Allen-Bradley

Fieldbus Solutions for Rockwell Automation's Integrated Architecture

ProcessLogix, ControlLogix, and PLC5

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of these products must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards. In no event will Allen-Bradley be responsible or liable for indirect or consequential damage resulting from the use or application of these products.

Any illustrations, charts, sample programs, and layout examples shown in this publication are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this publication, notes may be used to make you aware of safety considerations. The following annotations and their accompanying statements help you to identify a potential hazard, avoid a potential hazard, and recognize the consequences of a potential hazard:

WARNING	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
ATTENTION	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

About this Document

Contents guide

The following table summarizes each chapter in this document.

Table P.A Content Summary

Read this chapter:	If you need to:
Chapter 1, The Fieldbus Communication Model or network layer?	become familiar with the Fieldbus Foundation is or what constitutes the **FOUNDATION Fieldbus**® technology. This section also includes descriptions of some standard fieldbus function blocks and describes the role of Device Descriptions and block parameters for general reference.
Chapter 2, Integrating Fieldbus into ProcessLogix R400.0	gain some insight on what functional relationships result from the integration of fieldbus devices with a ProcessLogix system. The information in this section will be helpful background for planning and configuring your control strategy.
Chapter 3, Planning Considerations	be responsible for setting up the hardware infrastructure to support fieldbus devices. This section identifies the things you should consider before installing any equipment and provides detailed procedures for how to install the Fieldbus Interface Module (FIM) and its companion Remote Termination Panel (RTP).
Chapter 4, Configuration	be the one configuring the control strategy through Control Builder. This section provides detailed procedures for including fieldbus functional components in your overall control strategy. It includes creating hardware blocks, making templates, associating blocks, assigning modules, assigning devices, and loading components
Chapter 5, Operation	be monitoring system operation. This section provides an overview of functions you can monitor through Station displays and the Monitoring tab in Control Builder.
Chapter 6, General Maintenance, Checkout and Calibration	be responsible for maintaining and trouble shooting system operation. This section provides information about replacing components, upgrading firmware in uncommissioned devices, and checking device calibration.
Chapter 7, Using the ControlNet-to-FOUNDATION Fieldbus H1 Linking Device	use the 1788-CN2FF H1 Linking Device.
Appendix A	reference the standard function block parameters.
Appendix B	reference Fieldbus status display indications.
Appendix C	define the mode change conditions.
Appendix D	review general Fieldbus wiring considerations.
Appendix E	use Fieldbus Library Manager to create device template for Control Builder
Appendix F	follow a hands-on example explaining how to configure and monitor a field bus device using the 1788-CN2FF.

Conventions

The following table summarizes the terms and type representation conventions used in this Guide.

Table P.B Convention Definitions

Term/Type Representation	**Meaning**	**Example**
Click	Click left mouse button once. (Assumes cursor is positioned on object or selection.)	Click the Browse button.
Double-click	Click left mouse button twice in quick succession. (Assumes cursor is positioned on object or selection.)	Double click the Station icon.
Drag	Press and hold left mouse button while dragging cursor to new screen location and then release the button. (Assumes cursor is positioned on object or selection to be moved.)	Drag the PID function block onto the Control Drawing.
Right-click	Click right mouse button once. (Assumes cursor is positioned on object or selection.)	Right-click the AND function block.
<F1>	Keys to be pressed are shown in angle brackets.	Press <F1> to view the online Help.
<Ctrl>+<C>	Keys to be pressed together are shown with a plus sign.	Press <Ctrl>+<C> to close the window.
File->New	Shows menu selection as menu name followed by menu selection	Click File->New to start new drawing.
>D:\setup.exe<	Data to be keyed in at prompt or in an entry field.	Key in this path location >D:\setup.exe<.

Rockwell Automation Technical Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors, and 260 authorized systems integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Obtain Technical Product Support

If you need to contact Rockwell Automation for technical assistance, first call your local Rockwell Automation representative, then:

If you need to contact Rockwell Automation for technical assistance, try one of the following methods:

Type of technical support:	Access at:
Personalized Service	Call your local Rockwell Automation representative
Pre-sales Technical Support	1.440.646.3638 (3NET)
Post-sales Technical Support	1.440.646.5800
Email your questions	racleasktheexpert@ra.rockwell.com
Internet	www.ab.com
Publications	www.theautomationbookstore.com

Your Questions or Comments about This Manual

If you find a problem or have a comment about this manual, please notify us of it on the enclosed How Are We Doing? form (at the back of this manual).

If you have any suggestions about how we can make this manual more useful to you, please contact us at the following address:

Rockwell Automation, Allen-Bradley Company, Inc.
Control and Information Group
Technical Communication
1 Allen-Bradley Drive
Mayfield Heights, OH 44124-6118

Notes:

Chapter 2

Integrating Fieldbus into Rockwell Automation Logix System

Chapter 3

1757-FIM Planning Considerations

Chapter 4

Configurating the 1757-FIM

Chapter 5

Operating the 1757-FIM

Chapter 6

1757-FIM General Maintenance, Checkout, and Calibration

Chapter 7

Using the 1788-CN2FF, ControlNet-to-FOUNDATION Fieldbus H1 Linking Device

Appendix A

Standard Function Block Parameters

Appendix B

Fieldbus Status Display Indications

Appendix C

Mode Change Conditions

Appendix D

Fieldbus Wiring Considerations

Appendix E

Fieldbus Library Manager

Appendix F

1788-CN2FF Installation Example

List of Figures

The Fieldbus Communication Model

Fieldbus Organization

About the Fieldbus Foundation

The Fieldbus Foundation is a not-for-profit corporation made up of nearly 120 leading suppliers and customers of process control and manufacturing automation products. Since its inception in 1994, it is totally dedicated to developing one standard, "open," interoperable field communication model known as **FOUNDATION**™ Fieldbus[1].

Want more information?

Visit the Fieldbus Foundation web site at www.fieldbus.org, or the following address, for more information:

9390 Research Blvd.
Suite II-250
Austin, TX 78759-9780

What is Fieldbus?

There are many digital communication technologies being promoted as the future replacement for the venerable 4–20 mA analog standard, and most are self-described as fieldbus. With the exception of FOUNDATION fieldbus, virtually all of these technologies were developed for non-process environments such as automotive manufacturing, building automation, or discrete parts manufacturing, and later adapted to process control.

Generally, they are well suited to the applications for which they were originally developed. Some of these technologies are open, some are proprietary. Every communication technology provides a method for transmitting data between various devices and a host, and some provide communications directly between devices. The various schemes differ in how well they are optimized for moving data quickly, their suitability for real-time control, the cost of hardware implementations, their networking capability for branches, spurs and long distances, and for how power is distributed.

(1) Sections of this publication has been provided by FOUNDATION Fieldbus.

Comparisons among “fieldbus technologies” typically reduces to comparisons of data rates, message length, number of devices on a segment, etc. These are all important communications issues and each technology represents a particular set of trade-offs which adapt it to its original application, and each is rooted in the technology that was available or in vogue at the time of its development.

Using a strategy exactly opposite of FOUNDATION fieldbus, these various communications technologies minimize dependence on local intelligence in deference to minimum device cost, and maximize reliance on a centralized control architecture. Measurement instruments in such structures communicate to a central computing system at the request of that central system. A proprietary control application, running on the central system processes the field data and distributes control signals to other devices back in the field. Regardless of how open the communication scheme may be, the control application is always proprietary.

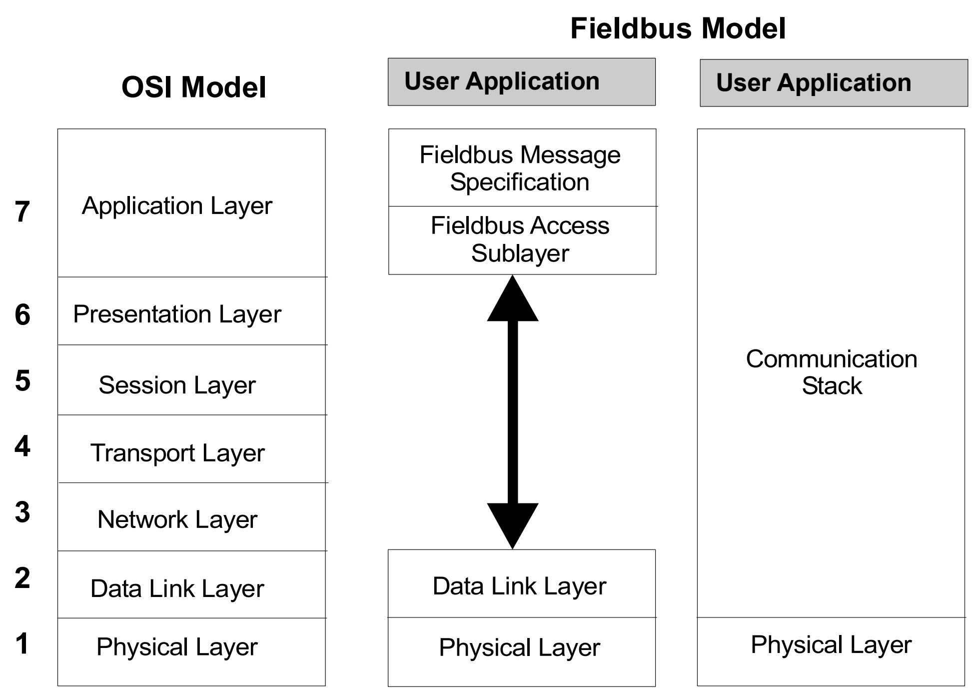
The key distinctions between these technologies and FOUNDATION fieldbus are; FOUNDATION fieldbus provides an open specification for both communications and the control application. FOUNDATION fieldbus distributes control functionality across the bus, making maximum use of local intelligence to improve performance and reduce total system cost. Devices are required to be interoperable, providing the user with tools to implement a control system with products from multiple manufacturers without custom programming. With FOUNDATION fieldbus, the network is the control system.

Open Communications Architecture

FOUNDATION Fieldbus is an enabling technology for dynamically integrating dedicated field devices with digitally based control systems. It defines how all “smart” final control devices are to communicate with other devices in the control network. The technology is based upon the International Standards Organization's Open System Interconnection (OSI) model for layered communications.

As shown in Figure 1.1, OSI layer 1 is the Physical Layer, OSI layer 2 is the Data Link Layer, and OSI layer 7 is the application layer or the Fieldbus Message Specification. A Fieldbus Access Sublayer maps the Fieldbus Message Specification onto the Data Link Layer. Fieldbus does not use OSI layers 3 to 6, and layers 2 and 7 form the Communication Stack. Also, the OSI model does not define a User Application, but the Fieldbus Foundation does.

Figure 1.1 OSI versus Fieldbus communication model



Communication Layer Description

The following table provides a summarized description of the communication layers that make up the FOUNDATION Fieldbus. The Fieldbus Foundation maintains a complete library of detailed reference specifications including a Technical Overview, and Wiring and Installation Guides.

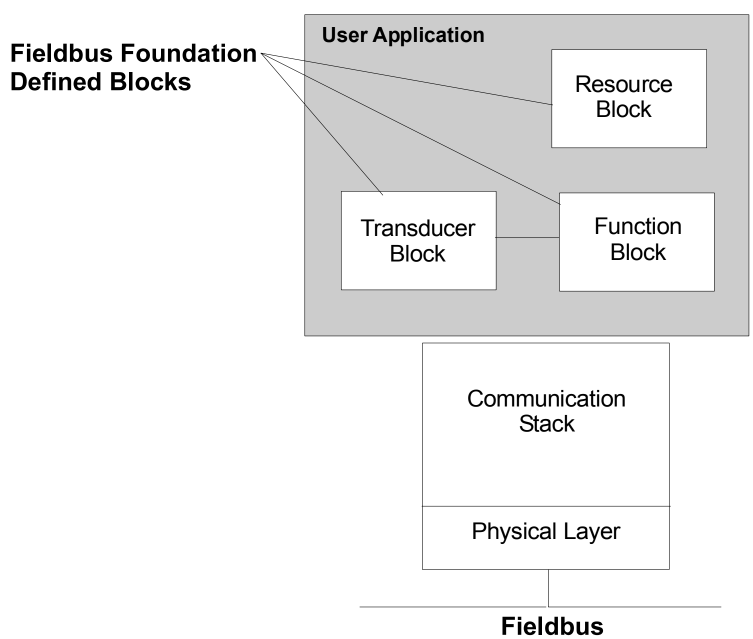
Table 1.A Communication Layer Descriptions

Layer	Functional Description	Associated Terms
Physical	Defines the transmission medium for fieldbus signals and the message conversion tasks to/from the Communication Stack. Based on the Manchester Biphase-L Encoding technique, so a FOUNDATION Fieldbus (FF) device interprets a positive transition in the middle of a bit time as logical "0" and a negative transition as logical "1". Complies with existing International Electrotechnical Commission (IEC 1158-2) and the Instrumentation, Systems, and Automation Society (ISA S50.02) physical layer standards. And, it can be used with existing 4 to 20mA wiring.	H1, 31.25 kbit/s signal rate H1 Link H1 Segment HSE, High Speed Ethernet
Data Link (DLL)	Defines how messages are transmitted on a multi-drop network. It uses a deterministic centralized bus scheduler called a Link Active Scheduler (LAS) to manage access to the fieldbus. It controls scheduled and unscheduled communications on the fieldbus in a publish/subscribe environment. Identifies device types as Basic Device, Link Master, or Bridge. A Link Master device type can become a Link Active Scheduler (LAS) for the network.	Compel Data (CD) message Pass Token (PT) message Time Distribution (TD) message Live List Link Active Scheduler (LAS)
Fieldbus Access Sublayer (FAS)	Defines the types of services used to pass information to the Fieldbus Message Specification layer. The types of services are defined as Virtual Communication Relationships (VCR). The VCR types are Client/Server, Report Distribution, and Publisher/Subscriber. The Client/Server type handles all operator messages. The Report Distribution type handles event notification and trend reports. The Publisher/Subscriber type handles the publishing of User Application function block data on the network.	Virtual Communication Relationship (VCR)
Fieldbus Message Specification (FMS)	Defines how fieldbus devices exchange User Application messages across the fieldbus using a set of standard message formats. It uses object descriptions that are stored in an object dictionary (OD) to facilitate data communication. The OD also includes descriptions for standard data types such as floating point, integer, Boolean, and bitstring. A Virtual Field Device (VFD) mirrors local device data described in the OD. A physical device may have more than one VFD. Provides these communication services to standardize the way the User Applications such as function blocks communicate over the fieldbus - Context Management, Object Dictionary, Variable Access, Event, Upload/Download, and Program Invocation. Uses a formal syntax description language called Abstract Syntax Notation 1 (ASN-1) to format FMS messages and applies special behavioral rules for certain types of objects.	Object Dictionary (OD) Virtual Field Device (VFD) Network Management Information Base (NMIB) System Management Information Base (SMIB)

Table 1.A Communication Layer Descriptions

Layer	Functional Description	Associated Terms
User Application or Function Block Application Process (FBAP)	Defines blocks to represent different types of application functions. The three types of blocks are the Resource block, the Function block, and the Transducer block. See Figure 1.2. The Resource block is used to describe characteristics of the fieldbus device such as the device name, manufacturer, and serial number. Each fieldbus device requires one Resource block. The Function block is used to define the specific characteristics of the process control function. The Fieldbus Foundation provides a set of pre-defined function blocks. A single fieldbus device can include many Function blocks to achieve the desired control functionality. See the following section, *Standard Function Blocks* for more information. The Transducer block is used to interface Function blocks with local input/output devices. They read sensors and command outputs, and contain information such as calibration date and sensor type. One Transducer block is usually included for each input or output Function block. These associated objects are also defined in the User Application: Link Objects, Trend Objects, Alert Objects, and View Objects. They provide linking between internal Function block inputs and outputs, trending of Function block parameters, reporting of alarms and events, viewing of predefined block parameter sets through one of four defined views. The four defined views are View 1 - Operation Dynamic, View 2 - Operation Static, View 3 - All Dynamic, and View 4 - Other Static.	Resource block Function block Transducer block Link Objects Trend Objects Alert Objects View Objects View 1 - Operation Dynamic View 2 - Operation Static View 3 - All Dynamic View 4 - Other Static

Figure 1.2 Function Block Application Process based on blocks



Standard Function Blocks

The key to fieldbus interoperability is the User Application or Function Block Application Process (FBAP) that defines standard function blocks that can reside in field devices and be interconnected as a distributed process control system. A function block is a named entity that has inputs, outputs, and parameters. It performs certain functions that operate on its inputs and produce outputs in accordance with its assigned parameters. The Fieldbus Foundation Function Blocks are similar in nature to the Function Blocks used to build control strategies in the Control Builder application in the ProcessLogix system.

