB16	CompactLogix-Digitalausgang mit 16 Kanälen
6ES7 421-1BL01-0AA0	S7-400-Digitaleingang mit 32 Kanälen	1756-IB32	ControlLogix-Digitaleingang mit 32 Kanälen
6ES7 422-1BH01-0AA0	S7-400-Digitalausgang mit 16 Kanälen	1756-OB16E	ControlLogix-Digitalausgang mit 16 Kanälen

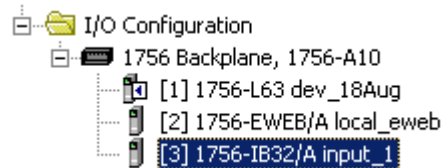
Ausführlichere Konvertierungstabellen für E/A-Module finden Sie in [Anhang A](#).

Auswahl und Konfiguration von Logix-E/A-Komponenten

Über den Zweig „E/A-Konfiguration“ der Projektstruktur kann auf die Logix-Bibliothek für Geräteprofile zugegriffen werden. Diese Profile bieten eine vollständig auf Assistenten basierende Konfiguration für eine vollständige, bedienerfreundliche Integration in die Datentafel und intuitive Programmsteuerung über die Funktionalität jedes Moduls, z. B. Skalieren, Alarmieren und Diagnose.



Sobald Sie ein Element auswählen, wird es im Rack in Ihrer E/A-Konfiguration angezeigt.



Die Geräteprofil-Tags für das neue E/A-Modul wurden automatisch in der Tag-Datenbank für die Steuerung hinzugefügt.

+	Local:3:C		AB:1756_DI:C:0
+	Local:3:I		AB:1756_DI:I:0

In der nachstehenden Ansicht sind die Tags teilweise erweitert.

[-] Local:3:C			AB:1756_DI:C:0
[+] Local:3:C.FilterOffOn_0_7			SINT
[+] Local:3:C.FilterOnOff_0_7			SINT
[+] Local:3:C.FilterOffOn_8_15			SINT
[+] Local:3:C.FilterOnOff_8_15			SINT
[+] Local:3:C.FilterOffOn_16_23			SINT
[+] Local:3:C.FilterOnOff_16_23			SINT
[+] Local:3:C.FilterOffOn_24_31			SINT
[+] Local:3:C.FilterOnOff_24_31			SINT
[+] Local:3:C.COSONOffEn			DINT
[+] Local:3:C.COSONOffEn			DINT
[+] Local:3:I			AB:1756_DI:I:0

Das Profil enthält Konfigurations- und Statusdaten sowie E/A-Daten.

[-] Local:0:C			AB:1756_DI:C:0
[+] Local:0:C.FilterOffOn_0_7			SINT
[+] Local:0:C.FilterOnOff_0_7			SINT
[+] Local:0:C.FilterOffOn_8_15			SINT
[+] Local:0:C.FilterOnOff_8_15			SINT
[+] Local:0:C.FilterOffOn_16_23			SINT
[+] Local:0:C.FilterOnOff_16_23			SINT
[+] Local:0:C.FilterOffOn_24_31			SINT
[+] Local:0:C.FilterOnOff_24_31			SINT
[+] Local:0:C.COSONOffEn			DINT
[+] Local:0:C.COSONOffEn			DINT
[-] Local:0:I			AB:1756_DI:I:0
[+] Local:0:I.Fault			DINT
[+] Local:0:I.Data			DINT
[+] Local:0:O			DINT

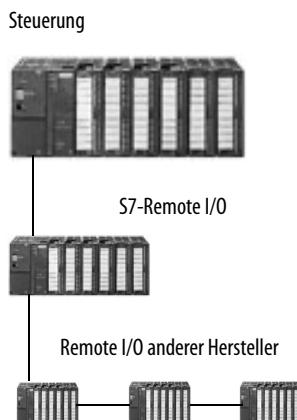
Weitere Informationen finden Sie in [Kapitel 4](#).

S7 Remote I/O

Oft wird bei I/O zwischen dem zentralen Rack der Steuerung und dezentralen I/O-Stationen unterteilt, wobei die Kommunikation über das Profibus DP-Netzwerk abläuft. Folgende Arten von Profibus DP-Netzknoten sind vorhanden:

- S7 Remote I/O. In diesem Fall werden S7-300-Standard-I/O-Module in einem dezentralen I/O-Schaltschrank montiert und über ein Spezialmodul mit dem Profibus DP-Bus verbunden. Die Steuerung nimmt diese I/O als zentrale I/O wahr und weist Standard-I/O-Adressen zu. Dies wird als ET200M bezeichnet.
- Andere Remote-Siemens-I/O wie ET200S (ähnlich dem POINT I/O-System) und ET200L (ähnlich dem FLEX I/O-System).
- Remote I/O anderer Hersteller. Verschiedene Hersteller von I/O-Modulen und Ventilen stellen Schnittstellen her, um ihre Systeme genau wie die dezentralen S7-E/A mit dem Profibus DP-Bus zu verbinden. Für diese Systeme muss eventuell eine spezielle Integrationsdatei (GSD-Datei) in Ihre STEP 7-Installation importiert werden.
- Einige Hersteller komplexerer Geräte wie Waagen oder Frequenzumrichter stellen Profibus DP-Schnittstellen für ihre Produkte her. Für diese Systeme muss in jedem Fall eine spezielle Integrationsdatei (GSD-Datei) in Ihre STEP 7-Installation importiert werden. Um die Bedeutung der Datenbereiche zu erfahren, muss meist in der Dokumentation des Hersteller nachgeschlagen werden.

Typische S7-I/O-Konfiguration



Konfiguration von Remote-S7-I/O mit Profibus DP

Profibus DP-Schnittstellenmodule können in der Hardware-Konfiguration durch Ziehen aus dem Hardware-Katalog in die Grafik des Profibus DP-Busses installiert werden. Nach der Installation kann das Schnittstellenmodul geöffnet werden, und ihm können S7-300-Standardmodule wie bei zentralen I/O hinzugefügt werden.

Slot	Module	Order Number	I Address	Q Address
1				
2	IM 153-1	6ES7 153-1AA03-0AB0	8167	
3				
4	A18x12Bit	6ES7 331-7KF02-0AB0	512...527	
5	A04x12Bit	6ES7 332-5HD01-0AB0		512...519
6	DI16xDC24V	6ES7 321-1BH02-0AA0	0...1	
7	DI8/DO8x24V/0.5A	6ES7 323-1BH00-0AA0	24	0
8				

In der Datentafel sind die dem Antrieb zugewiesenen I/O-Adressen definiert. Symbole für diese Adressen werden in der Symboltabelle manuell hinzugefügt. Die Hardware-Konfiguration ist damit abgeschlossen.

Es ist möglich, Remote-Geräte im Profibus DP-Netzwerk neben Logix zu verwenden, dabei gelten jedoch die gleichen Einschränkungen wie in der S7-Umgebung.

Verteilte Logix-E/A

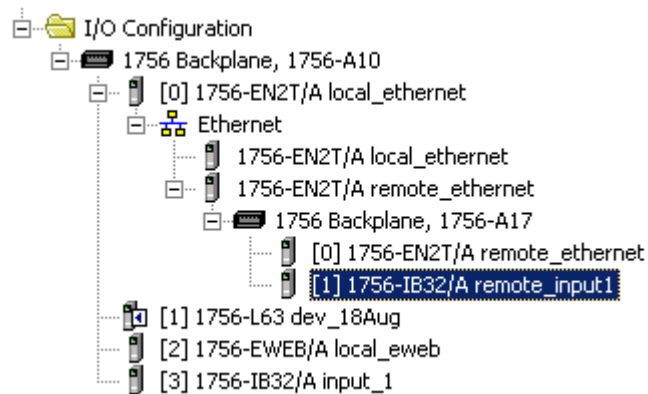
Verteilte E/A von Rockwell Automation umfassen dezentrale E/A mit E/A-Modulen der Serien 1756 oder 1769 sowie zahlreiche verteilte E/A-Plattformen, z. B. POINT I/O-, FLEX I/O-, ArmorPoint- und ArmorBlock-Systeme.

Die E/A-Module sind über ein Kommunikationsmodul oder einen Kommunikationsadapter oder direkt durch eine integrierte Kommunikationsschnittstelle mit dem Netzwerk verbunden.

Konfiguration von dezentralen Logix-E/A

Die gesamte E/A-Konfiguration erfolgt in der Projektstruktur der Software RSLogix 5000. Fügen Sie im Zweig „E/A-Konfiguration“ ein Kommunikationsmodul für den gewählten Netzwerktyp ein.

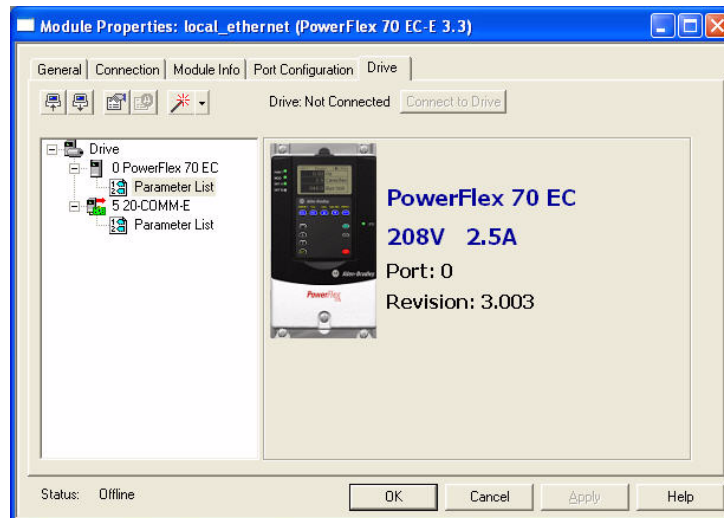
Auf dem Bildschirmfoto ist das Hinzufügen eines dezentralen E/A-Moduls 1756-IB32 zu sehen, das über ein EtherNet/IP-Netzwerk verbunden ist.



Die entsprechenden Tags für das dezentrale E/A-Modul wurden automatisch in der Tag-Datenbank für die Steuerung hinzugefügt.

[-] remote_ethernet:I			AB:1756_ENET_17SLOT:I:0
[+] remote_ethernet:I.SlotStatusBits			DINT
[+] remote_ethernet:I.Slot			AB:1756_ENET_SLOT:I:0[17]

Im Netzwerk vorhandene Frequenzumrichter wie PowerFlex-Frequenzumrichter können auf gleiche Weise hinzugefügt werden.



Auch hierbei generiert die Software RSLogix 5000 die Tags für jedes Gerät automatisch, das in RSLogix 5000 über ein Profil verfügt und über ein EtherNet/IP- oder ein ControlNet-Netzwerk verbunden ist. Beim DeviceNet-Netzwerk werden GuardLogix Safety-E/A ebenso integriert. Andere DeviceNet-Geräte müssen mit der Konfigurationssoftware RSNetWorx und EDS-Dateien eingerichtet werden. Dies entspricht im Wesentlichen der Verwaltungssoftware STEP 7 Profibus und GSD-Dateien.

Nachstehend sind Geräteprofil-Tags in der Software RSLogix 5000 dargestellt, die für Hunderte von Rockwell Automation-Geräten verfügbar sind.

-	PowerFlex_Drive:I	
+	PowerFlex_Drive:I.DriveStatus	
	PowerFlex_Drive:I.DriveStatus_Ready	
	PowerFlex_Drive:I.DriveStatus_Active	
	PowerFlex_Drive:I.DriveStatus_CommandDir	
	PowerFlex_Drive:I.DriveStatus_ActualDir	
	PowerFlex_Drive:I.DriveStatus_Accelerating	
	PowerFlex_Drive:I.DriveStatus_Decelerating	
	PowerFlex_Drive:I.DriveStatus_Alarm	
	PowerFlex_Drive:I.DriveStatus_Faulted	
	PowerFlex_Drive:I.DriveStatus_AtSpeed	
	PowerFlex_Drive:I.DriveStatus_LocalID0	
	PowerFlex_Drive:I.DriveStatus_LocalID1	
	PowerFlex_Drive:I.DriveStatus_LocalID2	
	PowerFlex_Drive:I.DriveStatus_SpdRefID0	
	PowerFlex_Drive:I.DriveStatus_SpdRefID1	
	PowerFlex_Drive:I.DriveStatus_SpdRefID2	
	PowerFlex_Drive:I.DriveStatus_SpdRefID3	
+	PowerFlex_Drive:I.OutputFreq	
-	PowerFlex_Drive:O	
+	PowerFlex_Drive:O.DriveLogicRslt	
	PowerFlex_Drive:O.DriveLogicRslt_Stop	
	PowerFlex_Drive:O.DriveLogicRslt_Start	

Netzwerke

In diesen Abschnitten finden Sie Informationen über die Netzwerke.

Netzwerke in S7

Profibus DP-Netzwerk, DPV1, DPV3

In der S7-Welt ist das hauptsächlich verwendete Netzwerk für die Kommunikation mit Geräten das Profibus DP-Netzwerk in verschiedenen Implementierungen. Einige übergeordnete S7-300-Steuerungen und alle S7-400-Steuerungen weisen integrierte Profibus-Masteranschlüsse auf.

Profibus-Netzwerk – Andere

Profibus FMS und FDL dienen der Datenkommunikation zwischen Steuerungen. Sie erfüllen eine ähnliche Funktion wie das industrielle Ethernet-Netzwerk, und die Konfiguration ist nahezu identisch. Die Unterschiede bestehen darin, dass anstelle von Ethernet-Netzwerkprozessoren solche für das Profibus-Netzwerk sowie Profibus-Kabel verwendet werden.

Profibus DPv2 kann für Verbindungen mit Servoantrieben in den Steuerungen S7-315T und S7-317T für einfache Achssteuerungsanwendungen verwendet werden.

Industrial Ethernet-Netzwerk

Das Industrial Ethernet-Netzwerk von Siemens ist die hauseigene Variante des Ethernet-Netzwerks in einer industriellen Umgebung. Es wird hauptsächlich für die Kommunikation zwischen Steuerungen untereinander und zwischen Steuerungen und Programmiercomputern verwendet.

Abgesehen von einigen aktuellen Steuerungen, die für Profinet ausgestattet sind, weisen S7-Steuerungen keine integrierten Ethernet-Anschlüsse auf. S7-Systeme mit Industrial Ethernet enthalten in den Racks montierte Kommunikationsprozessoren.

Je nach Kommunikationsprozessor können folgende Protokolle verwendet werden:

- S7 (herstellerspezifisches Protokoll für die Kommunikation zwischen S7-Steuerungen)
- TCP (Transmission Control Protocol) Raw Sockets
- ISO-on-TCP (erweitertes TCP mit zusätzlichen Prüfungen)
- UDP (User Datagram Protocol) Raw Sockets

Für die Verwaltung der meisten Kommunikationsaspekte in diesen Netzwerken ist Anwendungscode notwendig.

In der Rockwell Automation-Umgebung kann diese Funktionalität mit integrierten EtherNet/IP-Anschlüssen, EtherNet/IP Bridge-Modulen oder EWEB-Modulen implementiert werden.

Profinet

Profinet bietet ähnliche Funktionalität wie Profibus DP in einem Industrial Ethernet-Netzwerk mit dem gleichen Programmieraufwand. Netzwerke mit Profinet sind ähnlich wie Profibus, nur dass unterschiedliche Kabel und Anschlüsse verwendet werden und anstelle von Profibus-Modulen auf Ethernet-Feldschnittstellenmodule zurückgegriffen wird. Die Verbindung mit dem Netzwerk erfolgt über Steuerungen mit integrierter Profinet-Schnittstelle oder einem für Profibus ausgelegten Kommunikationsprozessor.

Es kann auch ein vorhandenes Profibus DP-Netzwerk per Bridge mit Profinet verbunden werden, entweder über einen Proxy oder unter Verwendung des Profibus DP-Anschlusses an einer für Profinet ausgelegten Steuerung.

Einige Profinet-Feldschnittstellenmodule enthalten mehrere RJ45-Anschlüsse mit einem integriertem Switch, sodass bei Bedarf eine Bus-Topologie wie bei Profibus möglich ist.

Profinet bietet folgende drei Kommunikationsmöglichkeiten:

- Profinet CBA (Component Based Automation), hauptsächlich für die Kommunikation zwischen zwei Steuerungen unter Verwendung von Ethernet-Standard-Hardware und dem TCP/IP Software Stack genutzt.
- Profinet IO für regelmäßige Übertragung, z. B. bei Antrieben oder E/A-Modulen, unter Verwendung von Ethernet-Standard-Hardware und Umgehung des TCP/IP Software Stack.
- Profinet IRT (Isochronous Real Time) für Achssteuerungsanwendungen unter Verwendung von Profinet-spezifischer Hardware und Umgehung des TCP/IP Software Stack, nur in einem geschützten Netzwerksegment möglich.

Wenn das Profinet CBA-Framework verwendet wird, können Profibus-, Profinet- und Industrial Ethernet-Netzwerke per grafischer Konfiguration integriert werden, sodass weniger zusätzliche Programmierung benötigt wird. Rockwell Automation-EtherNet/IP-Netzwerke bieten diese Funktionalität mit Standard-Hardware und dem TCP/IP-Standard-Software Stack, wobei integrierte Funktionen wie der MSG-Befehl (Message) und produzierte/konsumierte Tags genutzt werden.

Netzwerke in Logix

Mit „NetLinx“ wird die Lösung von Rockwell Automation im Bereich der Netzwerktechnologien bezeichnet. Die folgenden Netzwerke sind die hauptsächlich in Logix-Systemen verwendeten:

- Ethernet/IP
- ControlNet
- DeviceNet

Diese Netzwerke weisen einige beachtliche Merkmale auf. Alle sind nach dem CIP-Protokoll (Common Industrial Protocol) entwickelt, sodass Sie über jedes NetLinx-Netzwerk steuern, konfigurieren und Daten erfassen können. Dadurch können Daten zwischen verschiedenen Netzwerken ausgetauscht werden, ohne Protokollübersetzung durch Software oder Proxys.

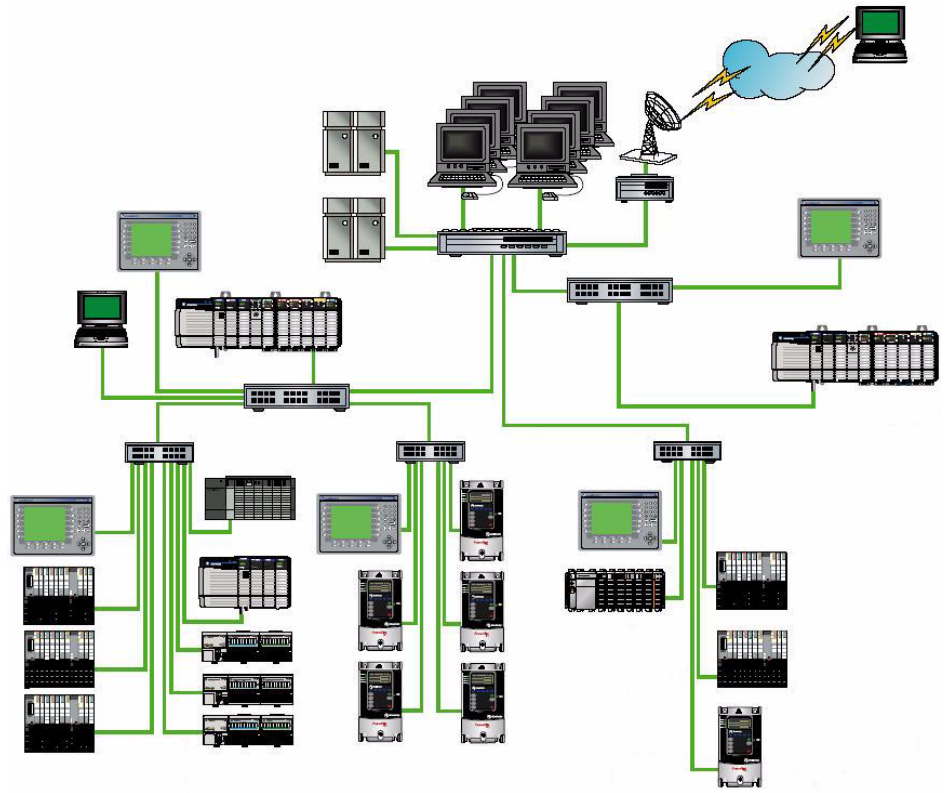
Mit Logix-Systemen vertraute Entwickler lassen sich möglicherweise von der integrierten Natur und der Eleganz bei der Konfiguration von Logix-Netzwerken beeindrucken.

EtherNet/IP-Netzwerk

Das EtherNet/IP-Netzwerk bietet eine vollständige Suite von Steuerungs-, Konfigurations- und Datenerfassungsdiensten. Für den allgemeinen Austausch von Nachrichten und Informationen wird TCP/IP verwendet und für die E/A-Nachrichtenübermittlung UDP/IP. Es wird überwiegend in den folgenden Arten von Konfigurationen verwendet:

- Allgemeine E/A-Steuerung
- Datenaustausch zwischen Steuerungen
- Verbinden vieler Computer
- Verbinden vieler Geräte
- Anschlussmöglichkeiten an Unternehmenssysteme
- Integration von Sicherheitsgeräten
- Achssteuerung (zukünftig)

Typisches Ethernet/IP-Beispiel

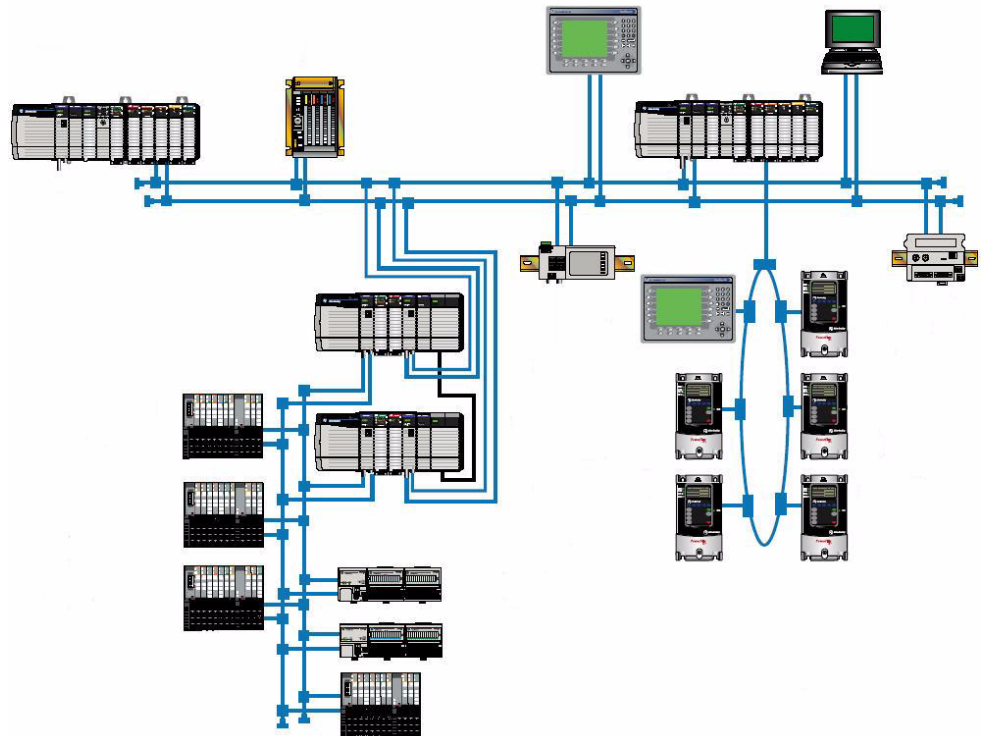


ControlNet-Netzwerk

ControlNet ist ein Echtzeit-Steuerungsnetzwerk, das die Übertragung von zeitkritischen E/A-Daten sowie von Verriegelungs- und Nachrichtenübermittlungsdaten ermöglicht, darunter Hoch-/Herunterladen von Programmier- und Konfigurationsdaten über einen einzigen physikalischen Medienverbund. Es wird überwiegend in den folgenden Arten von Konfigurationen verwendet:

- Allgemeine E/A-Steuerung
- Datenaustausch zwischen Steuerungen
- Backbone für mehrere verteilte DeviceNet-Netzwerke

Typisches ControlNet-Beispiel

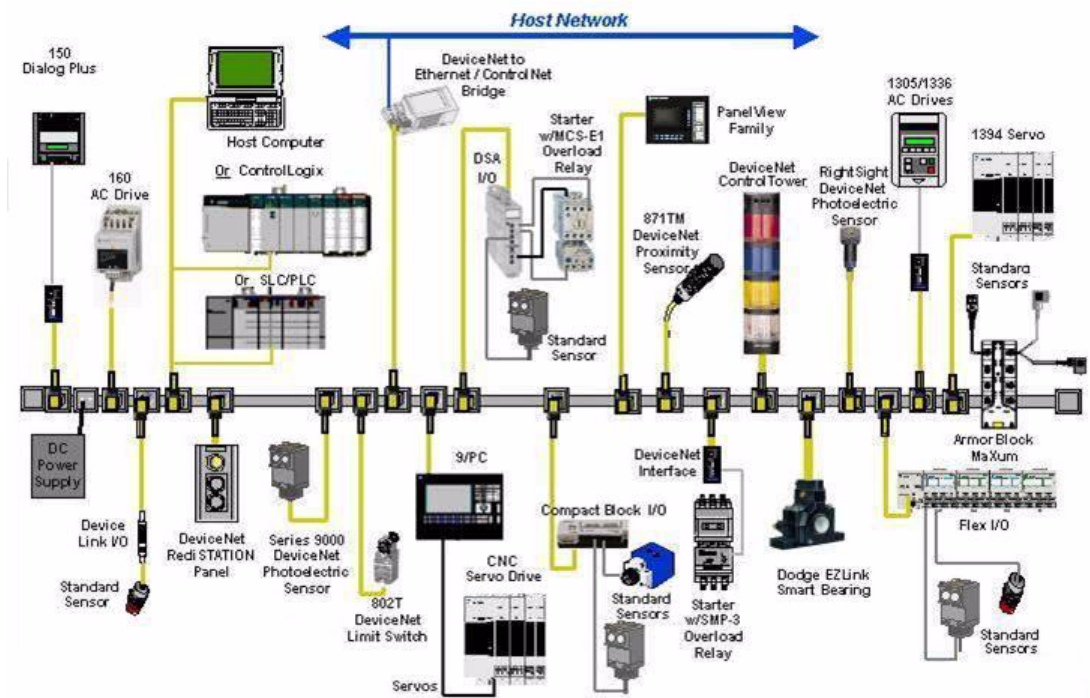


DeviceNet-Netzwerk

Das DeviceNet-Netzwerk ist eine Lösung zur Vernetzung von untergeordneten Industriegeräten. Es ist auf Geräte mit geringem Datenaufkommen pro Gerät für den Echtzeitbetrieb ausgelegt. Es wird überwiegend in den folgenden Arten von Konfigurationen verwendet:

- Anwendungen mit verteilten Geräten mit wenigen Punkten
- Netzwerk aus Antrieben und anderen „einfachen“ Geräten anderer Hersteller
- Systeme, in denen Geräte direkt mit dem Netzwerk verbunden sein und Datenaustausch und Stromversorgung über dieselbe Verbindung stattfinden müssen
- Anforderung nach erweiterten Diagnoseinformationen

Typisches DeviceNet-Beispiel



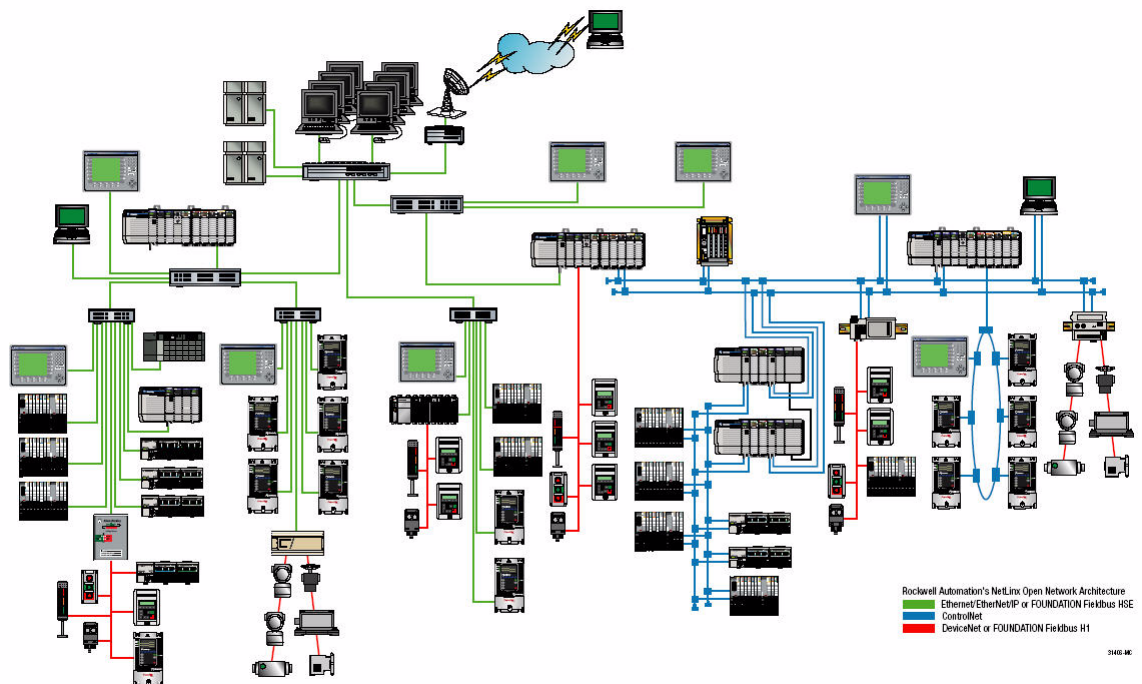
Miteinander verbundene NetLinx-Netzwerke

NetLinx-Netzwerke können auf zwei Arten miteinander verbunden werden.

- Kommunikations-Backplane für mehrere Netzwerkverbindungen auf einmal.
- Kommunikations-Verbindungsgeräte, durch die zwei Netzwerke nahtlos miteinander verbunden werden.

Bei keiner dieser Herangehensweisen sind Steuerungen oder Programmierung erforderlich.

Beispiel eines Steuerungssystems auf der Basis der NetLinx-Netzwerke



Konvertierung der Bedienerschnittstelle

Weitere Informationen finden Sie in [Anhang B](#).

Konvertierung von Systemen mit verteilten Steuerungen

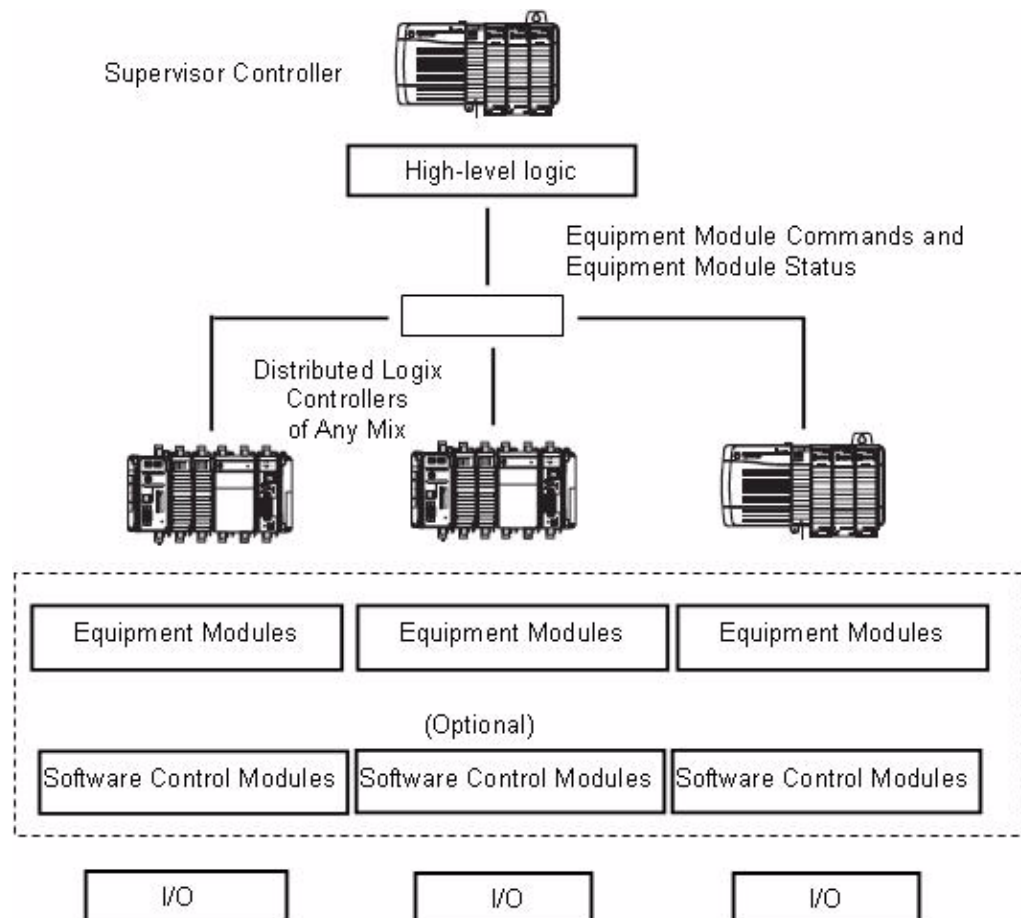
In diesem Abschnitt wird Folgendes behandelt:

- Aufbau einer allgemeinen diskreten Steuerungsanwendung, die eine Gruppe von Funktionseinheiten enthält, mit mehreren Steuerungen.
- Anwendung einer ähnlichen Methode auf eine Prozesssteuerungsanwendung, die gemäß dem S88-Standard entworfen ist.