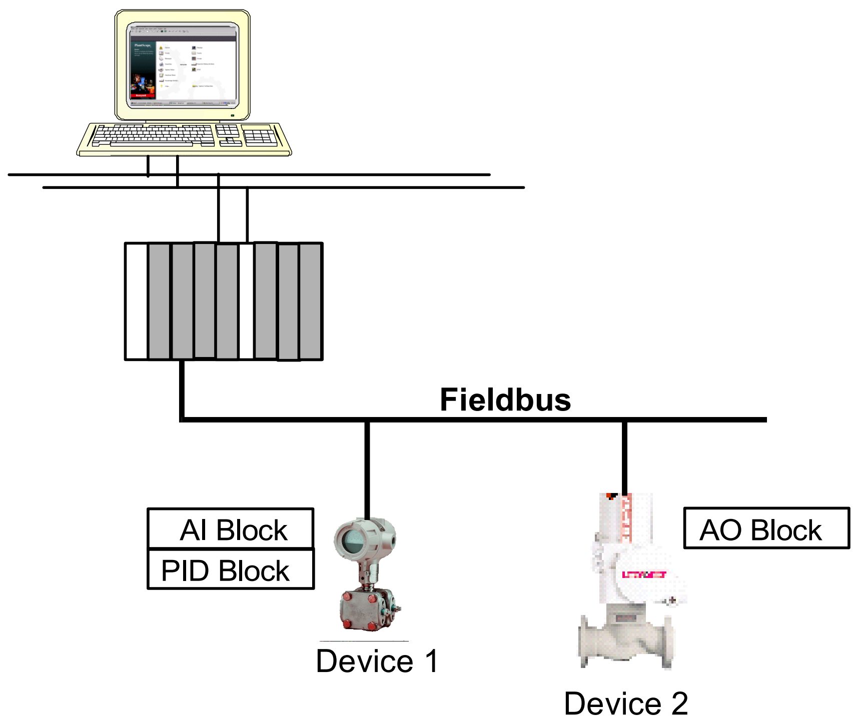
The Fieldbus Foundation provides the standard Function Blocks listed below for basic control functionality. They also support additional blocks for more complex applications. Please refer to the applicable Fieldbus Foundation specification for more information about these additional blocks.

Table 1.B Function Block Specifications

Function Block	Abbreviation	Class
Analog Input	AI	Input
Analog Output	AO	Output
Bias/Gain	BG	Control
Control Selector	CS	Control
Discrete Input	DI	Input
Discrete Output	DO	Output
Manual Loader	ML	Control
Proportional/Derivative	PD	Control
Proportional/Integral/Derivative	PID	Control
Ratio	RA	Control

Function blocks make it possible to build a control loop using fieldbus devices that include the appropriate Function block types. For example, a pressure transmitter that contains an Analog Input and Proportional/Integral/Derivative blocks can be used with a valve containing an Analog Output block to form a control loop, as shown in Figure 1.3.

Figure 1.3 Using Function Blocks in Fieldbus Devices to Form a Control Loop



About Modes of Operation

Every Function block includes a mode parameter with configured permitted modes. This structured parameter is composed of the actual mode, the target mode, the permitted mode, and the normal mode. The normal mode is the desired operating mode. The actual mode reflects the mode used during block execution. The target mode may be set and monitored through the mode parameter. The permitted mode defines the allowable target mode settings. The following table provides a summary of the available modes of operation and their effect on operation.

Table 1.C Modes of Operation

Mode	Abbreviation	Operation Effect
Out of Service	OOS	The block is not being evaluated. The output is maintained at the last value, an assigned failsafe value -last value or configured failsafe value. Set Point is maintained at last value.
Initialization Manual	IMan	The block output is being set in response to the back-calculation input parameter status. When status is no path to the final output element, control blocks must initialize to provide for bumpless transfer, when the condition clears. The Set Point may be maintained or initialized to the Process Variable parameter value.
Local Override	LO	Applies to control and output blocks that support a track input parameter. Also, manufacturers may provide a local lockout switch on the device to enable the Local Override mode. The block output is being set to track the value of the track input parameter. The algorithm must initialize to avoid a bump, when the mode switches back to the target mode. The Set Point may be maintained or initialized to the Process Variable parameter value.
Manual	Man	The block is not being calculated, although it may be limited. The operator directly sets it through an interface device. The algorithm must initialize to avoid a bump, when the mode switches. The Set Point may be maintained, initialized to the Process Variable parameter value, or initialized to the Set Point value associated with the previous (retained) target mode.
Automatic	Auto	The block's normal algorithm uses a local Set Point value to determine the primary output. An operator may set the value of the Set Point through an interface device.
Cascade	Cas	The block's normal algorithm uses a Set Point value fed through the Cascade input parameter from another block to determine the primary output value.
Remote Cascade	RCas	The block's Set Point is being set by a Control Application running on an interface device through the remote-cascade in parameter. The block's normal algorithm uses this Set Point to determine the primary output value. The block maintains a remote-cascade out parameter to support initialization of the control application, when the block mode is not remote-cascade.
Remote-Out	ROut	The block's output is being set by a Control Application running on an interface device through the remote-output in parameter. The algorithm must initialize to avoid a bump, when the mode switches. The block maintains a remote-output out parameter to support initialization of the Control Application, when the block mode is not remote-output. The Set Point may be maintained or initialized to the Process Variable parameter value.

Analog Input Block

Figure 1.4 Functional Schematic for Analog Input Function Block

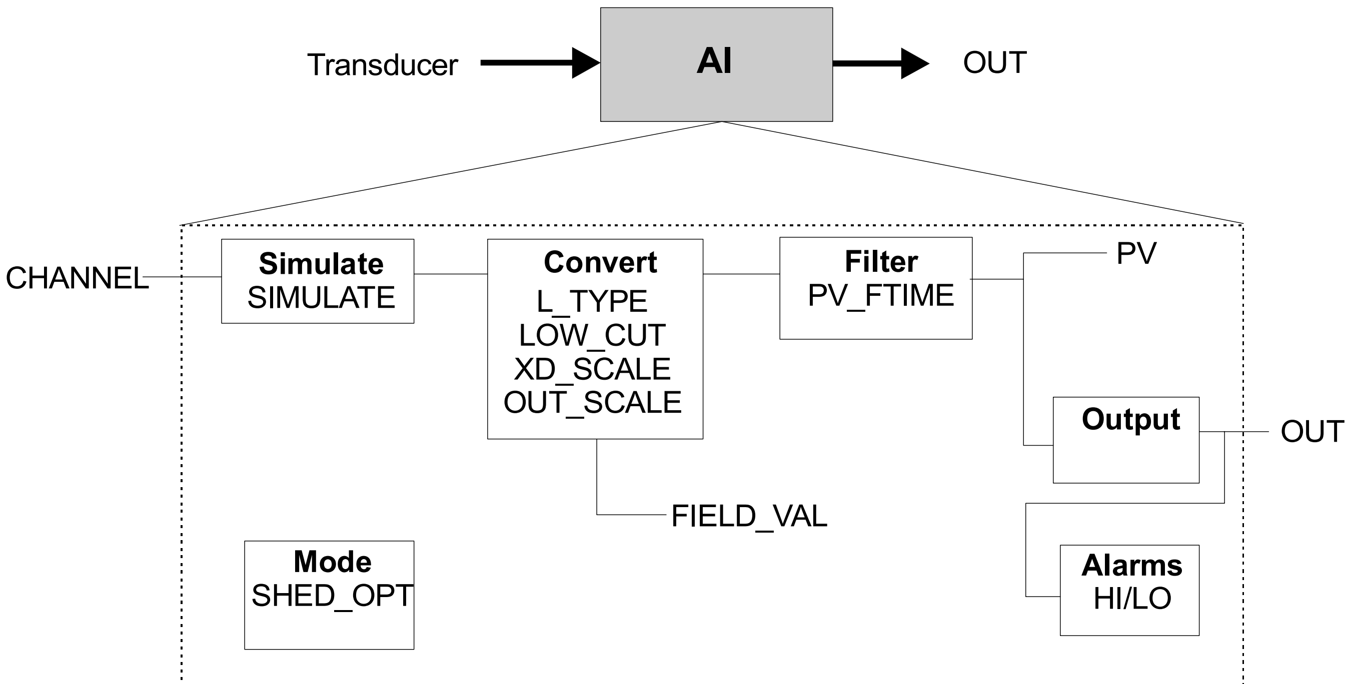


Table 1.D Analog Input Block Specifications

Description	The AI function block takes the input data from a Transducer block and calculates an output to be fed to other fieldbus function blocks. A functional schematic of the block is shown in Figure 1.4 for reference.
Function Notes	• Supports Out of Service (OOS), Manual (Man), and Automatic (Auto) modes. • The XD_SCALE units code must match the channel units code, or the block will remain in OOS mode after being configured. • The OUT_SCALE is normally the same as the transducer, unless the L_TYPE is set to Indirect or Ind Sqr Root, then the OUT_SCALE determines the conversion from FIELD_VAL to the output. • If the mode is Auto, the PV is the value the block puts in OUT. • If the mode is Man, an operator can write a value to OUT. • The SIMULATE parameter is for testing purposes only and always initializes in the disabled state.
Equation Options	FIELD_VAL = 100 x (channel value - EU@0%) / (EU@100% - EU@0%) [XD_SCALE] Direct: PV = channel value Indirect : PV = (FIELD_VAL / 100) x (EU@100% - EU@0%) + EU@0% [OUT_SCALE] Ind Sqr Root: PV = sqrt(FIELD_VAL / 100) x (EU@100% - EU@0%) + EU@0% [OUT_SCALE]
Parameters (see Appendix A for definitions of each parameter)	ACK_OPTION ALARM_HYS ALARM_SUM ALERT_KEY BLOCK_ALM BLOCK_ERR CHANNEL FIELD_VAL GRANT_DENY HI_ALM HI_HI_ALM HI_HI_LIM HI_HI_PRI HI_LIM HI_PRI IO_OPTS L_TYPE LO_ALM LO_LIM LO_LO_ALM LO_LO_LIM LO_LO_PRI LO_PRI LOW_CUT MODE_BLK OUT OUT_SCALE PV PV_FTIME SIMULATE ST_REV STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT XD_SCALE

Analog Output Block

Figure 1.5 Functional Schematic for Analog Output Function Block

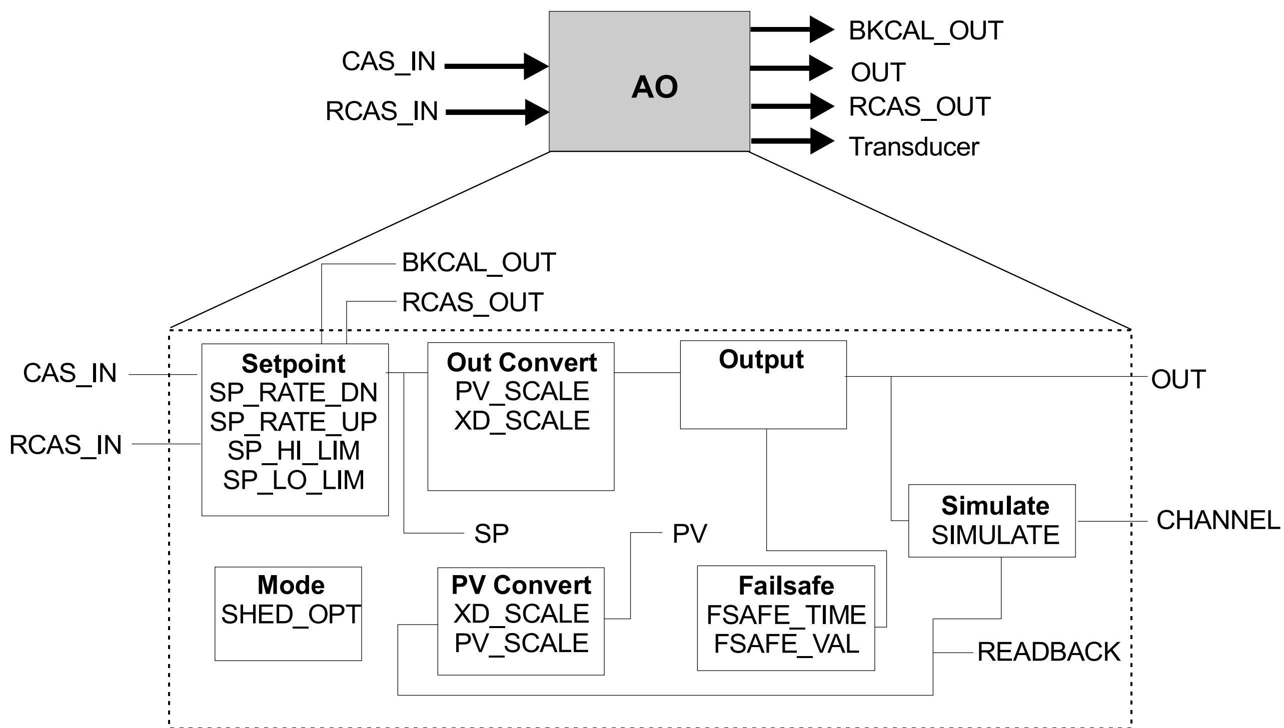


Table 1.E Analog Output Specifications

Description	The Analog Output function block converts the set point (SP) value to a number that can be used by the hardware associated with the CHANNEL selection. A functional schematic of the block is shown in Figure 1.5 for reference.			
Function Notes	• Can use either the Set point (SP) value after limiting or the Process Variable (PV) value for the BKCAL_OUT value. • Supports Out of Service (OOS), Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), and Remote Cascade (RCas) modes. • The conversion of Set point (SP) to percent of span is based on the PV_SCALE range. • The conversion of the percent of span to a compatible value for the hardware is based on the XD_SCALE range. • Use the Increase to Close Option in IO_OPTS to invert the span. • Use the Cascade mode to transfer the output of another block to the Set point of the AO block. • If the hardware, such as a valve positioner, supports a readback value, run this value backwards through the XD scaling to act as the PV for this block. If this is not supported, READBACK is generated from OUT. • In the Man mode, an operator can write a value to OUT. A manufacturer must put operational limits in the Transducer, where an operator cannot access them, to permit the Man mode. If Man mode is not permitted, it must be supported as a transition mode for exiting the OOS mode • The SIMULATE parameter is for testing purposes only and always initializes in the disabled state.			
Equation Options	Temp = (SP - EU@0%) / (EU@100% - EU@0%) [PV_SCALE] OUT = Temp x (EU@100% - EU@0%) + EU@0% [XD_SCALE] Temp = (READBACK - EU@0%) / (EU@100% - EU@0%) [XD_SCALE] PV = Temp x (EU@100% - EU@0%) + EU@0% [PV_SCALE]			
Parameters (see Appendix A for definitions of each parameter)	ALERT_KEY BKCAL_OUT BLOCK_ALM BLOCK_ERR CAS_IN CHANNEL FSAFE_TIME FSAFE_VAL	GRANT_DENY IO_OPTS MODE_BLK OUT PV PV_SCALE RCAS_IN RCAS_OUT	READBACK SHED_OPT SIMULATE SP SP_HI_LIM SP_LO_LIM SP_RATE_DN SP_RATE_UP	ST_REV STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT XD_SCALE