Hardware- und Software-Implementierung

Allgemeine diskrete Steuerung

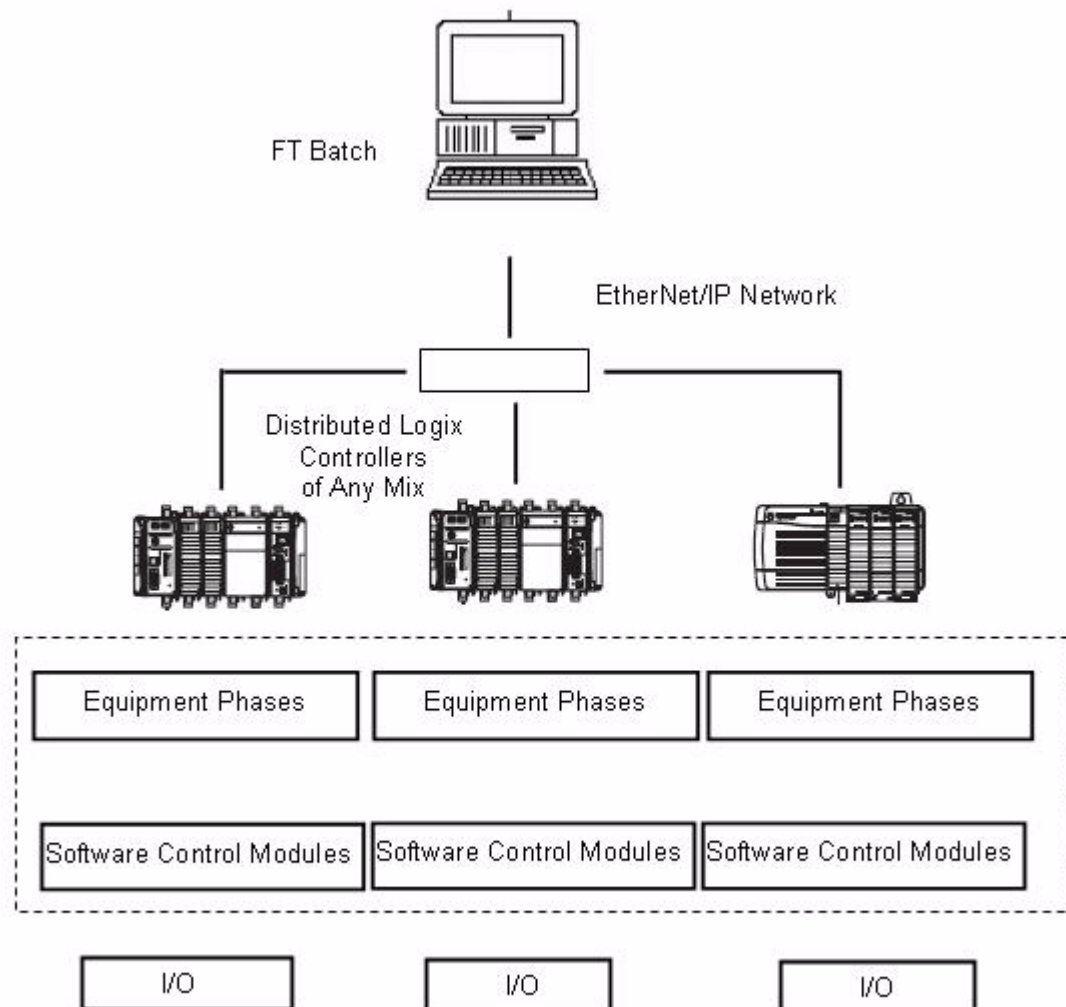
Das Hardware- und Softwaremodell für verteilte Logik zur allgemeinen diskreten Steuerung ist nachstehend dargestellt. In diesem Fall wird die Überwachungsfunktion von einer Steuerung übernommen. Zum Verbinden der Steuerungen untereinander kann ein EtherNet/IP- oder ControlNet-Netzwerk verwendet werden. Produzierte/konsumierte Tags oder explizite Nachrichtenübertragung sind für den Austausch von Daten im System möglich.



Prozesssteuerung

Das nachstehende Diagramm veranschaulicht die Hard- und Softwarestruktur für eine S88-Prozesssteuerungsanwendung. Auf dem PC wird die Software FactoryTalk Batch ausgeführt, mit der Produktionschargen mithilfe von Rezepten umgesetzt werden können. Die Software FactoryTalk Batch befindet sich auf einem PC und kommuniziert über das EtherNet/IP-Netzwerk mit den einzelnen Steuerungen.

Anlagenphasen werden wie später in [Kapitel 2](#) erläutert mit PhaseManager konfiguriert. Sie führen die Phasenlogik aus und kommunizieren über Steuerungsmodule mit den E/A des Steuerungssystems.



Verbinden von Geräten von Siemens und Rockwell Automation

In bestimmten Situationen müssen Sie Geräte von Siemens und Rockwell Automation miteinander verbinden. Wir empfehlen, Produkte von Partnerunternehmen zu verwenden, die im Encompass-Programm eine Gruppe bilden.

Steuerungen

Logix-Steuerungen können folgendermaßen mit S7-Netzwerken verbunden werden:

- über Module im Rack
- über eigenständige Kommunikations-Gateways

Verteilte Geräte

Einige der E/A-Systeme, PowerFlex-Frequenzumrichter und Bedienerchnittstellen-Terminals von Rockwell Automation können über Kommunikationsadapter, integrierte Schnittstellen oder Schnittstellenmodule eine Verbindung zu Profibus herstellen.

Merkmale von Logix, die S7-Benutzern eventuell nicht vertraut sind

Einführung

In diesem Kapitel werden Merkmale von Logix beschrieben, die S7-Benutzern eventuell nicht vertraut sind.

Thema	Seite
S7-Bausteine (OB) im Vergleich mit Logix-Tasks	36
Tags anstelle von Adressen	47
E/A und Alias-Tags	51
Programmiersprachen	53
Add-On-Befehle	57
Das Common Industrial Protocol (CIP)	58
Datenaustausch zwischen Steuerungen	60
Benutzerdefinierte Datentypen	61
Asynchrone E/A-Aktualisierung	62
Der Datentyp DINT	62
Phase Manager	63
Koordinierte Systemzeit (CST)	65
Eingänge mit Zeitstempel	65
Zyklische Ausgänge	65
Keine temporären Variablen	66
Keine Akkumulatoren oder besonderen Zählwerke erforderlich	66

Bestimmte Merkmale des Logix-Systems sind einfacher in der Anwendung und Verwaltung als bei S7 – beispielsweise sind Daten in Tag-Datenbanken ohne absolute Adressen organisiert, während bei S7 die Datenelemente absolute Adresse aufweisen, die vom Programmierer in definierten Speicherbereichen ausgewählt werden.

In anderer Hinsicht ist die Struktur von Logix der von S7 sehr ähnlich, wird allerdings anders dargestellt – zum Beispiel ähnelt die Task-Struktur unter der Oberfläche den Bausteinen von S7.

In diesem Kapitel werden die unterschiedlichen Merkmale gegenübergestellt (z. B. Tags) und die im Wesentlichen ähnlichen Merkmale verglichen (z. B. Tasks).

Dabei sind folgende Ziele gesetzt:

- S7-Benutzer, die zu Logix wechseln, sollen Informationen erhalten, die den Entwicklungsprozess erleichtern und beschleunigen.
- Die Möglichkeiten von Logix sollen gezeigt werden, sodass Entwickler bereits vorhandene Merkmale in der Firmware der Steuerung nicht neu entwickeln.

S7-Bausteine (OB) im Vergleich mit Logix-Tasks

In diesem Vergleich von Bausteinen mit Tasks wird S7-Benutzern die Struktur eines Logix-Programms vorgestellt.

Bausteine und Tasks ähneln einander, da beide vom Betriebssystem der Steuerung aufgerufen werden und nicht vom Anwenderprogramm. In STEP 7 (und Logix) gibt es drei Arten von Bausteinen (Tasks in Logix).

- Programmzyklus-OB (kontinuierliche Organisations-Task in Logix), bei der der OB nach dem Abschluss wieder von vorn beginnt.
- OB für zyklischen Interrupt (periodische Task in Logix), bei dem der OB in einem vorkonfigurierten Zeitraum ausgeführt wird.
- Hardware-Interrupt-OBs (Ereignis-Task in Logix) werden als Reaktion auf einen Hardwareanreiz ausgeführt.

Viele STEP 7-Programmierer greifen nicht auf OBs für zyklischen Interrupt zurück.

Logix bietet ein vom Anwender konfigurierbares Multitasking-Betriebssystem, das Prozessorzeit gemäß den Anforderungen der Anwendung zuordnen kann.

Bausteine in S7

Die Art eines OB ist durch seine Nummer definiert – er kann kontinuierlich (nur OB1), periodisch (OB30–OB38), für Ereignisse (OB40–OB47) oder beim Auftreten bestimmter Fehler ausgeführt werden. Bei Logix werden Tasks nicht nummeriert, sondern durch einen benutzerdefinierten Namen unterschieden.

Ein aussagekräftiger Name kann z. B. einem STEP 7-OB zugeordnet werden.

OB1-Programmzyklus

OB1 wird kontinuierlich erneut ausgeführt. Nach dem Ausführen werden die Werte der Ausgangsdatentafel an die Ausgänge gesendet, die Eingangsdatentafel wird von den Ausgängen aktualisiert, und OB1 beginnt erneut.

Ein STEP 7-Programm muss OB1 nicht einschließen, ist das jedoch der Fall, wird er kontinuierlich ausgeführt.

Typischer Ausschnitt aus OB1:

Network 3 : Title:

```
callup valve and motor control module
```

```
CALL "ValveMotor_Calls"
```

Network 4 : Title:

```
callup switch control module
```

```
CALL "Switch_Calls"
```

Network 5 : Title:

```
callup flow totalisers control module
```

```
CALL "Totaliser_calls"
```

Network 6 : Title:

```
callup analogue input control module
```

```
CALL "AnalogueIn_calls"
```

OB1 stellt den Stamm der Aufrufhierarchie für den gesamten kontinuierlich ausgeführten Code dar.

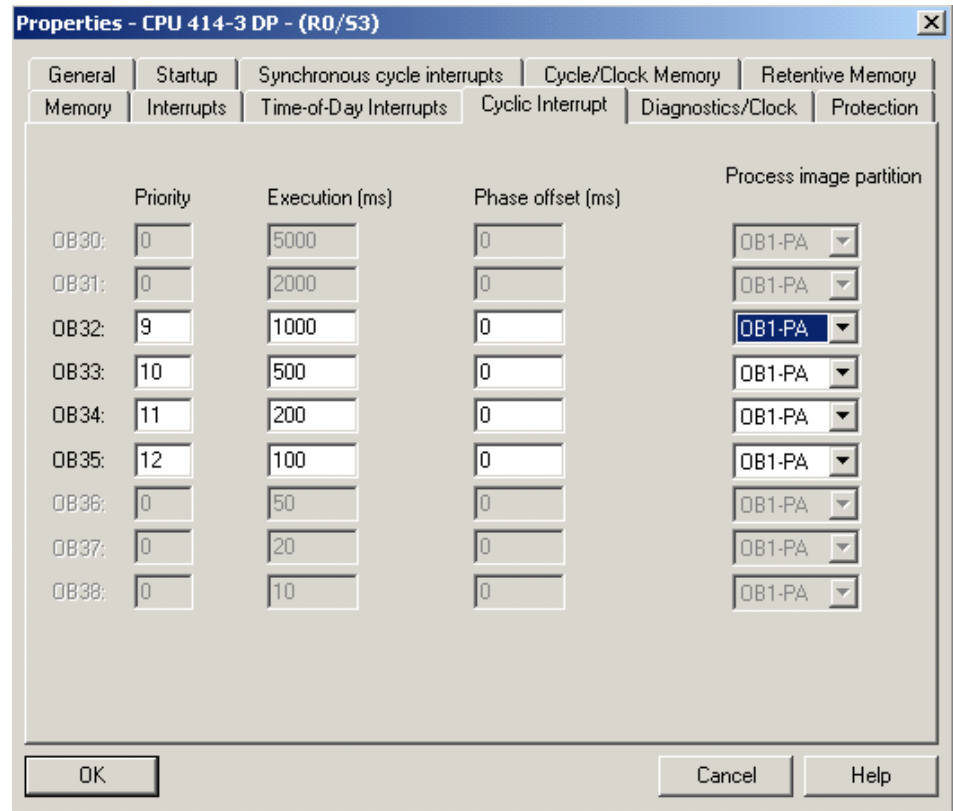
OB1 ähnelt der **kontinuierlichen Task** in Logix, von der selbstverständlich nur eine möglich ist. In der S7-Terminologie ist OB1 ein „Programmzyklus“.

Für Leser, die mit Logix vertrauter sind als mit STEP 7, ist es hilfreich zu wissen, dass in der Kontaktplanlogik von STEP 7 ein Netzwerk einem Logix-Strompfad entspricht. In der Anweisungsliste von STEP 7 sind die Netzwerke ebenfalls enthalten, dienen jedoch nur der besseren Übersichtlichkeit des Codes. Durch sie wird der Code in Abschnitte unterteilt und die Möglichkeit gegeben, Kommentare hinzuzufügen. Auf Wunsch könnte der gesamte Code in einem Netzwerk eingefügt werden – er ließe sich ohne Probleme kompilieren und ausführen.

OB30–OB38: Zyklische Interrupts

Diese OBs werden in festen, konfigurierbaren Intervallen ausgeführt. Auch ihre Priorität kann konfiguriert werden. OBs mit höherer Priorität unterbrechen solche mit niedrigerer Priorität, die gerade ausgeführt werden.

Konfiguration periodisch aufgerufener OBs



Die Anzahl der verfügbaren periodischen OBs ist von der Art der Steuerung abhängig. Eine niedrigere Zahl unter „Priority“ steht für eine höhere Interrupt-Priorität (die Priorität kann nur bei S7 400-Steuerungen ausgewählt werden). „Execution (ms)“ bezeichnet die Ausführungsdauer für den OB. Durch „Phase offset (ms)“ können periodische Interrupts mit Bezug aufeinander ausgelöst werden. Die Auswahl von „Process image partition“ ermöglicht die Unterteilung der E/A-Datentafel, die dann nur noch beim Auftreten des Interrupts aktualisiert wird. (Diese Funktion steht nur bei S7 400-Steuerungen zur Verfügung.) Standard ist die vollständige Tabelle. In Logix gibt es entsprechend die Auswahl der Task-E/A-Aktualisierung und die IOT-Befehle.

Der Inhalt eines periodischen Interrupt-OB ähnelt meist dem Inhalt von OB1. Er besteht aus Aufrufen von Funktionen und Funktionsblöcken, die gemäß der Periodizität des OB ausgeführt werden sollen.

Diese OBs ähneln **periodischen Tasks** in Logix. In der S7-Terminologie werden OB30–OB38 als zyklische Interrupt-OBs bezeichnet.

OB40–OB47: Hardware-Interrupt-OBs

Diese OBs können so konfiguriert werden, dass sie bei Eingangsereignissen ausgelöst werden. Auch ihre Priorität kann konfiguriert werden.

In Logix handelt es sich hierbei um **ereignisgesteuerte Tasks**. In S7 werden OB40–OB47 als Hardware-Interrupts bezeichnet.

Beispielsweise ist das einfachste Hardware-Ereignis, das von einem Hardware-Interrupt-OB (oder einer ereignisgesteuerten Task) behandelt werden kann, eine Zustandsänderung an einem Digitaleingang. Durch einen Hardware-Interrupt (oder eine ereignisgesteuerte Task) wird eine sehr schnelle Reaktion auf die Änderung gewährleistet.

Ereignisgesteuerte Tasks sind flexibler als Hardware-Interrupt-OBs, da die Auslösung nicht nur durch E/A, sondern auch durch Netzwerkereignisse, Programmbefehle und Achssteuerungsereignisse stattfinden kann.

Programmstruktur in STEP 7

Ein typisches Programm enthält Bausteine (OB), Funktionsbausteine (FB), Funktionen (FC) und Datenbausteine (DB). Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC) sind üblicherweise vorhanden.

- Von Bausteinen (Programmzyklus-OBs, zyklische Interrupt-OBs oder beide Arten) werden Funktionsbausteine und Funktionen aufgerufen.
- Ein Funktionsbaustein enthält Code und ist einem Datenbaustein zugeordnet, der die erforderlichen statischen Daten für den FB enthält. Zusätzlich zu den statischen Daten weist der FB temporäre Daten auf. Funktionsbausteine werden verwendet, wenn die Werte der Logik zwischen mehreren Ausführungen beibehalten werden müssen.
- Eine Funktion enthält Code, jedoch keine statischen Daten. Sie weist temporäre Daten auf. FCs werden verwendet, wenn die Logik nach einer einzelnen Ausführung abgeschlossen ist – Werte müssen nicht beibehalten werden.
- Datenbausteine sind Bereiche, in denen statische Daten gespeichert werden. Sie werden im nächsten Abschnitt erläutert.
- Bei SFBs und SFCs handelt es sich um Systemfunktionsbausteine und Systemfunktionen. Sie können aus bei der Installation von STEP 7 bereitgestellten Bibliotheken kopiert und in ein Projekt eingefügt werden.
- Anschließend können sie von beliebigen Stellen im Programm aufgerufen werden.

In STEP 7 gibt es keine Struktur, die den Programmen/Routinen von Logix gleicht. Der OB bildet den Stamm der Aufrufkette an FBs und FCs, die genaue Umsetzung ist jedoch dem Programmierer überlassen.

Tasks in Logix

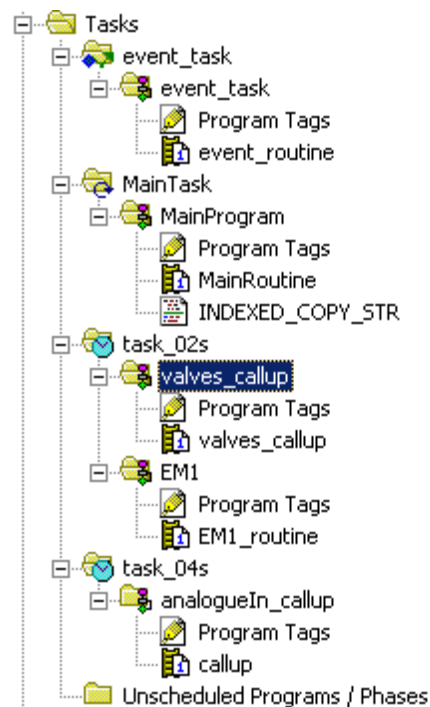
Tasks werden vom Betriebssystem aufgerufen. Sie bieten Zeitplanung und Prioritäten für Programme. Jedes Programm enthält einen Datenabschnitt und mindestens eine Coderoutine.

Die Tasks können periodisch, ereignisgesteuert oder kontinuierlich ausgeführt werden. Jeder Task kann eine Priorität zugewiesen werden. Die kontinuierliche Task erhält stets die niedrigste Priorität, sofern sie vorhanden ist.

Logix-Projekte weisen eine Task auf, deren Standardname „MainTask“ (Haupttask) lautet. Diese Task kann kontinuierlich, periodisch oder ereignisgesteuert sein. Sie können den Namen auch ändern.

Task- und Programmstruktur in Logix

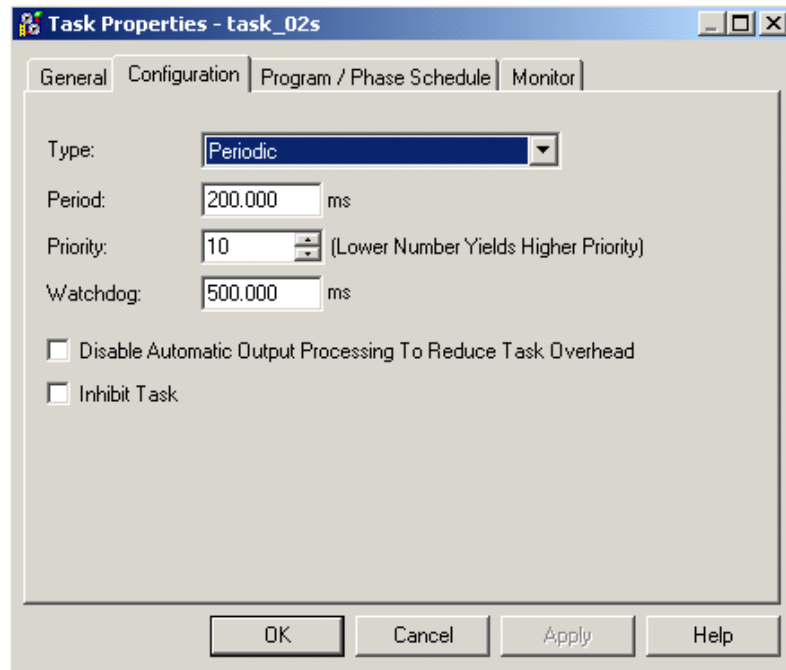
Dieser Ausschnitt aus einer Beispiel-Projektstruktur in RSLogix 5000 veranschaulicht die Strukturierung von Tasks und Programmen.



Im oben dargestellten Bildschirmfoto zeigt das Symbol links neben „event_task“ eine ereignisgesteuerte Task an. Das Symbol links neben „MainTask“ zeigt eine kontinuierliche Task an, und das Symbol links neben „task_02s“ zeigt eine periodische Task an.

Periodische Tasks

Periodische Tasks werden in konstanten, konfigurierten Intervallen ausgelöst. Die Konfiguration des Zeitabstands und der Priorität ist nachstehend dargestellt.



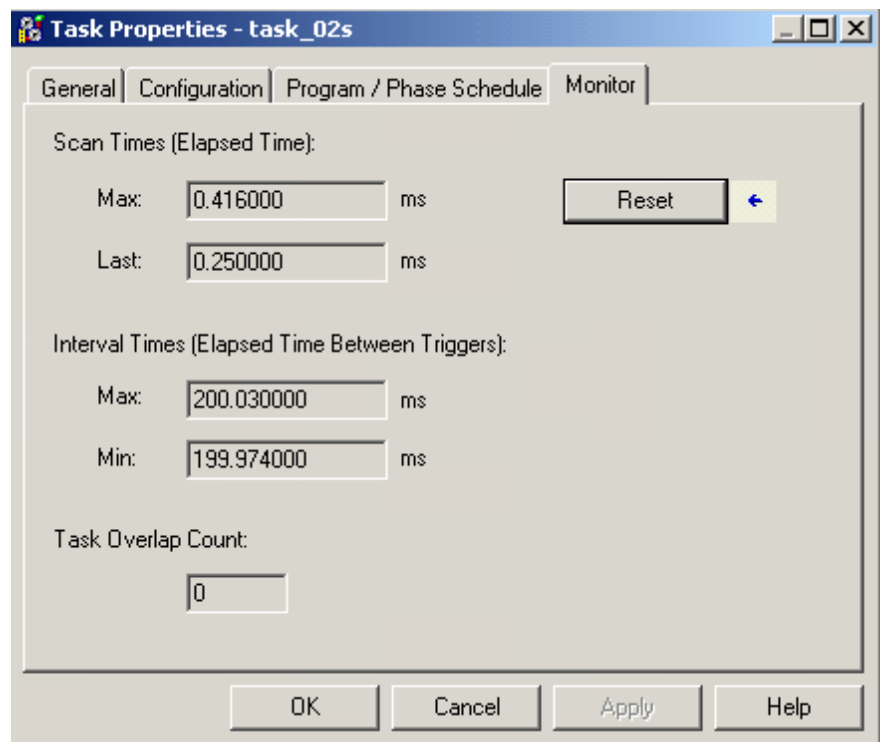
Die Konfiguration ähnelt der Konfigurationsseite für OB30–OB38, die im Abschnitt „OB30–OB38: Zyklische Interrupts“ beschrieben wurde.

Zeitplanung von periodischen Tasks

Das Task-System hat folgenden Zweck:

- Der Programmierer soll die Möglichkeit haben, geeignete Frequenzen für die Ausführung von Programmen auszuwählen. Indem Code nur so häufig wie nötig ausgeführt wird, steht die Prozessorleistung der Steuerung effizienter für Anwendungsprioritäten zur Verfügung.
- Mithilfe des Prioritätensystems können kritische Tasks solche mit niedrigerer Priorität unterbrechen, sodass die Wahrscheinlichkeit verbessert wird, dass sie mit der beabsichtigten Frequenz ausgeführt werden.

Diese Zeiten können einfach über die Task-Eigenschaften bzw. den Monitor überprüft werden.



Was geschieht, wenn während der Ausführung einer Task eine Auslösung erfolgt?

- Wenn die neue Auslösung eine Task mit höherer Priorität als die gerade ausgeführte betrifft, wird die ausgeführte Task durch die neue unterbrochen und nach Abschluss der Task mit höherer Priorität wieder aufgenommen.
- Wenn die neue Auslösung eine Task mit niedrigerer Priorität als die gerade ausgeführte betrifft, wird die ausgeführte Task fortgesetzt, und die neue Task wartet, bis keine Task mit höherer Priorität mehr ausgeführt wird.
- Wenn die neue Auslösung eine Task mit derselben Priorität wie die ausgeführte betrifft, führt die Steuerung beide Tasks aus, indem in Intervallen von 1 ms zwischen ihnen gewechselt wird.
- Wenn die Auslösung die gerade ausgeführte Task selbst betrifft, wird die Auslösung verworfen. Dabei handelt es sich um einen **Überlappungszustand**.

Die Anzahl von aufgetretenen Überlappungen seit dem letzten Reset des Zählers wird im Fenster für Taskprioritäten angezeigt. Eine Zahl außer null gibt an, dass der Interrupt-Zeitraum erhöht werden muss.

TIPP

Vermeiden Sie es, unnötig zwischen Tasks zu wechseln, da bei unnötigen Wechseln Prozessorleistung verschwendet wird.

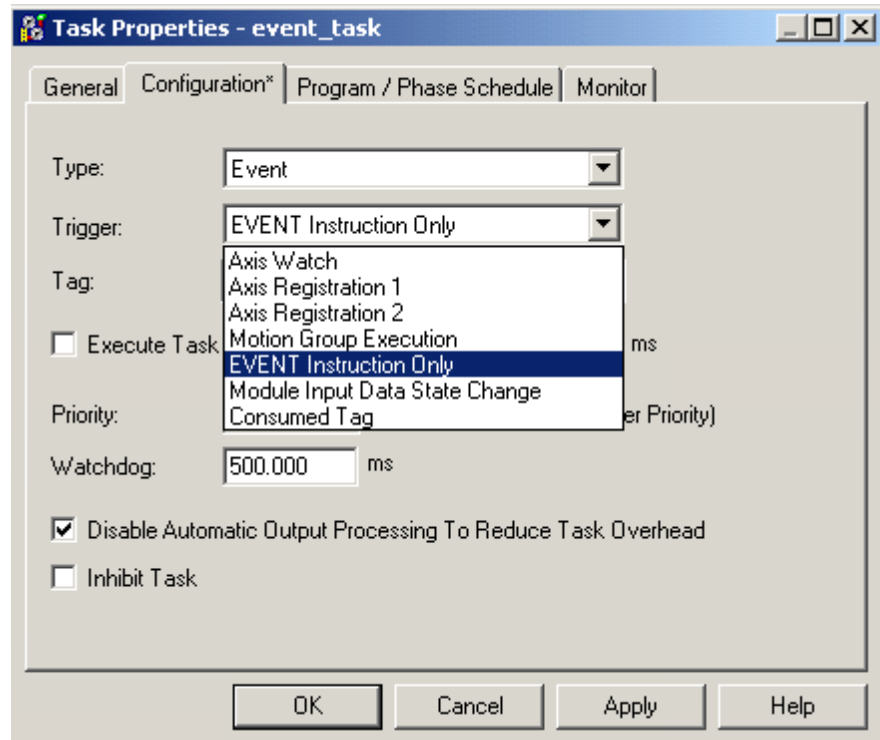
Beachten Sie bei der Programmierung periodischer Interrupts in Logix die folgenden Ähnlichkeiten und Unterschiede zu STEP 7:

- In STEP 7 erfolgen Aufrufe von dem OB, der für die Ausführung mit der gewünschten Frequenz konfiguriert ist, an die Funktionen und Funktionsblöcke, die Sie mit dieser Frequenz ausführen möchten. In Logix fügen Sie Programme und Routinen in der Projektstruktur unter der Task ein.
- In STEP 7 und Logix weist der eigentliche Anwendungscode keine wesentlichen Unterschiede vom Code in einer Task mit kontinuierlicher Ausführung auf. Beachten Sie, dass die konstante und bekannte Frequenz einer periodischen Task Programmierern die Möglichkeit gibt, durch eine einfache Inkrementierung einer Variablen ein Zeitwerk zu erzeugen.
- In beiden Systemen müssen Sie beim Entwickeln und Testen Ihres Codes auf Überlappungen achten. Die Ausführungszeit des OB oder der Task muss deutlich unter der Ausführungsfrequenz liegen.
- Bei Logix-Tasks lässt sich die Ausführungszeit einfach ermitteln. Verwenden Sie dazu den oben dargestellten Bildschirm für Task-Eigenschaften. In STEP 7 müssen Sie am Anfang und am Ende des OB die Systemuhr abtasten, die Werte subtrahieren und das Ergebnis zur Überwachung in einer Variable speichern.

- In S7-Steuerungen führen Überlappungen dazu, dass die Steuerung anhält, sofern kein Fehler-OB hinzugefügt wird, der den Fehler abfängt. Logix geht weniger streng vor und zählt lediglich die Anzahl von Überlappungen.
- In STEP 7 kann die Ausführung von periodischen OBs relativ zueinander abgestimmt werden. Dies ist bei Logix-Tasks nicht möglich.

Ereignisgesteuerte Tasks

Ereignisgesteuerte Tasks werden ausgeführt, wenn ein konfigurierbares Auslösungsereignis auftritt. Normalerweise erhalten sie eine höhere Priorität als periodische Tasks.



Die Konfiguration von ereignisgesteuerten Tasks wird durch Öffnen der Task-Eigenschaften einer Task und Auswählen des Typs für das Ereignis durchgeführt. Für verschiedene Logix-Steuerungen können verschiedene Arten von Auslösern für ereignisgesteuerte Tasks verwendet werden.

Kontinuierliche Task

Logix-Steuerungen unterstützen eine kontinuierliche Task, diese muss jedoch in einem Projekt nicht auftreten. Wenn Sie möchten, können Sie das gesamte Programm unter periodischen und ereignisgesteuerten Tasks ausführen.

Sie können konfigurieren, ob die kontinuierliche Task am Ende ihrer Ausführung die Ausgänge aktualisieren soll.

Auf Wunsch können Sie den Anteil der Prozessorzeit für ungeplante Kommunikation als prozentualen Anteil der für die kontinuierliche Task reservierten Zeit anpassen.

Task Monitor

Die Software RSLogix 5000 enthält ein Tool namens Task Monitor, der bei der Analyse von zyklischen Tasks und vielem mehr helfen kann.

Im nachstehenden Bildschirmfoto ist dargestellt, wie die Informationen über die Tasks einer Steuerung in einer Tabelle angezeigt werden können.

The screenshot shows the 'Logix5000 Task Monitor' window with the 'Performance' tab selected. The table below displays the following data:

Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Stat
MainTask	* 500,000us	0.70%	Lowest	3,524us	5,852us	500,000us	0	Runn
task_02s	200,000us	0.16%	9	548us	894us	500,000us	0	Runn
task_04s	400,000us	0.04%	10	764us	1,182us	500,000us	0	Runn
task_01s	100,000us	0.17%	5	134us	426us	500,000us	0	Runn
event	10,000us	0.00%	10	0	0	500,000us	0	Stopp

Die anderen Registerkarten bieten umfassende Informationen auf Systemebene über die Leistung der Steuerung. Das Tool ist als Standard auf dem Installationsmedium für RSLogix 5000 enthalten.

Tags anstelle von Adressen

Einer der größten Unterschiede, der S7-Benutzern bei der Arbeit mit Logix auffällt, ist das Fehlen von Adressen für Daten. Datenelemente werden in einer Tag-Datenbank erstellt, und die Software RSLogix 5000 weist Adressen im Hintergrund zu. Dadurch müssen Anwender keine Speicheradressen begreifen und verwalten. In diesem Abschnitt wird die Datenzuweisen der beiden Systeme erläutert.

Datenbereiche in S7

Datenbereiche in S7-Steuerungen

Adressbereich	S7-Benennung	Einheitengröße
Prozess-Eingangsdatentafel	I	Eingangsbit
	IB	Eingangsbyte
	IW	Eingangswort
	ID	Eingangs-Doppelwort
Prozess-Ausgangsdatentafel	Q	Ausgangsbit
	QB	Ausgangsbyte
	QW	Ausgangswort
	QD	Ausgangs-Doppelwort
Bitspeicher	M	Speicherbit
	MB	Speicherbyte
	MW	Speicherwort
	MD	Speicher-Doppelwort
Timer	T	
Zähler	C	
Datenbaustein	DBX	Datenbit
	DBB	Datenbyte
	DBW	Datenwort

In den nachstehenden Abschnitten werden die beiden am häufigsten verwendeten Bereiche bei der Programmierung näher erläutert – Bitspeicher und Datenblöcke.

Bitspeicher

„Bitspeicherorte“ werden in der Form „Mx“ benannt. Beispiele:

- M5.3 ist ein Bit.
- MB6 ist ein Byte (BYTE).
- MW8 ist ein 16-Bit-Wort (WORD).
- MD10 ist ein 32-Bit-Wort (DWORD).

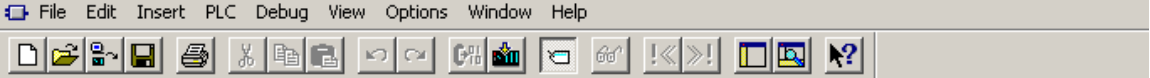
Bitspeicherorte können in der Symboltabelle (ähnlich einer PLC-5- oder SLC-Symboltabelle) mit einer Bezeichnung versehen werden, wie auf dem folgenden Bildschirmfoto zu erkennen ist.

Status	Symbol ▲	Address	Data type	Comment
	EXT_ZONE2_ON	Q 28.2	BOOL	EXTRUDER ZONE2 ON
	EXT_ZONE3_ON	Q 28.3	BOOL	EXTRUDER ZONE3 ON
	EXT_ZONE4_ON	Q 28.4	BOOL	EXTRUDER ZONE4 ON
	EXT_ZONE5_ON	Q 28.5	BOOL	EXTRUDER ZONE5 ON
	FALSE	M 0.0	BOOL	
	FlowTotaliser	UDT 10	UDT 10	
	FSL24001	I 10.0	BOOL	
	FSL24002	I 41.0	BOOL	
	FSL24003	I 10.6	BOOL	
	FT24001	PMW 530	INT	
	FT24002	PMW 526	INT	
	FT24006	PMW 570	INT	
	GET_INDEXED_REFE...	FC 111	FC 111	
	Global_Data	DB 99	DB 99	
	IFIX_alarms	DB 71	DB 71	
	INDEXED_COMPARE	FC 102	FC 102	
	INDEXED_COPY	FC 101	FC 101	
	Interlocks_Handler	FB 70	FB 70	
	Interrupt_Execution	OB 35	OB 35	
	JUNK_BIT	M 0.2	BOOL	
	KTRON_CmdsFromM...	DB 33	DB 33	Commands & Setpoints From Manufacturing PLC
	KTRON_StatusToMan...	DB 32	DB 32	Feeder Status to ManufacturingPLC
	LF24001A	Q 17.0	BOOL	PIG LAUNCH VALVE
	LF24002A	Q 18.5	BOOL	PIG LAUNCHER
	LF24003A	Q 10.4	BOOL	PIG LAUNCHER

Datenbausteine

Datenbausteine weisen einen ähnlichen Status wie andere Blöcke – Bausteine, Funktionsbausteine und Funktionen – auf, enthalten jedoch Daten anstelle von Programmcode. Der Speicher in Datenblöcken ist statisch – die Daten behalten ihren Wert bei, bis dieser geändert wird.