Bias/Gain Block

Figure 1.6 Functional Schematic for Bias/Gain Function Block

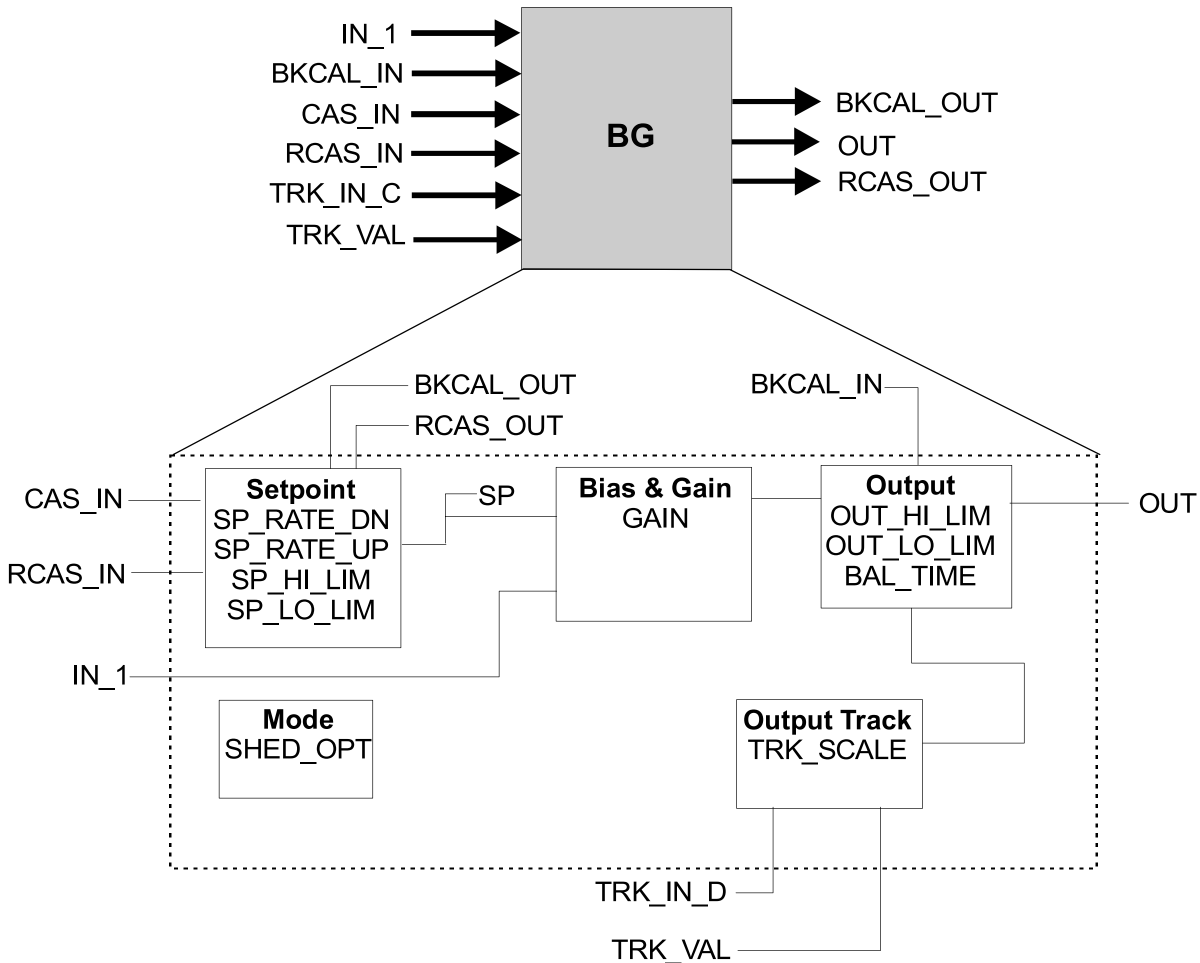


Table 1.F Bias/Gain Block Specifications

Description	The Bias/Gain function block can be used for biased external feedforward control or to set several unit controllers, such as boiler masters, from one controller output, such as a plant master. A functional schematic of the block is shown in Figure 1.6 for reference.
Function Notes	• Supports Out of Service (OOS), Initialization Manual (IMan) Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), and Remote Cascade (RCas) modes. • The output supports the track algorithm. • The Balance Ramp option is supported. • The CONTROL_OPTS selection Act on IR determines whether initialization requests are to be passed on or acted on locally by changing the BIAS value. • If the Act on IR option is false, a status of Not Invited (NI) or Initialization Request (IR) at BKCAL_IN will be passed to BKCAL_OUT. The BKCAL_OUT value will be calculated from the value of BKCAL_IN adjusted for SP and GAIN, as determined by the control or process status of IN_1. When the upstream block sends an Initialization Acknowledge (IA) status, this block will send IA status, since its output will now be nearly equal to the value of BKCAL_IN. • If the Act on IR option is true, a status of NI or IR at BKCAL_IN results in an adjustment to SP to balance OUT to the value of BKCAL_IN. The IA status can be sent as soon as IR is detected. BKCAL_OUT will not request initialization. • The TRK_VAL input brings in an external value or uses a constant. The TRK_SCALE values convert the TRK_VAL to a percent of output span value. If the CONTROL_OPTS Track Enable selection is true and TRK_IN_D is true, the converted TRK_VAL replaces the output (OUT), when the block is in Automatic, Cascade, or Remote Cascade mode. The CONTROL_OPTS Track in Manual selection must be true for this to occur in Manual mode. If the actual mode is OOS or IMan, the track request is ignored. • If the TRK_VAL replaces the OUT, its status becomes Locked Out with Limits set to Constant. The actual mode goes to LO. The status of RCAS_OUT goes to Not Invited (NI), if not already there. • If the status of TRK_IN_D is Bad, its last usable value will be maintained and acted upon. If the device restarts, losing the last usable value, it will be set to false. • If the status of TRK_VAL is Bad, the last usable value will be used. If there is no last usable value, the present value of the OUT will be used.
Equation Options	In Automatic mode: OUT = (IN_1 + SP) x GAIN If IN_1 has Non-Cascade status: BKCAL_OUT = (BKCAL_IN / GAIN) - IN_1 If IN_1 has Cascade status: BKCAL_OUT = (BKCAL_IN / GAIN) - SP
Parameters (see Appendix A for definitions of each parameter)	ALERT_KEY BAL_TIME BKCAL_IN BKCAL_OUT BLOCK_ALM BLOCK_ERR CAS_IN CONTROL_OPTS GAIN GRANT_DENY IN_1 MODE_BLK OUT OUT_HI_LIM OUT_LO_LIM OUT_SCALE RCAS_IN RCAS_OUT SHED_OPT SP SP_HI_LIM SP_LO_LIM SP_RATE_DN SP_RATE_UP ST_REV STATUS_OPTS STRATEGY TAG_DESC TRK_IN_D TRK_SCALE TRK_VAL UPDATE_EVT

Control Selector Block

Figure 1.7 Functional Schematic for Control Selector Function Block

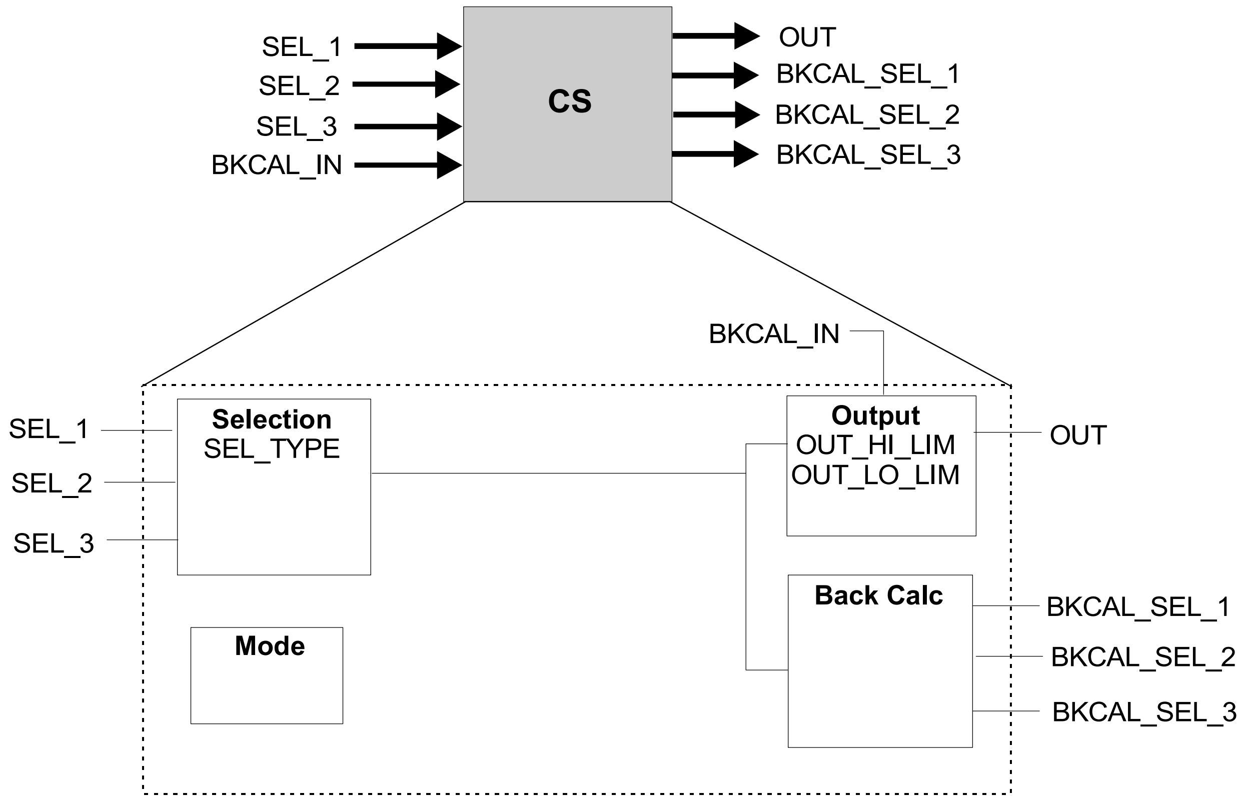


Table 1.G Control Selector Block Specifications

Description	The Control Selector function block accepts input from up to three control signals and selects one for output based on the SEL_TYPE setting of High, Middle, or Low. A functional schematic of the block is shown in Figure 1.7 for reference.			
Function Notes	• All inputs must have the same scaling as OUT, since any one can be selected for OUT. • Supports Out of Service (OOS), Initialization Manual (IMan) Local Override (LO), Manual (Man), and Automatic (Auto) modes. • If an input has a sub-status of Do Not Select, it will not be selected. • Three separate back calculation outputs (BKCAL_SEL_1, 2, 3) are available - one for each input (SEL_1, 2, 3). • The status will identify those inputs that are not selected. Control signals that are not selected are limited in one direction only as determined by the SEL_TYPE selection. • The value of each BKCAL_SEL_1, 2, 3 output is the same as OUT. The limits of back calculation outputs corresponding to not-selected inputs will be high for a low selection, low for a high selection, or one of each for a middle selection. • If the status of an input is Bad, it is not eligible for selection. If the status of an input is Uncertain, it is treated as Bad unless the STATUS_OPTS selection is Use Uncertain as Good. • When all inputs are Bad, the actual mode goes to Manual. This condition will set Initiate Failsafe (IFS) in the output status, if the STATUS_OPTS setting is IFS if BAD IN. • If SEL_TYPE selection is Middle and only two inputs are good, the higher input will be selected. • If the status of BKCAL_IN is Not Invited (NI) or Initialization Request (IR), it is passed back on all three back calculation outputs. This causes all initializable inputs to initialize to the BKCAL_IN value. Otherwise, if the status of BKCAL_IN is not normal, it is passed back on the BKCAL_SEL_N, where N is the number of the selected input. The back calculation outputs for not-selected inputs just have the Not Selected status with the appropriate high or low limit set. • When the mode is Manual, no input is selected. All three back calculation outputs will have a Not Invited status and Constant limits, with a value equal to OUT.			
Parameters (see Appendix A for definitions of each parameter)	ALERT_KEY BKCAL_IN BKCAL_SEL_1 BKCAL_SEL_2 BKCAL_SEL_3 BLOCK_ALM	BLOCK_ERR GRANT_DENY MODE_BLK OUT OUT_HI_LIM OUT_LO_LIM	OUT_SCALE SEL_1 SEL_2 SEL_3 SEL_TYPE ST_REV	STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT

Discrete Input Block

Figure 1.8 Functional Schematic for Discrete Input Function Block

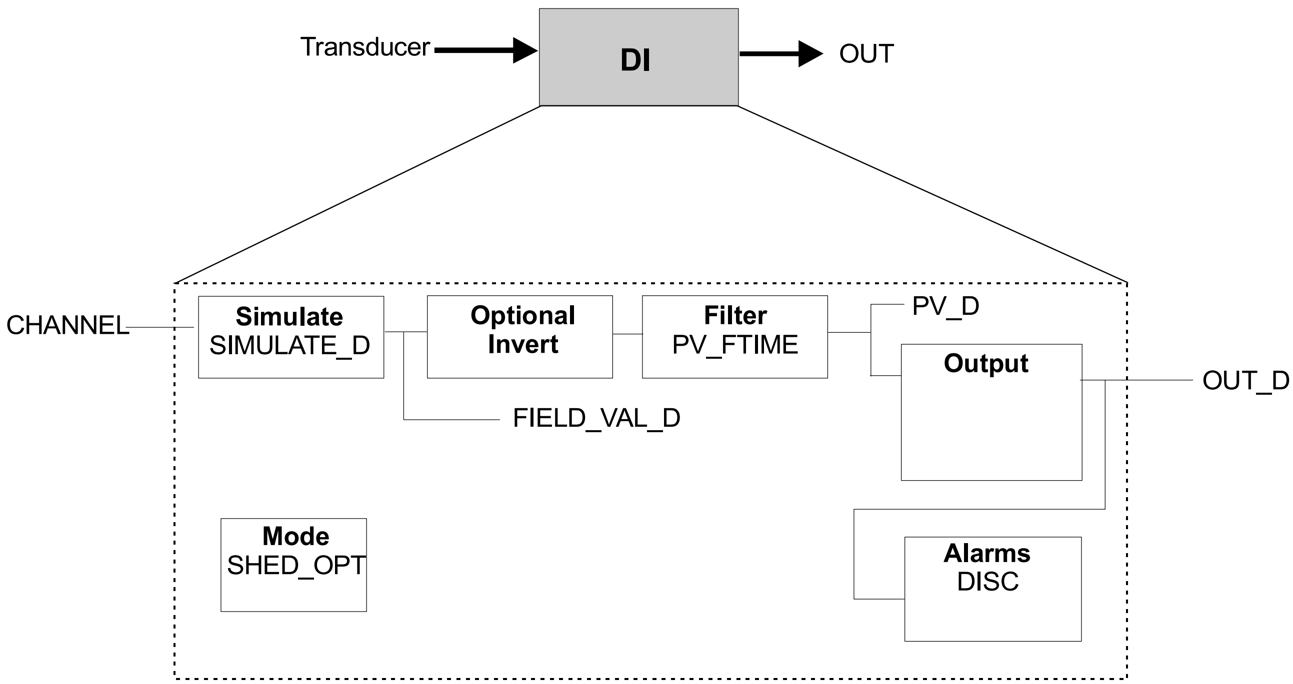


Table 1.H Discrete Input Block Specifications

Description	The Discrete Input function block takes the discrete input data from a selected Transducer block channel and provides it as an output for other fieldbus function blocks. A functional schematic of the block is shown in Figure 1.8 for reference.		
Function Notes	• Supports Out of Service (OOS), Manual (Man), and Automatic (Auto) modes. • The FIELD_VAL_D represents the true ON/OFF state of the value from the Transducer, using XD_STATE. • Use the IO_OPTS Invert selection to do a Boolean NOT function between the field value and the output. • Use the PV_FTIME to set the time that the input must be in one state before it gets passed to the PV_D. • The PV_D is always the value that the block places in OUT_D, when the mode is Automatic. • In Manual mode, if allowed, an operator can write a value to OUT_D. • The SIMULATE_D parameter is for testing purposes only and always initializes in the disabled state.		
Parameters (see Appendix A for definitions of each parameter)	ACK_OPTION ALARM_SUM ALERT_KEY BLOCK_ALM BLOCK_ERR CHANNEL DISC_ALM DISC_LIM	DISC_PRI FIELD_VAL_D GRANT_DENY IO_OPTS MODE_BLK OUT_D OUT_STATE PV_D	PV_FTIME SIMULATE_D ST_REV STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT XD_STATE

Discrete Output Block

Figure 1.9 Functional Schematic for Discrete Output Function Block

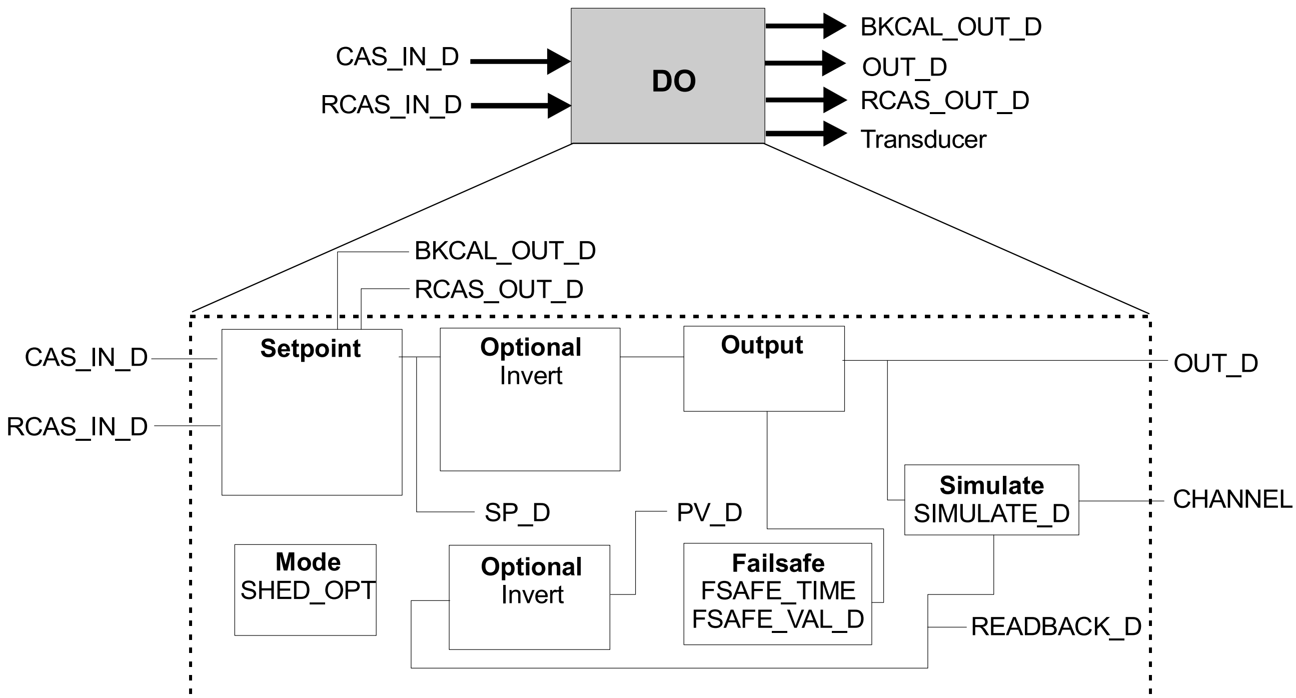


Table 1.I Discrete Output Block Specifications

Description	The Discrete Output function block converts the value in SP_D to something useful for the hardware linked to the CHANNEL selection. A functional schematic of the block is shown in Figure 1.9 for reference.		
Function Notes	• Supports Out of Service (OOS), Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), and Remote Cascade (RCas) modes. • The Set point (SP_D) supports the full cascade sub-function. • Use the Cascade mode to transfer the output of another block to the Set point (SP_D) of the DO block. • Use the IO_OPTS Invert selection to do a Boolean NOT function between the field value and the output. • Use the IO_OPTS Invert selection to do a Boolean NOT function between the SP_D and the output. • If the hardware supports a readback value, it is used for READBACK_D, and, after accounting for the IO_OPTS Invert selection, acts as the PV_D for this block. If this is not supported, READBACK is generated from OUT_D. • In the Man mode, an operator can force the output, in a programmable logic controller sense. If Man mode is not permitted, it must be supported as a transition mode for exiting the OOS mode • The SIMULATE_D parameter is for testing purposes only and always initializes in the disabled state.		
Parameters (see Appendix A for definitions of each parameter)	ALERT_KEY BKCAL_OUT_D BLOCK_ALM BLOCK_ERR CAS_IN_D CHANNEL FSAFE_TIME FSAFE_VAL_D GRANT_DENY	IO_OPTS MODE_BLK OUT_D PV_D PV_STATE RCAS_IN_D RCAS_OUT_D READBACK_D SHED_OPT	ST_REV STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT XD_STATE SIMULATE_D SP_D

Manual Loader Block

Figure 1.10 Functional Schematic for Manual Loader Function Block

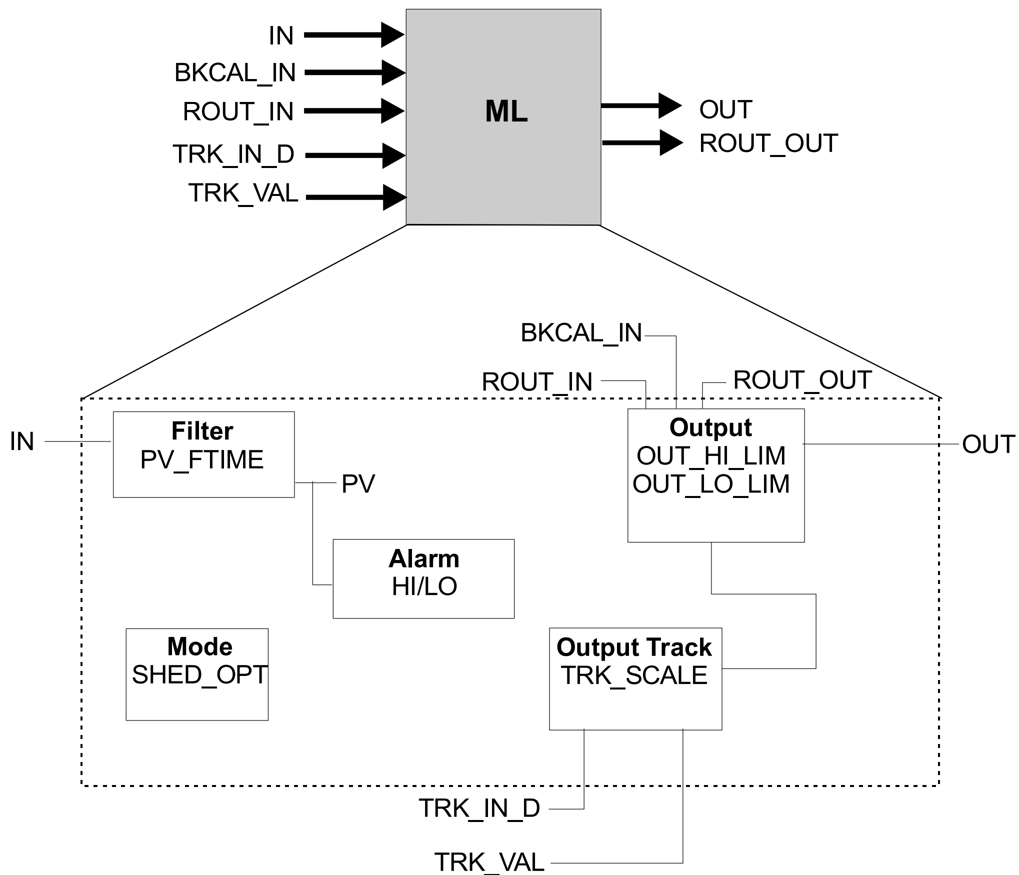


Table 1.J Manual Loader Block Specifications

Description	The Manual Loader function block output is not set by the block's algorithm. Its output can be set by an operator in the Manual mode or a program in the Remote-Out mode. A functional schematic of the block is shown in Figure 1.10 for reference.
Function Notes	• Supports Out of Service (OOS), Initialization Manual (IMan), Local Override (LO), Manual (Man), and Remote-Out (ROut) modes. • Accepts output from an AI block as its input (IN) to get a PV filtered by PV_FTIME. • The block's algorithm uses value and status for alarming only. • If selected, the STATUS_OPTS of IFS if BAD IN will work. • The BKCAL_IN value and status can force balancing of the output. • The TRK_VAL input brings in an external value or uses a constant. The TRK_SCALE values convert the TRK_VAL to a percent of output span value. If the CONTROL_OPTS Track Enable selection is true and TRK_IN_D is true, the converted TRK_VAL replaces the output (OUT), when the block is in Remote-Out (ROut) mode. The CONTROL_OPTS Track in Manual selection must be true for this to occur in Manual mode. If the actual mode is OOS or IMan, the track request is ignored. • If the TRK_VAL replaces the OUT, its status becomes Locked Out with Limits set to Constant. The actual mode goes to LO. The status of ROUT_OUT goes to Not Invited (NI), if not already there. • If the status of TRK_IN_D is Bad, its last usable value will be maintained and acted upon. If the device restarts, losing the last usable value, it will be set to false. • If the status of TRK_VAL is Bad, the last usable value will be used. If there is no last usable value, the present value of the OUT will be used.
Parameters (see Appendix A for definitions of each parameter)	ACK_OPTION ALARM_HYS ALARM_SUM ALERT_KEY BKCAL_IN BLOCK_ALM BLOCK_ERR CONTROL_OPTS GRANT_DENY HI_ALM HI_HI_ALM HI_HI_LIM HI_HI_PRIHI_LIM HI_PRI IN LO_ALM LO_LIM LO_LO_ALM LO_LO_LIM LO_PRI LO-LO_PRI MODE_BLK OUT OUT_HI_LIM OUT_LO_LIM OUT_SCALE PV PV_FTIME PV_SCALE ROUT_IN ROUT_OUT SHED_OPT ST_REV STATUS_OPTS STRATEGY TAG_DESC TRK_IN_D TRK_SCALE TRK_VAL UPDATE_EVT

Proportional/Derivative Block

Figure 1.11 Functional Schematic for Proportional/Derivative Function Block

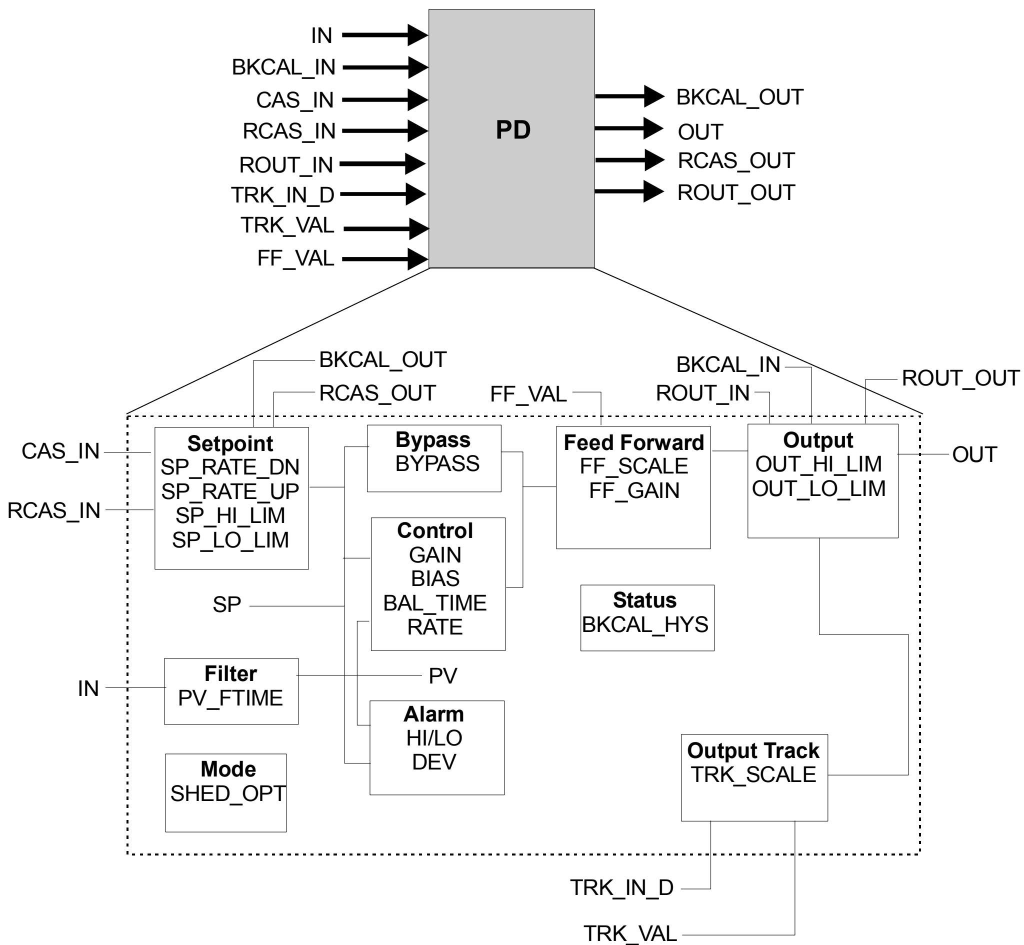


Table 1.K Proportional/Derivative Block

Description	The Proportional/Derivative function block provides classic two-mode control function for processes that handle their own integration. When the Process Variable deviates from the Set point, the PD function acts upon the error to move the output in a direction to correct the deviation. PD blocks support cascade applications to compensate for the difference in process time constants of a primary and secondary process measurement. A functional schematic of the block is shown in Figure 1.11 for reference
Function Notes	• Supports Out of Service (OOS), Initialization Manual (IMan), Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), Remote Cascade (RCas) and Remote-Out (ROut) modes. • The input (IN) passes through a filter with a time constant (PV_FTIME). The filtered value becomes the Process Variable (PV) to be used with the Set point (SP) in the block's algorithm. • The full cascade SP sub-function is used, with rate and absolute limits. Additional control options are available to have the SP value track the PV value, when the block's actual mode is IMan, LO, Man, or ROut. Limits do not cause SP-PV tracking. • The tuning constant used for the Proportional term is GAIN and RATE is used for the Derivative term. Some controllers use the inverse values of Proportional Band and repeats per minutes for their tuning constants. Users can choose which tuning constants they want to display.

Table 1.K Proportional/Derivative Block

Function Notes (cont.)	• A BYPASS switch function is available for operators to use, when secondary cascade controllers have a bad PV and the Bypass Enable (LSB) CONTROL_OPTS is ON. The Bypass Enable option is required, since some control schemes may become unstable when BYPASS is ON. An operator can only set the BYPASS switch, when the block is in the Man or OOS mode. While BYPASS is ON, the SP value, in percent of range, is passed directly to the target output, and the value of OUT is used for BKCAL_OUT. When block mode switches to Cascade, the upstream block is requested to initialize to the value of OUT. Upon transition to bypass OFF, the upstream block is requested to initialize to the PV value, regardless of the Use PV for BKCAL_OUT CONTROL_OPTS status. • Use the Balance Ramp CONTROL_OPTS to maintain the BIAS value, when the block is in Manual (Man) mode. An internal value follows the actual value required to maintain balance. When block mode changes to Automatic (Auto), the internal value ramps to zero contribution in BAL_TIME seconds. If Balance Ramp option is OFF or not used, the BIAS value immediately changes to follow the changes to the input or output, when the block is in Man mode. • Use the Act on IR CONTROL_OPTS to select whether to ignore initialization requests or act on them by changing the BIAS. If this option is ON, a status of Not Invited (NI) or Initialization Request (IR) at BKCAL_IN causes the BIAS term to be adjusted to balance OUT to the value of BKCAL_IN. • Use the Direct Acting CONTROL_OPTS to define how a change in PV relative to the SP affects the output. When Direct Acting is ON, the output increases when the PV exceeds the SP. When Direct Acting is OFF, the output decreases when the PV exceeds the SP. Be sure this option is set correctly and never changed while in the Automatic mode, since it makes the difference between positive and negative feedback. This option setting also affects the calculation of the limit states for BKCAL_OUT. • This block includes a Feed Forward algorithm. It accepts a value that is proportional to some disturbance in the control loop as its FF_VAL input. The FF_SCALE values convert the FF_VAL to a percent of output span value. The converted value is multiplied by the FF_GAIN and added to the target output of the block's algorithm. If the status of FF_VAL is Bad, the last usable value will be used to prevent a bump in the output. When the status returns to Good, the block adjusts its BIAS term to maintain the previous output. • The TRK_VAL input brings in an external value or uses a constant. The TRK_SCALE values convert the TRK_VAL to a percent of output span value. If the CONTROL_OPTS Track Enable selection is true and TRK_IN_D is true, the converted TRK_VAL replaces the output (OUT), when the block is in Automatic (Auto), Cascade (Cas), Remote Cascade (RCas), or Remote-Out (ROut) mode. The CONTROL_OPTS Track in Manual selection must be true for this to occur in Manual mode. If the actual mode is OOS or IMan, the track request is ignored. • If the TRK_VAL replaces the OUT, its status becomes Locked Out with Limits set to Constant. The actual mode goes to LO. The status of BKCAL_OUT, RCAS_OUT and ROUT_OUT goes to Not Invited (NI), if not already there. • If the status of TRK_IN_D is Bad, its last usable value will be maintained and acted upon. If the device restarts, losing the last usable value, it will be set to false. • If the status of TRK_VAL is Bad, the last usable value will be used. If there is no last usable value, the present value of the OUT will be used. • Use the Obey SP limits if Cas or RCas CONTROL_OPTS to use SP value after limiting in Cas or RCas mode. • Use the Use PV for BKCAL_OUT CONTROL_OPTS to the PV value for the BKCAL_OUT value.