Beispiel für einen Datenbaustein



Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	GrantrezSalts_SP	REAL	0.000000e+000	Grant
+4.0	GrantrezSalts_FeedFact	REAL	0.000000e+000	Grant
+8.0	CMC_SP	REAL	0.000000e+000	CMC S
+12.0	CMC_FeedFact1	REAL	0.000000e+000	CMC I
+16.0	SiliconDioxide_SP	REAL	0.000000e+000	Silic
+20.0	SiliconDioxide_FeedFact	REAL	0.000000e+000	Silic
+24.0	MaleicAcid_SP	REAL	0.000000e+000	Malei
+28.0	MaleicAcid_FeedFact	REAL	0.000000e+000	Malei
+32.0	Sequence_Cmnds	STRUCT		
+0.0	PrimeMaterials	BOOL	FALSE	Loads
+0.1	StartProduction	BOOL	FALSE	Start
+0.2	EndProduction	BOOL	FALSE	EndPr
+0.3	Stop	BOOL	FALSE	Stop
+0.4	FaultAck	BOOL	FALSE	Ackno
+0.5	Spare1	BOOL	FALSE	

Datenbausteinsymbole werden nicht in der Symboltabelle angezeigt, der Name des Datenbausteins wird hingegen angezeigt.

Datenbausteine können für die Daten in Funktionsblöcken verwendet werden. Sie werden dann als Instanz-Datenbausteine bezeichnet.

Daten in Logix

In der RSLogix 5000-Programmierungsumgebung werden Daten in einer Tag-Datenbank eingerichtet. Die Speicheradressen sind vor dem Programmierer verborgen, was ihm die Arbeit erleichtert.

Tag-Datenbank

Controller Tags - dev_18Aug(controller)						
Scope: dev_18Aug		Show...	Show All			
Name	Alias For	Base Tag	Data Type	Style	Description	
+ analogIn_1			DINT	Decimal		
Blue_Button	Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal		
+ CompactLogix_1_consume			UDT_STEP_SEQUENCE		Data - step sequ...	
+ ControlLogix_1_produce			UDT_STEP_SEQUENCE		Data - step sequ...	
+ Drive:I			AB:PowerFlex70EC_Driv...			

Auswählen eines Tags in einem Pulldown-Menü während der Programmierung

The screenshot illustrates the process of selecting a tag from a dropdown menu during programming. The dropdown menu is open, displaying a list of tags with columns for Name, Data Type, and Description. The tag 'em1Data.step[0]' is selected. The background shows a ladder logic diagram with a 'JMP' instruction and a 'Data - step sequence EM sequencer step' label.

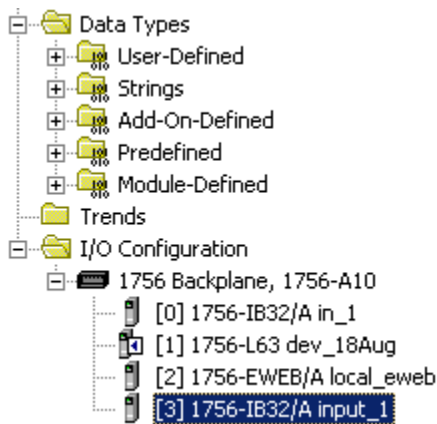
In Logix ist jedem Programm eine **Steuerungsbereich-Tag-Datenbank** und eine **Programmbereich-Tag-Datenbank** zugeordnet.

- Tags in der Steuerungsbereich-Datenbank sind global und können von Routinen in jedem Teil des Programms verwendet werden.
- Programmbereich-Tags können nur von Routinen in diesem Programm abgerufen werden.

E/A und Alias-Tags

Mit einem Alias-Tag können Sie ein anderes Tag darstellen, wobei beide Tags denselben Wert aufweisen. Aliase dienen u. A. dazu, auf E/A-Tags wie weiter unten beschrieben zu verweisen.

E/A-Module können einem Projekt hinzugefügt werden, indem das Modul im Projektordner der Steuerungs-Backplane hinzugefügt wird.

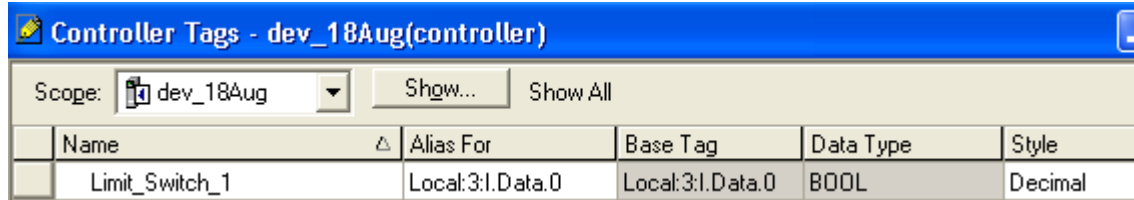


In diesem Fall wurde eine Eingangskarte mit 32 Punkten an Steckplatz 3 hinzugefügt. Die Steckplatznummer ist am Anfang der Zeile in eckigen Klammern angegeben. „1756-IB32/A“ ist die Bestellnummer der Karte. „input_1“ ist der Name für die Karte, der beim ersten Hinzufügen der Karte im Rack konfiguriert wird.

Nach dem Hinzufügen der Karte generiert die Software RSLogix 5000 die entsprechenden Geräteprofil-Tags automatisch in der Steuerungsbereich-Tag-Datenbank. Es handelt sich um die unten dargestellten Tags „Local:3:I“ (Eingang) und „Local:3:C“ (Konfiguration).

Controller Tags - dev_18Aug(controller)					
Scope: <input type="text" value="dev_18Aug"/>		Show...		Show All	
Name	Alias For	Base Tag	Data Type	Style	
[-] Local:3:I			AB:1756_DI:1:0		
[+] Local:3:I.Fault			DINT	Binary	
[-] Local:3:I.Data			DINT	Binary	
[-] Local:3:I.Data.0			BOOL	Decimal	
[-] Local:3:I.Data.1			BOOL	Decimal	
[-] Local:3:I.Data.2			BOOL	Decimal	

Sie können ein neues Alias-Tag mit einem aussagekräftigeren Namen erstellen. Beispielsweise kann ein Alias für den ersten Eingang den Namen „Begr_schalter_1“ erhalten, der diesen Eingang physisch beschreibt.



Name	Alias For	Base Tag	Data Type	Style
Limit_Switch_1	Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal

In STEP 7 werden Adressen einer E/A-Karte durch das Hardware-Konfigurationstool zugewiesen, wenn die Karte dem System hinzugefügt wird. Beispielsweise werden einer Digitaleingabekarte möglicherweise die Bytes I16 und I17 zugewiesen. Der Programmierer erkennt dann die Bitadressen jeder Eingabe und gibt einen Namen in der Symboltabelle an. Anschließend weist das Programm automatisch I16.5 = „ZSC2036“ zu.

Programmiersprachen

In diesem Abschnitt werden die Programmiersprachen beschrieben, die für STEP 7 und RSLogix 5000 verfügbar sind. Alle Sprachen müssen nicht vorhanden sein, sondern sind von der erworbenen Version der Software abhängig. Die Auswahl der für eine Aufgabe am besten geeigneten Logix-Sprache führt zu einem einfacheren Programmentwurf, schnellerer Programmierung und einem Programm, das einfacher zu verstehen ist.

Zwischen den Sprachen für S7 und Logix besteht ein wesentlicher Unterschied. In S7 stellt die Bemerkungsliste die systemeigene Sprache der Steuerung dar. Andere Sprachen werden in die Bemerkungsliste übersetzt. In Logix sind alle Sprachen in der Steuerung „systemeigene“ Sprachen – jede wird ohne Bezug auf die anderen kompiliert. Der Vorteil besteht darin, dass Programme beim Hochladen von der Steuerung in der Sprache angezeigt werden, in der sie programmiert wurden.

STEP 7 weist drei Standardsprachen auf:

- Anweisungsliste - AWL, (Statement List - STL) – ähnlich einem Assembler auf hoher Ebene.
- Kontaktplan - KOP, (Ladder Logix - LAD)
- Funktionsblockdiagramm (Function Block Diagram - FBD)

Hinzu kommen einige optionale Sprachen:

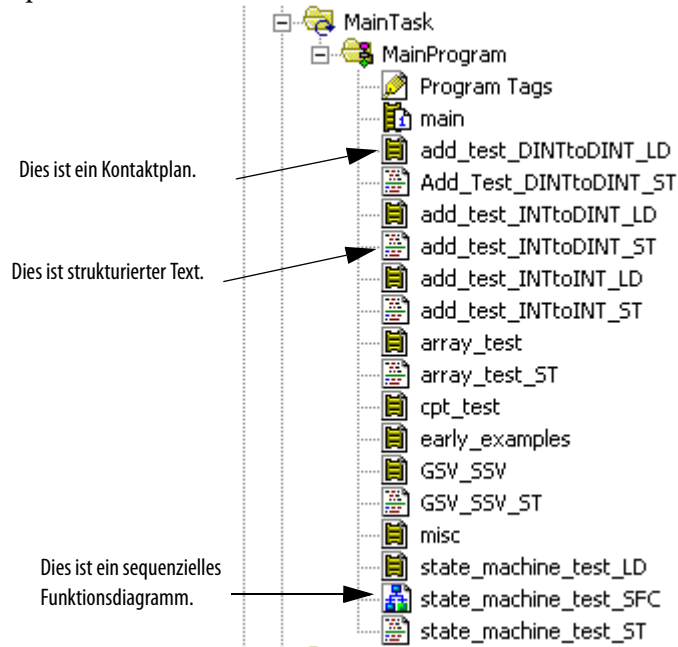
- Strukturierter Text (ST)
- CFC – Continuous Flow Chart – kontinuierliches Diagramm für Prozessanwendungen
- HiGraph – Ablaufsteuerung per Grafiksoftware
- ML – Motion Language – ähnlich GML in der älteren dedizierten Achssteuerung 1394 von Rockwell Automation

Ein Programm kann aus Funktionsblöcken und Funktionen bestehen, die in verschiedenen Sprachen geschrieben wurden.

Die Software RSLogix 5000 weist vier Programmiersprachen auf:

- Kontaktplan (LD) – vergleichbar mit Siemens LD mit einem erweiterten Befehlssatz.
- Strukturierter Text (ST) – äquivalent zu Siemens ST
- Funktionsblockdiagramm(FBD) – äquivalent zu Siemens CFC
- Sequenzielles Funktionsdiagramm (SFC) – vergleichbar mit Siemens hiGraph.

Eine Routine – der einfachste Abschnitt von Code in Logix – kann in jeder dieser Sprachen geschrieben werden, und ein Programm kann aus Routinen in verschiedenen Sprachen bestehen. Das folgende Bildschirmfoto stellt ein Beispiel dar.



Logix-Kontaktplan

Traditionell werden Kontaktpläne für die Implementierung von boolescher kombinatorischer Logik verwendet. In Logix stehen sie auch für sequenzielle Logik, Achsststeuerung, Datenmanipulation und mathematische Berechnungen zur Verfügung, auch wenn andere Sprachen für solche Aufgaben eventuell besser geeignet sind.

Strukturierter Text in Logix

Strukturierter Text ist eine prozedurale Sprache auf hoher Ebene, die für Programmierer mit Erfahrung in Basic, Pascal oder einer der C-Sprachen einfach zu erlernen ist. Sie wird hauptsächlich für die Datenmanipulation und mathematische Berechnungen verwendet, ist aber auch gut für Achsststeuerung, kombinatorische und sequenzielle Logik geeignet.

Logix-Funktionsblockdiagramm

In einem Funktionsblockdiagramm wird eine (boolesche oder mathematische) Funktion grafisch unter Bezug von Eingangs- und Ausgangsvariablen dargestellt. Die Eingangs- und Ausgangsvariablen werden über Verbindungslinien mit Blöcken verbunden. Ein Ausgang eines Blocks kann auch mit einem Eingang eines anderen Blocks verbunden werden.

PID-Regelkreise sollten in FBD programmiert werden. Für die Prozesssteuerung stellt es die praktischste Sprache dar.

Sequenzielles Funktionsdiagramm in Logix

SFC ist ein grafisches Werkzeug für die Beschreibung von sequenzieller Logik als Kombination aus Zuständen und Transitionen. Einem Zustand können Ausgänge zugewiesen werden, und Transitionen in andere definierte Zustände können boolesche Bedingungen zugewiesen werden.

Konvertierung von STEP 7-Code in Logix

- Wenn Sie STEP 7-Kontaktplancode in Logix konvertieren möchten, stellt LD die erste Wahl dar. Die Bedeutung von LD ist in beiden Systemen ähnlich.
- Wenn Sie STEP 7-Funktionsblockdiagramm-Code in Logix konvertieren möchten, stellt FBD die erste Wahl dar.
- Beachten Sie, dass Standard-Logix-FBD mehr Möglichkeiten als STEP 7-FBD bietet und der optionalen STEP 7-Sprache CFC entspricht.
- Wenn Sie STEP 7-Anweisungslisten-Code in Logix konvertieren möchten, ist die geeignetste Sprache von der Art des STL-Blocks abhängig. Wenn der STL-Block viele boolesche Ausdrücke enthält, ist die beste Logix-Sprache für die Konvertierung wahrscheinlich LD. Enthält der STL-Block Zeiger zum Zugreifen auf Daten und zum Manipulieren von Daten oder es werden mathematische Berechnungen durchgeführt, ist die beste Logix-Sprache für die Konvertierung wahrscheinlich ST. Wenn der STL-Block sequenzielle Logik enthält, sollte SFC in Betracht gezogen werden, sequenzielle Logik kann jedoch auch in ST und LD einfach implementiert werden.

Datenfelder anstelle von Zeigern

In STEP 7 können Datenfelder genau wie in Pascal oder C definiert werden, die Grundsprachen (STL, LD und FBD) weisen jedoch keine Unterstützung auf hoher Ebene zum Zugriff auf Datenfelder auf. Stattdessen müssen Zeigerroutinen konstruiert werden.

STEP 7-Bibliotheksfunktionen bieten keine Unterstützung für den Zugriff auf Datenfelder. Programmierer, die Erfahrung mit Zeigern haben, können eigene Funktionen wie FC101 „INDEXED_COPY“ (siehe unten) schreiben, dazu sind jedoch umfassende Kenntnisse und Zeit erforderlich.

„INDEXED_COPY“ in STEP 7 erreicht das gleiche Ziel wie der Logix-Befehl COP für das indizierte Kopieren.

```
CALL "INDEXED_COPY"           FC101
  indexSrc:=#index_in
  source  :="Instance_FB2".table P#DB4.DBX0.0
  indexDst:=1
  dest    :="Instance_FB2".target P#DB4.DBX96.0
  len     :=8
```

FC111 unten greift auf ein Datenfeld zu.

```
CALL "GET_INDEXED_REFERENCE"   FC111
  refArray :="Instance_FB2".table P#DB4.DBX0.0
  index    :=#index_in
  byteIncr :=32
  startIndex:=TRUE
  retVal   :=#ptr
```

Der Zeiger auf das Objekt wird in Parameter #ptr zurückgegeben. Dieses Referenz kann zum Abrufen der Daten aufgelöst werden.

In Logix können Datenfelder auf die übliche Weise für Computersprachen auf hoher Ebene definiert und abgerufen werden, wie im nachstehenden Codeausschnitt veranschaulicht wird.

```
// copy a string from a table of strings #table
// to a target string #target. The index is #index_in

COP(table[index_in], target, target.LEN);
```


Add-On-Befehle

Zusammenfassung der Add-On-Befehle

Add-On-Befehle sind das Äquivalent von STEP 7-Funktionsblöcken. Sie weisen private Daten und erweiterte Parameterwahlmöglichkeiten auf. Insbesondere ermöglicht der Parametertyp INOUT (oder das „Übergeben per Verweis“) das effiziente Übergeben von Datenstrukturen an den Code.

Da Add-On-Befehle den STEP 7-Funktionsblöcken so ähnlich sind, werden S7-Programmierer, die zu Logix wechseln, sie wahrscheinlich bereitwillig verwenden.

Vergleich zwischen FBs und Add-On-Befehlen:

- Beide können sie von beliebigen Stellen im Programm als benannte Funktionen aufgerufen werden.
- Beide enthalten einen privaten Datenbereich von statischen Daten, auch wenn dieser im Fall von STEP 7 nicht vollkommen privat ist.
- Ein STEP 7-Funktionsbaustein weist auch einen temporären Datenbereich auf.
- Im Add-On-Befehl wird dies durch lokale statische Daten erreicht.

Beide weisen drei Arten von Parametern auf – Eingangs- (übergeben durch Wert), Ausgangs- (übergeben durch Wert) und Ein-Aus-Parameter (übergeben durch Verweis). Der Parameter zum Übergeben durch Verweis stellt einen beträchtlichen Vorteil dar, da er die effiziente Übergabe von großen Datenstrukturen ermöglicht.

Beim Add-On-Befehl wird automatisch eine Änderungshistorie geführt, da zum Zeitpunkt der Änderung ein Zeitstempel und der Windows-Benutzername aufgezeichnet werden. Dies ist bei STEP 7-Funktionsblöcken nicht möglich.

Für den Add-On-Befehl kann eine Vorabtestroutine konfiguriert werden, die ausgeführt wird, wenn die Steuerung vom Program-Modus in den Run-Modus wechselt oder im Run-Modus eingeschaltet wird. Unter diesen Bedingungen wird die Vorabtestroutine einmal ausgeführt und wird daher meist zum Initialisieren von Daten verwendet. In STEP 7 wird mit dem Baustein OB100 das gleiche erreicht, der Vorabtestcode kann jedoch nicht spezifisch einem FB zugeordnet werden.

Wenn der Add-On-Befehl aus einem SFC-Schritt aufgerufen wird und das sequenzielle Funktionsdiagramm für automatische Rückstellung konfiguriert ist, wird beim Beenden dieses Schritts eine im Add-On-Befehl definierte Nachabtestroutine einmal ausgeführt. Sie kann beispielsweise zum Rücksetzen von Daten verwendet werden. In STEP 7-Funktionsblöcken steht kein integriertes Äquivalent zur Verfügung (ist allerdings einfach zu programmieren).

Ein Add-On-Befehl kann mit einer Routine „EnableInFalse“ ausgestattet werden, die ggf. aufgerufen wird, wenn die Strompfadbedingung beim Aufruf des Add-On-Befehls „Falsch“ ist. In diesem Fall übergeben die Eingangs- und Ausgangsparameter Werte. In STEP 7-Funktionsblöcken gibt es kein Äquivalent.

Add-On-Befehle werden in [Kapitel 4](#) näher erläutert.

Backing-Tags

In vielen Befehlen und Datentypen werden Backing-Tags verwendet – Tags, die speziell für die erstellte Instanz des Befehls oder Datentyps erstellt werden. Für Add-On-Befehle, Timer, Zähler, Nachrichten und PID-Steuerung sind Backing-Tags vorhanden. Die Software RSLogix 5000 generiert die entsprechende Struktur der Elemente, wenn Sie ein Tag dieses Typs erstellen. Sie müssen die Elemente daher nicht selbst erstellen.

Name	Alias For	Base Tag	Data Type	Style
-Timer1			TIMER	
+ Timer1.PRE			DINT	Decimal
+ Timer1.ACC			DINT	Decimal
- Timer1.EN			BOOL	Decimal
- Timer1.TT			BOOL	Decimal
- Timer1.DN			BOOL	Decimal

Das Common Industrial Protocol (CIP)

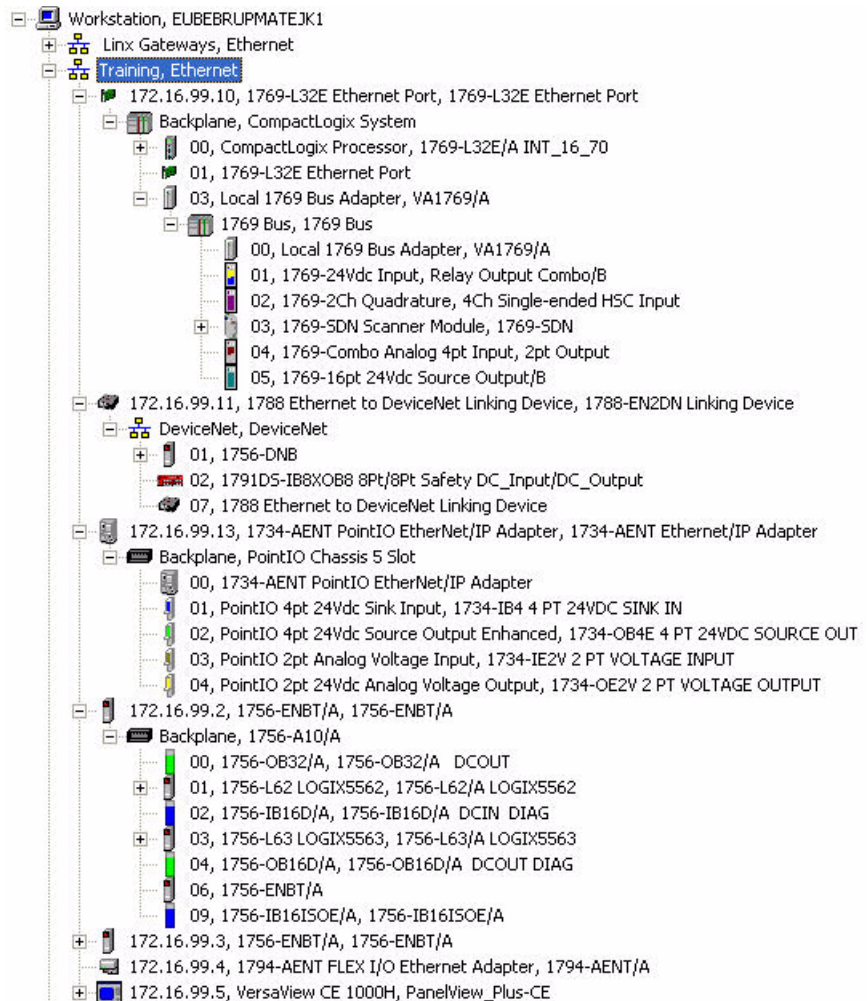
Logix greift auf drei Hauptnetzwerke zurück – Ethernet/IP, ControlNet und DeviceNet. Jedes weist Merkmale auf, die für bestimmte Anwendungsbereiche geeignet sind. Die drei Netzwerktypen verwenden ein gemeinsames modernes Industrieprotokoll, das „Common Industrial Protocol“.

Durch das CIP ist es möglich, Daten in allen drei von Logix unterstützten Netzwerken mit nahezu identischer Konfiguration und Programmierungsschnittstelle zu übertragen. Daten können außerdem über ein Netzwerk übertragen werden, das mehrere der drei Netzwerktypen umfasst, ohne dass der Programmierer Protokolle übersetzen muss.

S7 arbeitet traditionell mit den beiden Hauptprotokollen Industrial Ethernet für die Vernetzung mit IT und anderen Steuerungen und Profibus DP für die Vernetzung mit Feldsystemen. Diese beiden Protokolle sind auf Hardwareebene und Datenebene getrennt. Bei der aktuellen Hard- und Software von S7, „Profinet CBA“, sind Industrial Ethernet, Profinet und Profibus integriert.

Anzeigen des Netzwerks

S7-Benutzer finden die Netzwerkconfiguration und -verwaltung von Logix möglicherweise bemerkenswert. Beispielsweise sind in der nachstehenden Struktur die tatsächlich mit dem System verbundenen Geräte aufgeführt. Diese Struktur wurde durch einfaches Online-Schalten hergestellt – es erfolgte keinerlei Konfiguration.



Netzwerke werden in [Kapitel 1](#) weiter beschrieben.

Datenaustausch zwischen Steuerungen

Senden/Empfangen in STEP 7

Zum Vorbereiten der Kommunikation zwischen Steuerungen in STEP 7 werden die folgenden Schritte durchgeführt.

1. Die dezentralen Stationen werden grafisch in einer STEP 7-Komponente namens NetPro konfiguriert.
2. In NetPro wird eine Verbindungstabelle erstellt, in der die Protokolle und Parameter für jede Verbindung angegeben sind.
3. Die Bibliotheksfunktionen FC5 AG_SEND und FC6 AG_RECV werden in das Projekt kopiert.
4. Aus dem Anwenderprogramm werden AG_SEND und AG_RECV aufgerufen, wobei Verbindungsparameter und die Datenbereiche für die Erfassung und den Empfang der Daten angegeben werden.

Produzierte/konsumierte Tags in Logix

Über produzierte und konsumierte Tags werden jeweils nach einem definierten Zeitraum kritische Daten zwischen in einem Netzwerk verbundenen Logix-Steuerungen übertragen. Produzierte und konsumierte Tags können über Ethernet/IP oder ControlNet und auf der Backplane von ControlLogix-Steuerungen übertragen werden.

Produzierte und konsumierte Tags sind Tags, die beim Erstellen als produziert oder konsumiert konfiguriert werden. Wenn ein Tag als produziert markiert ist, wird der Wert per Multicast an das EtherNet/IP- oder ControlNet-Netzwerk übertragen, mit dem die Steuerung verbunden ist. Ist es als konsumiert markiert, wird die Steuerung, von der das Tag Daten benötigt, als Teil der Konfiguration identifiziert, und das konsumierte Tag empfängt seinen Wert von dem gleichwertigen produzierten Tag in dieser Steuerung.

Zum Senden und Empfangen sind getrennte Kanäle vorhanden. Das Ändern des Wertes eines konsumierten Tags hat keine Auswirkung auf das produzierende Tag. Dieses Verfahren ähnelt der Kommunikation zwischen Steuerungen in S7 und unterscheidet sich von der Kommunikation zwischen Steuerung und SCADA, bei der jede Änderung am anderen Ende wiedergegeben wird.

Zum Einrichten von Verbindungen zwischen produzierten und konsumierten Tags ist keine Programmierung erforderlich. Dies weicht von S7 ab, da dort eine gewisse Programmierung für die Kommunikation zwischen Steuerungen (SENDEN/EMPFANGEN) erforderlich ist.

Benutzerdefinierte Datentypen

In Logix können benutzerdefinierte Datentypen konfiguriert werden. Dadurch lässt sich die Struktur eines komplexen Datentyps als Typ deklarieren. Instanzen dieses Typs können anschließend im Programm definiert werden.

Die benutzerdefinierten Datentypen von Logix weisen eine ähnliche Konfiguration und Nutzung wie die benutzerdefinierten Datentypen von STEP 7 auf.

Benutzerdefinierte Datentypen (UDT) in Logix

Name:

Description:

Ramps a real variable from its current value to a new value at a specified rate.

Members: Data Type Size: 28 byte(s)

	Name	Data Type	Style	Description
	initial_output	REAL	Float	saved initial output
	increment	REAL	Float	calculated increment
	RAMP_RATE_ABS	REAL	Float	per second - (set always +ve)
	RAMP_TARGET	REAL	Float	final value - (set)
	change	REAL	Float	calculated change over ramp
	counter	DINT	Decimal	internal counter
	complete	BOOL	Decimal	ramping is complete
	_enable	BOOL	Decimal	for enable one shot
	enabled	BOOL	Decimal	ramper enabled
100 010				

Asynchrone E/A-Aktualisierung

In Logix-Systemen werden E/A asynchron in Bezug auf die Zeiträume der Programmausführung aktualisiert. Dies steht im Gegensatz zur traditionellen SPS-Herangehensweise wie bei S7, bei der eine E/A-Datentafel zu Beginn des Zyklus aktualisiert wird und die Eingangswerte sich während der Ausführung des Programms nicht ändern.

Ein Logix-Programmierer muss sich entscheiden, ob Eingangsdaten gepuffert werden müssen, sodass ihre Werte während der Programmausführung konstant bleiben.

Häufig werden die Eingänge nur einmal abgerufen, indem sie als Parameter an ein Codemodul übergeben werden. Die Eingänge werden an keiner anderen Stelle des Programms verwendet. Dabei besteht keine Notwendigkeit für Pufferung. Nähere Informationen finden Sie im Steuerungsmodul-Beispiel in [Kapitel 4](#).

Der Datentyp DINT

Logix-Steuerungen können mit DINT-Tags (32-Bit-Ganzzahl) effizienter betrieben werden als mit INT (16 Bit) oder SINT (8 Bit). Verwenden Sie möglichst DINT, auch wenn der Bereich der möglichen Werte in ein INT- oder SINT-Tag passen würden. Diese Datentypen werden aus Gründen der Kompatibilität mit IEC61131-3 bereitgestellt, jedoch intern nach DINT konvertiert, bevor das Programm auf sie zugreift, damit der Code in den meisten Situationen effizienter ausgeführt werden kann.

Phase Manager

Phasenverwaltung in STEP 7

STEP 7 verfügt nicht über integrierte Tools zur Phasenverwaltung. Die notwendigen Strukturen müssen in einem Satz von Routinen programmiert werden, die meist als PLI oder Phasenlogikschnittstelle bezeichnet werden. Die Komponenten eines PLI-Programms auf der Basis von S88 sind:

- Ein Schrittschaltwerk, dessen Verhalten dem S88-Zustandsmodell entspricht. Der S88-Zustand ist durch bestimmte Schritte oder Bereiche von Schritten definiert. Schrittschaltwerk-Befehle werden ebenfalls durch S88 spezifiziert, und das Schrittschaltwerk reagiert nur, wenn das Zustandsmodell dies zulässt. Ein Schrittschaltwerk mit diesen Eigenschaften wird als Phase bezeichnet.
- Ein Satz von Daten für jede Phase, der zum Aufzeichnen des Phasenstatus und zum Empfangen eingehender Befehle vom Rezeptmanager verwendet wird. Der Rezeptmanager kommuniziert mit diesen Daten. Das Format der Daten ist vom Rezeptmanager abhängig.
- Ein Logikmodul, das den Phasenstatus in das vom Rezeptmanager benötigte Format und Befehle vom Rezeptmanager in Phasenbefehle übersetzt.

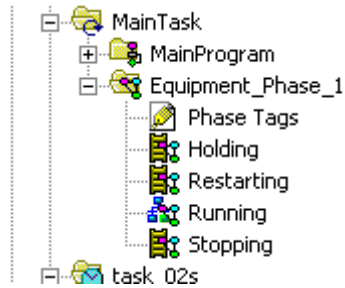
PhaseManager in Logix

S88-Gerätephasen weisen bestimmte Zustände der Phase sowie Transitionen zwischen diesen Zuständen auf. PhaseManager ist eine Funktionalität der Software RSLogix 5000, mit der Sie drei Dinge erreichen können:

- Zuordnen des Codes für jeden Phasenzustand zu einer anderen Routine
- Ausführen eines Zustandsautomaten im Hintergrund, der die Transitionen zwischen den Zuständen der Phase verarbeitet
- Verwalten der Ausführung der Phase mittels eines bestimmten Logix-Befehlssatzes

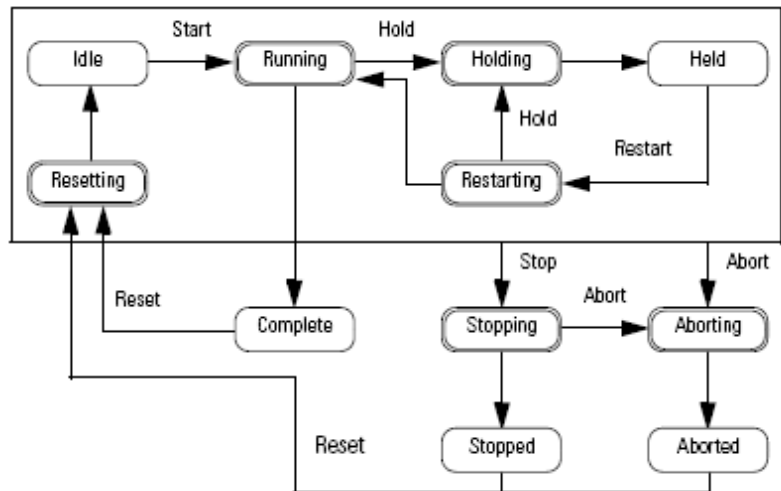
Diese wird in zahlreichen Anwendungen verwendet, beispielsweise bei der Prozesssteuerung und in der Verpackung, da eine saubere Trennung zwischen der Geräte-/Anlagensteuerung und der Verfahrenssteuerung und somit ein erheblich modularerer und effizienter Code erreicht wird, besonders bei größeren Systemen mit Standardisierung.

Anlagenphase in der Projektstruktur



Der Code für jeden Zustand der Phase kann in jeder der Logix-Sprachen geschrieben werden.

Dies ist der Phasen-Zustandsautomat. Er entspricht nahezu vollständig dem S88-Zustandsmodell.



Wenn Sie einen S88-konformen STEP 7-Phasenmanager bzw. eine PLI-Routine programmiert haben und diese in Logix konvertieren möchten, kann eine Übersetzung eventuell mit dem Logix-PhaseManager vermieden werden.

Koordinierte Systemzeit (CST)

S7 weist eine Systemuhr auf, die mit 32 Bit repräsentiert wird und in Millisekunden zählt. Ihr Wert kann durch einen Aufruf an das Betriebssystem abgerufen (und gespeichert) werden. Das ist für eine präzise Messung von Zeitintervallen praktisch.

In Logix wird die koordinierte Systemzeit verwendet, bei der es sich um eine 64-Bit-Zahl handelt, die die Anzahl von Mikrosekunden seit dem letzten Start der Steuerung angibt. Wie bei S7 können Intervalle durch Aufrufe des Betriebssystems zum Abrufen des CST-Wertes gemessen werden. Dies bildet die Grundlage für Taktsynchronisierung von Systemen mit mehreren Prozessoren, präzise Achssteuerungsfunktionalität, zyklische Ausgangsschaltung mit 100 µs Genauigkeit, Eingangseignis-Zeitstempel, zyklische Analogabtastung, Sicherheits-E/A-Überwachung und -Kommunikation, Positionsberechnungen von Kurvenscheiben bei der Achssteuerung und Uhrzeit.

Eingänge mit Zeitstempel

Der Zeitstempel ist eine Funktionalität, bei der eine Änderung der Eingangsdaten mit einer relativen Zeit des Auftretens dieser Änderung aufgezeichnet wird. Bei Digitaleingangsmodulen können Sie einen Zeitstempel für Datenänderungen konfigurieren. Sie können den CST-Zeitstempel zum Vergleichen der relativen Zeit zwischen Datenabtastungen verwenden.

Dadurch steht dem Programmierer eine unerreichte Genauigkeit beim Verknüpfen von Eingangssignalen mit Zeitverweisen zur Verfügung, wie sie z. B. für Anwendungen der Achssteuerung benötigt wird, ohne die Kommunikations- und Logikverarbeitungssysteme und den zugehörigen Anwendungscode stark zu belasten.

Zyklische Ausgänge

Bei Digitalausgangsmodulen können Sie das Modul für das Einstellen der Ausgänge zu einem geplanten Zeitpunkt konfigurieren.

Dadurch steht dem Programmierer eine unerreichte Genauigkeit beim Verknüpfen von Ausgängen mit Zeitverweisen zur Verfügung, wie sie z. B. für Anwendungen wie Achsenpositionen in der Achssteuerung oder Prozesssteuerungsfunktionen benötigt wird, ohne die Kommunikations- und Logikverarbeitungssysteme und den zugehörigen Anwendungscode stark zu belasten.