Parameters (see Appendix A for definitions of each parameter)	ACK_OPTION ALARM_HYS ALARM_SUM ALERT_KEY BAL_TIME BIAS BKCAL_HYS BKCAL_IN BKCAL_OUT BLOCK_ALM BLOCK_ERR BYPASS CAS_IN CONTROL_OPTS DV_HI_ALM DV_HI_LIM DV_HI_PRI	DV_LO_ALM DV_LO_LIM DV_LO_PRI FF_GAIN FF_SCALE FF_VAL GAIN GRANT_DENY HI_ALM HI_HI_ALM HI_HI_LIM HI_HI_PRI HI_LIM HI_PRI IN LO_ALM LO_LIM	LO_LO_ALM LO_LO_LIM LO_LO_PRI LO_PRI MODE_BLK OUT OUT_HI_LIM OUT_LO_LIM OUT_SCALE PV PV_FTIME PV_SCALE RATE RCAS_IN RCAS_OUT ROUT_IN ROUT_OUT	SHED_OPT SP SP_HI_LIM SP_LO_LIM SP_RATE_DN SP_RATE_UP ST_REV STATUS_OPTS STRATEGY TAG_DESC TRK_IN_D TRK_SCALE TRK_VAL UPDATE_EVT

Proportional/Integral/Derivative Block

Figure 1.12 Functional Schematic for Porportional/Integral/Derivative Function Block

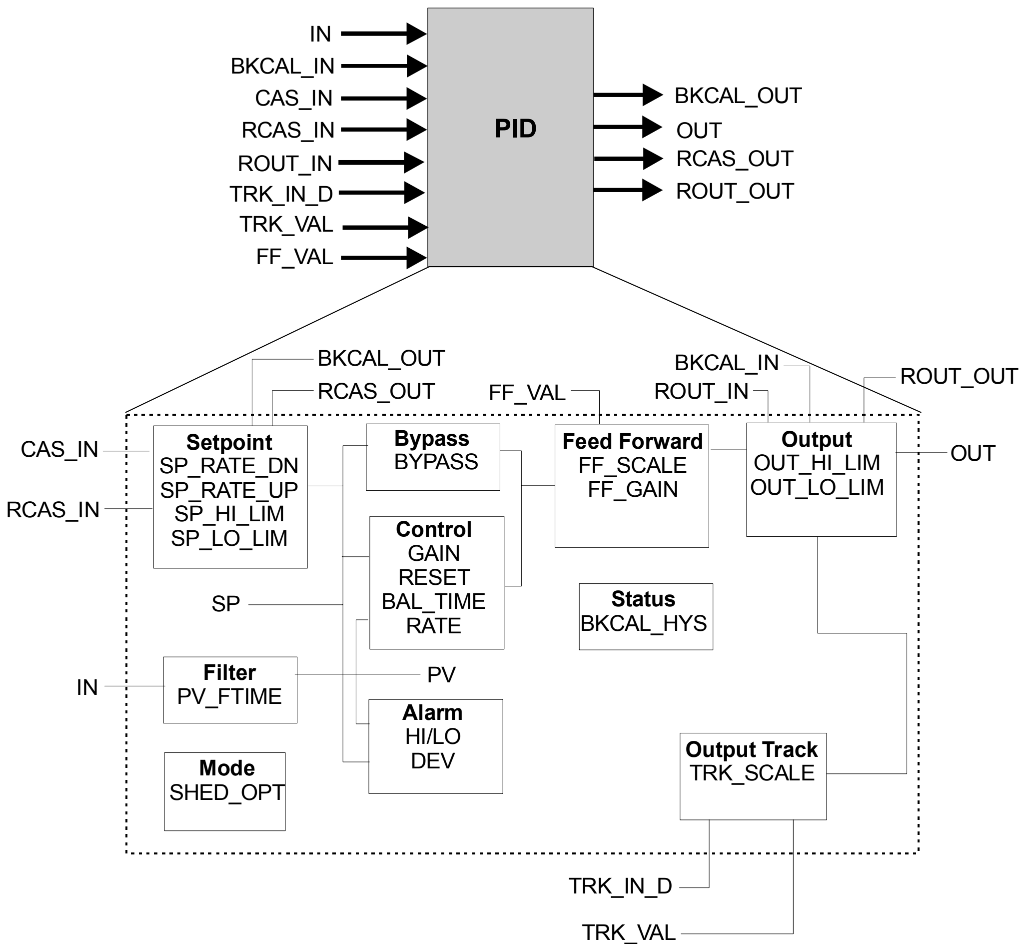


Table 1.L Proportional/Integral/Derivative Block Specifications

Description	The Proportional/Integral/Derivative function block provides classic three-mode control function for closed-loop control applications. When the Process Variable deviates from the Set point, the PID function acts upon the error to move the output in a direction to correct the deviation. PID blocks support cascade applications to compensate for the difference in process time constants of a primary and secondary process measurement. A functional schematic of the block is shown in Figure 1.12 for reference.
Function Notes	• Supports Out of Service (OOS), Initialization Manual (IMan), Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), Remote Cascade (RCas) and Remote-Out (ROut) modes. • The input (IN) passes through a filter with a time constant (PV_FTIME). The filtered value becomes the Process Variable (PV) to be used with the Set point (SP) in the block's algorithm. A PID algorithm will not integrate, if the limit status of the input (IN) is constant. • The full cascade SP sub-function is used, with rate and absolute limits. Additional control options are available to have the SP value track the PV value, when the block's actual mode is IMan, LO, Man, or ROut. Limits do not cause SP-PV tracking.

Table 1.L Proportional/Integral/Derivative Block Specifications

<table>
<tr><td>Function Notes (cont.)</td><td>

The tuning constant used for the Proportional term is GAIN, RESET is used for the Integral term, and RATE is used for the Derivative term. Both RESET and RATE are time constants expressed in seconds. Some controllers use the inverse values of Proportional Band and repeats per minutes for their tuning constants. Users can choose which tuning constants they want to display.
A BYPASS switch function is available for operators to use, when secondary cascade controllers have a bad PV and the Bypass Enable (LSB) CONTROL_OPTS is ON. The Bypass Enable option is required, since some control schemes may become unstable when BYPASS is ON. An operator can only set the BYPASS switch, when the block is in the Man or OOS mode. While BYPASS is ON, the SP value, in percent of range, is passed directly to the target output, and the value of OUT is used for BKCAL_OUT. When block mode switches to Cascade, the upstream block is requested to initialize to the value of OUT. Upon transition to bypass OFF, the upstream block is requested to initialize to the PV value, regardless of the Use PV for BKCAL_OUT CONTROL_OPTS status.
Use the Direct Acting CONTROL_OPTS to define how a change in PV relative to the SP affects the output. When Direct Acting is ON, the output increases when the PV exceeds the SP. When Direct Acting is OFF, the output decreases when the PV exceeds the SP. Be sure this option is set correctly and never changed while in the Automatic mode, since it makes the difference between positive and negative feedback. This option setting also affects the calculation of the limit states for BKCAL_OUT.
This block includes a Feed Forward algorithm. It accepts a value that is proportional to some disturbance in the control loop as its FF_VAL input. The FF_SCALE values convert the FF_VAL to a percent of output span value. The converted value is multiplied by the FF_GAIN and added to the target output of the block's algorithm. If the status of FF_VAL is Bad, the last usable value will be used to prevent a bump in the output. When the status returns to Good, the block adjusts its Integral (RESET) term to maintain the previous output.
The TRK_VAL input brings in an external value or uses a constant. The TRK_SCALE values convert the TRK_VAL to a percent of output span value. If the CONTROL_OPTS Track Enable selection is true and TRK_IN_D is true, the converted TRK_VAL replaces the output (OUT), when the block is in Automatic (Auto), Cascade (Cas), Remote Cascade (RCas), or Remote-Out (ROut) mode. The CONTROL_OPTS Track in Manual selection must be true for this to occur in Manual mode. If the actual mode is OOS or IMan, the track request is ignored.
If the TRK_VAL replaces the OUT, its status becomes Locked Out with Limits set to Constant. The actual mode goes to LO. The status of BKCAL_OUT, RCAS_OUT and ROUT_OUT goes to Not Invited (NI), if not already there.
If the status of TRK_IN_D is Bad, its last usable value will be maintained and acted upon. If the device restarts, losing the last usable value, it will be set to false.
If the status of TRK_VAL is Bad, the last usable value will be used. If there is no last usable value, the present value of the OUT will be used.
Use the Obey SP limits if Cas or RCas CONTROL_OPTS to use SP value after limiting in Cas or RCas mode.
Use the Use PV for BKCAL_OUT CONTROL_OPTS to the PV value for the BKCAL_OUT value.

</td></tr>
<tr><td>Parameters (see Appendix A for definitions of each parameter)</td><td>
<table>
<tr><td>ACK_OPTION
ALARM_HYS
ALARM_SUM
ALERT_KEY
BAL_TIME
BKCAL_HYS
BKCAL_IN
BKCAL_OUT
BLOCK_ALM
BLOCK_ERR
BYPASS
CAS_IN
CONTROL_OPTS
DV_HI_ALM
DV_HI_LIM
DV_HI_PRI
DV_LO_ALM</td>
<td>DV_LO_LIM
DV_LO_PRI
FF_GAIN
FF_SCALE
FF_VAL
GAIN
GRANT_DENY
HI_ALM
HI_HI_ALM
HI_HI_LIM
HI_HI_PRI
HI_LIM
HI_PRI
IN
LO_ALM
LO_LIM
LO_LO_ALM</td>
<td>LO_LO_LIM
LO_LO_PRI
LO_PRI
MODE_BLK
OUT
OUT_HI_LIM
OUT_LO_LIM
OUT_SCALE
PV
PV_FTIME
PV_SCALE
RATE
RCAS_IN
RCAS_OUT
RESET
ROUT_IN
ROUT_OUT</td>
<td>SHED_OPT
SP
SP_HI_LIM
SP_LO_LIM
SP_RATE_DN
SP_RATE_UP
ST_REV
STATUS_OPTS
STRATEGY
TAG_DESC
TRK_IN_D
TRK_SCALE
TRK_VAL
UPDATE_EVT</td></tr>
</table>
</td></tr>
</table>

Ratio Block

Figure 1.13 Functional schematic for Ratio function block.

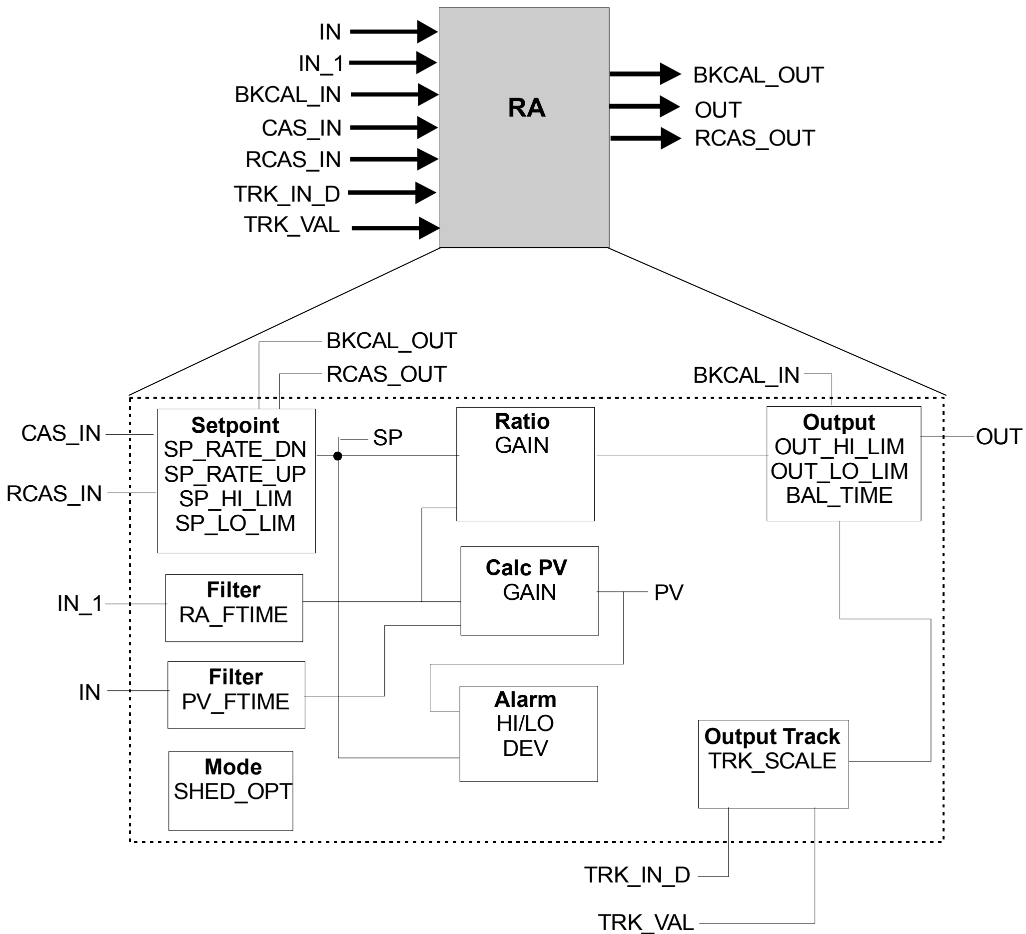


Table 1.M Ratio Block Specifications

Description	The Ratio function block set point is the ratio of its output to its input. A ratio set point of 0.5 produces an output that is one half of its input. The input (IN_1) is either a wild flow or the output of a blend-pacing controller. The output can be used as the set point for a secondary flow controller. An input (IN) from the secondary measurement is used to calculate the actual ratio, which is displayed as the PV. A functional schematic of the block is shown in Figure 1.13 for reference.
Function Notes	• Supports Out of Service (OOS), Initialization Manual (IMan), Local Override (LO), Manual (Man), Automatic (Auto), Cascade (Cas), and Remote Cascade (RCas) modes. • The input 1 (IN_1) value to be ratioed passes through a filter with a time constant of RA_FTIME. The filtered value is multiplied by the Set point (SP) and GAIN to become the target output. The GAIN controls the number of zeros in the SP display. • The input (IN) value is the actual value of the ratioed variable and it passes through a filter with a time constant of PV_FTIME. The filtered IN value is divided by the filtered IN_1 value and the GAIN to become the PV. The units of IN are **not** PV, but OUT. The units of IN_1 are OUT units divided by PV units.

Table 1.M Ratio Block Specifications

Function Notes (cont.)	• The full cascade SP sub-function is used, with rate and absolute limits. Additional control options are available to have the SP value track the PV value, when the block's actual mode is IMan, LO, Man, or ROut. Limits do not cause SP-PV tracking. • Use the Act on IR CONTROL_OPTS to select whether to pass initialization requests or act on them locally by changing the SP value. If this option is OFF or to pass, a status of Not Invited (NI) or Initialization Request (IR) at BKCAL_IN will be passed to BKCAL_OUT. The BKCAL_OUT value will be calculated from the value of BKCALC_IN divided by GAIN and IN_1. When the upstream block sends Initialization Acknowledge (IA) status, the block sends the IA status, since its output will now be nearly identical to the value BKCAL_IN. If this option is ON or to act, a status of NI or IR at BKCAL_IN adjusts the SP to balance the output to the value of BKCAL_IN. The IA status is sent as soon as IR is detected. The BKCAL_OUT will not request initialization. • Use the Balance Ramp CONTROL_OPTS to maintain the ratio SP value, when the block is in Manual (Man) mode. An internal value follows the actual value required to maintain balance. When block mode changes to Automatic (Auto), the internal value ramps to zero contribution in BAL_TIME seconds. If Balance Ramp option is OFF or not used, the ratio SP value immediately changes to follow the changes to the input or output, when the block is in Man mode. • The TRK_VAL input brings in an external value or uses a constant. The TRK_SCALE values convert the TRK_VAL to a percent of output span value. If the CONTROL_OPTS Track Enable selection is true and TRK_IN_D is true, the converted TRK_VAL replaces the output (OUT), when the block is in Automatic (Auto), Cascade (Cas), or Remote Cascade (RCas) mode. The CONTROL_OPTS Track in Manual selection must be true for this to occur in Manual mode. If the actual mode is OOS or IMan, the track request is ignored. • If the TRK_VAL replaces the OUT, its status becomes Locked Out with Limits set to Constant. The actual mode goes to LO. The status of BKCAL_OUT, and RCAS_OUT goes to Not Invited (NI), if not already there. • If the status of TRK_IN_D is Bad, its last usable value will be maintained and acted upon. If the device restarts, losing the last usable value, it will be set to false. • If the status of TRK_VAL is Bad, the last usable value will be used. If there is no last usable value, the present value of the OUT will be used. • Use the Obey SP limits if Cas or RCas CONTROL_OPTS to use SP value after limiting in Cas or RCas mode. • Use the "Use PV for BKCAL_OUT" CONTROL_OPTS to the PV value for the BKCAL_OUT value.			
Equation Options	If Auto mode, OUT = IN_1 (filtered) x SP x GAIN PV = IN (filtered) / IN_1 (filtered) / GAIN If IN_1 has non-cascade status, BKCAL_OUT = BKCAL_IN / GAIN / IN_1 (filtered) If IN_1 has cascade status, BKCAL_OUT = BKCAL_IN / GAIN / SP			
Parameters (see Appendix A for definitions of each parameter)	ACK_OPTION ALARM_HYS ALARM_SUM ALERT_KEY BAL_TIME BKCAL_IN BKCAL_OUT BLOCK_ALM BLOCK_ERR CAS_IN CONTROL_OPTS DV_HI_ALM DV_HI_LIM DV_HI_PRI DV_LO_ALM	DV_LO_LIM DV_LO_PRI GAIN GRANT_DENY HI_ALM HI_HI_ALM HI_HI_LIM HI_HI_PRI HI_LIM HI_PRI IN IN_1 LO_ALM LO_LIM LO_LO_ALM	LO_LO_LIM LO_LO_PRI LO_PRI MODE_BLK OUT OUT_HI_LIM OUT_LO_LIM OUT_SCALE PV PV_FTIME PV_SCALE RA_FTIME RCAS_IN RCAS_OUT SHED_OPT	SP SP_HI_LIM SP_LO_LIM SP_RATE_DN SP_RATE_UP ST_REV STATUS_OPTS STRATEGY TAG_DESC TRK_IN_D TRK_SCALE TRK_VAL UPDATE_EVT

Device Descriptions and Block Parameters

About Device Descriptions

Device Descriptions (DD) are absolutely critical to the interoperability of fieldbus devices. They define the data needed to establish communications among different fieldbus devices from multiple vendors and with control system hosts. The DD provides an extended description of each object in the User Application Virtual Field Device (VFD).

The Fieldbus Foundation provides Device Descriptions for all standard Function Blocks and Transducer Blocks on a CD-ROM. Manufacturer's provide an “Incremental” DD that references the standard DDs and describes manufacturer specific features such as calibration and diagnostic procedures added to their devices.

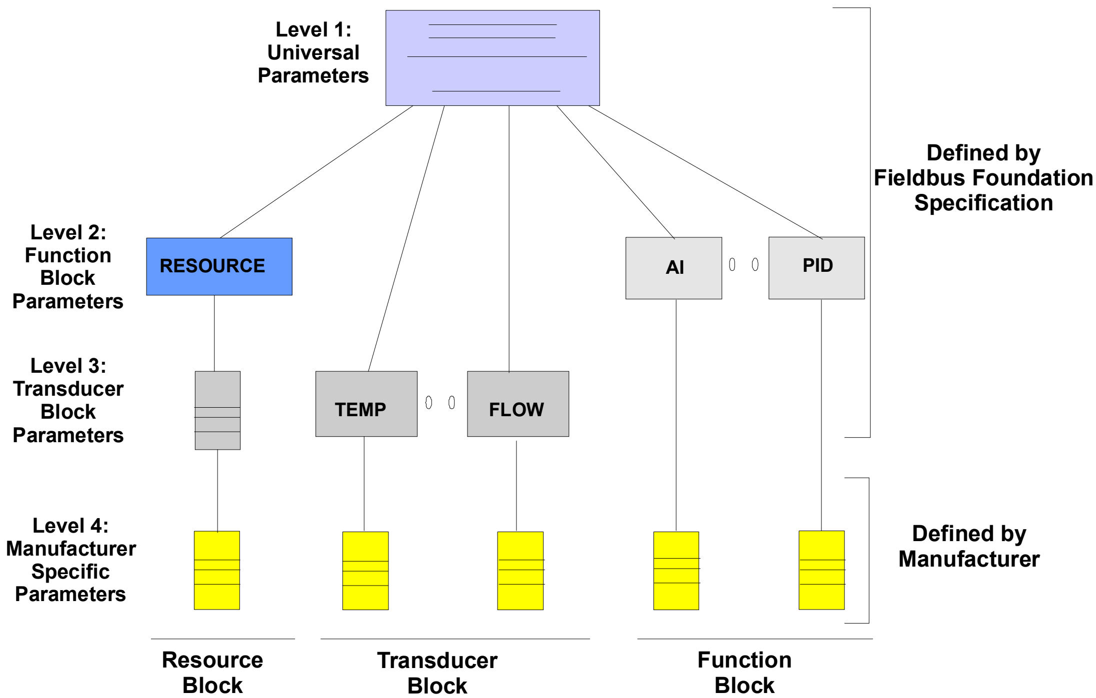
Device Description Language

The Device Description Language (DDL) is a structured text language used to write a DDL source file. A DDL source file describes each device function, parameter, and special feature as well as how a field device can interact with a host application and other field devices. A completed DDL source file is converted into a binary formatted DD output file. The DD output file information can be provided in object form in the device itself, or on a removable storage media delivered with the device. A field device's Object Dictionary (OD) can be transferred from a device to a host using standard Fieldbus Message Specification services.

Device Description infrastructure

The Fieldbus Foundation defines a four-level infrastructure for Device Descriptions for the sake of consistency. See Figure 1.14 for a graphical representation of the DD infrastructure.

Figure 1.14 Device Descriptions infrastructure.



Levels 1, 2, and 3 are the Device Descriptions that the Fieldbus Foundation provides on CD-ROM.

Level 1 consists of Universal Parameters that define common attributes such as Tag, Revision, and Mode. All blocks must include Universal Parameters.

Level 2 consists of Function Block Parameters that define parameters for all standard Function Blocks including the standard Resource Block.

Level 3 consists of Transducer Block Parameters that define parameters for the standard Transducer block. In some cases, the Transducer Block specification may add parameters to the standard Resource Block.

Level 4 is the Manufacturer Specific Parameters that define the parameters a manufacturer has added to the standard Function Block and Transducer Block parameters. These added parameters will be included on the manufacturer's Incremental DD.

Foundation Fieldbus Performance

Foundation Fieldbus is a powerful network providing both communication and distributed control capability. However, fast response is not one of its great capabilities. The screen capture below reflects the time allocated for 18 function blocks to publish their outputs on Fieldbus. The average time is about 40 ms. per published value.

Therefore, in the application reflected in this schedule, you should plan on a loop closure time of on the order of 1 second if you want new data from all devices each time you run the loop calculation.

Part of the reason that fieldbus is slow is that Fieldbus devices operate on very small amounts of current. 10 to 20 ma. per device is typical. This translates into slow computations in the transmitters. It typically takes 100 ms for a fieldbus transmitter to make a new measurement of an input with all the associated calculations completed. Therefore, when attempting to determine the performance of a fieldbus system, please recognize these facts.

Performance Calculation Considerations

Some pressure transmitters will read their transducers and create a new floating point digital readings of the PV every 100 ms. That value can only be read every 40 to 50 milliseconds because of the Fieldbus data rate, and of the Fieldbus protocol.

The data rate is 31.25 Kbps, or 31 bits per millisecond. Very very slow by comparison with ControlNet or Ethernet.

- A minimum Fieldbus message uses 99 bits.
- A minimum response uses 150 bits.

Just to put those messages on the wire takes 8 ms. The protocol says that you must allow time for each device to send nonscheduled messages, in addition to the Publishing of the Precess Variables, that are scheduled. The protocol also says that you must allow significant time for a Fieldbus device to respond to a request for data or information. The result of the slow data rate and the protocol dictate that Fieldbus configuration tools allow 40 to 50 milliseconds for the transmission of data from each Function Block.

Also, many pressure transmitters measures both the pressure and the temperature. If the application dictates that both values must be used, then 80 to 100 milliseconds will be allocated to communicating with those two function blocks, in that one pressure transmitter. Both the Pressure and the Temperature interface with other Fieldbus devices through independent function blocks, so each require their own 40 to 50 milliseconds.

The 1788-CN2FF operates on the Fieldbus side at the max. speed of the Fieldbus, and at the ControlNet rate on the ControlNet side. Therefore, the 1788-CN2FF is not a limiting factor in a Fieldbus systems performance. When a 1788-CN2FF operates, the Fieldbus side and the CN side run asynchronously. When the CN2FF receives data, it is stored in the CN2FF and is Produced on CN at the NUT rate. Therefore, in a typical CN2FF Fieldbus system, the controller will be receiving a lot of redundant data.

In a PLX system, with a FIM fieldbus interface, the Fieldbus side operates at the Fieldbus data rate, and the controller side operates at the backplane rate, so again, it is not a restriction on the performance of a fieldbus system.

Integrating Fieldbus into Rockwell Automation Logix System

Overview

Background - the goals of integration

The following table summarizes the major areas of consideration that were key to defining the goals for bringing Fieldbus into ProcessLogix.

Table 2.A

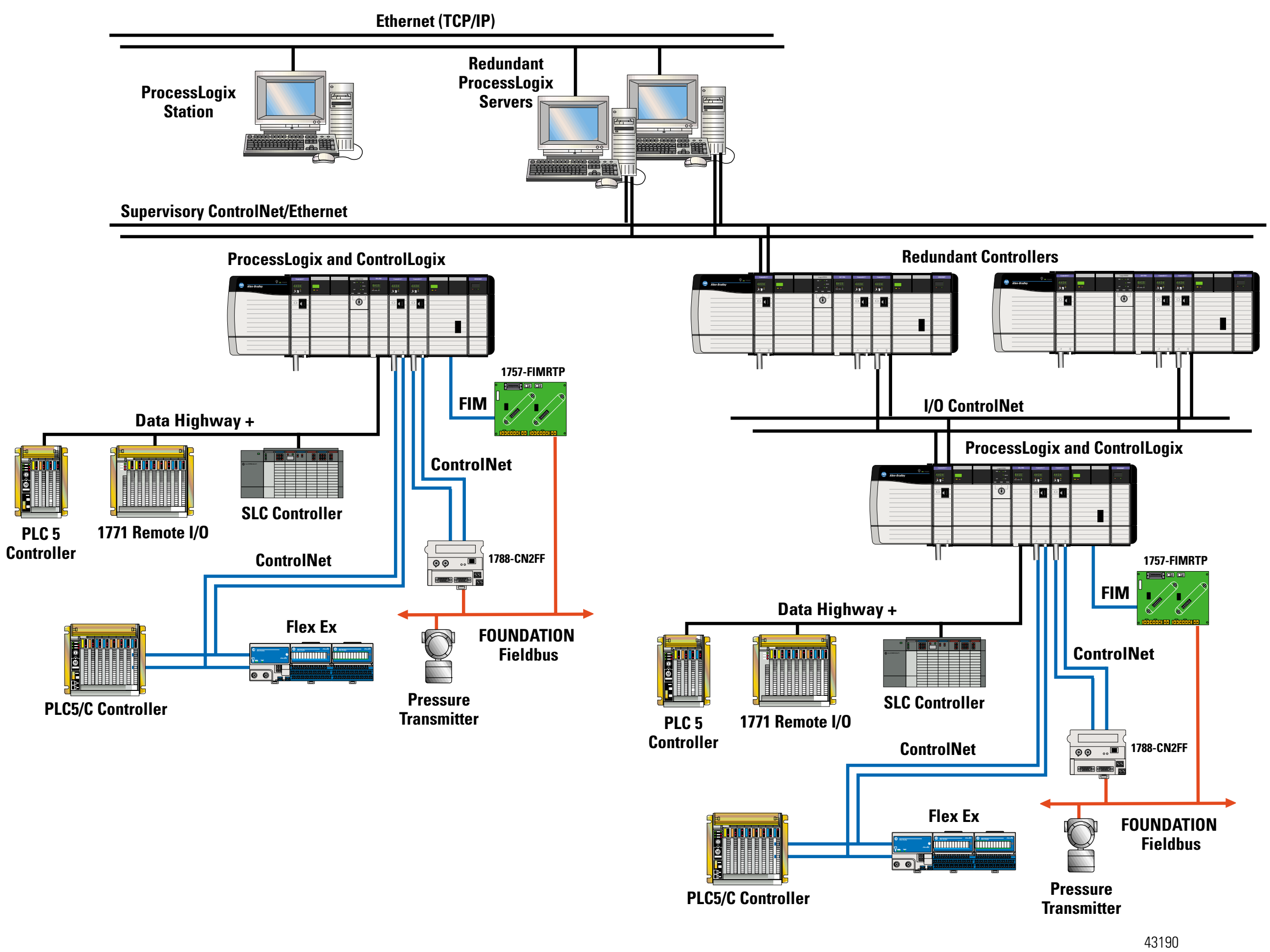
Function	Goal
Connection of Foundation Fieldbus devices to a Logix system.	Integrate fieldbus devices on an H1 link with Supervisory level ControlNet or Ethernet network, and/or the I/O ControlNet network.
Configuration of Foundation Fieldbus devices through Tools system.	Integrate configuration of fieldbus devices through the NetLinx strategy.
Integration of Foundation Fieldbus Devices process, maintenance, and alarm data with notification and display functions in control systems.	Integrate data from fieldbus devices into Detail, Group, Trend, Maintenance, and Alarm displays through the Station application in ProcessLogix as well as the Monitoring tab of the Control Builder application.

Fieldbus Integrated Architecture

As shown in Figure 2.1, Foundation Fieldbus devices can be connected or integrated into a Rockwell Logix system. ProcessLogix, Release 400.0 and later can be configured with a Fieldbus Interface Module (FIM). The FIM serves as the communication gateway between the Supervisory ControlNet/Ethernet and/or I/O ControlNet network and the Foundation Fieldbus H1 communications medium. It works with a Remote Termination Panel (RTP) for connecting and powering up to two fieldbus H1 links.

For applications that require a more highly distributed connection and/or communications directly with a PLC5 or ControlLogix 5500 processor, you can us the 1788-CN2FF ControlNet to Foundation Fieldbus Gateway device.

Figure 2.1 Logix system architecture for Fieldbus integration.



Fieldbus Interface Modules - The Key to an Integrated System

The 1757-FIM, Fieldbus Interface Module is the key to bringing the Foundation Fieldbus system into a ProcessLogix system. The FIM has been designed to operate as a stand-alone Foundation Fieldbus interface or as a bridge between the ProcessLogix control environment and the fieldbus devices. It supports both the publish/subscribe and the client/server communication methods to communicate with fieldbus function blocks. The control connections must be downstream only.

The FIM is a doublewide module that plugs into a non-redundant Controller or remote I/O chassis. It connects up to two Fieldbus H1 links through a companion Remote Termination Panel (RTP). These independent links each have their own link schedule, link master and time master functions. The RTP is designed for DIN rail mounting within an enclosure. It optionally accepts a 24 Vdc input from an external power supply to provide low-level power to fieldbus devices on the H1 links.