Keine temporären Variablen

S7 verfügt über eine Kategorie von Variablen, die als temporäre Variablen bezeichnet werden. Ihr Gültigkeitsbereich ist der Programmblock, in dem sie definiert wurden, und ihre Gültigkeitszeit ist die Ausführung des Programmblocks, in dem sie definiert wurden.

Logix weist kein Äquivalent zu den temporären Variablen auf. Alle Variablen sind statisch – sie behalten ihren Wert bei, bis er geändert wird.

Um die meist in S7-Anwendungen gewünschten Funktionen zu erreichen, gehen Sie beispielsweise folgendermaßen vor:

- Verwenden Sie Programmbereich-Tags.
- Wenn Sie einen Add-On-Befehl programmieren, verwenden Sie lokale Tags (Teil der Add-On-Befehlsdaten).

Keine Akkumulatoren oder besonderen Zählwerke erforderlich

Wenn Sie in der STEP 7-Bemerkungsliste programmieren, werden Ihnen die Akkumulatoren und die Zeiger-Zählwerke AR1 und AR2 vertraut sein. In Logix gibt es dazu keine Äquivalente. Alle Operanden sind Tags.

Um die meist in S7-Anwendungen gewünschten Funktionen zu erreichen, gehen Sie beispielsweise folgendermaßen vor:

- Verwenden Sie Programmbereich-Tags.
- Wenn Sie einen Add-On-Befehl programmieren, verwenden Sie lokale Tags (Teil der Add-On-Befehlsdaten).
- Wägen Sie ab, ob Sie Logix-Äquivalente der Akkumulatoren und besonderen Zählwerke von S7 benötigen. Sie sind aufgrund der untergeordneten Natur der S7-Bemerkungsliste vorhanden. In einer Sprache wie strukturiertem Text sind sie mit hoher Wahrscheinlichkeit nicht erforderlich.

Konvertierung von Systemsoftware und Standardfunktionen

Einführung

In diesem Kapitel werden die am häufigsten verwendeten S7-Systemfunktionen aufgeführt, und es wird anhand zahlreicher spezifischer Beispiele erläutert, wie das Äquivalent in Logix erreicht wird.

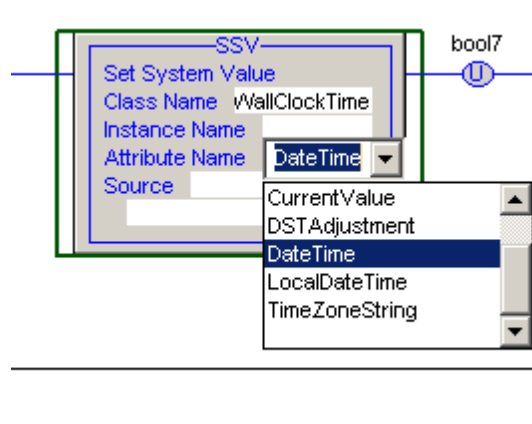
Thema	Seite
Logix-Systemfunktionen	68
Kopieren	68
Einstellen und Lesen des Datums und der Uhrzeit	69
Lesen der Systemzeit	69
Verarbeitung von Interrupts	70
Fehler	70
Status – Steuerung	71
Status – Modul	71
Status – für Bausteine und Tasks	72
Timer	72
Konvertierungsroutinen	73
Zeichenketten-Verarbeitungsroutinen	73
Beispiele für Systemfunktionsaufrufe	74

Dieses Kapitel hat den Zweck, Ihnen die in Logix verfügbaren dedizierten Befehle vorzustellen, sodass Sie keine Zeit mit der Entwicklung von Lösungen verschwenden, die bereits existieren.

Logix-Systemfunktionen

In Logix können die meisten S7-Systemfunktionen durch die Befehle GSV (Get System Value – Systemwert abrufen) und SSV (Set System Value – Systemwert festlegen) abgebildet werden. Diese Befehle greifen auf eine Hierarchie von Objekten (Klassen, Instanzen und Attribute) zu, die in Logix-Steuerungen integriert sind. Beim Programmieren von GSV und SSV wird Ihnen die Parameterauswahl durch Dropdown-Menüs erleichtert.

SSV-Befehl



Nachdem die Grundlagen von GSV und SSV verstanden sind, stellen viele neue Logix-Benutzer fest, dass der Zugriff auf das Betriebssystem einfacher als mit den sequenziellen Funktionsdiagrammen (SFC) in S7 ist.

Kopieren

Verwendet zum Kopieren komplexer Datenstrukturen – Datenfelder von Instanzen von Anwenderdatentypen.

S7	Kommentar	Logix	Kommentar
SFC20 BLKMOV	Bei BLKMOV müssen die Adressen zur Kompilierungszeit definiert werden.	COP (Befehl)	Wenn mit COP zwischen Datenfeldern kopiert wird, kann der Startpunkt des Blocks (Quelle oder Ziel) einen Datenfeldindex enthalten, der auf das Element verweist, dessen Wert zur Laufzeit ausgewertet wird.
SFC81 UBLKMOV	Nicht unterbrechungsfähige Version – stellt sicher, dass Quelldaten während des Kopiervorgangs nicht geändert werden können.	CPS (Befehl)	Nicht unterbrechungsfähige Version – stellt sicher, dass Quelldaten während des Kopiervorgangs nicht geändert werden können.
SFC14 DPRD_DAT	Wenn das Profibus DP-Gerät einen Kommunikations-Datenbereich > 4 Byte aufweist, stellt das SFC einheitliche Lesevorgänge sicher.	CPS (ControlNet und Ethernet /IP)	Für DeviceNet nicht erforderlich
SFC15 DPWR_DAT	Wenn das Profibus DP-Gerät einen Kommunikations-Datenbereich > 4 Byte aufweist, stellt das SFC einheitliche Schreibvorgänge sicher.	CPS (ControlNet und Ethernet /IP)	Für DeviceNet nicht erforderlich

Einstellen und Lesen des Datums und der Uhrzeit

Die Steuerungen beider Systeme weisen Echtzeituhren auf, die gelesen oder eingestellt werden können.

S7	Kommentar	Logix	Kommentar
SFC0 SET_CLK	Werte werden in einer Instanz des Typs DT (DateTime) übergeben.	SSV (Set System Value – Systemwert festlegen)	SSV-Klasse – WallClockTime SSV-Attribut – DateTime SSV-Quelle – Element[0] von DINT[7]
SFC1 READ_CLK	Werte werden in einer Instanz des Typs DT (DateTime) zurückgegeben.	GSV (Get System Value – Systemwert abrufen)	GSV-Klasse – WallClockTime GSV-Attribut – DateTime GSV-Ziel – Element[0] von DINT[7]

Lesen der Systemzeit

Die Steuerungen beider Systeme weisen eine Systemuhr auf, die beim Starten der Steuerung gestartet wird. Im S7-System wird die Zeit in Millisekunden angegeben, in Logix in Mikrosekunden.

S7	Kommentar	Logix	Kommentar
SFC64 TIME_TCK	Gibt die Systemzeit im Bereich von 0 bis 2,31 ms zurück	GSV (Get System Value – Systemwert abrufen)	Gibt die Systemzeit im Bereich von 0 bis 2,63 µs zurück GSV-Klasse – CST GSV-Attribut – CurrentValue GSV-Ziel – Element[0] von DINT[2] DINT[0] – untere 32 Bit DINT[1] – obere 32 Bit

Verarbeitung von Interrupts

Interrupts können im Anwenderprogramm durch Aufrufe von Systemfunktionen aktiviert und deaktiviert werden.

S7	Kommentar	Logix	Kommentar
SFC39 DIS_IRT	Deaktiviert Interrupts, die von einem bestimmten OB verarbeitet werden. Interrupt-Anforderungen gehen verloren.	SSV Sperrt die angegebene Task.	SSV-Klasse – Task SSV-Instanz – Task-Name SSV-Attribut – InhibitTask SSV-Quelle – auf 1 gesetzte DINT-Variable
SFC39 EN_IRT	Aktiviert Interrupts, die von einem bestimmten OB verarbeitet werden.	SSV Gibt die spezifizierte Task frei.	SSV-Klasse – Task SSV-Instanz – Taskname SSV-Attribut – InhibitTask SSV-Quelle – auf 0 gesetzte DINT-Variable
SFC41 DIS_AIRT	Deaktiviert Interrupts, die von einem bestimmten OB verarbeitet werden. Interrupt-Anforderungen werden verzögert.	UID	Deaktiviert die Unterbrechung der aktuellen Task durch eine Task höherer Priorität
SFC42 EN_AIRT	Aktiviert Interrupts, die von einem bestimmten OB verarbeitet werden. Durch SFC41 verzögerte Interrupts werden ausgeführt.	UIE	Aktiviert Interrupts der aktuellen Task

Fehler

Diese Systemaufrufe geben Bitfelder (S7) bzw. Ganzzahlen (Logix) zurück, in denen Fehlercodes enthalten sind.

S7	Kommentar	Logix	Kommentar
SFC38 READ_ERR	Lies und löscht Fehlerbits. Die Art des abzufragenden Fehlers kann durch ein Filterfeld ausgewählt werden.	GSV (Verwendung des SSV zum Zurücksetzen von Zählern oder Fehlern.)	GSV-Klasse – FaultLog GSV-Attribut: MajorEvents – Anzahl schwerwiegender Ereignisse MinorEvents – Anzahl geringfügiger Ereignisse MajorFaultBits – aktueller schwerwiegender Fehler MinorFaultBits – aktueller geringfügiger Fehler GSV-Ziel – INT oder DINT zum Empfangen von Daten

Status – Steuerung

Die SFC- (S7) und GSV-Aufrufe (Logix) geben Daten zur Steuerung zurück.
Hinweis – SFC51 erfordert eine gewisse Einarbeitung vor der Verwendung.
GSV ist in diesem Fall einfacher anzuwenden.

S7	Kommentar	Logix	Kommentar
SFC51 RDSYSST	Eingangsparameter geben die zu lesende Informationsklasse an, sowie möglicherweise eine Instanznummer, wenn mehrere Objekte vorhanden sind. Ausgangsparameter sind Zeiger auf eine Liste mit den zurückgegebenen Informationen, sowie die Anzahl und die Größe der Elemente in der Liste.	GSV	Module mit direkter Verbindung: Untersuchen der Elemente „Fault“ oder „ChannelFault“, falls vorhanden. Module mit für ein Rack optimierter Verbindung: Untersuchen des Elements „SlotStatusBits“ der Adapter- Eingangsdaten oder des Elements „Fault“ der Karte wie oben. Bei allen anderen Karten: Führen Sie GSV aus: Klasse – Module Instanz – ModuleName Attribut – EntryStatus

Status – Modul

Die SFC- (S7) und GSV-Aufrufe (Logix) geben Daten zu den installierten Modulen zurück.

S7	Kommentar	Logix	Kommentar
SFC51 RDSYSST	Eingangsparameter geben die zu lesende Informationsklasse an, sowie möglicherweise eine Instanznummer, wenn mehrere Objekte vorhanden sind. Ausgangsparameter sind Zeiger auf eine Liste mit den zurückgegebenen Informationen, sowie die Anzahl und die Größe der Elemente in der Liste.	GSV	GSV-Klasse – Module GSV-Attribut: EntryStatus (Beziehung des Modulobjekts zum Modul) FaultCode FaultInfo ForceStatus LEDStatus Mode (auch SSV) GSV-Ziel – abhängig vom ausgewählten Attribut

Sie können Fehlerinformationen in den Logix-Tags überwachen, die beim Einfügen des Moduls in die E/A-Konfiguration erstellt werden. Bei STEP 7 werden Fehlerinformationen für Module auf ähnliche Weise angezeigt, wenn Sie in die Hardware-Konfiguration wechseln und „Open ONLINE“ auswählen.

Status – für Bausteine und Tasks

S7	Kommentar	Logix	Kommentar
OB-Header	Statusdaten für Bausteine werden in temporären Variablen gespeichert, die im OB-Header automatisch generiert werden. Auf diese kann über den OB-Code direkt zugegriffen werden. Wenn ein Zugriff von außerhalb des OB erforderlich ist, können die Daten in statische Datenbereiche übertragen werden. Weiter unten finden Sie ein Beispiel.	GSV/SSV	GSV-Klasse – Task GSV-Instanz – Task-Name GSV-Attribut: DisableUpdateOutputs (am Ende der Task) EnableTimeOut InhibitTask Instance LastScanTime (Mikrosekunden) MaxInterval (zwischen aufeinander folgenden Ausführungen der Task) OverlapCount (bei der Ausführung ausgelöst) Priority Rate (Periode in Mikrosekunden) StartTime (Wert von WallClockTime beim letzten Starten der Task) Status (3 Statusbits) Watchdog (Mikrosekunden) GSV-Quelle/-Ziel – abhängig vom ausgewählten Attribut

Timer

S7	Kommentar	Logix	Kommentar
SFB4 TON	Einschaltverzögerter Timer	TON (LD) TONR (ST & FBD)	Einschaltverzögerter Timer
		RTO (LD) RTOR (LD & ST)	Remanenter einschaltverzögerter Timer
SFB5 TOF	Ausschaltverzögerter Timer	TOF (LD) TOFR (ST & FBD)	Ausschaltverzögerter Timer
SFB3 TP	Erzeugt einen Impuls, der bedingungslos weiter läuft	Bit des Speichers eines frei ausgeführten TON	

Konvertierungsroutinen

S7	Kommentar	Logix	Kommentar
Bibliotheksfunktionen		Befehle	
FC16 I_STRNG	Ganzzahl in Zeichenkette	DTOS	INT kann anstelle von DINT als Quell-Tag verwendet werden
FC5 DI_STRNG	Doppelganzzahl in Zeichenkette	DTOS	DINT in Zeichenkette
FC30 R_STRG	Realzahl in Zeichenkette	RTOS	Realzahl in Zeichenkette
FC38 STRG_I	Zeichenkette in Ganzzahl	DTOS	
FC37 STRG_DI	Zeichenkette in Doppelganzzahl	STOD	Zeichenkette in DINT
FC39 STRG_R	Zeichenkette in Realzahl	STOR	Zeichenkette in Realzahl

Zeichenketten-Verarbeitungs-routinen

S7	Kommentar	Logix	Kommentar
	Bibliotheksfunktionen		Befehle
FC10 EQ_STRNG	Zeichenketten auf Gleichheit überprüfen	EQU	Zeichenketten auf Gleichheit überprüfen
FC13 GE_STRNG	Zeichenketten auf >= überprüfen	GEQ (LD) >= (ST)	Zeichenketten auf >= überprüfen
FC15 GT_STRNG	Zeichenketten auf > überprüfen	GRT (LD)	Zeichenketten auf > überprüfen
FC19 LE_STRNG	Zeichenketten auf <= überprüfen	LEQ (LD) <= (ST)	Zeichenketten auf <= überprüfen
FC24 LT_STRNG	Zeichenketten auf < überprüfen	LES (LD) < (ST)	Zeichenketten auf < überprüfen
FC29 NE_STRNG	Zeichenketten auf <> überprüfen	NEQ (LD) <> (ST)	Zeichenketten auf <>überprüfen
FC21 LEN	Länge der Zeichenkette	.LEN	Eigenschaft jeder Zeichenketteninstanz
FC26 MID	Gibt einen Mittelbereich der Zeichenkette zurück	MID	Gibt einen Mittelbereich der Zeichenkette zurück
FC2 CONCAT	Verknüpft zwei Zeichenketten	CONCAT	Verknüpft zwei Zeichenketten
	Kann mit FC31 REPLACE erreicht werden	DELETE	Einen Bereich einer Zeichenkette löschen
FC17 INSERT	Quellzeichenkette in Zielzeichenkette einfügen	INSERT	Quellzeichenkette in Zielzeichenkette einfügen
FC31 REPLACE	n Zeichen der Zielzeichenkette durch Quellzeichenkette ersetzen	DELETE/INSERT verwenden	
FC11 FIND	Zeichenkette in anderer Zeichenkette suchen	FIND	Zeichenkette in anderer Zeichenkette suchen

In STEP 7 gibt es kein Äquivalent zu den Logix-Befehlen für die serielle ASCII-Schnittstelle – weder im Befehlssatz, noch in der Funktionsbibliothek. Diese müssen bei Bedarf in STL programmiert werden.

Beispiele für Systemfunktionsaufrufe

Diese Beispiele sind hauptsächlich zum Veranschaulichen der Verwendung von GSV/SSV-Befehlen gedacht.

Einstellen der Uhr

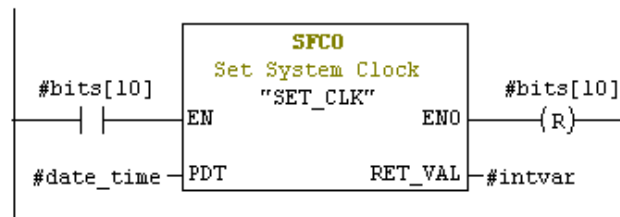
STEP 7

Dieser Aufruf von SFC0 stellt die Uhr ein. Uhrzeit und Datum werden in #date_time eingegeben.

Das Datum und die Uhrzeit werden in 8 Bytes nach #date_time im BCD-Format gespeichert.

Network 14 : Title:

```
set the clock to the value stored in "date_time"
```



0 – Jahr

1 – Monat

2 – Tag

3 – Stunde

4 – Minute

5 – Sekunde

6 – 2 höchstwertigste Ziffern der Millisekunden

7 – 1 niedrigstwertigste Ziffer für Millisekunden und Wochentag

Logix

Die Werte für Datum und Uhrzeit werden in den sieben DINTs nach #date_time gespeichert.



0 – Jahr

1 – Monat

2 – Tag

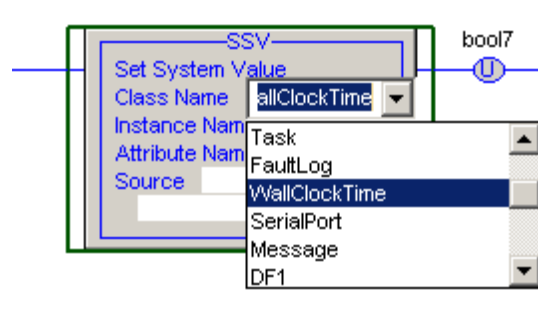
3 – Stunde

6 – Minute

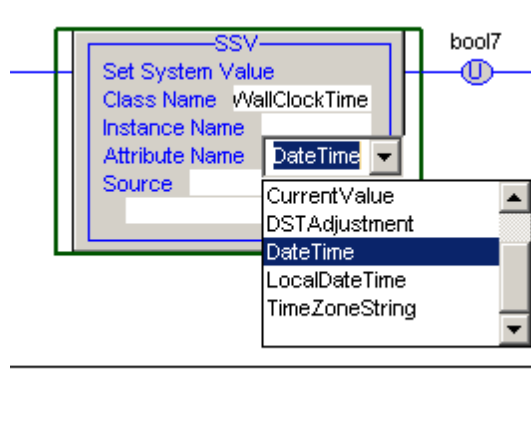
5 – Sekunde

6 – Mikrosekunde

Das Bildschirmfoto für Logix zeigt die Datenstruktur bei GSV und SSV. Wählen Sie die Klasse wie folgt in einem Pulldown-Menü aus.



Wählen Sie das Attribut wie folgt im Pulldown-Menü aus.



Wählen Sie schließlich das Tag aus, das die Quelle (SSV) bzw. das Ziel (GSV) für die Daten darstellt.

Deaktivieren von Interrupts

STEP 7

Network 2 : Title:

```
Disable interrupts for the Interrupt Execution (ie Periodic) 0B35
```

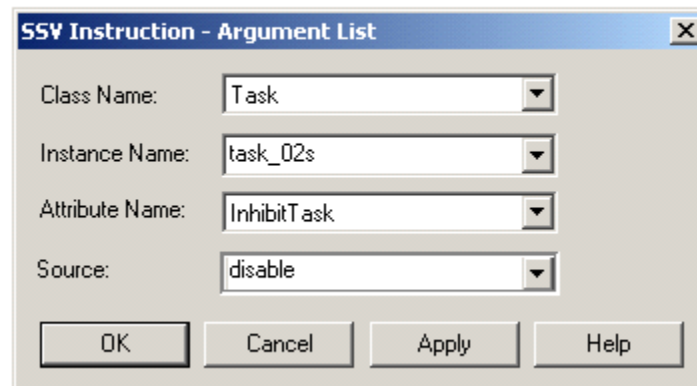
```
CALL "DIS_IRT"           SFC39           -- Disable New Int  
MODE   :=B#16#2  
OB_NR  :=35  
RET VAL:=#intVar
```

Logix

In diesem Beispiel wird SSV in strukturiertem Text gezeigt.

Wenn Sie zuerst „gsv“ eingeben und dann ALT-A drücken, wird der folgende Bildschirm für Parameterauswahl geöffnet.

```
// disable task_0.2s  
ssv().
```



The screenshot shows a dialog box titled "SSV Instruction - Argument List". It contains four dropdown menus for selecting parameters:

- Class Name: Task
- Instance Name: task_02s
- Attribute Name: InhibitTask
- Source: disable

At the bottom of the dialog are four buttons: OK, Cancel, Apply, and Help.

Klicken Sie nach dem Eingeben der Parameter auf „OK“. Die tatsächlichen Parameter werden nun vervollständigt.

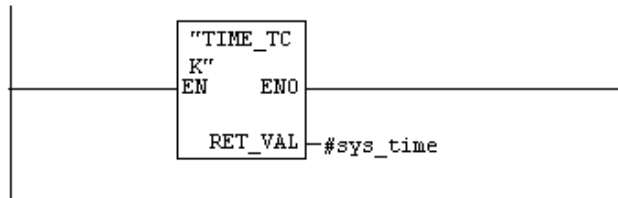
```
// disable task_0.2s  
ssv(Task,task_02s,InhibitTask,disable);
```

Lesen der Systemzeit

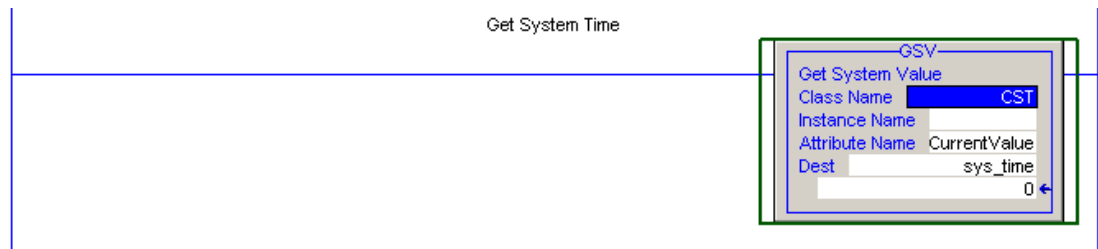
STEP 7

Network 15: Title:

```
read system time
```



Logix

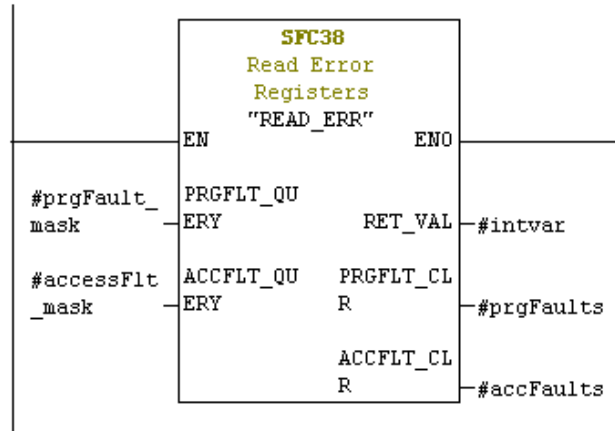


Abrufen von Fehlern

STEP 7

Network 16 : Title:

Get programming faults and I/O access faults



Das Bitmuster der Eingangsparameter bildet einen Filter zum Auswählen der abzufragenden Fehler. Die zurückgegebenen Fehler sind die **maskierten** Fehler – durch die Maskierung wird verhindert, dass die Steuerung abbricht oder einen Fehler-OB aufruft.

Logix



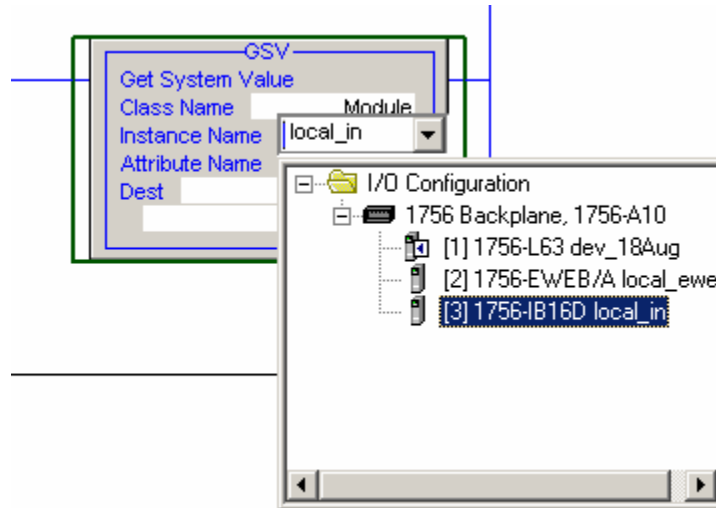
Modulinformationen

Die einfachste Möglichkeit besteht im Untersuchen der Geräteprofil-Tags für das Modul, die sämtliche Fehler-/Diagnoseinformationen enthalten.

Tag für Thermoelement-Analogeingangskarte 1756-IT612

Name	Alias For	Base Tag	Data Type
Local:4:C			AB:1756_AI6_Float:C:0
Local:4:I			AB:1756_AI6_CJ_Float:I:0
Local:4:I.ChannelFaults			INT
Local:4:I.Ch0Fault			BOOL
Local:4:I.Ch1Fault			BOOL
Local:4:I.Ch2Fault			BOOL
Local:4:I.Ch3Fault			BOOL
Local:4:I.Ch4Fault			BOOL
Local:4:I.Ch5Fault			BOOL
Local:4:I.ModuleFaults			INT
Local:4:I.AnalogGroupFault			BOOL
Local:4:I.InGroupFault			BOOL
Local:4:I.Calibrating			BOOL
Local:4:I.CalFault			BOOL
Local:4:I.CJUnderrange			BOOL
Local:4:I.CJOverrange			BOOL
Local:4:I.Ch0Status			SINT
Local:4:I.Ch0CalFault			BOOL
Local:4:I.Ch0Underrange			BOOL
Local:4:I.Ch0Overrange			BOOL
Local:4:I.Ch0RateAlarm			BOOL
Local:4:I.Ch0LAlarm			BOOL
Local:4:I.Ch0HAlarm			BOOL
Local:4:I.Ch0LLAlarm			BOOL
Local:4:I.Ch0HHAlarm			BOOL

Eine andere Möglichkeit ist das Lesen der Modulobjekte mit dem GSV-Befehl. Im nachstehenden Bildschirmfoto wird das Abrufen von Informationen zum Digitaleingangsmodul 1756-IB16D unter Verwendung des GSV gezeigt.



Abrufen der Abtastzeit

STEP 7

Dies ist ein Bildschirmfoto des Headers für temporäre Variablen in OB1.

Contents Of: 'Environment\Interface\TEMP'				
Name	Data Type	Address	Comment	
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)	
OB1_PRIORITY	Byte	2.0	Priority of OB Execution	
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)	
OB1_RESERVED_1	Byte	4.0	Reserved for system	
OB1_RESERVED_2	Byte	5.0	Reserved for system	
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)	
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)	
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)	
OB1_DATE_TIME	Date_And...	12.0	Date and time OB1 started	

#OB1_PREV_CYCLE ist die Abtastzeit. Als temporäre Variable existiert sie nach Abschluss der Ausführung von OB1 nicht mehr. Um die Abtastzeit zu speichern, kopieren Sie #OB1_PREV_CYCLE an einen statischen Speicherort.

Logix

Die Ausführungszeit kann für jede Logix-Task abgerufen werden.



Bei S7 können Sie die Abtastzeit für OB1 direkt aus #OB1_PREV_CYCLE entnehmen. Für periodische OBs gibt es jedoch kein Äquivalent zu #OB1_PREV_CYCLE. Um die Ausführungszeit für periodische Bausteine zu ermitteln, müssen Sie Aufrufe von SFC64 TIME_TCK zu Beginn und am Ende des OB einfügen und dann die vom SFC zurückgegebenen Systemuhrzeiten subtrahieren.

Konvertierung typischer Programmstrukturen

Einführung

In diesem Abschnitt soll vorgeführt werden, wie einige typische Programmierungsaufgaben in STEP 7 in der Software RSLogix 5000 durchgeführt werden können. Die Erläuterung beruft sich dabei überwiegend auf Codeausschnitte, es werden jedoch auch einige vollständige Beispiele angegeben.

Thema	Seite
Beispiele für konvertierten Code	83
Andere Themen bezüglich der Programmierung	120
Ein umfangreicheres Beispiel – Feldgerät	121

Es werden auch Themen der Programmierung angeschnitten, z. B. der Gültigkeitsbereich und die Sichtbarkeit von Variablen und die Zeitplanung von Codeabschnitten.

Beispiele für konvertierten Code In diesen Beispielen wird konvertierter Code gezeigt.

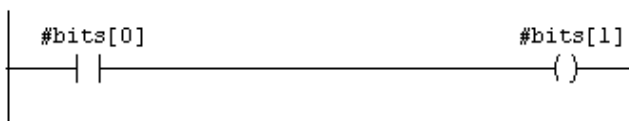
Übersetzung von Kontaktplanlogik

In diesem Abschnitt werden einige Beispiele für Vergleiche zwischen Kontaktplanlogik in STEP 7 (LAD) und in Logix (LD) aufgeführt.

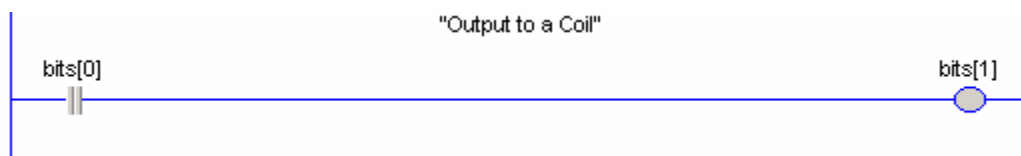
Schreiben in eine Spule

STEP 7

Output to a coil



LOGIX

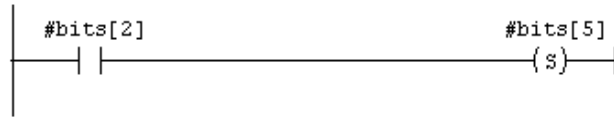


Setzen und Rücksetzen

STEP 7

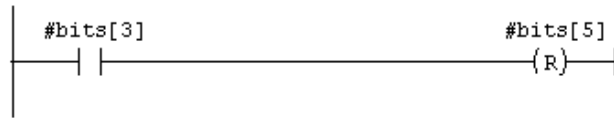
Network 3 : Title:

```
set bit
```

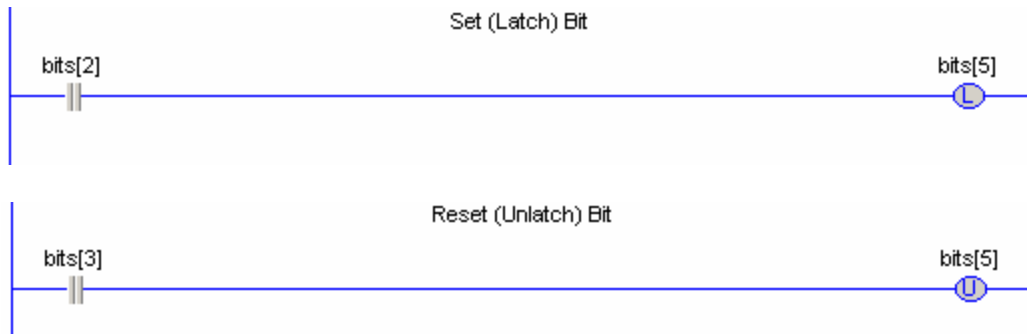


Network 4 : Title:

```
reset bit
```



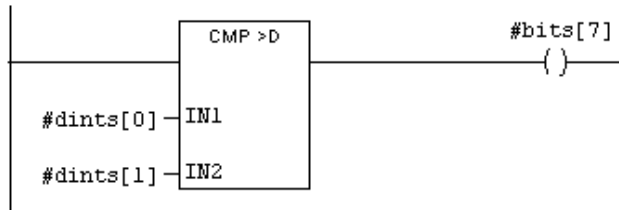
LOGIX



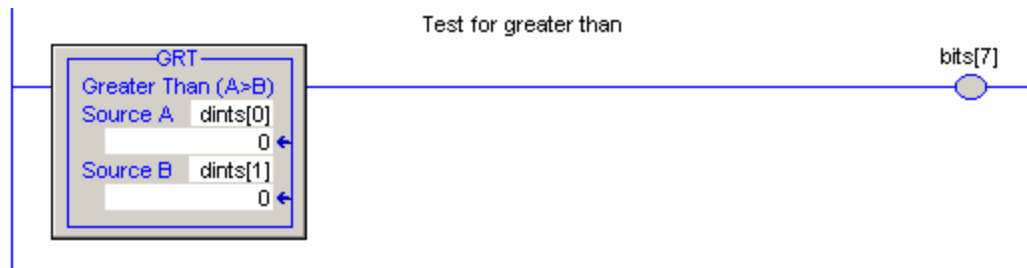
Testen auf „größer als“

STEP 7

test for greater than



LOGIX

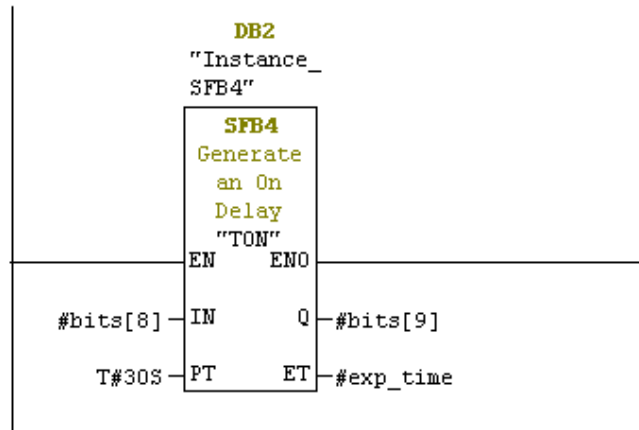


Nutzen Sie wieder den CMP-Befehl, wenn der Ausdruck komplexer als ein einfacher Vergleich zweier Zahlen ist.