The Fieldbus Interface Module functions as a dual network bridge using a dynamic data cache to facilitate the exchange of data between the ControlNet/Ethernet network and the Fieldbus H1 links. It supports both publish/subscribe and client/server communications methods to implement control connections between ProcessLogix function blocks and fieldbus function blocks.

FIM capability includes converting ProcessLogix value-status structure to fieldbus value-status by mapping similar fields to one another and defaulting others. This means ProcessLogix can monitor fieldbus control functions, fully integrate with control functions, or provide a combination that includes using fieldbus based control as backup for selected ProcessLogix control functions.

The FIM uses low and high priority send queues to make sure that publish/subscribe data normally used for control is processed before less important display access data. Publish/subscribe requests are placed in the high priority send queue and client/server requests are placed in the low one.

Configuration Tools

1788-CN2FF Linking Device

The ControlNet-to-Foundation Fieldbus H1 linking device (1788-CN2FF) connects a ControlNet™ network with one or two FOUNDATION Fieldbus H1 (Fieldbus) networks. Each H1 network consists of multiple Fieldbus devices. Each field device has one or more function blocks. Each function block performs an elementary control function such as analog input, analog output, discrete input, or discrete output. The ControlNet network consists of controllers, such as PLC® processors, HMIs, drives, I/O devices, and so on. The linking device has two broad functions, supporting the following:

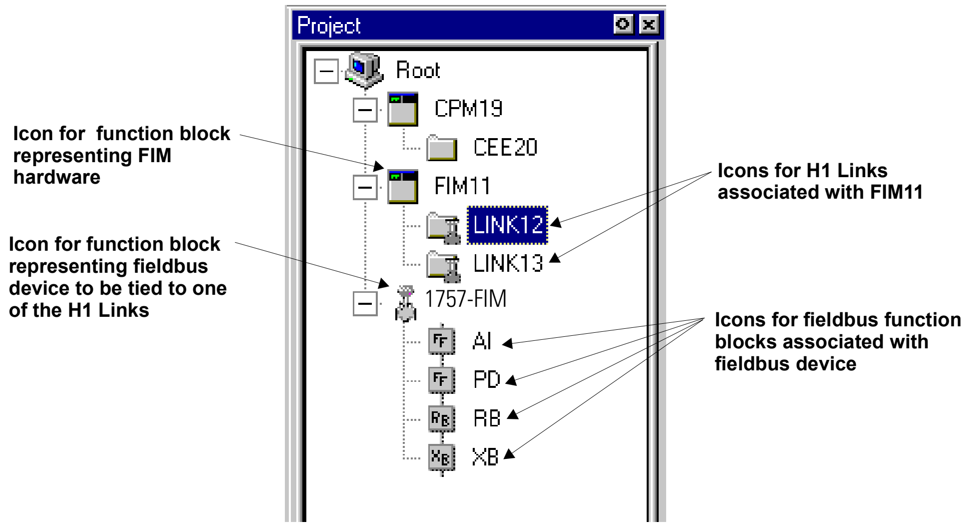
- closed-loop control
- configuration and monitoring

ControlBuilder

The ProcessLogix R400.0 Control Builder application supports integral configuration of fieldbus function blocks with ProcessLogix function blocks to incorporate fieldbus devices in a unified ProcessLogix Control Strategy. This means ProcessLogix function blocks and fieldbus function blocks can be easily interconnected, so control can reside on the fieldbus link, in the Control Processor/Control Execution Environment (CEE), or cascaded from CEE to the fieldbus device.

An integrated Fieldbus Library Manager lets users read the manufacturer's Device Descriptions for fieldbus devices to be tied to an H1 Link and create individual templates for each fieldbus device including their function blocks. The fieldbus device templates will reside in the Engineering Repository Database for ProcessLogix. Once a fieldbus device template is created, the fieldbus device is easily associated with the appropriate FIM H1 Link through the Project tab in Control Builder. The following figure shows how icons are used to readily identify FIM, H1 Links, and fieldbus devices in the Control Builder Project tab.

Figure 2.2 Project tab in Control Builder has new icons for Fieldbus components.



Foundation Fieldbus Configuration Tool

To configure the 1788-CN2FF, you use 1788-FFCT Configuration Software tool. This Fieldbus configuration software uses RSLinx to connect via ControlNet (supports redundant ControlNet) to any CN2FF devices on the Network.

Using the FFCT software you can configure any Foundation Fieldbus device, as well as view, display, and monitor all Foundation Fieldbus parameters. You can also use this tool to setup the data exchange to PLCs, ControlLogix, and ProcessLogix processors.

Centralized Operator Interface

The ProcessLogix R400.0 Station application includes Detail Displays dedicated to the configured FIM, associated H1 Links, fieldbus device, and associated fieldbus function blocks. They provide access to the same parameters that are accessible through the control charts and configuration forms in the Monitoring tab of Control Builder. This includes manufacturer specific parameters, where applicable.

The reporting of alarm conditions and retrieval of process data for inclusion in group, trend, history, and schematic displays is closely integrated with ProcessLogix's existing notification management system. The existing access authorization levels apply and will take precedence over fieldbus restrictions specified in Device Descriptions.

Network Management description

Network Management provides the following capabilities for managing the communication system of a fieldbus device.

- Loading a Virtual Communication Relationship (VCR) list or single entries in this list; (A VCR represents a communication channel through the complete communication stack.)
- Configuring the communication stack;
- Loading the Link Active Schedule (LAS);
- Monitoring performance; and
- Monitoring fault detection.

The collection of managed variables is called the Network Management Information Base (NMIB).

System Management Description

System Management provides the following functions to coordinate the operation of various devices in a distributed fieldbus system.

- Assigning node addresses for devices;
- Synchronizing the application clock;
- Distributing application scheduling across the link; and
- Providing support for locating application tags.

It provides the needed facilities for bringing new devices on the link to an operational state and for controlling the overall system operation. Information, which is used to control system management operation, is organized as objects stored in the System Management Information Base (SMIB).

About the Device Object

The device object represents a physical device entity connected to the fieldbus link. It provides access to the device's Network Management (NM) and System Management (SM) parameters. The client/server VCR is configured in the FIM to access the Management Interface Base (MIB) of the device as soon as it joins the network. The Control Builder does not configure the MIB VCR explicitly. Once the MIB VCR is configured and opened, FIM retrieves MIB information, SMdirectory, and NM directory. Knowledge of these directories allows FIM to transform writes into domain object variables into proper sequence of domain download operations. The SM directory is also used to determine the number of application VFDs. The NM directory is key in attempting to configure VCRs to access Function Block Application Process VFDs in the device.

About the VFD Object

The Virtual Field Device object represents an application VFD and provides parameter access to that VFD. Each physical device may have one or more application VFDs. The FIM attempts to build a client/server VCR to every VFD in the device, when it is added to the network. If the VCR configuration is successful, the FIM obtains VFD and resource identification from the device's VFD. During device download, you can overwrite VCR configuration used to access VFD parameters through the Control Builder application.

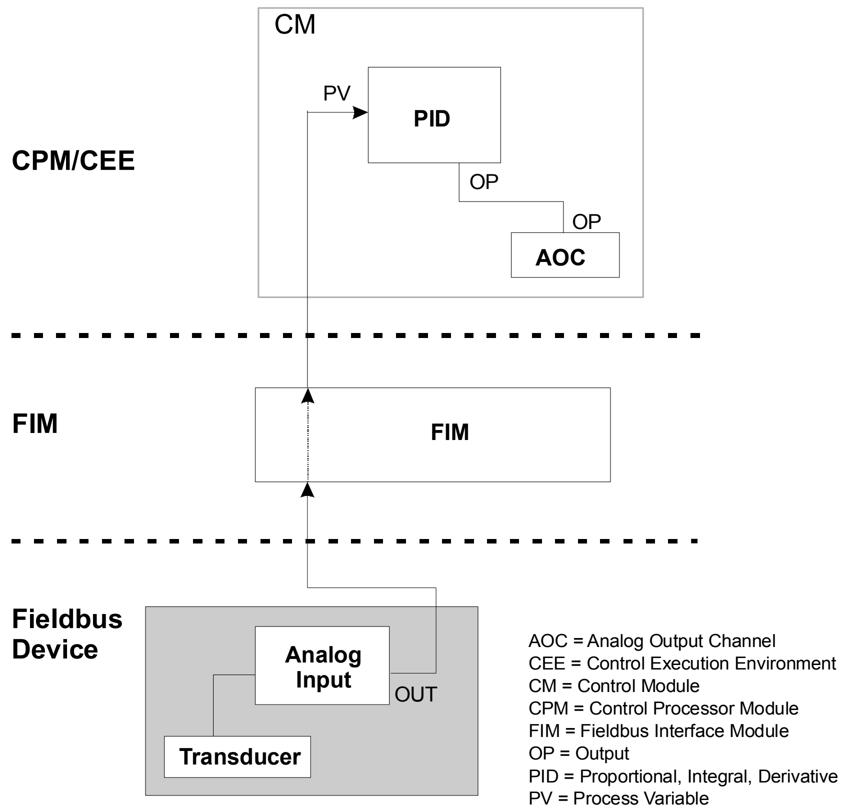
Fieldbus Device Analog Input Integration

A user can functionally wire the output from an Analog Input (AI) function block in a fieldbus device residing on an H1 link to the input of a regulatory control type function block contained in a Control Module in the ProcessLogix Control Builder application. The Proportional, Integral, Derivative (PID) function block is a typical regulatory control type function block.

The Fieldbus Library Manager (FLM) in ProcessLogix R400.0 Control Builder makes this possible. The FLM reads the manufacturer's DD for the fieldbus device and creates a device template that is included in the Project tab of Control Builder. The device template includes the device's fieldbus function blocks, so it can be configured and integrated with control strategies through Control Builder.

Figure 2.3 shows a simplified functional diagram of how the output from an Analog Input function block in a fieldbus compliant transmitter is integrated with a PID function block in a Control Module that is assigned and loaded to the CEE in the Control Processor Module (CPM).

Figure 2.3 Integration of fieldbus device analog input signal with ProcessLogix control strategy



Fieldbus Analog Input data manipulation

When the OUT from the fieldbus analog input function block is wired to the PV input for a PID function block, the Control Builder creates a CEE input agent to handle the analog input from the fieldbus block. The block-like input agent maps the data structure (DS-65) of the OUT parameter to the ProcessLogix PV with status parameter. It interprets the value portion in fieldbus terms and converts it to ProcessLogix representation. The floating-point representation is identical, in most cases, but the fieldbus +/-infinity value must be converted to a ProcessLogix representation.

If the fieldbus status byte indicates “BAD”, the value must be converted to Not a Number (NaN) for ProcessLogix representation.

The fieldbus data quality of good, bad, and uncertain is mapped to the appropriate ProcessLogix parameter of PVSTS, PVSTSFL.NORM, PRSTSFL.BAD, or PVSTSFL.UNCER.

The fieldbus limit indications of no-limit, limited-low, limited-high, and constant are mapped to the same four indications for ProcessLogix. The fieldbus data substatus indicator maps only the limited number of substatus conditions that have corresponding ProcessLogix indications.

The handshaking provided by the substatus associated with Good [cascade] status is **not** supported from an upstream Fieldbus device. This means that control may **not** originate in the field and cascade into the ProcessLogix Controller.

Fieldbus device Analog Output or PID integration

A user can functionally “wire” the output from a regulatory control type function block contained in a Control Module in the ProcessLogix Control Builder application to the input of an Analog Output (AO) or Proportional, Integral, Derivative (PID) function block in a fieldbus device residing on an H1 link. The Proportional, Integral, Derivative (PID) function block is a typical ProcessLogix regulatory control type function block. The Fieldbus Library Manager (FLM) included in the R400 Control Builder makes this possible. The FLM reads the manufacturer's DD for the fieldbus device and creates a device template that is included in the Project tab of Control Builder. The device template includes the device's fieldbus function blocks, so it can be configured and integrated with control strategies through Control Builder.

Figure 2.4 shows a simplified functional diagram of how the output from a PID function block in a Control Module that is assigned and loaded to the CEE in the Control Processor Module (CPM) is integrated with an Analog Output function block in a fieldbus compliant device.

Figure 2.4 Integration of a Fieldbus device analog output signal with ProcessLogix control strategy

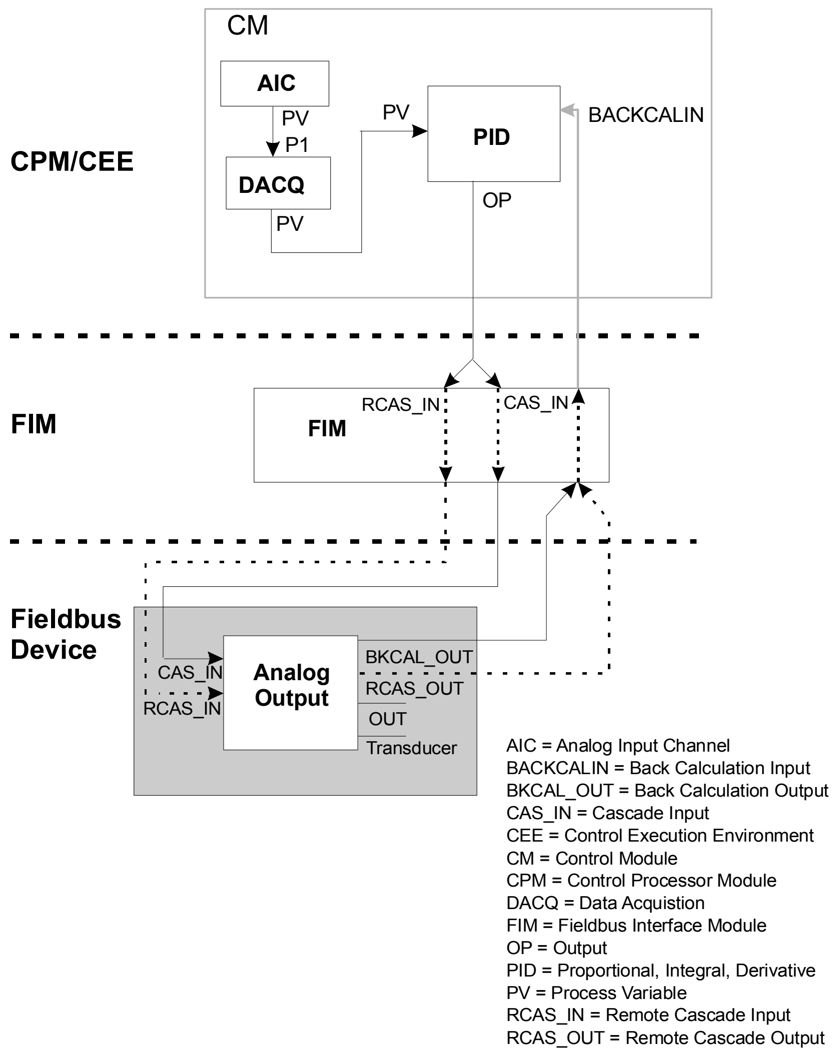
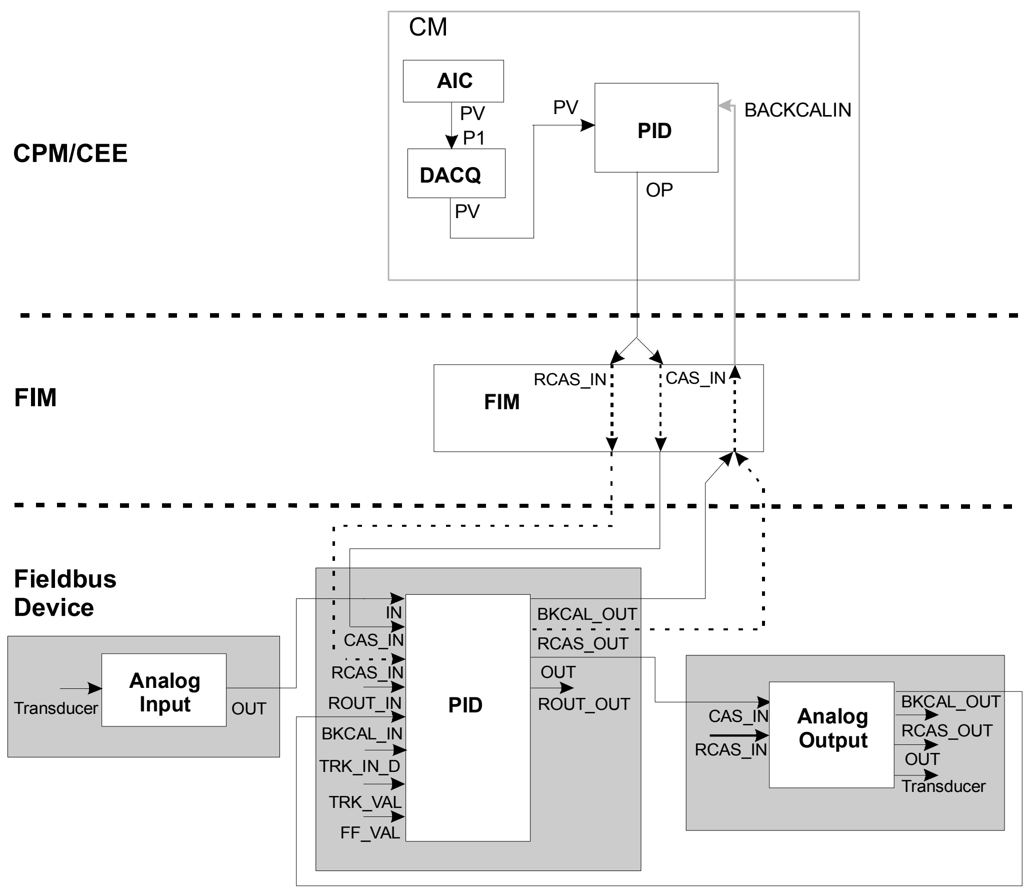


Figure 2.5 shows a simplified functional diagram of how the output from a PID function block in a Control Module that is assigned and loaded to the CEE in the Control Processor Module (CPM) is integrated with a cascaded Proportional, Integral, Derivative function block in a fieldbus compliant device.