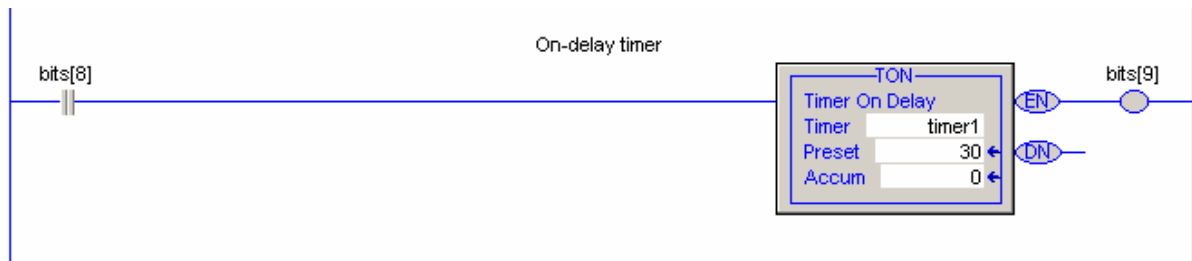
Einschaltverzögerter Timer

STEP 7

On delay timer



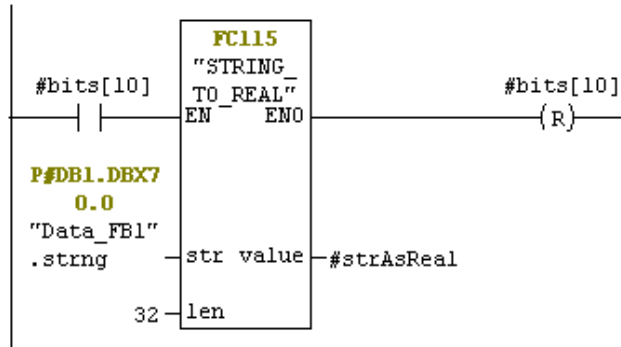
LOGIX



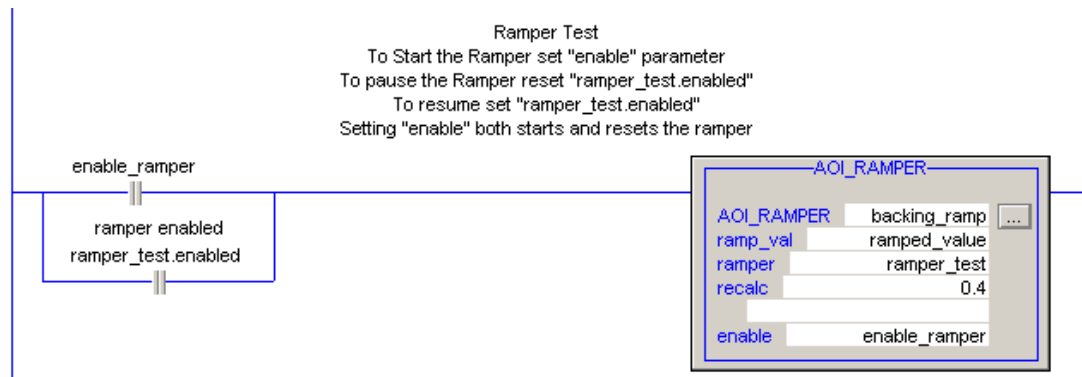
Aufruf einer Anwenderfunktion

STEP 7

user-function call

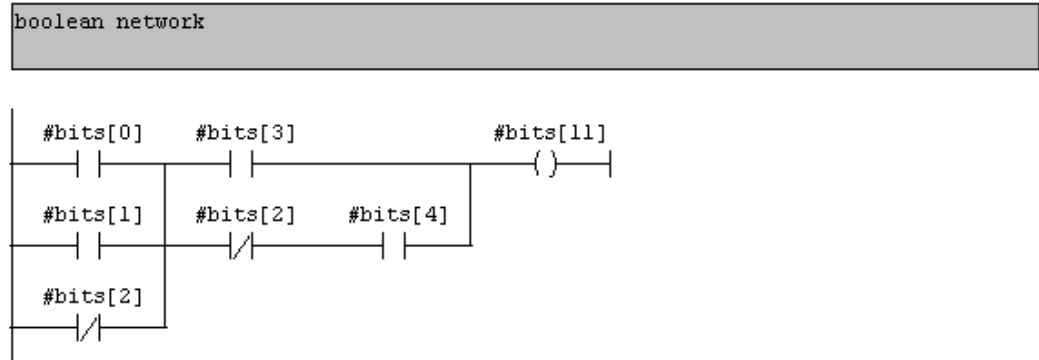


LOGIX

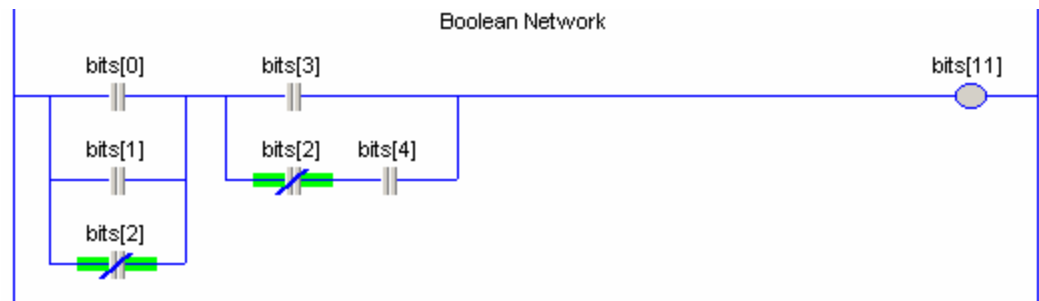


Boolesches Netzwerk

STEP 7



LOGIX

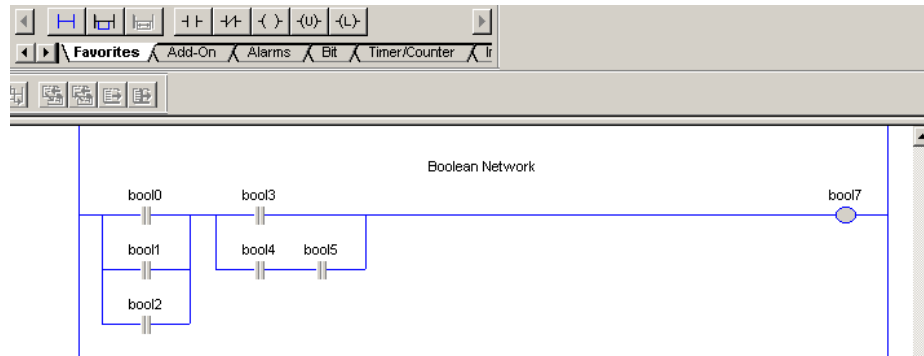


LAD in STEP 7 und LD in Logix weisen so viele Gemeinsamkeiten auf, dass die Übersetzung auf Routineebene recht geradlinig abläuft.

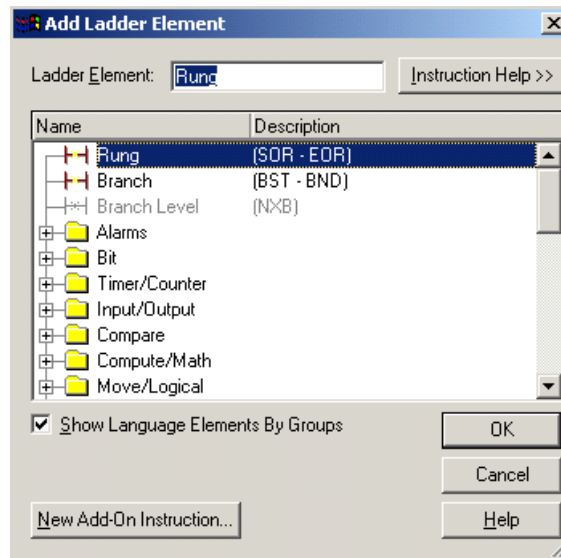
Der LD-Editor von Logix

LD-Befehle können auf sieben verschiedene Arten aufgerufen werden. Zwei Methoden, die der Vorgehensweise in STEP 7 sehr ähnlich sind, werden im Folgenden beschrieben.

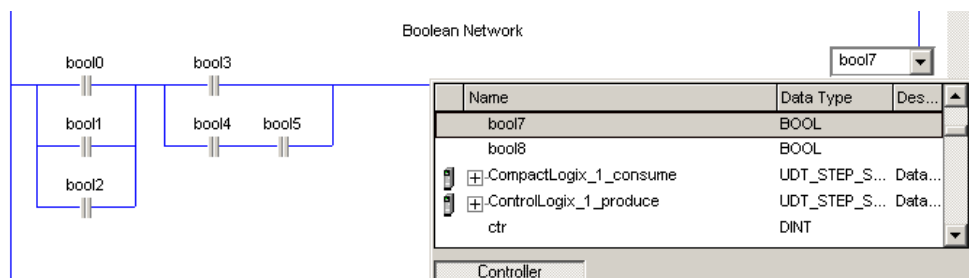
Sie können aus einer Palette über dem LD-Arbeitsblatt wählen.



Wenn Sie ALT+EINFG drücken, wird dieses Auswahlfenster eingeblendet.



Beim Konfigurieren von Befehlen sind Pulldown-Menüs verfügbar, in denen Sie das einzugebende Tag auswählen können.



Sprünge und Entscheidungen

STEP 7 – Herkömmliche Sprungabfolge

Die folgende Beispielaufgabe wird im Netzwerkkommentar erläutert. Es sind zwei S7 Versionen gezeigt, da beide häufig verwendet werden.

Network 1: Multi-way selection

```
if #input is 5 set #target to 8
else if #input is 6 set #target to 10
else if #input is 7 set #target to 16
else set #input to 0
```

```

L   #input
L   5
==I
JCN _001

L   8
T   #target

JU   end

_001: L   #input
L   6
==I
JCN _002

L   10
T   #target

JU   end

_002: L   #input
L   7
==I
JCN _003

L   16
T   #target

JU   end

_003: L   0
T   #target

end: NOP 0
```

Der Wert von #input wird mit den Konstanten verglichen, bis eine Übereinstimmung vorliegt. Dann wird die Aktion durchgeführt, und der Vergleichsvorgang endet. Wenn keine Übereinstimmung von #input mit den Werten im Satz vorliegt, wird eine Standardaktion durchgeführt.

STEP 7 – Sprungliste

In diesem Beispiel wird die gleiche Aufgabe ausgeführt, dazu wird jedoch eine Sprungliste verwendet. Dies ist ähnlich einer Sprungtabelle für einen Mikroprozessor, wobei hier die Ausführung entsprechend dem Wert einer Variablen an eine Marke übertragen wird.

Network 2 : Title:

```
if #input is 5 set #target to 8
else if #input is 6 set #target to 10
else if #input is 7 set #target to 16
else set #input to 0
```

```

L      #input
L      5
-I
JL     rng
JU     d5
JU     d6
JU     d7
rng: L  0
T      #target
JU     cont

d5: L  8
T      #target
JU     cont

d6: L  10
T      #target
JU     cont

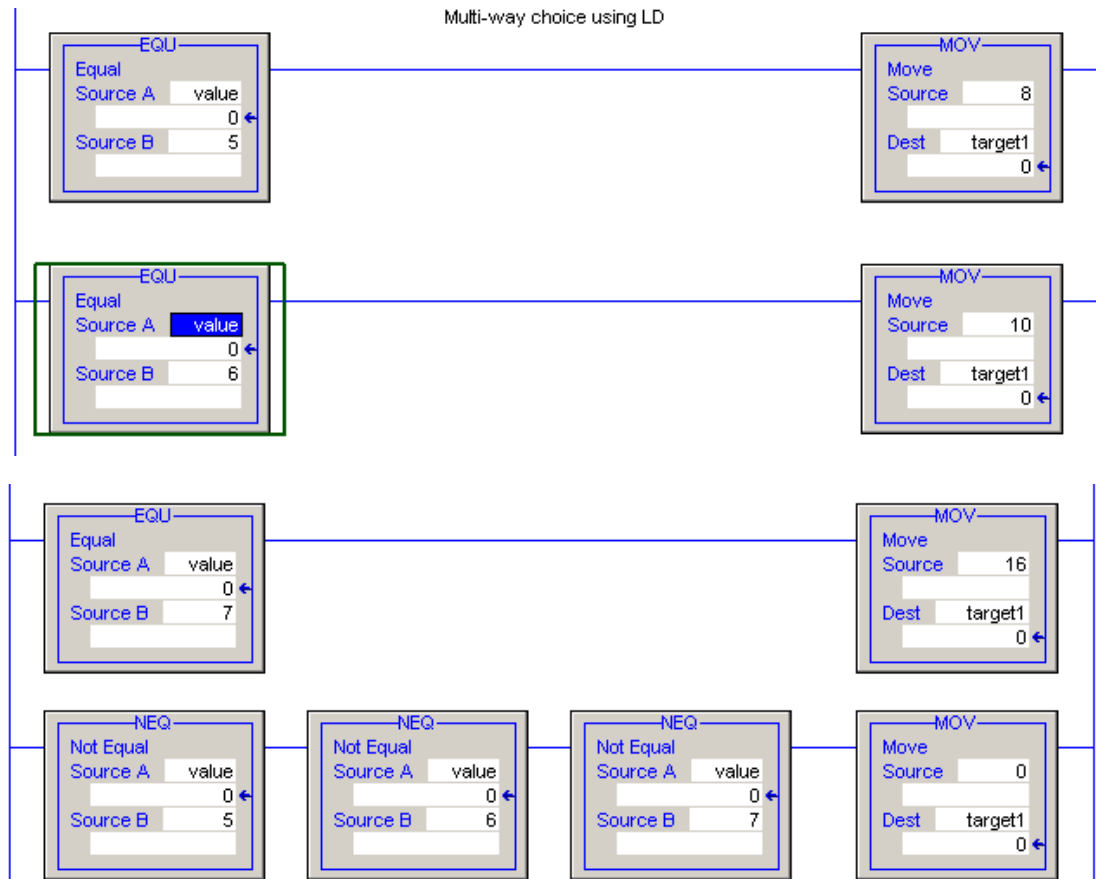
d7: L  16
T      #target

cont: NOP 0
```

Dies ist lesbarer als die herkömmliche Sprungabfolge und auch effizienter, da nur der Code an der Zielmarke ausgeführt wird.

Logix – Kontaktplanlogik

Hier wird eine Auswahl mit mehreren Möglichkeiten unter Verwendung von LD gezeigt.



Logix – Strukturierter Text – If ... Then ... Else

Mit einer Programmiersprache der Familien Basic/Pascal/C vertraute Programmierer werden dieses Beispiel sofort verstehen.

```
// multi-way choice using Structured Text

if (value = 5) then target := 8;
elsif (value = 6) then target := 10;
elsif (value = 7) then target := 16;
else target := 0;
end_if;
```

Die Klammern um die „if“-Bedingung sind nicht zwingend.

Logix – Strukturierter Text – CASE-Anweisung

Dies ist eine weitere Variante in ST für die gleiche Aufgabe. Da sie recht kompakt und deutlich ist, sind keine weiteren Kommentare notwendig.

```
// multi-way choice using Structured Text CASE

case value of
  5: target := 8;
  6: target := 10;
  7: target := 16;
else target := 0;
end_case;
```

Alle Lösungen funktionieren, dies ist jedoch die bevorzugte Lösung in Logix. Da sie recht kompakt und deutlich ist, ist keine weitere Dokumentation notwendig.

Datenfelder

STEP 7 und Logix unterstützen beide die Erstellung von Datenfeldern einfacher oder komplexer Objekte im Speicher. Logix bietet Unterstützung auf höherer Ebene für den Zugriff auf Datenfelder. In STEP 7 wird jedoch Programmierung auf niedriger Ebene benötigt.

Datenfelderstellung in STEP 7

Auf dem folgenden Bildschirmfoto sind zwei Datenfelder zu erkennen, die in einem instanziierten Datenbaustein erstellt wurden. Simple_array ist ein Datenfeld von 10 Elementen. UDT_array ist ein Datenfeld von 10 Strukturen des Typs test_UDT1, wobei test_UDT1 ein nicht gezeigter Anwenderdatentyp ist, der einige andere Typen enthält.

Contents Of: 'Environment\Interface\STAT'				
Name	Data Type	Address	Initial Value	
input	Int	6.0	0	
target	Int	8.0	0	
simple_array	Array [0..9...]	10.0		
UDT_array	Array [0..9...]	50.0		
state	Int	170.0	0	
error	Bool	172.0	FALSE	

Datenfelderstellung in Logix

Dies funktioniert in Logix genau gleich.

Scope: program_02s						
Name	Alias For	Base Tag	Data Type	Style	Description	
target			DINT	Decimal		
value			DINT	Decimal		
simple_array			DINT[10]	Decimal		
UDT_array			test_UDT1[10]		For testing Step7->Controllo...	
index			DINT	Decimal		

Syntax der Datenfelddeklaration

In STEP 7 wird die Deklarationssyntax „ARRAY[0...15] OF REAL“ verwendet. In Logix wird „REAL[15]“ verwendet.

STEP 7 weist eine spezielle Syntax für Zeichenketten auf. In STEP 7 ist „STRING[32]“ eine Zeichenkette mit 32 Zeichen, in Logix dagegen ist „STRING[32]“ ein Datenfeld von 32 Zeichenketten, die je 82 Zeichen aufweisen.

Zugriff auf Datenfelder in STEP 7

In diesem Beispiel wird eine einfache Aufgabe mit den beiden Datenfeldern `simple_array[]` und `UDT_array[]` ausgeführt. Die Aufgabe wird im Netzwerkkommentar beschrieben.

In STEP 7 ist es nicht möglich, mit der normalen Notation „array[]“ auf ein Datenfeld zuzugreifen. Stattdessen müssen Sie Zeigerbefehle auf niedriger Ebene verwenden. Im nachstehenden Ausschnitt wird die Aufgabe durch die Funktion „GET_INDEXED_REFERENCE“ stark vereinfacht, indem ein Zeiger für den Zugriff auf das Datenfeld zurückgegeben wird.

Network 3: Title:

```
array operations
-----
if (simple_array[2] = simple_array[5]) then
  UDT_array[8].boolean1 := 1;
else
  UDT_array[8].boolean1 := 0;
end_if;

// 1. compare simple_array[2] with simple_array[5]
CALL "GET_INDEXED_REFERENCE"          FC111
  refArray := "Data_test".simple_array P#DB1.DBX10.0
  index    := 2
  byteIncr := 4
  startIndex:=FALSE
  retVal   := #ptr1                                [BOOL]

CALL "GET_INDEXED_REFERENCE"          FC111
  refArray := "Data_test".simple_array P#DB1.DBX10.0
  index    := 5
  byteIncr := 4
  startIndex:=FALSE
  retVal   := #ptr2

OPN "Data_test"                               DB1
L   DID [#ptr1]
L   DID [#ptr2]
==D
=   #compare

// 2. get pointer to UDT_array
CALL "GET_INDEXED_REFERENCE"          FC111
  refArray := "Data_test".UDT_array P#DB1.DBX44.0
  index    := 8
  byteIncr := 12                                [INT]
  startIndex:=FALSE
  retVal   := #ptr1

L   #ptr1
LAR1

// 3. set or reset the bit
A   #compare
=   DIX [AR1,P#0.0]
```

In diesem Fall wurde der tatsächliche Logix-Code in strukturiertem Text als Netzwerkkommentar verwendet, um zu demonstrieren, wie intuitiv der Code in Logix ist.

STEP 7 – Durchwandern von Datenfeldelementen in einer Schleife

In diesem Beispiel soll das Fließkommafeld in jeder Struktur von UDT_array[] gelöscht werden. Dies ist nicht schwierig, es sind jedoch solide Fähigkeiten bei der Verwendung von Zeigern erforderlich.

```

Array Operations
-----
Clear all float elements at UDT offset P#6.0 in array UDT_array

```

```

// transfer pointer to UDT_array to AR1
L    P##UDT_array
LAR1
// initialise counter
L    0
T    #ctr

// end if #ctr > 9
loop: L    #ctr
      L    9
      >I
      JC    end2
// clear the float field at offset p#6.0
L    0.000000e+000
T    DID [AR1,P#6.0]
// increment AR1 by size of the UDT
+AR1 P#12.0
// increment counter
L    #ctr
INC  1
T    #ctr
// loop back
JU   loop

end2: NOP  0

```


Logix – Datenfeldvorgänge in strukturiertem Text

Mit dem folgenden ST-Ausschnitt wird die in den voranstehenden beiden Abschnitten beschriebene Aufgabe durchgeführt.

```
// array access in ST
if (simple_array[2] = simple_array[5]) then
    UDT_array[8].boolean1 := 1;
else
    UDT_array[8].boolean1 := 0;
end_if;

// clearing array elements
if (simple_array[0] = 5) then
    index := 0;
    while (index <= 9) do
        UDT_array[index].reall := 0.0;
        index := index + 1;
    end_while;
end_if;
```

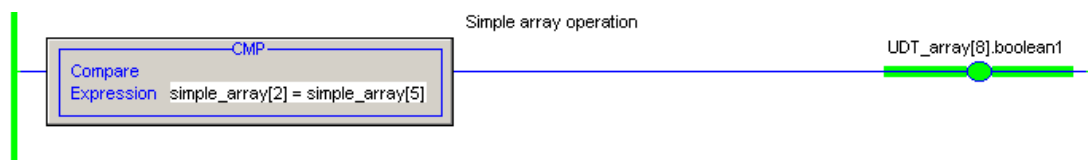
Zur Beschreibung der Funktionsweise sind keine weiteren Kommentare notwendig.

Wenn Sie sich dabei ertappen, dass Sie boolesche Variablen durch „if ... then ... else“-Bemerkungen austauschen, sollten Sie möglicherweise stattdessen eine boolesche Gleichung schreiben:

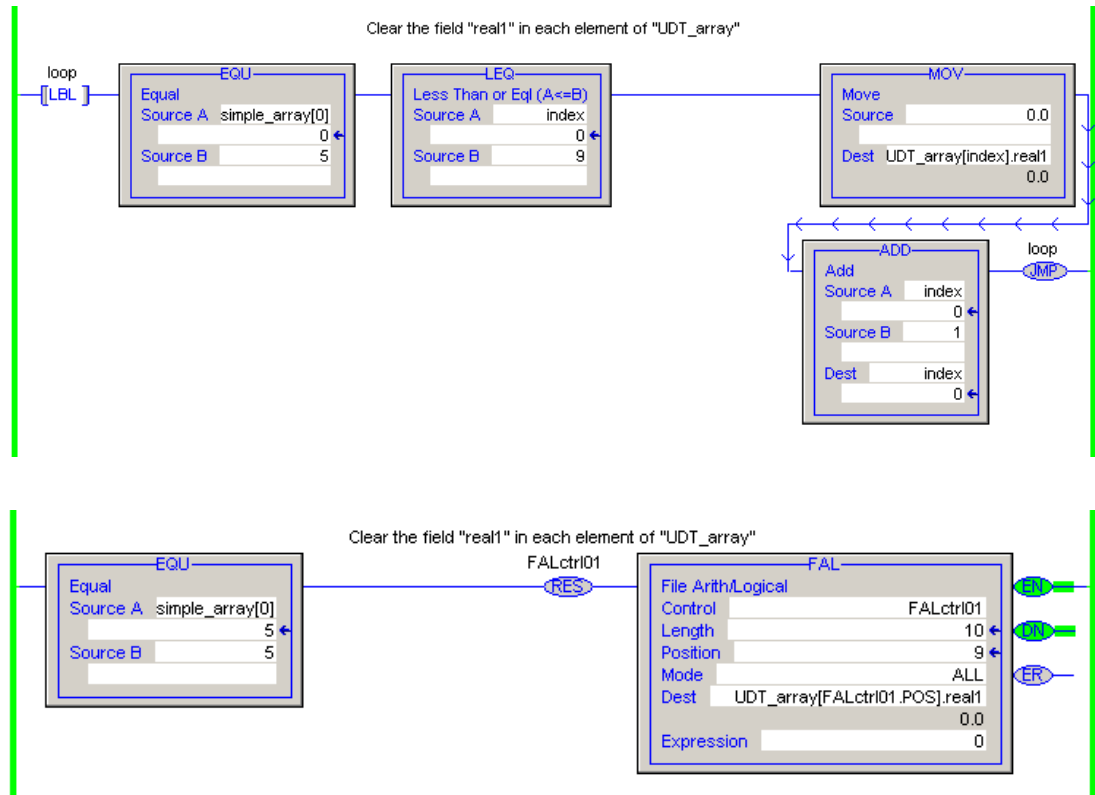
```
// array access in ST
UDT_array[8].boolean1 := simple_array[2] = simple_array[5];
```

Logix – Datenfeldvorgänge im Kontaktplan

Die Beispiele im vorgangegangenen Abschnitt können wie folgt in LD mit dem CMP-Befehl (Compare – Vergleich) geschrieben werden.



Das zweite (Löschen des Realzahlenfelds im Datenfeld von Anwenderdatentypen) kann auf diese beiden Arten erreicht werden.

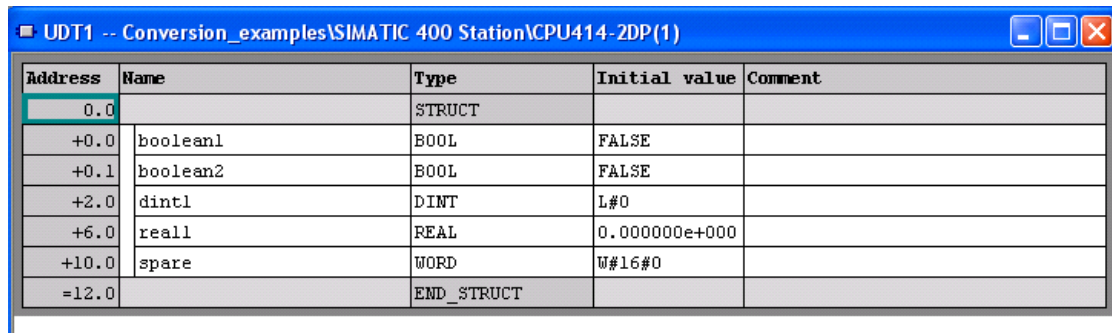


Die erste Herangehensweise an das Löschen der Datenfeldelemente ist eine Übersetzung der „While“-Schleife aus dem ST-Code. Bei der zweiten wird der erweiterte FAL-Befehl für Datenfeldvorgänge verwendet.

Anwenderdatentypen

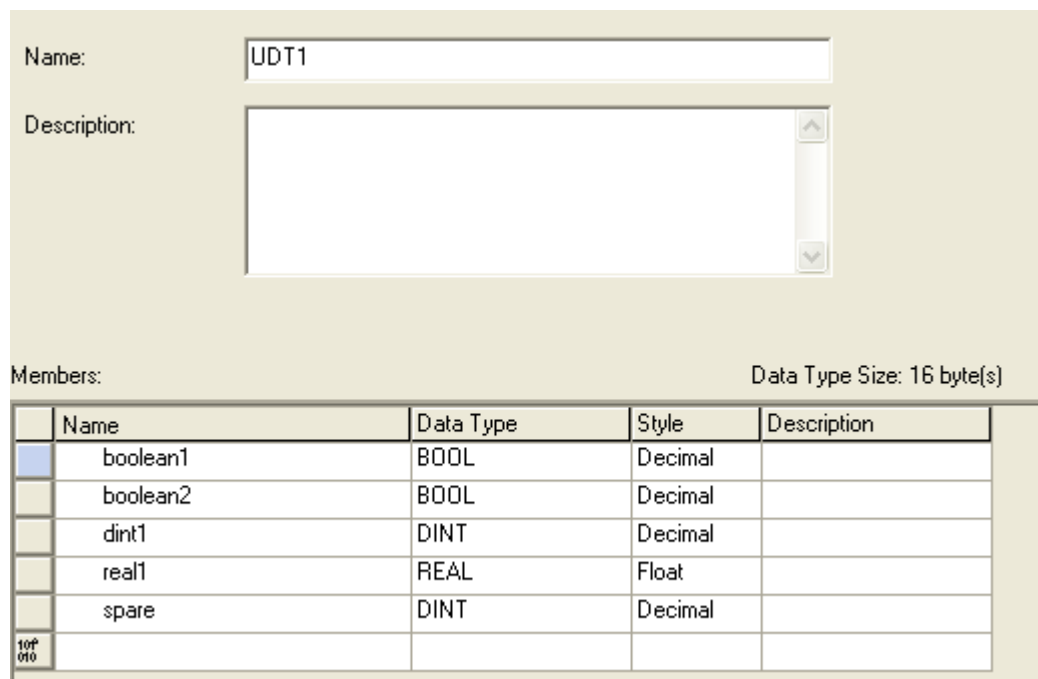
Das Konfigurieren und Verwenden von Anwenderdatentypen (UDTs) ist in STEP 7 und Logix sehr ähnlich.

Nachstehend ist ein UDT in STEP 7 abgebildet.



Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	boolean1	BOOL	FALSE	
+0.1	boolean2	BOOL	FALSE	
+2.0	dint1	DINT	L#0	
+6.0	real1	REAL	0.000000e+000	
+10.0	spare	WORD	W#16#0	
=12.0		END_STRUCT		

Nachstehend ist ein UDT in Logix abgebildet.



Name:

Description:

Members: Data Type Size: 16 byte(s)

	Name	Data Type	Style	Description
	boolean1	BOOL	Decimal	
	boolean2	BOOL	Decimal	
	dint1	DINT	Decimal	
	real1	REAL	Float	
	spare	DINT	Decimal	
10f 010				

In beiden Systemen können mit UDTs Variablen deklariert und definiert werden.

Hier ist eine Deklaration mit einem UDT in STEP 7.

Contents Of: 'Environment\Interface\STAT'

Name	Data Type	Address	Initial Value
input	Int	6.0	0
target	Int	8.0	0
simple_array	Array [0..9] Of Dint	10.0	
UDT_array	Array [0..9] Of UDT 1	50.0	

Hier ist eine Deklaration mit einem UDT in Logix.

Name	Alias For	Base Tag	Data Type	Style
Limit_Switch_1	Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal
Local:3:C			AB:1756_DI:C:0	
Local:3:I			AB:1756_DI:I:0	
conveyor_1			UDT1	
conveyor_1.boolean1			BOOL	Decimal
conveyor_1.boolean2			BOOL	Decimal
conveyor_1.dint1			DINT	Decimal
conveyor_1.real1			REAL	Float
conveyor_1.spare			DINT	Decimal

Zwischen den beiden Systemen besteht der folgende geringfügige Unterschied:

In STEP 7 können Variablen des Typs „struct“ deklariert werden.

Contents Of: 'Environment\Interface\STAT'

Name	Data Type	Address	Initial Value
target	Int	8.0	0
simple_array	Array [0..9] Of Dint	10.0	
UDT_array	Array [0..9] Of UDT 1	50.0	
state	Int	170.0	0
error	Bool	172.0	FALSE
transition01	Bool	172.1	FALSE
transition12	Bool	172.2	FALSE
transition13	Bool	172.3	FALSE
transition24	Bool	172.4	FALSE
transition43	Bool	172.5	FALSE
transition31	Bool	172.6	FALSE
str	String[46]	174.0	"
table	Struct	222.0	

Beachten Sie den Eintrag „table“ des Typs „Struct“. In „table“ können jegliche Kombinationen von Typen (geordnet oder ungeordnet) aufgenommen werden.

In Logix würde dazu „Struct“ als UDT mit der gewünschten Datenstruktur definiert und dann „table“ als Typ Struct deklariert werden.

Name	△	Alias For	Base Tag	Data Type	Style
Limit_Switch_1		Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal
+ Local:3:C				AB:1756_DI:C:0	
+ Local:3:I				AB:1756_DI:I:0	
- conveyor_1				UDT1	
- conveyor_1.boolean1				BOOL	Decimal
- conveyor_1.boolean2				BOOL	Decimal
+ conveyor_1.dint1				DINT	Decimal
- conveyor_1.real1				REAL	Float
+ conveyor_1.spare				DINT	Decimal
- table				Struct	
- table.status_bit1				BOOL	Decimal
- table.status_bit2				BOOL	Decimal
- table.status_bit3				BOOL	Decimal
+ table.dwell_timer1				TIMER	
+ table.dwell_timer2				TIMER	
- table.speed				REAL	Float

Zeiger und Datenfelder

Ein STEP 7-Programm kann Zeiger auf jedes Datenobjekt enthalten. Der indirekte Zugriff auf Datenbausteine ist ebenfalls zulässig, es gibt jedoch keine Zeiger auf Funktionen (außer auf eingeschränkte Weise durch den JL-Befehl (Jump List – Sprungliste)). Der Datenzeiger ist ungewöhnlich, da er einen Zeiger auf ein Bit darstellt. Daher ist sein Wert das Achtfache eines normalen Zeigers auf ein Byte. Dies gibt die große Bedeutung von Bits in der Programmierung von Steuerungssystemen wieder.

In Logix gibt es keine Zeiger. Datenfelder erfüllen die gleiche Funktion wie Zeiger, sind jedoch einfacher und sicherer.

Der S7-Programmierer kann sämtliche Aufgaben in Logix ohne Zeiger durchführen. In der Computerprogrammierung werden Zeiger auf Daten grundsätzlich zu drei Zwecken verwendet:

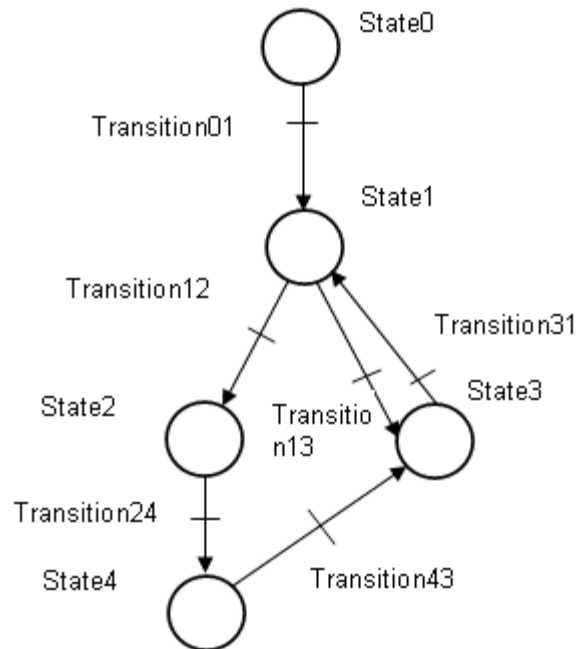
- Vorgänge mit sequenziell geordneten Datenelementen (Datenfelder von Objekten, Zeichenketten)
- Zuweisen, Zugreifen auf und Löschen von dynamisch zugewiesenen Objekten
- Übergeben von Verweisen auf Objekte als Parameter in Funktionsaufrufen

In Logix wird der erste Zweck durch Datenfelder erfüllt. Der zweite Zweck ist bei Steuerungssoftware nicht relevant, da keine dynamisch zugewiesenen Objekte vorhanden sind. Der dritte ist durch „inout“-Parameter in STEP 7-Funktionsblöcken und Logix-Add-On-Befehlen abgedeckt.

Daraus kann gefolgert werden, dass das Fehlen von expliziten Zeigern für Logix-Programmierer keine Einschränkung darstellt. STEP 7-Programmierer sollten auch erkennen, dass die Programmierung mit Datenfeldern in strukturiertem Text schneller möglich ist als in STL mit Zeigern.

Zustandsautomat

Der Zustandsautomat ist ein wichtiges Element in der Software für Steuerungssysteme, da er die Aufgabe der Programmierung einer sequenziellen Steuerung stark vereinfacht.



STEP 7-Zustandsautomat

STEP 7 bietet ein grafisches sequenzielles Funktionsdiagramm als optionalen Zusatz zur Grundanwendung. Wenn das grafische SFC nicht verfügbar ist, erfüllt die Bemerkungsliste diese Aufgabe.

Network 4: Title:

```
state machine
```

```

L      #state
JL     rngl
JU     st0
JU     st1
JU     st2
rngl: SET
S      #error
BEU

st0: L   1
      A   #transition01
      JC  next
      JU  ovr

st1: L   2
      A   #transition12
      JC  next

      L   3
      A   #transition13
      JC  next
      JU  ovr

st2: L   4
      A   #transition24
      JC  next
      JU  ovr

st3: L   0
      A   #transition31
      JC  next
      JU  ovr

st4: L   3
      A   #transition43
      JC  next
      JU  ovr

next: T   #state

ovr: NOP 0

```


Die Variable #state enthält die Zustandsnummer. Der Sprunglistenbefehl führt dazu, dass die Ausführung mit der für den Wert von #state relevanten Marke fortfährt. Wenn eine Transitionsbedingung von diesem Zustand „wahr“ ist, wird der neue Zustandswert in den Speicher geladen, und die Ausführung fährt mit der Marke „next“ fort, an der die neue Zustandsnummer an die Variable #state übertragen wird.

Logix – Zustandsautomat in strukturiertem Text

Hier ist der gleiche Zustandsautomat in strukturiertem Text mit der CASE-Bemerkung. Wie bei den anderen ST-Beispielen kann schwerlich eine verständlichere Beschreibung als der eigentliche Code abgegeben werden.

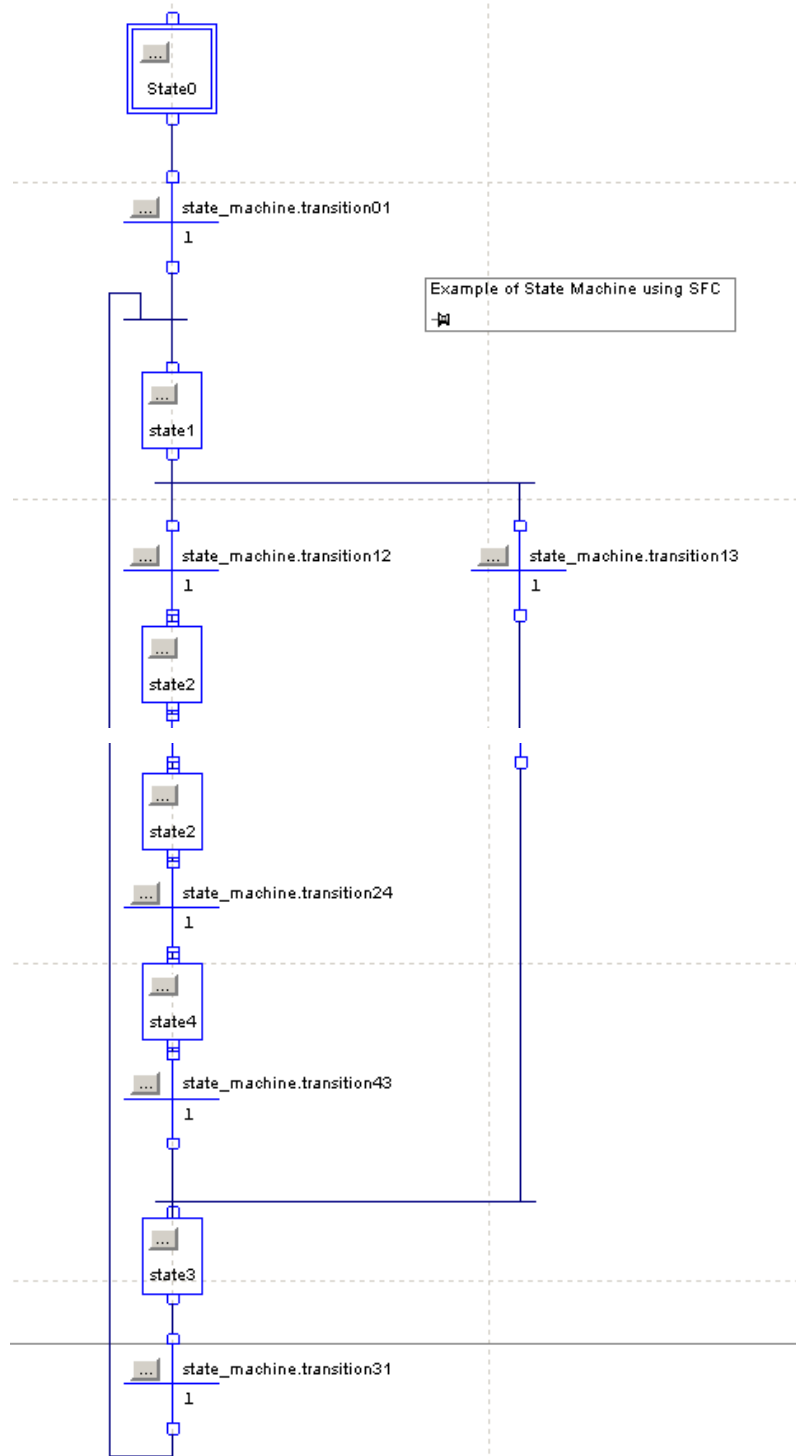
```
// implementation of State Machine using CASE in ST

case state_machine.state of
  0: if state_machine.transition01 then
      state_machine.state := 1;
    end_if;
  1: if state_machine.transition12 then
      state_machine.state := 2;
    elsif state_machine.transition13 then
      state_machine.state := 3;
    end_if;
  2: if state_machine.transition24 then
      state_machine.state := 4;
    end_if;
  3: if state_machine.transition31 then
      state_machine.state := 1;
    end_if;
  4: if state_machine.transition43 then
      state_machine.state := 3;
    end_if;
end_case;
```

Logix-Zustandsautomat in sequenziellem Funktionsdiagramm

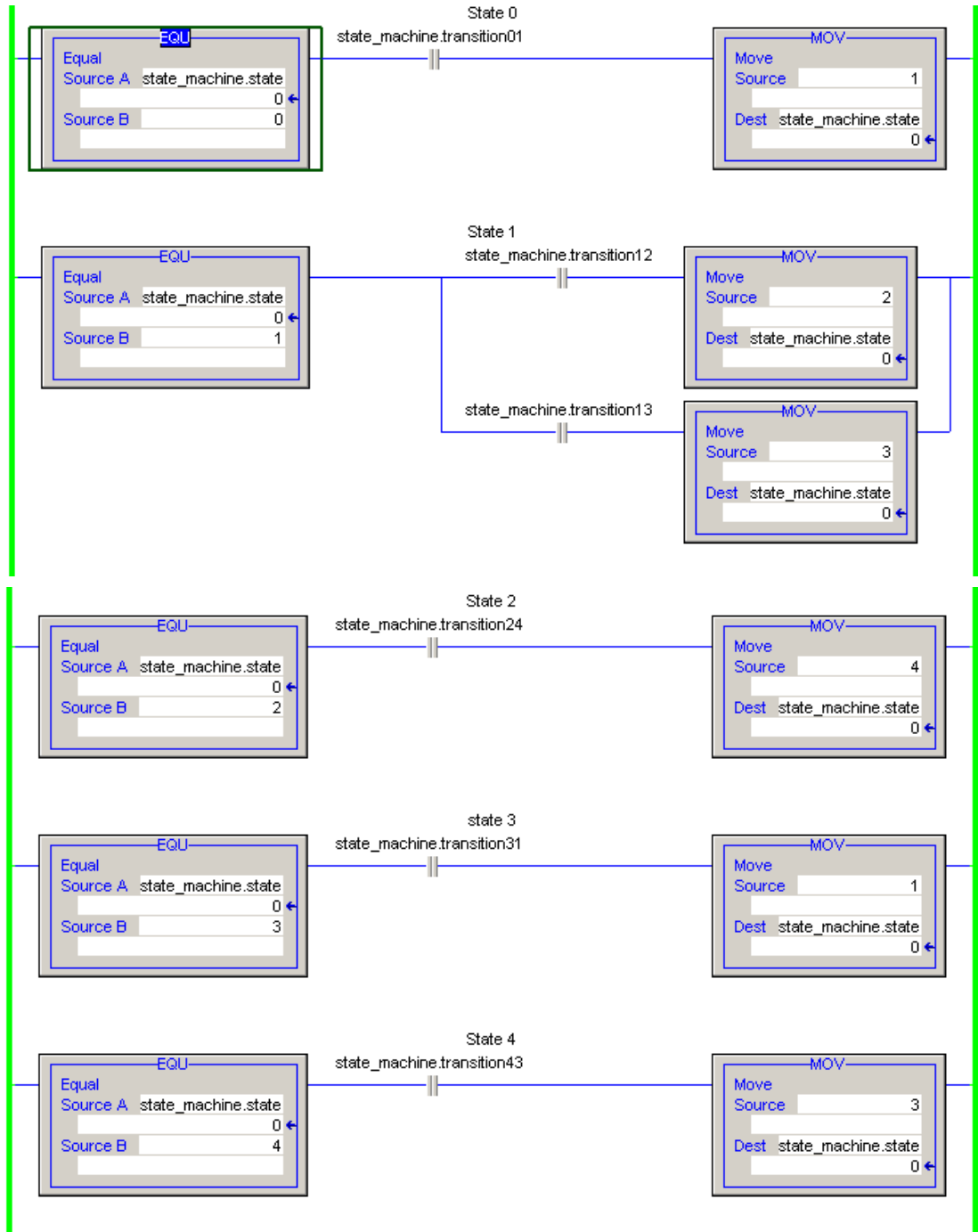
Logix bietet ein grafisches SFC als Teil der Standard-Suite von Sprachen. Nachstehend ist der Zustandsautomat in SFC dargestellt.

Implementierung des Zustandsautomaten mit der SFC-Tabelle



Zustandsautomat im Kontaktplan

Im nachstehenden Bildschirmfoto ist eine Implementierung des Zustandsautomaten in Kontaktplanlogik (LD) gezeigt.



Zeichenketten

Zeichenkettendefinition in STEP 7

Contents Of: 'Environment\Interface\STAT'				
Name	Data Type	Address	Initial Value	
transition12	Bool	172.2	FALSE	
transition13	Bool	172.3	FALSE	
transition24	Bool	172.4	FALSE	
transition43	Bool	172.5	FALSE	
transition31	Bool	172.6	FALSE	
str	String[46]	174.0	'This is an example string'	
table	Struct	222.0		

Der Daten-Header zeigt die Definition von Zeichenketten. Die Länge der Zeichenkette wird nach dem Datentyp „String“ in eckigen Klammern [] angegeben. Der Ausgangswert der Zeichenkette wird in der Spalte „Initial Value“ eingegeben.

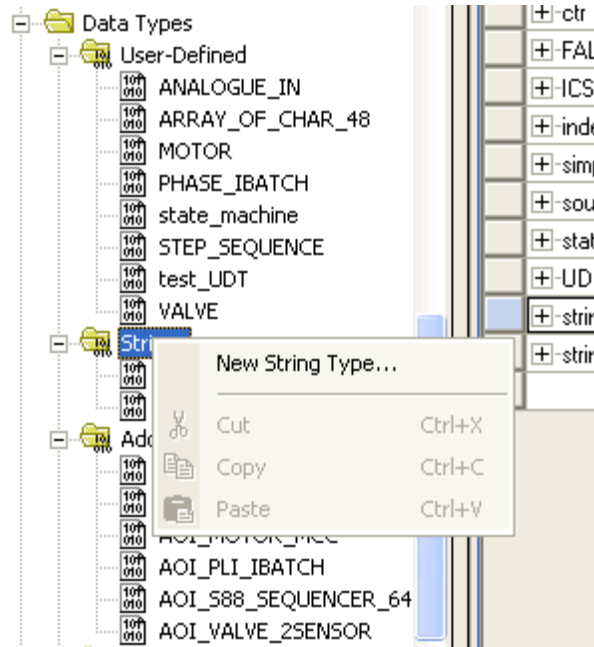
Es ist möglich ein Datenfeld von Zeichenketten zu erstellen, dabei kann jedoch nicht jede einen eigenen Ausgangswert erhalten. Eine alternative Definition zum Umgehen dieses Problems ist im Eintrag „table“ im Daten-Header gezeigt. „table“ ist eine Struktur. Den Inhalt der Struktur (nicht abgebildet) bilden fünf Instanzen von string[46], wobei jede einen Ausgangswert erhalten hat.

Zeichenkettendefinition in Logix

In dem nachstehenden Auszug aus der Tag-Konfigurationstabelle ist gezeigt, wie Zeichenketten in Logix definiert werden.

+UDT_array		test_UDT[10]
+string_of_82char		STRING
+string_of_40char		STRING 40

Wenn Sie eine Zeichenkette einer anderen Länge als 82 Zeichen (Standard) erstellen möchten, klicken Sie mit der rechten Maustaste in der Projektstruktur auf „strings“ (siehe unten).



Konfigurieren Sie dann die Eigenschaften wie unten gezeigt.

Name:

Description:

Maximum Characters:

Members: Data Type Size: 52 byte(s)

Name	Data Type	Style	Description
LEN	DINT	Decimal	
DATA	SINT[48]	ASCII	

Anschließend können Sie Instanzen des neuen Typs definieren.

<input type="checkbox"/> string_of_8zchar			STRING	
<input type="checkbox"/> string_of_48char			STRING_48	

Instanzen des Typs STRING oder STRING_48 weisen ein Feld „LEN“ auf, das beim Eingeben einer Zeichenkettenkonstante oder beim Manipulieren der Zeichenkette mit den Befehlen ASCII oder STRING automatisch aktualisiert wird.

Temporäre Variablen in STEP 7

Eine der Variablenkategorien in STEP 7 ist die temporäre Variable. Diese Variablen können nach Belieben in Bausteinen, Funktionen und Funktionsblöcken erstellt werden.

Temporäre Variablen werden für die lokale, vorübergehende Speicherung von Zwischenwerten sowie für Zeiger verwendet. Sie sind nur während der Ausführung ihres Blocks vorhanden, und nach Beendigung des Blocks gehen ihre Werte verloren.

Logix verfügt nicht über temporäre Variablen. Sämtlicher Speicher ist statisch, d. h. Werte werden zwischen den Ausführungen des Codes beibehalten.

Bei der Verwendung von Add-On-Befehlen fällt auf, dass für Add-On-Befehle lokale Variablen erstellt werden können. Diese Variablen können genau so verwendet werden wie temporäre Variablen.

Funktionen

Wenn ein Programmierer in STEP 7 die Bemerkungsliste verwendet, muss er eventuell Routinen auf niedriger Ebene entwickeln, die langwierig geschrieben und aufwändig getestet werden müssen. Funktionen sind wichtig, da die Entwicklung solcher Routinen nur einmal erledigt werden muss und anschließend der Ersteller der Funktion und andere Programmierer die gleiche Aufgabe in einem Bruchteil der Zeit erfüllen können.

In diesem Abschnitt wird beschrieben, wie Funktionen in Logix implementiert werden.

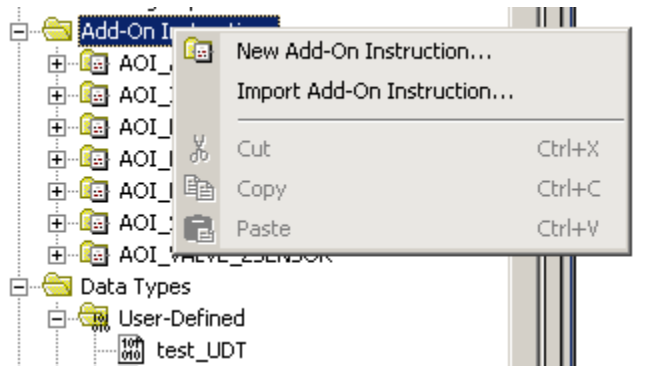
Funktion als Add-On-Befehl in Logix

STEP 7-Funktionen und -Funktionsbausteine weisen eine ähnliche Struktur auf wie der Logix-Add-On-Befehl. Der Add-On-Befehl hat die gleichen Parametertypen wie der FB (Input, Output und InOut) und verfügt über einen eigenen Datenbereich. Nachdem ein Add-On-Befehl programmiert und getestet wurde, kann er von jeder Stelle im Programm aus verwendet werden und ist in sich so weit abgeschlossen, dass er in andere Projekte exportiert oder in eine Code-Bibliothek übernommen werden kann.

Beispiel – Rampenfunktion

In diesem Beispiel wird eine Realzahlvariable linear in einer Rampe mit festgelegter Rate vom aktuellen Wert zu einem neuen Wert gebracht.

Wechseln Sie in den Zweig der Projektstruktur für Add-On-Befehle, und klicken Sie mit der rechten Maustaste auf „Add-On Instruction“.



Das gezeigte Formular wird eingeblendet.

New Add-On Instruction

Name:

Description:

Type:

Revision: Major Minor Extended Text

Revision Note:

Vendor:

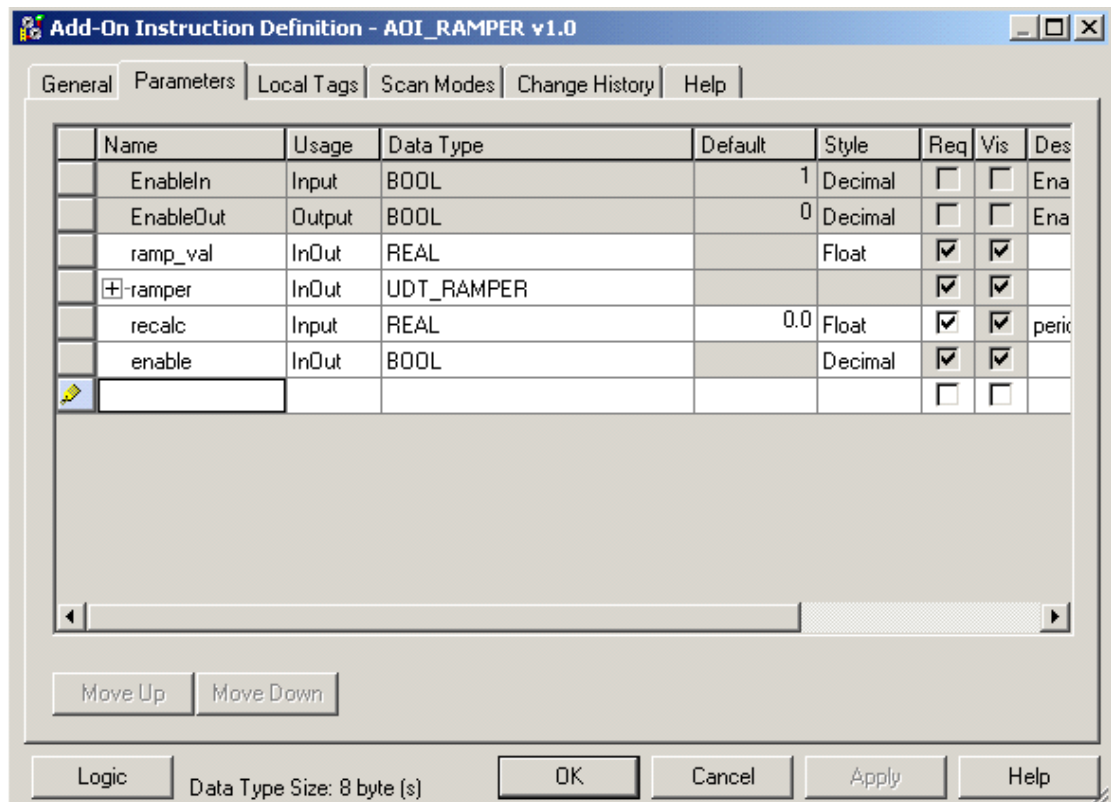
Open Logic Routine

Open Definition

Buttons: OK, Cancel, Help

Geben Sie den Namen des Add-On-Befehls ein, und legen Sie die Sprache fest, in der der Code geschrieben werden soll.

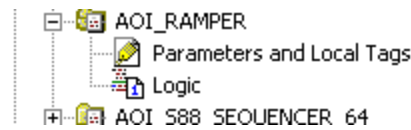
Wechseln Sie auf die Registerkarte „Parameters“.



Wie in STEP 7 sind die Eingangsparameter Werte vom Programm an den Add-On-Befehl, Ausgangsparameter sind Werte vom Add-On-Befehl an das Programm und InOut-Parameter dienen für Variablen, die durch den Add-On-Befehl geändert werden. Wenn Datenstrukturen vorhanden sind, wählen Sie in jedem Fall den Typ InOut, da diese per Verweis übergeben werden und dies effizienter ist.

Members:		Data Type Size: 28 byte(s)		
Name	Data Type	Style	Description	
initial_output	REAL	Float	saved initial output	
increment	REAL	Float	calculated increment	
RAMP_RATE_ABS	REAL	Float	per second - (set always +ve)	
RAMP_TARGET	REAL	Float	final value - (set)	
change	REAL	Float	calculated change over ramp	
counter	DINT	Decimal	internal counter	
complete	BOOL	Decimal	rampinn is complete	

In der Projektstruktur für AOI_RAMPER befindet sich ein Abschnitt „Logic“.



Öffnen Sie ihn, um den Code für diesen Add-On-Befehl anzuzeigen.

```

// Ramps a real variable from its current value to a new value at a
// specified rate.

// Parameters:
//   ramp_val   - variable to be ramped
//   ramper     - instance of UDT UDT_RAMPER
//   recalc     - code recalculation period (s)
//   enable     - start signal

// To use - set the target value in ramper.RAMP_TARGET_ABS
//         - set the ramp rate in ramper.RAMP_RATE_ABS
//         - to Start the Ramper set "enable" parameter
//         - to pause the Ramper reset "ramper.enabled"
//         - to resume set "ramper.enabled"
//         - setting "enable" both starts and resets the ramper

// on completion, the UDT field "complete" is set and the UDT field
// "enabled" is reset

// when enable is set, initialise
if (enable & (enable xor ramper._enable)) then
  ramper.initial_output := ramp_val;
  ramper.change := ramper.RAMP_TARGET - ramp_val;
  ramper.increment := ramper.change / abs(ramper.change)
                    * ramper.RAMP_RATE_ABS * recalc;

  ramper.counter := 0;
  ramper.complete := 0;
  enable := 0;
  ramper.enabled := 1;
end_if;

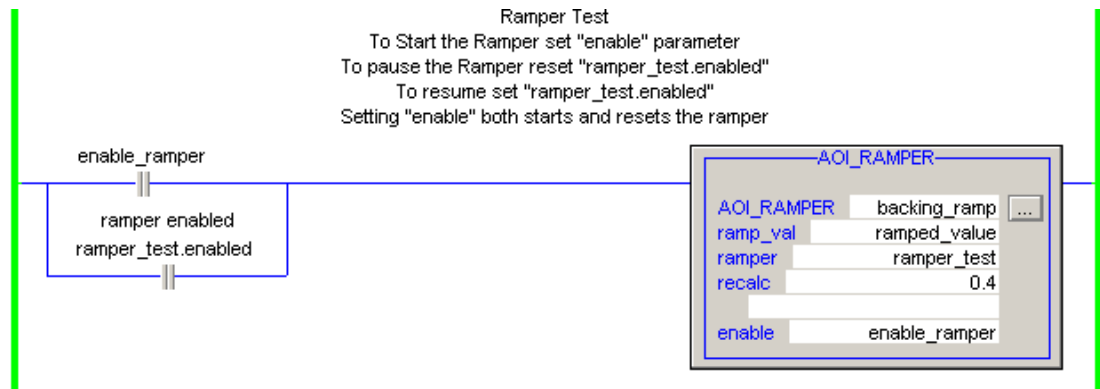
ramper._enable := enable;

// ramp calculations
if (ramper.enabled) then
  ramp_val := ramper.initial_output + (ramper.counter
                                       * ramper.increment);

  ramper.counter := ramper.counter + 1;
  if (abs(ramper.counter * ramper.increment)
      > abs(ramper.change)) then
    ramp_val := ramper.RAMP_TARGET;
    ramper.complete := 1;
    ramper.enabled := 0;
  end_if;
end_if;

```

Der Add-On-Befehl kann von jeder Routine aus aufgerufen werden.



Beachten Sie, dass Sie bei Add-On-Befehlen ein Tag des Typs „Add-On Instruction“ in einem Datenbereich erstellen müssen, der für die Routine sichtbar ist. Dieses wird als Backing-Tag bezeichnet.

Bevor Sie einen Add-On-Befehl schreiben, gehen Sie die Befehlshilfe in der Software RSLogix 5000 durch. Möglicherweise finden Sie dort einen Befehl, der die Aufgabe bereits erledigen kann. Im folgenden Abschnitt wird dies veranschaulicht.

Blockkopie, COP und CPS

In STEP 7 wird häufig mit der Systemfunktion SFC20 „BLKMOV“ ein Datenbaustein zwischen Speicherorten kopiert.

```

CALL "BLKMOV"
SRCBLK := "Data_EMI".stepMsgs.step5
RET_VAL := #intVar
DSTBLK := "Data_EMI".actualStep
    
```

Der Befehl kopiert die Zeichenkette von der fünften Position in einem Datenfeld von Zeichenketten zu einer Ziel-Zeichenkette.

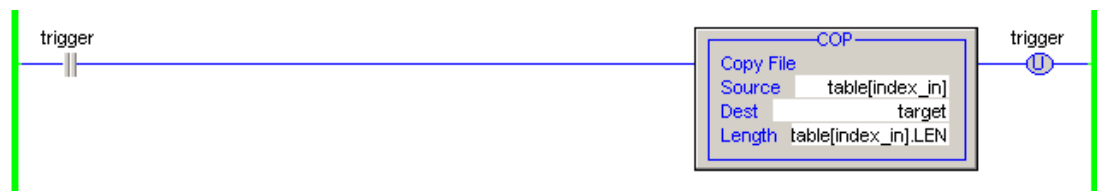
Oft soll das x-te Element aus dem Datenfeld kopiert werden, wobei „x“ während der Ausführung des Programms variieren kann. „BLKMOV“ ist dafür nicht geeignet.

Der STEP 7-Programmierer müsste zu diesem Zweck eine Funktion schreiben.

```
// copy step number descriptor to SCADA display area (EM faceplate)
CALL "INDEXED_COPY"
indexSource := "Data_EMs".EM1.stepNumber
sourceRef   := "Data_EM1".stepMsgs
indexDest   := 1
destRef     := "Data_EM1".actualStep
recordLength:=8           // 32 bytes
```

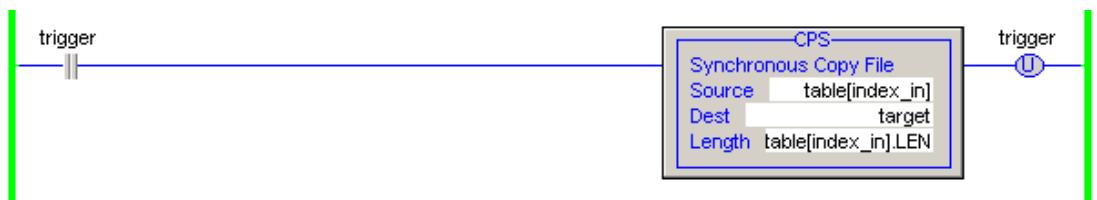
In diesem Fall betrifft der Kopiervorgang zwei Datenfelder, und die Indizes sind durch indexSource und indexDest definiert.

In Logix können alle diese Aufgaben über den integrierten COP-Befehl erfolgen.



Da die Angaben für die Quelle und das Ziel auch Indizes auf Variablendatenfelder sein können, ist der COP die geeignete Lösung. Das entspricht „INDEXED_COPY“.

Der CPS-Befehl gleicht COP, jedoch mit einem Unterschied.



Der Befehl kann nicht unterbrochen werden. Daher müssen die Quell- und Zieldaten während der Ausführung konstant bleiben. Wenn Sie Daten verschieben möchten, die sich möglicherweise ändern, verwenden Sie CPS.

Beispiele:

- Kopieren von Eingangsdaten in einen Puffer, in dem das Programm die Daten bearbeitet.
- Kopieren von konsumierten Tags in einen Puffer, in dem das Programm die Daten bearbeitet.

Mathematische Ausdrücke

In diesem Abschnitt wird beschrieben, wie der S7-Programmierer mathematische Berechnungen in Logix durchführen kann. Als Beispiel wird der Ausdruck „ $\sqrt{\cos(x)^2 + \sin(x)^2}$ “ verwendet. Das Ergebnis dieses Ausdrucks ist stets genau 1, daher kann einfach überprüft werden, ob der Code die richtige Antwort erzeugt.

STEP 7 – STL

Mathematischer Code ist in STEP 7 STL effizient, allerdings nicht unbedingt einfach für eine Person zu verstehen, die mit STL nicht vertraut ist.

Network 1: Title:

```
calculate:
```

```
(SIN(x)^2 + COS(x)^2) ^0.5
```

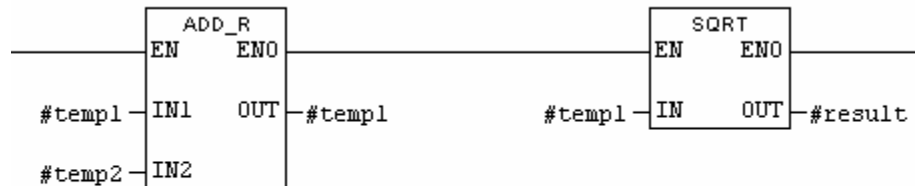
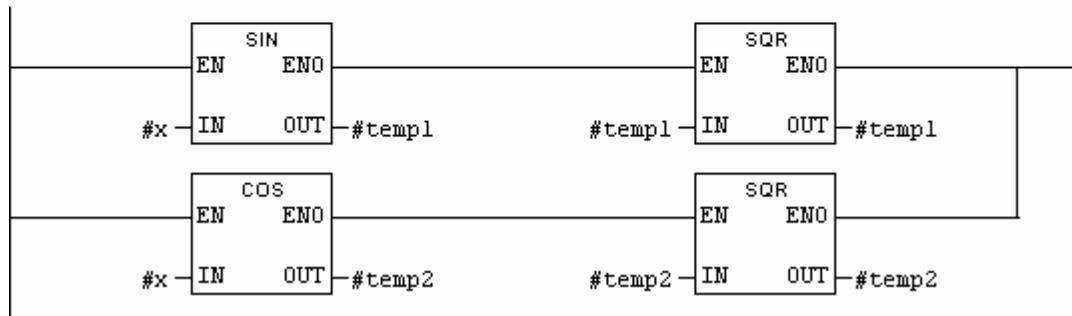
```
L      #x  
SIN  
SQR  
L      #x  
COS  
SQR  
+R  
SQRT  
T      #result
```

STEP 7 – LAD

Mathematische Auswertungen in LAD folgen einem konventionellen Muster der Verknüpfung von Funktionen.

Network 1: Title:

```
calculate:
(SIN(x)^2 + COS(x)^2) ^0.5
```



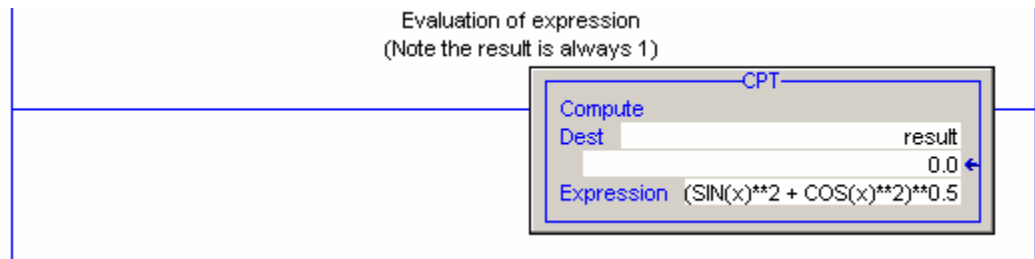
TEMP: REAL

Logix – ST

Der Ausdruck wird genau wie bei jeder anderen Hochsprache eingegeben.

```
// evaluation of mathematical expression in Structured Text
result := (SIN(x)**2 + COS(x)**2)**0.5;
```

Logix – LD



Durch den CPT-Befehl kann der Ausdruck auf hoher Ebene eingegeben werden, sodass die meisten Personen ihn einfacher verstehen als ein Netzwerk (einen Strompfad) von getrennten Befehlen.

STEP 7 – Anwenderfunktion

Dieser Funktionsbaustein erfüllt im Wesentlichen denselben Zweck wie CPT in Logix.

```
CALL "CPT" , "Data_CPT"           FB119 / DB119
  str := "test_formulae".test_string15 P#DB16.DBX476.0
  x   := 3.300000e+001
  a   := 0.000000e+000
  b   := 0.000000e+000
  c   := 0.000000e+000
  result:=#realVar
```

```
-- ((cos x)^2 + (sin x)^2) ^0.5 --
```

IN		OUT	
+3.300e+001			
+0.000e+000			
+0.000e+000			
+0.000e+000			
			+1.000e+000

Er liest eine Ausdruckszeichenkette, die in einem Datenbaustein gespeichert ist, und wertet diese aus. Im Vergleich mit Logix CPT hat er eine Einschränkung: Der Ausdruck muss in umgekehrter polnischer Notation geschrieben werden. Vielen Programmieren sagt diese Notation nicht zu.

Die Hauptprobleme beim Schreiben eines Funktionsbausteins wie diesem sind die benötigte Zeit und die vorausgesetzten Programmierkenntnisse. Bei Logix ist der CPT-Befehl sofort nach dem Installieren der Software RSLogix 5000 allgemein verfügbar.

Typenüberprüfung

Bei STEP 7 und Logix werden die Typen von Parametern für Funktionen, Funktionsblöcke, Befehle und Add-On-Befehle vom Compiler strikt geprüft.

Bei mathematischen Ausdrücken gibt es Unterschiede.

Logix unterscheidet zwischen numerischen und booleschen Werten. Der Compiler weist Ausdrücke ab, in denen numerische und boolesche Werte auf unlogische Weise gemischt sind. Wenn Ausdrücke von gemischtem numerischem Typ auftreten, werden diese so konvertiert, dass das Ergebnis den Typ der deklarierten Ergebnisvariable aufweist. Daher wird * als ganzzahlige Multiplikation interpretiert, wenn das Ergebnis eine Ganzzahl sein soll, und als Realzahl-Multiplikation, wenn das Ergebnis eine Realzahl sein soll.

In STEP 7 muss die Art der arithmetischen Vorgänge angegeben werden. Beispielsweise gibt es *I (Multiplizieren zweier 16-Bit-Ganzzahlen) *D (Multiplizieren zweier 32-Bit-Ganzzahlen) und *R (Multiplizieren zweier Realzahlen). Dabei muss der Programmierer sicherstellen, dass die beiden Operanden eines *R-Befehls Realzahlen sind. Ist dies nicht der Fall, beanstandet der Compiler dies nicht, das Ergebnis ergibt jedoch keinen Sinn.

Schlussfolgerung

Die Logix-Methodik zur Programmierung mathematischer Ausdrücke ist klarer, und durch die Trennung des mathematischen Codes von anderer Logik werden Tests und Gültigkeitsprüfungen vereinfacht.

Andere Themen bezüglich der Programmierung

Gültigkeitsbereich von Variablen

In diesem Bereich weicht Logix beträchtlich von STEP 7 ab.

Regeln für STEP 7

- Temporäre Variablen sind außerhalb des Blocks, in dem sie deklariert sind, nicht sichtbar.
- Globale statische Variablen sind im gesamten Programm sichtbar.
- Statische Variablen, die als Instanzdaten für einen Funktionsbaustein deklariert werden, weisen im FB einen besonderen Status auf, können jedoch von anderen Teilen des Programms abgerufen werden.

Regeln für Logix

Die Ausführung in Logix ist in Tasks unterteilt. Jede Task kann mehrere Programme aufweisen, und jedes Programm kann mehrere Routinen aufweisen. Jedes Programm kann einen eigenen Tag-Bereich haben.

- Tags mit Gültigkeitsbereich der Steuerung sind in allen Routinen aller Programme sichtbar.
- Tags mit Gültigkeitsbereich des Programms sind nur in den Routinen des Programms sichtbar, in dem sie definiert sind. Das bedeutet, dass eine Routine in einem Programm nur dann Daten mit einer Routine in einem anderen Programm gemeinsam verwenden kann, wenn es sich um Daten mit Gültigkeitsbereich der gesamten Steuerung handelt.
- Lokale Tags für Add-On-Befehle sind nur in der Logik dieses Add-On-Befehls sichtbar.

Bausteine, Tasks und Zeitplanung

Bausteine, Tasks und Zeitplanung werden in [Kapitel 2](#) beschrieben.

Ein umfangreicheres Beispiel – Feldgerät

In diesem Beispiel werden einige der in den voranstehenden Abschnitten beschriebenen Themen zusammengefasst. Der Begriff „Feldgerät“ (CM für engl. Control Module) stammt vom einflussreichen Batch-Steuerungs-Standard S88. S88 hat dazu geführt, dass Steuerungssoftware „objektorientierter“ geworden ist. Dieses Feldgerät ist für ein Binärventil ausgelegt. Für diese Art der Programmierung ist der Add-On-Befehl geeignet.

Komponenten des Feldgeräts (CM)

Folgende Komponenten sind vorhanden:

- ein Anwenderdatentyp (UDT) mit der Bezeichnung UDT_VALVE
- ein Add-On-Befehl mit der Bezeichnung AOI_VALVE_2SENSOR
- ein neues Programm unter „task_02s“ mit dem Namen „valves_callup“, das einen Programm-Tag-Abschnitt und eine Routine enthält

Anwenderdatentyp „Valve“

Der UDT ist nachstehend dargestellt.

Name:

Description:

Members: Data Type Size: 24 byte(s)

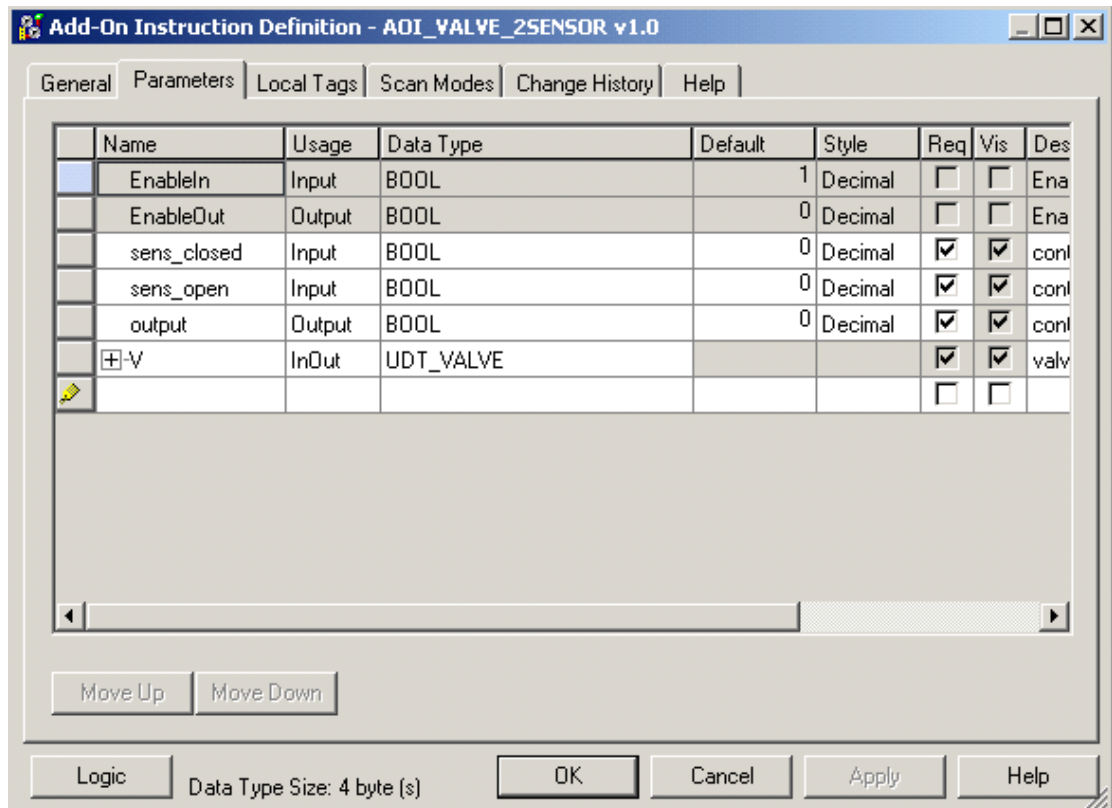
	Name	Data Type	Style	Description
	opening_preset	DINT	Decimal	max preset for opening
	closing_preset	DINT	Decimal	max preset for closing
	state	DINT	Decimal	state of valve (for internal logic)
	state_saved	DINT	Decimal	for evaluation of edge
	timecount	DINT	Decimal	valve timer
	auto	BOOL	Decimal	auto mode (set from SCADA)
	manual	BOOL	Decimal	manual mode (set from SCADA)
	closed	BOOL	Decimal	state of valve
	open	BOOL	Decimal	state of valve
	fault_closing	BOOL	Decimal	closed sensor feedback not received
	fault_opening	BOOL	Decimal	open sensor feedback not received
	fault_sensors	BOOL	Decimal	sensors and logical state of valve do not agree
	acquired	BOOL	Decimal	acquired by EM
	interlocked	BOOL	Decimal	interlocked - de-energise
	fail_open	BOOL	Decimal	property - fails open

Die Erstellung des UDT sollte der erste Schritt sein – er enthält alle Daten, die für die Modellierung des Ventils erforderlich sind.

Der Add-On-Befehl

Parameter des Add-On-Befehls

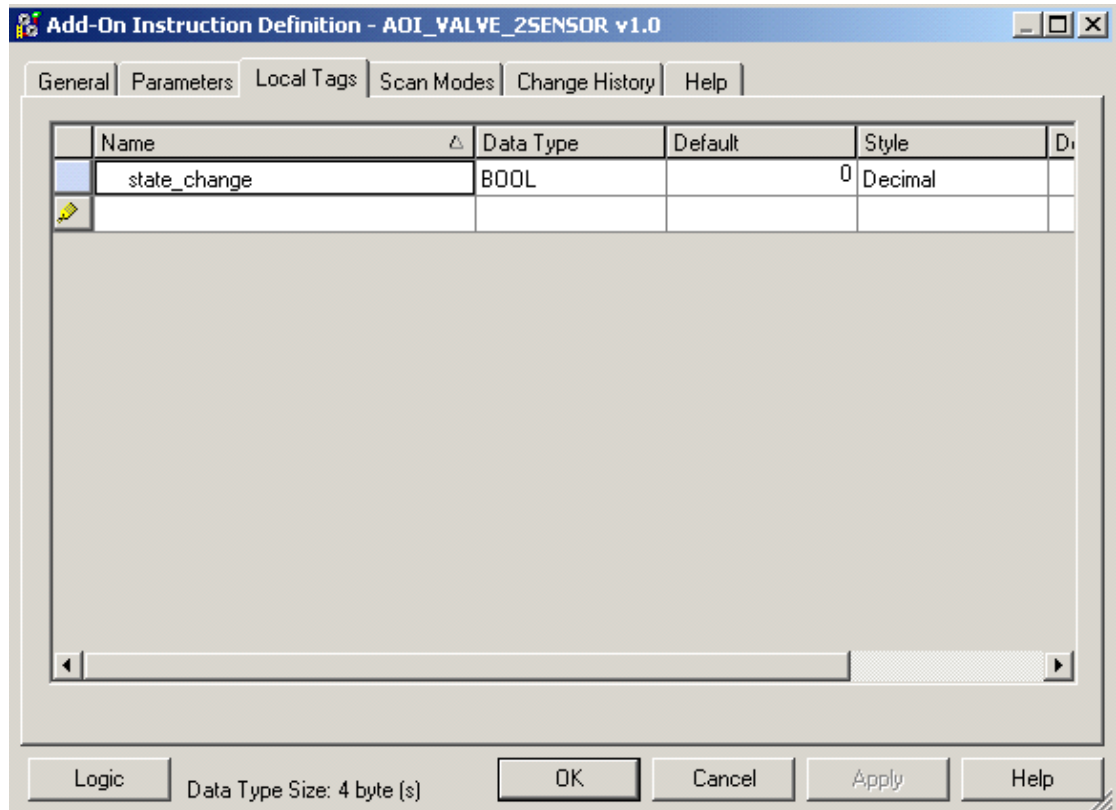
Auf dem Bildschirmfoto ist der Parameter-Konfigurationsbildschirm zu sehen.



Die hinzugefügten Parameter sind die E/A für das Ventil und ein Objekt des Typs „UDT_VALVE“. „V“ muss ein InOut-Parameter sein.

Lokale Daten des Add-On-Befehls

Auf dem nachstehenden Bildschirmfoto ist die Konfigurationsseite für die lokalen Daten des Add-On-Befehls zu sehen.



Logik des Add-On-Befehls

Auf dem nachstehenden Bildschirmfoto ist die Logik für diesen Add-On-Befehl gezeigt.

```

// Control Module Valve_2sensor
// -----
// Implements logic for a valve with an open and a closed sensor and one output

// See UDT Valve for data structure.

// Note the open/close command V.open_command must be set or reset externally
// and then left until the next activation is required. Do not continuously
// hold the flag set or reset.

// increment timer counter
V.timecount := V.timecount + 1;

// evaluate change of state (state machine)
state_change := V.state <> V.state_saved;
V.state_saved := V.state;

// set output
output := (V.fail_open xor V.open_command) and not
           (V.interlocked or V.faulted);

// valve is faulted
V.faulted := V.fault_opening or V.fault_closing or V.fault_sensors;

// action on fault or interlock
if V.faulted or V.interlocked then
  if V.fail_open then
    V.state := 3;
    V.open_command := 1;
  else
    V.state := 0;
    V.open_command := 0;
  end_if;
end_if;

// state machine:
// the state machine does not set outputs - it monitors inputs
// to set status and faults.
case V.state of
  // state 0 - valve is closed - wait for open command
  0: V.closed := 1;
     V.open := 0;
     if (V.open_command) then
       V.state := 1;
     // fault sensors
     else
       V.fault_sensors := (not sens_closed) or (sens_open);
     end_if;
  // state 1 -
  1: V.state := 2;
  // state 2 - waiting for open sensor
  2: if (sens_open & not sens_closed) then
       V.state := 3;

```

```

        // possible close command while waiting to open
        elsif not V.open_command then
            V.state := 0;
        // fault opening
        else
            V.fault_opening := (V.timecount > V.opening_preset);
        end_if;
// state 3 - open - wait for close command
3: V.closed := 0;
   V.open := 1;
   if (not V.open_command) then
       V.state := 4;
   // fault sensors
   else
       V.fault_sensors := (sens_closed) or (not sens_open);
   end_if;
// state 4 -
4: V.state := 5;
// state 5 - wait for closed sensor
5: if (sens_closed & not sens_open) then
    V.state := 0;
    // possible open command while waiting to close
    elsif V.open_command then
        V.state := 3;
    // fault closing
    else
        V.fault_closing := (V.timecount > V.closing_preset);
    end_if;
else;
end_case;
// end state machine

// reset timer if change of state
if (state_change) then V.timecount := 0;
end_if;

// external fault reset
if (V.clear_faults) then
    V.fault_opening := 0;
    V.fault_closing := 0;
    V.fault_sensors := 0;
    V.clear_faults := 0;
end_if;

```

Die in dieser Logik referenzierten Tags sind Parameter oder lokale Tags. Das bedeutet, dass der Add-On-Befehl in jedem Programm verwendet werden könnte (sofern auch der UDT „Valve“ vorhanden ist).

Aufruf

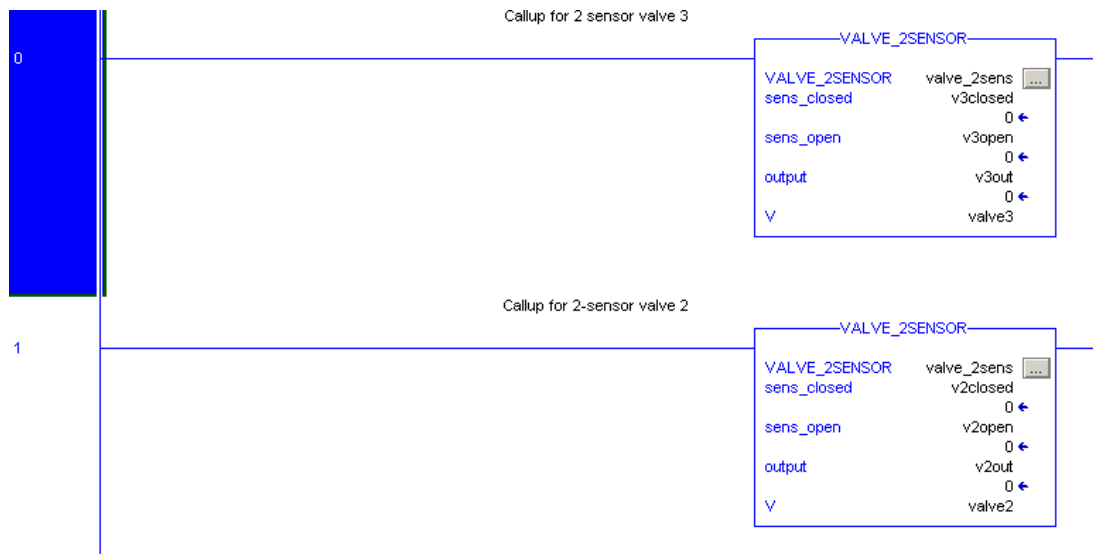
Sowohl der Aufrufcode als auch die Instanzen des UDT „Valve“ befinden sich im Programm „valves_callup“, das unter task_02s ausgeführt wird. Die Frequenz, mit der der Aufrufcode ausgeführt wird, ist von der Anwendung und der Größe des Ventils abhängig.

Auf dem nachstehenden Bildschirmfoto sind die Dateninstanzen abgebildet.

Scope: valves_callup				
Show... Show All				
Name	△	Alias For	Base Tag	Data Type
+valve_2sens				valve_2sensor
+valve1				Valve
+valve2				Valve
+valve3				Valve
+valve4				Valve
+valve5				Valve

Fügen Sie für jedes physische Ventil eine Instanz des Typs „Valve“ hinzu. Das erste Tag ist das erforderliche Backing-Tag für den Add-On-Befehl.

Auf dem nachstehenden Bildschirmfoto ist der Aufrufcode abgebildet.



Rufen Sie den Add-On-Befehl für jedes Ventil einmal auf. Die Parameter sind die tatsächlichen E/A-Tags für die Sensoren und die Magnetspule des Ventils und die Instanz vom UDT „Valve“.

Die E/A-Tags treten nur im Aufruf des Add-On-Befehls auf. Sie sind an keiner anderen Stelle des Programms zu finden. Dies ist nicht nur aus der Sicht der Softwarestruktur ordentlicher, sondern verhindert zudem jedes Risiko von Problemen, die durch eine asynchrone Aktualisierung der E/A auftreten könnten.

Denken Sie daran, dass bei Logix-Steuerungen die E/A asynchron abgetastet werden.

Häufige Fehler beim Konvertieren nach Logix

Einführung

Dieser Abschnitt hat das Ziel, einige Fehler im Entwurf und in der Programmierung hervorzuheben, die S7-Benutzern beim Konvertieren von Anwendungen nach Logix häufig unterlaufen. Diese Fehler wurden bei Untersuchungen von Logix-Programmen identifiziert, die von STEP 7 konvertiert wurden.

Thema	Seite
Auswählen nicht geeigneter Hardware	129
Unterschätzen der Auswirkungen von Task-Planung	130
Durchführen einer Übersetzung anstelle einer Konvertierung	130
Verwenden ungeeigneter Logix-Sprachen	130
Implementierung der falschen Datentypen – DINT gegenüber INT	131
Anwendercode zur Emulation vorhandener Befehle	132
Falsche Verwendung von COP, MOV und CPS	133
Falsche Verwendung von CPT	133
Handhabung von Zeichenketten auf nicht optimale Weise	133
Übermäßige Verwendung von Jump-Befehlen	133
Keine Verwendung von Alias-Tags	133

Programmierfehler fallen in diese beiden Kategorien:

- Programmierung, die zu einer geringeren Effizienz der Steuerung führt
- Programmierung, die zu einem schwierig zu verstehenden, zu verwaltenden und zu entwickelnden Steuerungssystem führt

In den meisten Fällen verbessert eine effizienzorientierte Programmierung auch die Lesbarkeit und Modularität des Programms. Umgekehrt führt eine verbesserte Programmstruktur auch zu einer höheren Programmeffizienz.

Auswählen nicht geeigneter Hardware

Dieses Kapitel beschäftigt sich hauptsächlich mit Software. Denken Sie jedoch daran, dass die richtige Auswahl der Hardware eine Voraussetzung für einen zufrieden stellenden Betrieb ist. Möglicherweise ist die Anzahl von Steuerungen und Racks nicht die gleiche wie bei einem entsprechenden S7-System.

Näheres über Hardware finden Sie in [Kapitel 1](#) und [Anhang A](#). Weitere Informationen sind in den [Anhängen A](#) und [B](#) zu finden.

Unterschätzen der Aus-wirkungen von Task-Planung

Im Bereich der Zeitplanung und der Interrupts weisen die beiden Systeme sehr ähnliche Fähigkeiten auf. Bei Logix wird jedoch Zeitplanung aktiv gefördert.

STEP 7-Programmierer vernachlässigen häufig die Zeitplanung bei der Arbeit mit Logix-Steuerungen. Eine ausführlichere Darstellung der Zeitplanung in Logix finden Sie in [Kapitel 2](#).

Durchführen einer Übersetzung anstelle einer Konvertierung

Ein verbreiteter Fehler besteht darin, ein STEP 7-Programm Zeile für Zeile nach Logix zu übersetzen.

Stattdessen wird ein umfassenderes Verfahren benötigt, das als Konvertierung bezeichnet wird. Dabei werden die Auswahl der Sprache, die Zeitplanung und die Auswahl der Coderoutinen berücksichtigt.

Indem Sie Ihre STEP 7-Programme konvertieren und nicht nur übersetzen, können Sie die Fähigkeiten des Logix-Systems besser nutzen.

Verwenden ungeeigneter Logix-Sprachen

Oft vernachlässigen Programmierer die anderen Logix-Sprachen neben der Kontaktplanlogik.

Eine Auseinandersetzung mit der Auswahl einer Logix-Sprache finden Sie in [Kapitel 2](#) und Beispiele für in Logix übersetzten STEP 7-Code in [Kapitel 4](#).

Implementierung der falschen Datentypen – DINT gegenüber INT

Es wird meist empfohlen, DINT anstelle von INT zu verwenden.

Im nachstehenden Beispiel wird das Hinzufügen zweier DINTs gegenüber dem Hinzufügen zweier INTs gezeigt.

Hinzufügen von DINTs

```
// add two DINTs and assign to a third DINT
for index := 0 to 999 do
    result_DINT := operandA_DINT + operandB_DINT;
end_for;
```

Hinzufügen von INTs

```
// add two INTs and assign to a third INT
for index := 0 to 999 do
    result_INT := operandA_INT + operandB_INT;
end_for;
```

Zeitmessungsergebnisse

In der Tabelle sind relative Zeiten aufgeführt (kleinere Zahlen stehen für kürzere Zeiten). Die Zahlen dienen nur dem Vergleich mit anderen Zahlen in der Tabelle. Sie sollten nicht mit Einträgen in anderen Tabellen verglichen werden.

Methode	Relative Zeiten
Hinzufügen von DINTs mit ST für einen Regelkreis	53
Hinzufügen von INTs mit ST für einen Regelkreis	100

Der gleiche Test wurde zum Vergleich mit einer S7-Steuerung durchgeführt. In diesem Fall waren die Ergebnisse für DINTs und INTs identisch.

Daraus ergibt sich, dass für alle Vorgänge mit Ganzzahlen in Logix DINT verwendet werden sollte. Verwenden Sie INT oder SINT nur, wenn Sie eine Verbindung zu einem externen System herstellen, bei dem die Verwendung von INTs bzw. SINTs vorgeschrieben ist.

Anwendercode zur Emulation vorhandener Befehle

Programmierer schreiben häufig Anwendercode, wenn für die Aufgabe bereits ein geeigneter Befehl vorhanden ist. Vergleichen Sie beispielsweise das Kopieren eines Datenfelds mit Anwendercode und mit dem COP-Befehl.

Anwendercode

```
for index := 0 to 99 do
    target_DINT[index] := source_DINT[index];
end_for;
```

COP-Befehl

```
cop(source_DINT[0], target_DINT[0], 100);
```

Nachstehend finden Sie die relativen Zeitmessungen für die beiden Verfahren. Auch hier dienen die Zahlen nur dem Vergleich mit anderen Zahlen in der Tabelle. Sie sollten nicht mit Einträgen in anderen Tabellen verglichen werden.

Methode	Relative Zeitmessung
Kopieren eines DINT-Datenfelds mit strukturiertem Text	100
Kopieren eines DINT-Datenfelds mit COP-Befehl	18

Für Vorgänge wie das Kopieren von Datenfeldern werden STEP 7-Bibliotheksfunktionen verwendet, die in der Bemerkungsliste geschrieben sind. Wenn die Bibliotheksfunktion nicht den geforderten Zweck erfüllt, kann eine neue geschrieben werden. Die geschriebenen Funktionen sind oft nahezu genauso effizient wie die von STEP 7 bereitgestellten.

In Logix ist es jedoch einem Programmierer nicht möglich, eine Kopierfunktion zu schreiben, die so effizient wie der integrierte COP-Befehl ist. S7-Programmierer sollten daher die Befehlshilfe in der Software RSLogix 5000 sorgfältig lesen, bevor sie eigene Routinen schreiben.

Falsche Verwendung von COP, MOV und CPS

Mit MOV wird ein einfacher Wert (direkt oder ein Tag) in einen einfachen Tag-Typ kopiert – DINT, INT, SINT oder REAL. COP erreicht das gleiche wie MOV (bei der Quelle kann es sich nicht um einen direkten Wert handeln), ist jedoch vor allen Dingen für das Kopieren komplexer Datentypen ausgelegt.

Mit COP einfache Datentypen zu kopieren, gilt als geringfügiger Programmierfehler.

Ein häufig auftretender Fehler ist die Verwendung mehrerer MOV-Befehle zum Kopieren einer Datenstruktur anstelle eines COP-Befehls.

Wenn Ihre Quelldaten während des Kopiervorgangs durch asynchrone E/A-Aktualisierungen geändert werden könnten, verwenden Sie stattdessen CPS. Dieser Befehl kann nicht unterbrochen werden, sodass die Quelldaten während des Kopierens konstant bleiben.

Falsche Verwendung von CPT

In Logix können Ausdrücke mit dem CPT-Befehl ausgewertet werden. Der Ausdruck wird in einem der Felder im Befehl eingegeben. Das ist sehr praktisch.

CPT sollte jedoch nur verwendet werden, wenn zum Auswerten des Ausdrucks mehrere arithmetische Befehle erforderlich wären. Wenn ein einzelner Befehl ausreicht, ist dies schneller als CPT.

Mehr über CPT erfahren Sie in [Kapitel 4](#).

Handhabung von Zeichenketten auf nicht optimale Weise

Wenn Sie einen neuen Zeichenkettentyp definieren möchten, beispielsweise mit einer anderen Anzahl von Zeichen als die standardmäßigen 82, wäre das Erstellen eines neuen „Anwenderdatentyps“ ein Fehler. Erstellen Sie stattdessen einen neuen Zeichenketten-Datentyp. Der Vorteil dieser Herangehensweise besteht darin, dass das Feld „LEN“ automatisch aktualisiert wird, wenn sich die Länge der Zeichenkette ändert.

Übermäßige Verwendung von Jump-Befehlen

In Logix können Sprungbefehle nur in der Kontaktplanlogik auftreten. Es wird empfohlen, den JMP-Befehl sparsam einzusetzen. Durch Sprünge in der Kontaktplanlogik wird das Programm häufig schwer lesbar.

Keine Verwendung von Alias-Tags

Denken Sie daran, für die E/A-Tags, die die Software RSLogix 5000 automatisch erstellt, Alias-Tags anzulegen. Das Programm ist dann einfacher zu lesen. Weitere Informationen finden Sie in [Kapitel 2](#).

Notizen:

Glossar: S7 nach Logix

Einführung

Dieses Kapitel enthält ein Glossar von S7-Begriffen und deren Logix-Äquivalenten.

Hardware-Terminologie

S7-Begriff	Definition	Logix-Begriff	Definition
Kommunikationsprozessor	Kommunikationsmodul	Bridge	
Controller	Die Steuerung	Steuerung	
CPU	Zentraleinheit	CPU oder Steuerung	
Failsafe-CPU	CPU 315F-2 DP implementiert PROFISAFE-Version von DP	GuardLogix	L61S, L62S, L63S
Industrial Ethernet	Siemens-Version von Ethernet	Ethernet/IP ControlNet	Beide weisen die gleiche (oder bessere) Funktionalität wie (als) Industrial Ethernet auf.
MPI	Multi-Point Interface – ein serieller Bus	Seriell	Protokolle DF1 und DH485
Speicherprogrammierbare Steuerung		Steuerung oder programmierbare Automatisierungssteuerung	
PROFIBUS DP	Häufig verwendeter Feldbus	Ethernet/IP ControlNet DeviceNet	
PROFIBUS PA	Profibus-Variante, die auf Prozessautomatisierung ausgelegt ist	Wie Profibus DP	
PROFINET	Profibus über Ethernet	Ethernet/IP	
PROFISAFE	Fehlersichere Ausführung von PROFIBUS DP	GuardLogix	
S7-200	Steuerungen im unteren Bereich	MicroLogix	
S7-300	Steuerungen im mittleren Bereich	CompactLogix	
S7-400	Steuerungen im oberen Bereich	ControlLogix	
SIMATIC	Markenname für Automatisierungsprodukte von Siemens	Logix	

Software-Terminologie

S7-Begriff	Definition	Am ehesten entsprechender Logix-Begriff	Definition
Speicher	In STL verwendet	-/-	In Logix-Sprachen muss nicht auf Strukturen der untergeordneten Prozessorebene zugegriffen werden.
AR1, AR2	Zeigerregister	-/-	In Logix-Sprachen muss nicht auf Strukturen der untergeordneten Prozessorebene zugegriffen werden.
Datenfeld	Syntax „ARRAY[0..7] OF REAL“	Datenfeld	Syntax „REAL[8]“ Die Indizierung beginnt immer bei 0.
Bitspeicher	Adressen M...	-/-	Verwenden Sie Tags.
Blocktransfer	Kopieren eines Datenbausteins. SFC20 BLK_MOV	COP	Befehl (Verwenden Sie für einfache Variablen MOV.)
BOOL		BOOL	
Byte	8-Bit-Wort	SINT	Von der Verwendung wird abgeraten (sie ist langsamer als bei DINT), sofern sie nicht erforderlich ist (z. B. bei Zeichen einer Zeichenkette).
CFC	Optionale Prozesssteuerungssprache	FBD	Standard-Funktionsblocksprache
CHAR	Byte als Zeichen	SINT	
Cycle_Execution	OB1 – kontinuierlich ausgeführt	Kontinuierliche Task	Kontinuierlich ausgeführt
Datenbaustein	Statische Datenspeichereinheit	Tag-Datenbank mit Gültigkeitsbereich der Steuerung oder Tag-Datenbank mit Gültigkeitsbereich des Programms	Global sichtbar im Programm, dem die Datenbank zugeordnet ist
DINT	Doppelganzzahl	DINT	Doppelganzzahl
DWORD	32-Bit-Wort	DINT	
FBD	Funktionsblockdiagramm	FBD	Funktionsblockdiagramm
Funktion	Programmeinheit mit temporärem Speicher aber ohne statischen Speicher	Routine AOI (Add-On-Befehl)	Beide können auf eine Funktion bezogen sein.
Funktionsbaustein	Programmeinheit mit temporärem Speicher und statischem Speicher	Routine AOI (Add-On-Befehl) Programm	Alle können auf einen Funktionsbaustein bezogen sein.
GRAPH	Optionale grafische Sprache	sequenzielle Ablaufsprache	Grafische Standardsprache
HW Config	Hardwarekonfiguration – Komponente von STEP 7	E/A-Konfiguration	Zweig der Steuerungsübersicht
INT	Ganzzahl	INT	Von der Verwendung wird abgeraten (langsamer als DINT).
Interrupt_Execution	Periodisch ausgeführter OB	Periodische Task	Periodisch ausgeführte Task

S7-Begriff	Definition	Am ehesten entsprechender Logix-Begriff	Definition
LAD	Kontaktplan	LD	Kontaktplan
Bibliothek	Systemfunktionen	GSV, SSV	Befehle – Get System Value – Systemwert abrufen Set System Value – Systemwert festlegen
NetPro	Netzwerkconfiguration	-/-	Teil des E/A-Konfigurations-Zweigs der Steuerungsübersicht
Baustein	Vom Betriebssystem aufgerufene Programmeinheit	Task	Vom Betriebssystem aufgerufene Programmeinheit
Zeiger	In STL verwendeter Datenzeiger	-/-	Verwenden Sie Datenfelder
REAL	32-Bit-Fließkommazahl	REAL	32-Bit-Fließkommazahl
SCL	Optionale Hochsprache	Strukturierter Text	Standardsprache
Simatic Manager	Komponente von STEP 7	Steuerungsübersicht	Komponente von RSLogix 5000
STEP 7	Entwicklungs- und Überwachungssoftware für S7	RSLogix 5000	Entwicklungs- und Überwachungssoftware für Logix
STL	Bemerkungsliste	-/-	Verwenden Sie strukturierten Text oder Kontaktplanlogik oder das sequenzielle Funktionsdiagramm
STRING	Abfolge von CHARs. Standardlänge 254.	STRING	Abfolge von SINTs. Standardlänge 82. Zeichenkettenobjekt enthält auch seine Länge als Eigenschaft „LEN“.
STRUCT	Sammlung von Daten ohne Typ	-/-	In Logix ist eine Struktur eine Instanz eines Typs (UDT).
Symbol	Name für Datenspeicheradresse	Tag	In einem Tag wird die Struktur der Variablen definiert <i>und</i> Speicher reserviert.
Temporärer Speicher	Auf dem Laufzeit-Stack erstellter Speicher	-/-	Verwenden Sie Tags.
WORD	16-Bit-Wort	INT	
UDT	User Data Type – Anwenderdatentyp	UDT	User Data Type – Anwenderdatentyp

Notizen:

S7 300- und S7 400-Komponenten und die RA-Äquivalente

Einführung

In diesem Anhang werden Siemens-Produkte und ihre Rockwell Automation-Äquivalente aufgeführt.

Thema	Seite
Kompakt-CPU S7 300	140
Standard-CPU S7 300	140
Technologie-CPU S7 300	141
Fail-Safe-CPU S7 300	142
S7 300-Digitaleingabebaugruppen	142
S7 300-Digitalausgabebaugruppen	143
S7 300-Relaisausgabebaugruppen	143
S7 300-Digitalein-/ausgabebaugruppen	144
S7 300-Analogeingabebaugruppen	144
S7 300-Analogausgabebaugruppen	145
S7 300-Analogein-/ausgabebaugruppen	145
S7 300-Analogausgabebaugruppen	146
Redundante und Fail-Safe-Steuerungen	146
Digitaleingabebaugruppen	147
Digitalausgabebaugruppen	147
Analogeingabebaugruppen	147
Analogausgabebaugruppen	148

Kompakt-CPU S7 300

Siemens-Bestellnummer	Siemens-Kurzreferenz	Speicher	Komm.-Anschlüsse			Max. MMC-Größe	Integrierte E/A				RA-Lösung
			MPI	DP	seriell		DI	DO	AI	AO	
6ES7 312-5BE0x-xxxx	S7-312C	32 000	J	N	N	4 MB	10	6			1769-L31 + Compact I/O ML1500
6ES7 313-5BF0x-xxxx	S7-313C	64 000 Ja Nein Nein	J	N	N	8 MB	24	16	4	2	1769-L31 + Compact I/O ML1500
6ES7 313-6BF0x-xxxx	S7-313C- PtP	64 000	J	N	RS-422/485	8 MB	16	16			1769-L31 + Compact I/O ML1500
6ES7 313-6CF0x-xxxx	S7-313C- DP	64 000	J	J	N	8 MB	16	16			1769-L31 + Compact I/O ML1500
6ES7 314-6BG0x-xxxx	S7-314C- PtP	96 000	J	N	RS-422/485	8 MB	24	16	4	2	1769-L31 + Compact I/O ML1500
6ES7 314-6CG0x-xxxx	S7-314C- DP	96 000 Ja Ja Nein 8 MB	J	J	N	8 MB	24	16	4	2	1769-L31 + Compact I/O ML1500

Standard-CPU S7 300

Siemens-Bestellnummer	Siemens-Kurzreferenz	Speicher	Komm.-Anschlüsse			Max. Last-Speichergröße (RAM)	RA-Lösung
			MPI	DP	PN		
6ES7 312-1AE1x-xxxx	S7-312	32 000	J	N	N	4 MB	1769-L31
6ES7 314-1AG1x-xxxx	S7-314	96 000	J	N	N	8 MB	1769-L31
6ES7 315-2AG1x-xxxx	S7-315-2 DP	128 000	J	J	N	8 MB	1769-L3xE oder 1769-L3xC
6ES7 315-2EH1x-xxxx	S7-315-2 PN/DP	256 000	J	J	J	8 MB	1769-L3xE oder 1769-L3xC

Siemens-Bestellnummer	Siemens-Kurzreferenz	Speicher	Komm.-Anschlüsse			Max. Last-Speichergröße (RAM)	RA-Lösung
			MPI	DP	PN		
6ES7 317-2AJ1x-xxxx	S7-317-2 DP	512 000	J	J	N	8 MB	1769-L3xE oder 1769-L3xC
6ES7 317-2EK1x-xxxx	S7-317-2 PN/DP	1 MB	J	J	J	8 MB	1769-L3xE oder 1769-L3xC
6ES7 319-3EL0x-xxxx	S7-319-3 PN/DP	1,4 MB	J	J	J	8 MB	1769-L3xE oder 1769-L3xC

Technologie-CPU S7 300

Siemens-Bestellnummer	Siemens-Kurzreferenz	Speicher	Komm.-Anschlüsse			Max. Last-Speichergröße (RAM)	RA-Lösung
			MPI	DP	PN		
6ES7 315-6TG1x-xxxx	S7-315T-2 DP	128 000	J	J	J	4 oder 8 MB	1768-L43
6ES7 317-6TJ1x-xxxx	S7-317T-2 DP	512 000	J	J	J	4 oder 8 MB	1768-L43

Fail-Safe-CPU S7 300

Siemens-Bestellnummer	Siemens-Kurzreferenz	Speicher		Komm.-Anschlüsse		Max. Last-Speichergroße (RAM)	RA-Lösung (ControlLogix)
			MPI	DP	PN		
6ES7 315-6FF1x-xxxx	S7-315F-2 DP	192 000	J	J	N	8 MB	GuardLogix oder SmartGuard 600
6ES7 315-2FH1x-xxxx	S7-315F-2 PN/DP	256 000	J	J	J	8 MB	GuardLogix oder SmartGuard 600
6ES7 317-6FF0x-xxxx	S7-317F-2 DP	1 MB	J	J	N	8 MB	GuardLogix oder SmartGuard 600
6ES7 317-2FK1x-xxxx	S7-317F-2 PN/DP	1 MB	J	J	J	8 MB	GuardLogix oder SmartGuard 600

S7 300-Digitaleingabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Bereich	RA-Lösung	Kommentare
6ES7 321-1BH0x-xxxx	20 Stifte	16	24 V DC	1769-IQ16 1769-IQ16F	
6ES7 321-1BH5x-xxxx	20 Stifte	16	24 V DC	1769-IQ16 1769-IQ16F	
6ES7 321-1BL0x-xxxx	40 Stifte	32	24 V DC	1769-IQ32 1769-IQ32T	
6ES7 321-1CH0x-xxxx	40 Stifte	16	24–48 V	-/-	
6ES7 321-1CH2x-xxxx	20 Stifte	16	48–125 V DC	-/-	
6ES7 321-1BH1x-xxxx	20 Stifte	16	24 V DC	1769-IQ16 1769-IQ16F	
6ES7 321-7BH0x-xxxx	20 Stifte	16	24 V DC	1769-IQ16 1769-IQ16F	
6ES7 321-1FH0x-xxxx	20 Stifte	16	120–230 V AC	1769-IA16	1769-IA16 ist nur für 120 V AC geeignet
6ES7 321-1FF0x-xxxx	20 Stifte	8	120–230 V AC	1769-IM12	1769-IM12 ist nur für 230 V AC geeignet
6ES7 321-1FF1x-xxxx	40 Stifte	8	120–230 V AC	1769-IA8I	1769-IA8I ist nur für 120 V AC geeignet
6ES7 321-1EL0x-xxxx	40 Stifte	32	120 V AC	-/-	
-/-		16	5 V DC TTL	1769-IG16	

S7 300-Digitalausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Bereich	Ausgangsstrom	RA-Lösung	Kommentare
6ES7 332-1FH0x-xxxx	20 Stifte	16	120/230 V AC	0,5 A	1769-0A16	
6ES7 332-1FF0x-xxxx	20 Stifte	8	120/230 V AC	2 A	1769-0A8	S7-300 weist eine Sicherung pro Gruppe auf
6ES7 332-5FF0x-xxxx	40 Stifte	8	120/230 V AC	2 A	1769-0A8	S7-300 wird in Gruppen à 1 geliefert
6ES7 322-1BH0x-xxxx	20 Stifte	16	24 V DC	0,5 A	1769-0B16 1769-0B16P	
6ES7 322-1BH1x-xxxx	20 Stifte	16	24 V DC	0,5 A	-/-	Hochgeschwindigkeit
6ES7 322-1BL0x-xxxx	40 Stifte	32	24 V DC	0,5 A	1769-0B32 1769-0B32T	
6ES7 322-1BF0x-xxxx	20 Stifte	8	24 V DC	2 A	1769-0B8	
6ES7 322-8BF0x-xxxx	20 Stifte	8	24 V DC	0,5 A	1769-0B8	
6ES7 332-1FL0x-xxxx	2 x 20 Stifte	32	120 V AC	1 A	-/-	
6ES7 332-5GH0x-xxxx	40 Stifte	16	24/48 V	0,5 A	-/-	
6ES7 332-1CF0x-xxxx	20 Stifte	8	48–125 V DC		-/-	
-/-		16	5 V DC TTL		1769-0G16	
-/-		16	24 V DC		1769-0V16	
-/-		32	24 V DC		1769-0V32T	
-/-		16	24 V DC		1769-0B16P	

S7 300-Relaisausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Ausgangsstrom	RA-Lösung	Kommentare
6ES7 322-1HH0x-xxxx	20 Stifte	16	2 A	1769-0W16	
6ES7 322-1HF0x-xxxx	20 Stifte	8	5 A	1769-0W8	
6ES7 322-1HF1x-xxxx	40 Stifte	8	5 A	1769-0W8I	
6ES7 322-5HF0x-xxxx	40 Stifte	8	8 A	1769-0W8I	S7-300-Module werden mit RC-Filter und Überspannungsschutz geliefert

S7 300-Digitalein/-ausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Bereich Eingänge	Ausgangsstrom	RA-Lösung	Kommentare
6ES7 323-1BH0x-xxxx	20 Stifte	8/8	24 V DC	24 V DC/0,5 A	1769-IQ6XOW4	Compact I/O bietet weniger E/A, und die Ausgänge sind Relais
6ES7 323-1BL0x-xxxx	40 Stifte	16/16	24 V DC	24 V DC/0,5 A	-/-	
6ES7 327-1BH0x-xxxx	20 Stifte	8/8	24 V DC	24 V DC/0,5 A	-/-	8 Eingänge; 8 Eingänge oder Ausgänge (konfigurierbar)

S7 300-Analogeingabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Auflösung (Bit)	Typ	Compact I/O-Lösung	Kommentare
6ES7 331-1KF0x-xxxx	40	8	13	Spannung, Strom, Widerstand Temperatur	1769sc-IF8U 1769-IF8U	
6ES7 331-7KF0x-xxxx	20	8	9/12/14	Spannung, Strom, Widerstand Temperatur	1769sc-IF8U 1769-IF8U	
6ES7 331-7KB0x-xxxx	20	2	9/12/14	Spannung, Strom, Widerstand Temperatur	1769sc-IF8U 1769-IF4	
6ES7 331-7NF0x-xxxx	40	8	16	Spannung Spannung	1769-IF8	
6ES7 331-7NF1x-xxxx	40	8	16	Spannung Spannung	1769-IF8	Beinhaltet Hardware-Interrupt am Ende des Zyklus, im Gegensatz zu 6ES7 331-7NF0x-xxxx
6ES7 331-7HF0x-xxxx	20	8	14	Spannung Spannung	1769-IF8	
6ES7 331-7PF0x-xxxx	40	8		Widerstandstemperaturfühler (RTD) Widerstand	1769-IR6	
6ES7 331-7PF1x-xxxx	40	8		Thermoelement	1769-IT6	
-/-					1769-IF4I	

S7 300-Analogausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Auflösung (Bit)	Typ	RA-Lösung	Kommentare
6ES7 332-5HD0x-xxxx	40	4	12	Spannung Strom	1769-OF4VI 1769-OF4CI	
6ES7 332-7ND0x-xxxx	20	4	16	Spannung Strom	1769-OF4VI 1769-OF4CI	
6ES7 332-5HB0x-xxxx	20	2	12	Spannung Strom	1769-OF2	
6ES7 332-5HF0x-xxxx	20	8	12	Spannung Strom	1769-OF8V 1769-OF8C	

S7 300-Analogein-/ausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Auflösung (Bit)	Typ	RA-Lösung	Kommentare
6ES7 334-0KE0x-xxxx	20	4/2	12	Spannung Strom Pt 100		Ausgänge nur Spannung
6ES7 334-0CE0x-xxxx	20	4/2	8	Spannung und Strom (Eingänge und Ausgänge)	1769-IF4X0F2	

S7 400-Standardsteuerungen

Siemens-Bestellnummer	Siemens-Kurzreferenz	Arbeits-Speichergröße		Komm.-Anschlüsse		Max. Last-Speichergröße (RAM)	RA-Lösung (ControlLogix)
			MPI	DP	PN		
6ES7 412-1XF04-0AB0	CPU 412-1	144 KB	J	J	N	64 MB	1756-L61
6ES7 412-2GX04-0AB0	CPU 412-2	256 KB	J	J	N	64 MB	1756-L61
6ES7 414-2GX04-0AB0	CPU 414-2	512 KB	J	J	N	64 MB	1756-L62
6ES7 414-3XJ04-0AB0	CPU 414-3	1,4 MB	J	J	N	64 MB	1756-L63
6ES7 414-3EM05-0AB0	CPU 414-3 PN/DP	2,8 MB	J	J	J	64 MB	1756-L63
6ES7 416-3XK04-0AB0	CPU 416-2	2,8 MB	J	J	N	64 MB	1756-L63
6ES7 416-3XL04-0AB0	CPU 416-3	5,6 MB	J	J	N	64 MB	1756-L64
6ES7 416-3ER05-0AB0	CPU 416-3 PN/DP	11,2 MB	J	J	J	64 MB	1756-L64
6ES7 417-4XL04-0AB0	CPU 417-4	20 MB	J	J	N	64 MB	1756-L64

Redundante und Fail-Safe-Steuerungen

Siemens-Bestellnummer	Siemens-Kurzreferenz	Arbeits-Speichergröße		Komm.-Anschlüsse			Max. Last-Speichergröße (RAM)	RA-Lösung (ControlLogix)
			MPI	DP	PN	Synchr.-Anschlüsse		
6ES7 414-4HJ04-0AB0	CPU 414-4H	1,4 MB	J	J	N	J	64 MB	1756-L63
6ES7 417-4HL04-0AB0	CPU 417-4H	20 MB	J	J	N	J	64 MB	1756-L64
6ES7 416-2FK04-0AB0	CPU-416F-2	2,6 MB	J	J	N	N	64 MB	1756-L61S

Digitaleingabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Bereich	RA-Lösung	Kommentare
6ES7 421-7BH01-0AB0 (Interrupt/Diagnose)	48 Stifte	16	24 V DC	1756-IB16D	
6ES7 421-1BL01-0AA0	48 Stifte	32	24 V DC	1756-IB32	
6ES7 421-1EL00-0AA0	48 Stifte	32	120 V AC/DC	1756-IA32	
6ES7 421-1FH20-0AA0	48 Stifte	16	230 V AC/DC	1756-IM161	
6ES7 421-7DH00 0AB0 (Interrupt/Diagnose)	48 Stifte	32	24–60 V AC/DC		

Digitalausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Punkte	Bereich	Strom	RA-Lösung	Kommentare
6ES7 422-1FH00-0AA0	48 Stifte	16	230 V AC	2 A	1756-0A16	
6ES7 422-1HH00-0AA0	48 Stifte	16	60 V DC 230 V AC (Relais)	5 A	1756-0W16I	
6ES7 422 1BH11-0AA0	48 Stifte	16	24 V DC	2 A	1756-0B16E	
6ES7 422-1BL00-0AA0	48 Stifte	32	24 V DC	0,5 A	1756-0B32	
6ES7 422-7BL00-0AB0 (Diagnose)	48 Stifte	32	24 V DC	0,5 A	1756-0B16D 1756-0B32	

Analogeingabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Kanäle	Auflösung (Bit)	Typ	RA-Lösung	Kommentare
6ES7 431-0HH0-0AB0	48 Stifte	16	13	Spannung Strom	1756-IF16	16 Bit
6ES7 431-1KF00-0AB0	48 Stifte	8	13	Spannung Strom Impedanz	1756-IF8	16 Bit 4 Differenzialeingänge
6ES7 431-1KF10-0AB0	48 Stifte	8	14-16	Spannung Strom Thermoelement Thermowiderstand Impedanz	1756-IR6I 1756-IT6I	6 RTD 6 Thermoelemente Je 16 Bit
6ES7 431-1FK20-0AB0	48 Stifte	8	14	Spannung Strom Impedanz	1756-IF16	16 Bit

6ES7 431-7QH00-0AB0 (Interrupt)	48 Stifte	16	16	Spannung Strom Thermoelement Thermowiderstand Impedanz	1756-IR6I 1756-IT6I	6 RTD 6 Thermoelemente
6ES7 431-7KF00-0AB0	48 Stifte	8	16	Spannung Strom Thermoelement	1756-IT6I	6 Kanäle
6ES7 431-7KF01-0AB0	48 Stifte	8	16	Thermowiderstand	1756-IR6I	5 Kanäle

Analogausgabebaugruppen

Siemens-Bestellnummer	Vorderer Anschluss	Kanäle	Auflösung (Bit)	Typ	RA-Lösung	Kommentare
6ES7 432-1HF00-0AB0	48 Stifte	8	13	Spannung Strom	1756-OF8	15 Bit

Referenztable für die Siemens-HMI

Mithilfe dieses Anhangs können Sie Panels von Rockwell Automation mit bestimmten Panel-Arten von Siemens vergleichen.

Thema	Seite
SIMATIC Micro-Panels und Rockwell Automation-Äquivalente	149
SIMATIC-Panels, 7x-Serie, und Rockwell Automation-Äquivalente	151
SIMATIC-Panels, 17x-Serie, und Rockwell Automation-Äquivalente	152
SIMATIC-Panels, 27x-Serie, und Rockwell Automation-Äquivalente	154
SIMATIC-Multi-Panels, 27x-Serie, und Rockwell Automation-Äquivalente	156
SIMATIC-Multi-Panels, 37x-Serie, und Rockwell Automation-Äquivalente	158

SIMATIC Micro-Panels und Rockwell Automation-Äquivalente

SIMATIC Micro-Panels					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6640-OBA11-0AX0	SIMATIC OP 73MICRO	3-Zoll-STN-Monochrom-Anzeige, 160 x 48 Bildpunkte, Tastatur, nur 24 V DC	128 KB	1 x RS485, S7-200-kompatibel, kein Druckeranschluss	2711P-K4M5D	PanelView Plus 400, Graustufen, Tastatur	3,8-Zoll- STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-OAA15-2AX0	SIMATIC TP070 Im April 2007 aus der Produktion genommen	5,7-Zoll-STN-Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	128 KB	1 x RS485, S7-200-kompatibel, kein Druckeranschluss	2711P-T6M5D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll- STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Touchscreen, 24 V DC, USB-Druckfähigkeiten
6AV6640-OCA01-0AX0	SIMATIC TP 170MICRO Im April 2007 aus der Produktion genommen	5,7-Zoll-STN-Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC, begrenzte Anwendungsfunktionalität	256 KB	1 x RS485, S7-200-kompatibel, kein Druckeranschluss	2711P-T6M5D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll- STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Touchscreen, 24 V DC, USB-Druckfähigkeiten

SIMATIC Micro-Panels					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6640-OCA11-0AX0	SIMATIC TP 177MICRO	5,7-Zoll-STN-Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	256 KB	1 x RS485, S7-200-kompatibel, kein Druckeranschluss	2711P-T6MSD	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Touchscreen, 24 V DC, USB-Druckfähigkeiten
6AV6610-0AA01-1CA8	WINCC FLEXIBLE MICRO-Software	Konfigurations- und Programmiersoftware nur für Simatic Panels	-/-	-/-	9701-VWSTMENE	Software RSView Studio for Machine Edition	Konfigurationssoftware RSView Studio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

SIMATIC-Panels, 7x-Serie, und Rockwell Automation-Äquivalente

SIMATIC-Panels – 7x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6641-0AA11-0AX0	SIMATIC OP73	3-Zoll-STN-Monochrom-Anzeige, 160 x 48 Bildpunkte, Tastatur, nur 24 V DC	256 KB	1 x RS485, kompatibel mit S7-200, S7-300/400, kein Druckeranschluss	2711P-K4M5D	PanelView Plus 400, Graustufen, Tastatur	3,8-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6641-0BA11-0AX0	SIMATIC OP77A	4,5-Zoll-STN-Monochrom-Anzeige, 160 x 64 Bildpunkte, Tastatur, nur 24 V DC	256 KB	1 x RS422, 1 x RS485, S7-200, S7-300/400, kein Druckeranschluss	2711P-K4M5D	PanelView Plus 400, Graustufen, Tastatur	3,8-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6641-0CA01-0AX0	SIMATIC OP77B	4,5-Zoll-STN-Monochrom-Anzeige, 160 x 64 Bildpunkte, Tastatur, nur 24 V DC	1 MB	1 x RS232, 1 x RS422, 1 x RS485, USB, S7-200, S7-300/400, Druckeranschluss erhältlich	2711P-K4M5D	PanelView Plus 400, Graustufen, Tastatur	3,8-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6621-0AA01-0AA0	WINCC FLEXIBLE COMPACT-Software	Konfigurations- und Programmiersoftware für Simatic OP77-, OP/TP170- und Micro-Panels	-/-	-/-	9701-VWSTMENE	Software RSView Studio for Machine Edition	Konfigurationssoftware RSView Studio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

SIMATIC-Panels, 17x-Serie, und Rockwell Automation-Äquivalente

SIMATIC-Panels– 17x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6545- OBA15-2AX0	SIMATIC TP170A Blaumodus Im April 2007 aus der Produktion genommen	5,7-Zoll- STN- Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	320 KB	1 x RS232, 1 x RS422, 1 x RS485, S5, S7-200, S7-300/400 und Steuerungen von Drittanbie- tern, kein Drucker- anschluss	2711P- T6M20D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll- STN-32- Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232- Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545- OBB15-2AX0	SIMATIC TP170B Blaumodus Im April 2007 aus der Produktion genommen	5,7-Zoll- STN- Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	768 KB	2 x RS232, 1 x RS422, 1 x RS485, S5, S7-200, S7-300/400 und Steuerungen von Drittanbie- tern, Drucker- anschluss erhältlich	2711P- T6M20D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll- STN-32- Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232- Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545- OBC15-2AX0	SIMATIC TP170B Farbe Im April 2007 aus der Produktion genommen	5,7-Zoll- STN- Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	768 KB	2 x RS232, 1 x RS422, 1 x RS485, S5, S7-200, S7-300/400 und Steuerungen von Drittanbie- tern, Drucker- anschluss erhältlich	2711P- T6C20D	PanelView Plus 600, Farbe, Touchscreen	5,5-Zoll- TFT- Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232- Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6542- OBB15-2AX0	SIMATIC OP170B Blaumodus Im April 2007 aus der Produktion genommen	5,7-Zoll- STN- Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Tastatur und Touchscreen, nur 24 V DC	768 KB	2 x RS232, 1 x RS422, 1 x RS485, S5, S7-200, S7-300/400 und Steuerungen von Drittanbie- tern, Drucker- anschluss erhältlich	2711P- B6M20D	PanelView Plus 600, Graustufen, Touchscreen und Tastatur	5,5-Zoll- STN-32- Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232- Kommunikation, Touchscreen und Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6642- ODC01-1AX0	SIMATIC OP177B Blaumodus	5,7-Zoll- STN- Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Tastatur und Touchscreen, nur 24 V DC	2 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbie- tern, Drucker- anschluss erhältlich	2711P- B6M20D	PanelView Plus 600, Graustufen, Touchscreen und Tastatur	5,5-Zoll- STN-32- Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232- Kommunikation, Touchscreen und Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten

SIMATIC-Panels – 17x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6642-0AA11-0AX0	SIMATIC TP177A Blaumodus	5,7-Zoll- STN-Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	512 KB	Kompatibel mit 1 x RS422, 1 x RS485, S7-200, S7-300/400, kein Druckeranschluss	2711P-T6M20D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6642-0BA01-1AX0	SIMATIC TP177B Farbe	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	2 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T6C20D	PanelView Plus 600, Farbe, Touchscreen	5,5-Zoll-TFT-Farbanzeige, 320 x 240 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6642-0BC01-1AX0	SIMATIC TP177B Blaumodus	5,7-Zoll- STN-Anzeige, Blaumodus (4 Stufen), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	2 MB	1 x RS422, 1 x RS485, USB, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T6M20D	PanelView Plus 600, Graustufen, Touchscreen	5,5-Zoll-STN-32-Graustufen-Anzeige, 320 x 240 Bildpunkte, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6642-8BA10-0AA0	SIMATIC TP177B Farbe, Edelstahl	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC, Frontblende aus Edelstahl	2 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T6C20D	PanelView Plus 600, Farbe, Touchscreen	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6642-0DA01-1AX0	SIMATIC OP177B Farbe	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Tastatur und Touchscreen, nur 24 V DC	2 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-B6C20D	PanelView Plus 600, Farbe, Touchscreen und Tastatur	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Touchscreen und Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6621-0AA01-0AA0	WINCC FLEXIBLE COMPACT-Software	Konfigurations- und Programmiersoftware für Simatic OP77-, OP/TP170- und Micro-Panels	-/-	-/-	9701-VWSTMENE	Software RSVIEW Studio for Machine Edition	Konfigurationssoftware RSVIEW Studio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

SIMATIC-Panels, 27x-Serie, und Rockwell Automation-Äquivalente

SIMATIC-Panels– 27x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6545-OCA10-OAX0	SIMATIC TP270 6 Zoll, Farbe Im Oktober 2006 aus der Produktion genommen	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	2 MB	2 x RS232, 1 x RS422, 1 x RS485, USB, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T6C20D	PanelView Plus 600, Farbe, Touchscreen	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-OCC10-OAX0	SIMATIC TP270 10 Zoll, Farbe Im Oktober 2006 aus der Produktion genommen	10,4-Zoll- STN-Anzeige, Farbe (256 Farben), 640 x 480 Bildpunkte, Touchscreen, nur 24 V DC	2 MB	2 x RS232, 1 x RS422, 1 x RS485, USB, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T10C4D1	PanelView Plus 1000, Farbe, Touchscreen	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6542-OCA10-OAX0	SIMATIC OP270 6 Zoll, Farbe Im Oktober 2006 aus der Produktion genommen	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Tastatur, nur 24 V DC	2 MB	2 x RS232, 1 x RS422, 1 x RS485, USB, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K6C20D	PanelView Plus 600, Farbe	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6542-OCC10-OAX0	SIMATIC OP270 10 Zoll, Farbe Im Oktober 2006 aus der Produktion genommen	10,4-Zoll- STN-Anzeige, Farbe (256 Farben), 640 x 480 Bildpunkte, Tastatur, nur 24 V DC	2 MB	2 x RS232, 1 x RS422, 1 x RS485, USB, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K10C4D1	PanelView Plus 1000, Farbe, Tastatur	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten

SIMATIC-Panels – 27x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6643-0AA01-1AX0	SIMATIC TP277 6 Zoll, Farbe	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	4 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T6C20D	PanelView Plus 600, Farbe, Touchscreen	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6643-0BA01-1AX0	SIMATIC OP277 6 Zoll, Farbe	5,7-Zoll- STN-Anzeige, Farbe (256 Farben), 320 x 240 Bildpunkte, Tastatur, nur 24 V DC	4 MB	1 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K6C20D	PanelView Plus 600, Farbe	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6622-0BA01-0AA0	WINCC FLEXIBLE STANDARD-Software	Konfigurations- und Programmiersoftware für Simatic OP/TP/MP270, MP370, OP77-, OP/TP170- und Micro-Panels	-/-	-/-	9701-VWSTMENE	Software RStudio Studio for Machine Edition	Konfigurationssoftware RStudio Studio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

SIMATIC-Multi-Panels, 27x-Serie, und Rockwell Automation-Äquivalente

SIMATIC-Multi-Panels– 27x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6542-OAG10-OAX0	SIMATIC MP270B Tastatur, 10 Zoll Im Oktober 2006 aus der Produktion genommen	10,4-Zoll- TFT-Anzeige, Farbe (65.536 Farben), 640 x 480 Bildpunkte, Tastatur, nur 24 V DC	5 MB	2 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K10C4D1	PanelView Plus 1000, Farbe, Tastatur	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-OAG10-OAX0	SIMATIC MP270B Touchscreen, 10 Zoll Im Oktober 2006 aus der Produktion genommen	10,4-Zoll- TFT-Anzeige, Farbe (65.536 Farben), 640 x 480 Bildpunkte, Touchscreen, nur 24 V DC	5 MB	2 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T10C4D1	PanelView Plus 1000, Farbe, Touchscreen	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-OAH10-OAX0	SIMATIC MP270B Touchscreen, 6 Zoll Im Oktober 2006 aus der Produktion genommen	5,7-Zoll- TFT-Anzeige, Farbe (65.536 Farben), 320 x 240 Bildpunkte, Touchscreen, nur 24 V DC	5 MB	2 x RS422, 1 x RS485, USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K6C20D	PanelView Plus 600, Farbe	5,5-Zoll- TFT-Farbanzeige, 320 x 240 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP, RS-232-Kommunikation, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6643-OCB01-1AX0	SIMATIC MP277 Touchscreen, 8 Zoll	7,5-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 640 x 480 Bildpunkte, Touchscreen, nur 24 V DC	6 MB	1 x RS422, 1 x RS485, 2 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T7C4D1	PanelView Plus 700, Farbe, Touchscreen ch	6.5-Zoll-TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6643- OCD01-1AX0	SIMATIC MP277 Touchscreen, 10 Zoll	10,4-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 640 x 480 Bildpunkte, Touchscreen, nur 24 V DC	6 MB	1 x RS422, 1 x RS485, 2 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T10C4D1	PanelView Plus 1000, Farbe, Touchscreen	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten

SIMATIC-Multi-Panels– 27x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
–	SIMATIC MP277 Touchscreen, 10 Zoll, Edelstahl	10,4-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 640 x 480 Bildpunkte, Touchscreen, nur 24 V DC, Frontblende aus Edelstahl, IP66	6 MB	1 x RS422, 1 x RS485, 2 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T10C4D1	PanelView Plus 1000, Farbe, Touchscreen	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6643-ODB01-1AX0	SIMATIC MP277 Tastatur, 8 Zoll	7,5-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 640 x 480 Bildpunkte, Tastatur, nur 24 V DC	6 MB	1 x RS422, 1 x RS485, 2 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K7C4D1	PanelView Plus 700, Farbe, Tastatur	6,5-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6643-ODD01-1AX0	SIMATIC MP277 Tastatur, 10 Zoll	10,5-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 640 x 480 Bildpunkte, Tastatur, nur 24 V DC	6 MB	1 x RS422, 1 x RS485, 2 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K10C4D1	PanelView Plus 1000, Farbe, Tastatur	10,4-Zoll- TFT-Anzeige, 640 x 480 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6622-OBA01-0AA0	WINCC FLEXIBLE STANDARD-Software	Konfigurations- und Programmiersoftware für Simatic OP/TP/MP270, MP370, OP77-, OP/TP170- und Micro-Panels	-/-	-/-	9701-VWSTMENE	Software RSView Studio for Machine Edition	Konfigurationssoftware RSView Studio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

SIMATIC-Multi-Panels, 37x-Serie, und Rockwell Automation-Äquivalente

SIMATIC-Multi-Panels– 37x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6542-ODA10-OAX0	SIMATIC MP370 Tastatur, 12 Zoll	12,1-Zoll- TFT-Anzeige, Farbe (256 Farben), 800 x 600 Bildpunkte, Tastatur, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 1 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K12C4D1	PanelView Plus 1250, Farbe, Tastatur	12,1-Zoll- TFT-Anzeige, 800 x 600 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-ODA10-OAX0	SIMATIC MP370 Touchscreen, 12 Zoll	12,1-Zoll- TFT-Anzeige, Farbe (256 Farben), 800 x 600 Bildpunkte, Touchscreen, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 1 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T12C4D1	PanelView Plus 1250, Farbe, Touchscreen	12,1-Zoll- TFT-Anzeige, 800 x 600 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-ODB10-OAX0	SIMATIC MP370 Touchscreen, 15 Zoll	15,1-Zoll- TFT-Anzeige, Farbe (256 Farben), 1024 x 768 Bildpunkte, Touchscreen, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 1 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T15C4D1	PanelView Plus 1500 Farbe, Touchscreen	15-Zoll- TFT-Anzeige, 1024 x 768 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6545-8DB10-OAA0	SIMATIC MP370 Touchscreen, 15 Zoll, Edelstahl	15,1-Zoll- TFT-Anzeige, Farbe (65 536 Farben), 1024 x 768 Bildpunkte, Touchscreen, nur 24 V DC, Frontblende aus Edelstahl, IP66	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 1 x USB, Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T15C4D1	PanelView Plus 1500, Farbe, Touchscreen	15-Zoll- TFT-Anzeige, 1024 x 768 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6 644-OAA01-2AX0	SIMATIC MP377 Touch, 12,1 Zoll	12,1-Zoll- TFT-Anzeige, 65.536 Farben, 800 x 600 Bildpunkte, Touch, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 2 x USB, 2 x Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T12C4D1	PanelView Plus 1250, Farbe, Touchscreen	12,1-Zoll- TFT-Anzeige, 800 x 600 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten

SIMATIC-Multi-Panels– 37x-Serie					Lösung von Rockwell Automation		
Siemens-Bestellnummer	Kurzreferenz	Beschreibung	Speicher	Komm.-Optionen	Rockwell Automation-Bestellnummer	Name	Beschreibung
6AV6 644-0BA01-2AX0	SIMATIC MP377 Tastatur, 12,1 Zoll	12,1-Zoll- TFT-Anzeige, 65 536 Farben, 800 x 600 Bildpunkte, Tastatur, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 2 x USB, 2 x Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-K12C4D1	PanelView Plus 1250, Farbe, Tastatur	12,1-Zoll- TFT-Anzeige, 800 x 600 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Tastatur, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6 644-0BA01-2AX0	SIMATIC MP377 Touch, 15 Zoll	15-Zoll- TFT-Anzeige, 65 536 Farben, 1024 x 768 Bildpunkte, Touchscreen, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 2 x USB, 2 x Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T15C4D1	PanelView Plus 1500, Farbe, Touchscreen	15-Zoll- TFT-Anzeige, 1024 x 768 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6 644-0BA01-2AX0	SIMATIC MP377 Touchscreen, 19 Zoll	19-Zoll- TFT-Anzeige, 65 536 Farben, 1280 x 1024 Bildpunkte, Touchscreen, nur 24 V DC	12,5 MB	1 x TTY, 2 x RS232, 1 x RS422, 1 x RS485, 2 x USB, 2 x Ethernet, S5, S7-200, S7-300/400 und Steuerungen von Drittanbietern, Druckeranschluss erhältlich	2711P-T15C4D1	PanelView Plus 1500, Farbe, Touchscreen	15-Zoll- TFT-Anzeige, 1024 x 768 Bildpunkte, 18 Bit Farbtiefe, EtherNet/IP und RS-232, Touchscreen, 24 V DC, 64 MB Flash, USB-Druckfähigkeiten
6AV6622-0BA01-0AA0	WINCC FLEXIBLE STANDARD-Software	Konfigurations- und Programmier- software für Simatic OP/TP/MP270, MP370, OP77-, OP/TP170- und Micro-Panels	-/-	-/-	9701-VWSTMENE	Software RStudio for Machine Edition	Konfigurationssoftware RStudio for Machine Edition zum Entwickeln und Testen von HMI-Anwendungen auf Maschinenebene

Notizen:

Kundendienst von Rockwell Automation

Rockwell Automation bietet Ihnen über das Internet Unterstützung zur Verwendung unserer Produkte. Unter <http://support.rockwellautomation.com> finden Sie technische Handbücher, eine Wissensdatenbank mit Antworten auf häufig gestellte Fragen, technische Hinweise und Applikationsbeispiele, Beispielcode sowie Links zu Software-Servicepaketen. Außerdem finden Sie dort die Funktion „MySupport“, über die Sie diese Tools individuell an Ihre Anforderungen anpassen können.

Zusätzlichen telefonischen Support für die Installation, Konfiguration und Fehlerbehebung erhalten Sie über unsere TechConnect Support-Programme. Wenn Sie weitere Informationen wünschen, wenden Sie sich an Ihren lokalen Distributor oder Ihren Rockwell Automation-Vertreter, oder gehen Sie auf unsere Internet-Seite <http://support.rockwellautomation.com>.

Unterstützung bei der Installation

Wenn innerhalb von 24 Stunden nach der Installation ein Problem auftritt, lesen Sie bitte die Informationen in diesem Handbuch. Sie können auch unter einer speziellen Kundendienst-Bearbeitungsnummer Unterstützung beim Einrichten und Inbetriebnehmen Ihres Moduls anfordern:

USA	+1 440 646 3434 Montag bis Freitag, 8.00 Uhr bis 17.00 Uhr EST
Außerhalb der USA	Bitte wenden Sie sich bei Fragen zur technischen Unterstützung an Ihren lokalen Rockwell Automation-Vertreter.

Rückgabeverfahren bei neuen Produkten

Rockwell Automation testet alle Produkte, um sicherzustellen, dass sie beim Verlassen des Werks voll betriebsfähig sind. Sollte das Produkt nicht ordnungsgemäß funktionieren und zurückgegeben werden müssen, gehen Sie wie folgt vor:

USA	Wenden Sie sich an Ihren Distributor. Sie müssen Ihrem Distributor eine Kundendienst-Bearbeitungsnummer angeben (diese erhalten Sie über die oben genannte Telefonnummer), damit das Rückgabeverfahren abgewickelt werden kann.
Außerhalb der USA	Bitte wenden Sie sich bei Fragen zu den Einsendevorschriften an Ihren lokalen Rockwell Automation-Vertreter.

www.rockwellautomation.com

Hauptverwaltung für Antriebs-, Steuerungs- und Informationslösungen

Amerika: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: +1 414 382 2000, Fax: +1 414 382 4444

Europa/Naher Osten/Afrika: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brüssel, Belgien, Tel: +32 2 663 0600, Fax: +32 2 663 0640

Asien/Australien/Pazifikraum: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, China, Tel: +852 2887 4788, Fax: +852 2508 1846

Deutschland: Düsselberger Straße 15, D-42781 Haan, Tel.: +49 (0)2104 960 0, Fax: +49 (0)2104 960 121

Schweiz: Buchserstrasse 7, CH-5001 Aarau, Tel.: +41(62) 889 77 77, Fax: +41(62) 889 77 11

Österreich: Kotzinastraße 9, A-4030 Linz, Tel.: +43 (0)732 38 909 0, Fax: +43 (0)732 38 909 61