Figure 2.5 Integration of fieldbus device PID control with ProcessLogix control strategy



AIC = Analog Input Channel
BACKCALIN = Back Calculation Input
BKCAL_IN = Back Calculation Input
BKCAL_OUT = Back Calculation Output
CAS_IN = Cascade Input
CEE = Control Execution Environment
CM = Control Module
CPM = Control Processor Module
DACQ = Data Acquistion
FIM = Fieldbus Interface Module
OP = Output
PID = Proportional, Integral, Derivative
PV = Process Variable
RCAS_IN = Remote Cascade Input
RCAS_OUT = Remote Cascade Output
ROUT_OUT = Remote Out Output

Fieldbus Analog Output or PID data manipulation

When the OP from the PID function block is wired to the CAS_IN input for a fieldbus Analog Output or Proportional, Integral, Derivative function block, the Control Builder automatically creates a CEE output agent to handle the analog output to the fieldbus block.

The block-like output agent maps the ProcessLogix OP with status parameter to the fieldbus data structure (DS-65) of the CAS_IN parameter. It interprets the value portion in ProcessLogix terms and converts it to fieldbus representation. The floating-point representation is identical, in most cases, but the ProcessLogix +/-infinity value must be converted to a fieldbus representation. If the status of OP is "BAD", its value must be converted from NaN to zero (0.0) for fieldbus representation or it may retain its previous good value, as long as the fieldbus status byte indicates "BAD".

The CEE output agent also accepts a single BKCAL_OUT parameter with the fieldbus data structure (DS-65) and maps it to the BACKCALIN parameter of the PID block in ProcessLogix terms.

ATTENTION

The ProcessLogix Control Builder application automatically makes the appropriate back calculation connections during configuration and the connections are "hidden" in Control Chart views.

Like the FIM, the output agent supports both publish/subscribe and client/server communication methods. The publish/subscribe method allows the FIM to appear as a fieldbus device on the H1 link. The FIM publishes the output (OP) for subscribing fieldbus device resident blocks such as Analog Output and Proportional, Integral, Derivative (PID) through their CAS_IN parameter input connection. This connection is generally used when the downstream control block is in the Cas (cascade) mode. This means that the fieldbus block's BKCAL_OUT parameter is published by the downstream block and subscribed to by the FIM.

The client/server method allows the FIM to appear as a computing device on the H1 link. The FIM writes the output (OP) to be read by fieldbus device resident blocks such as Analog Output and Proportional, Integral, Derivative (PID) through their RCAS_IN parameter input connection. This connection is generally used when the downstream control block is in the RCas (Remote Cascade) mode. This means that the fieldbus block's BKCAL_OUT parameter is written by the downstream block and read by the FIM.

The client/server method also allows the FIM to function in a Direct Digital Control (DDC) mode or the Remote Out mode in fieldbus terms. In this case, the FIM writes the output to be read by the fieldbus PID block though its ROUT_IN parameter input connection. In turn, the PID block publishes the ROUT_OUT or back calculation output value for the subscribing FIM.

The ProcessLogix data quality is converted to fieldbus data quality. The ProcessLogix Good indication is represented as fieldbus Good(Cascade).

The ProcessLogix limit indications of no-limit, limited-low, limited-high, and constant are mapped to the same four indications for fieldbus.

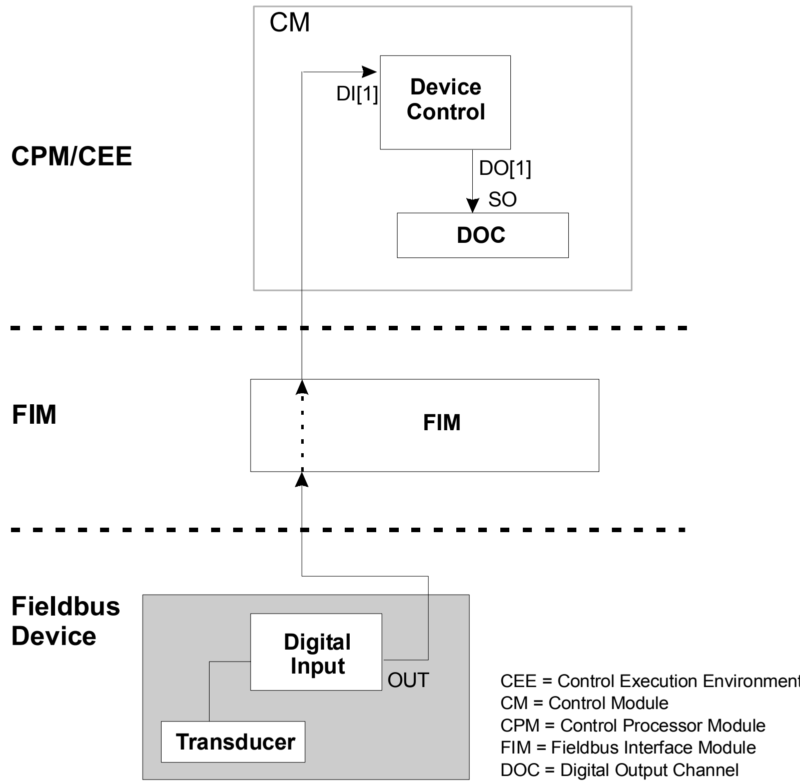
The ProcessLogix control initialization indicators map only to the limited number of substatus conditions that have corresponding indications in fieldbus Good(Cascade).

Fieldbus device Discrete Input integration

A user can functionally “wire” the output from a Discrete Input (DI) function block in a fieldbus device residing on an H1 link to the input of a Device Control (DEVCTL) function block or other block with a digital input contained in a Control Module in the ProcessLogix Control Builder application. The Fieldbus Library Manager (FLM) in ProcessLogix R400.0 Control Builder makes this possible. The FLM reads the manufacturer's DD for the fieldbus device and creates a device template that is included in the Project tab of Control Builder. The device template includes the device's fieldbus function blocks, so it can be configured and integrated with control strategies through Control Builder.

Figure 2.6 shows a simplified functional diagram of how the output from an Discrete Input function block in a fieldbus compliant transmitter is integrated with a Device Control (DEVCTL) function block in a Control Module that is assigned and loaded to the CEE in the Control Processor Module (CPM).

Figure 2.6 Integration of fieldbus device digital input signal with ProcessLogix control strategy



Fieldbus Discrete Input data manipulation

When the OUT from the fieldbus Discrete Input function block is wired to the DI[n] input for a DEVCTL function block, the Control Builder creates a CEE discrete input agent to handle the digital input from the fieldbus block. The block-like discrete input agent maps the data structure (DS-66) of the OUT parameter to the ProcessLogix DI[n] with status parameter. It interprets the value portion in fieldbus terms as a Boolean for Discrete Input (DI) block and as the appropriate multi-state representation for special fieldbus Device Control (DC) block. The value is converted and represented in ProcessLogix at the output.

The discrete input agent accepts inputs from either a published parameter or a client/server read parameter, depending upon the communication method used.

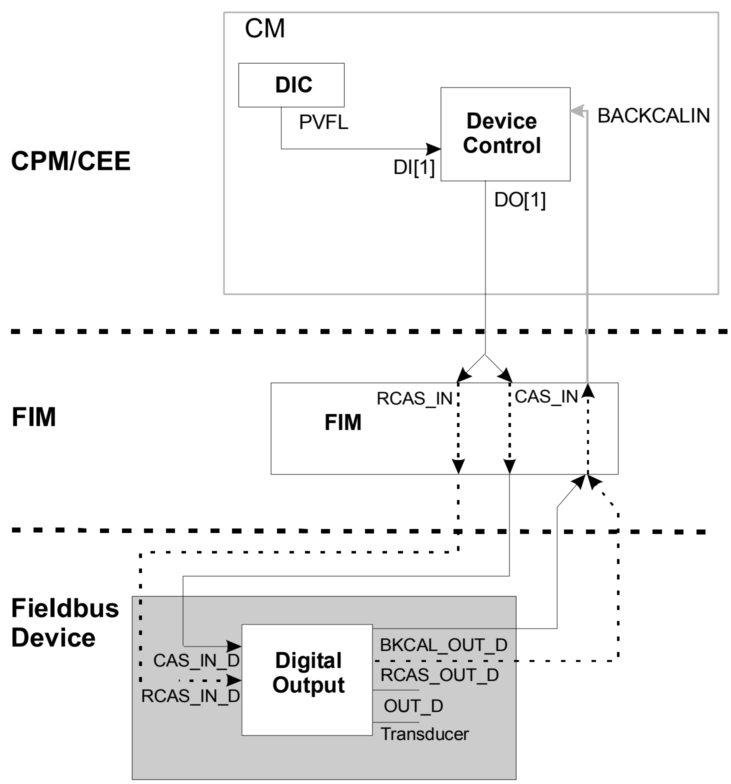
The fieldbus data quality of good(cascade), good(non-cascade), bad, and uncertain is mapped to the appropriate ProcessLogix parameter for good, bad, and uncertain.

Fieldbus device Discrete Output data integration

A user can functionally “wire” the output from a discrete process or control value producing ProcessLogix function block like Device Control to the input of a Discrete Output block in a fieldbus device residing on an H1 link. The Fieldbus Library Manager (FLM) included in the R400 Control Builder makes this possible. The FLM reads the manufacturer's DD for the fieldbus device and creates a device template that is included in the Project tab of Control Builder. The device template includes the device's fieldbus function blocks, so it can be configured and integrated with control strategies through Control Builder.

Figure 2.7 shows a simplified functional diagram of how the output from a Device Control (DEVCTL) function block in a Control Module that is assigned and loaded to the CEE in the Control Processor Module (CPM) is integrated with a Discrete Output function block in a fieldbus compliant device.

Figure 2.7 Integration of fieldbus device digital output signal with ProcessLogix control strategy



BACKCALIN = Back Calculation Input
BKCAL_OUT_D = Back Calculation Output Discrete
CAS_IN_D = Cascade Input Discrete
CEE = Control Execution Environment
CM = Control Module
CPM = Control Processor Module
DIC = Digital Input Channel
FIM = Fieldbus Interface Module
OP = Output
PVFL = Process Variable Flag
RCAS_IN_D = Remote Cascade Input Discrete
RCAS_OUT_D = Remote Cascade Output Discrete

Fieldbus Discrete Output data manipulation

When the DO[n] from the Device Control (DEVCTL) function block is wired to the CAS_IN_D input for a fieldbus Discrete Output function block, the Control Builder automatically creates a CEE output agent to handle the discrete output to the fieldbus block. The block-like output agent maps the ProcessLogix DO[n] with status parameter to the fieldbus data structure (DS-66) of the CAS_IN_D parameter. It interprets the value portion in ProcessLogix terms and converts it to fieldbus representation.

The CEE output agent also accepts a single BKCAL_OUT_D parameter with the fieldbus data structure (DS-66) and maps it to the BACKCALIN parameter of the DEVCTL block in ProcessLogix terms.

It sends the outgoing “control signal” either to a subscribed parameter or a client/server written parameter through the CAS_IN_D or RCAS_OUT_D connection. It can optionally receive the backcalculation signal from either the corresponding published parameter or client/server read parameter.

The ProcessLogix data quality is converted to fieldbus data quality. The ProcessLogix Good indication is represented as fieldbus Good(Cascade).

The ProcessLogix control initialization indicators map only to the limited number of substatus conditions that have corresponding indications in fieldbus Good(Cascade).

Interface Connections Summary

Since the downstream action with the upstream feedback is the same for all fieldbus blocks, there are essentially the following six types of interface connections through the FIM.

- Analog process value into the FIM.
- Discrete process value into the FIM.
- Analog process output from the FIM.
- Discrete process output from the FIM.
- Analog process output from the FIM with backcalculation feedback.

- Discrete process output from the FIM with backcalculation feedback.

Bit types 5 and 6 described in Table 2.B support publish/subscribe communications in Cascade mode or client/server communications in Remote Cascade mode. And, the analog values can also be used in the Remote Out mode.

Fieldbus also supports direct device-to-device (peer-to-peer) publish/subscribe connections independent of the FIM. The FIM can also monitor (subscribe to) the data published between the functions blocks of these fieldbus devices.

Fieldbus status data details

According to Foundation Fieldbus specifications, every fieldbus function block input and output connection must support a status byte that provides the following status indications.

- Data Quality (usability)
- Bad Data Cause
- Degraded Data Cause
- Limit Conditions
- Cascade Control Initialization, Rejection
- Fault-State Initiation, Indication
- Local Override Indication
- Worst Case Alarm Indication
- Upstream Block Class Identification

The status byte structure consists of a 2-bit quality, most significant bit, field; a 4-bit substatus field; and a 2-bit limits, least significant bit, field. The following table provides a breakdown of bit assignments for general reference. The value of the quality field determines the applicable substatus field indication.

Table 2.B Breakdoun of bit assignments

Bit	Quality	Substatus, if Quality field is				Limits
		BAD	UNCERTAIN	GOOD (Non-Cascade)(1)	GOOD (Cascade)(1)	
0	BAD Data Quality	Non-Specific	Non-Specific	Non-Specific	Non-Specific	No Limits
1	UNCERTAIN Data Quality	Configuration Error	Last Usable Value	Active Block Alarm	Initialization Acknow-ledge (IA)	Low Limit
2	GOOD (Non-Cascade) Data Quality	Not Connected	Substitute	Active Advisory Alarm	Initialization Request (IR)	High Limit
3	GOOD (Cascade) Data Quality	Device Failure	Initial Value	Active Critical Alarm	Not Invited (NI)	Constant
4		Sensor Failure	Sensor Conversion Not Accurate	Unacknow-ledged Block Alarm	Not Selected (NS)	
5		No Communication, with Last Usable Value	Engineering Unit Range Violation	Unacknow-ledged Advisory Alarm	Do Not Select (DNS)	
6		No Communication, with no Last Usable Value	Sub-Normal	Unacknow-leged Critical Alarm	Local Override (LO)	
7		Out-Of-Service			Fault-State Active (FSA)	
8					Initiate Fault-State (IFS)	

(1) The Good (non-cascade) substatus is used by output connections for fieldbus blocks such as Analog Input and Discrete Input. The Good (cascade) substatus is used by output connections for fieldbus blocks such as PID. Both of these substatuses are converted to the single ProcessLogix data quality of Good.

Fieldbus Status Indications

See Appendix B for list of possible display indications associated with a given fieldbus status.

Control Mode Interaction

Fieldbus Block Modes Versus Processlogix Modes

Every fieldbus function block including Resource and Transducer blocks contain the MODE_BLK parameter. This structured parameter consists of the Actual, Target, Permitted, and Normal modes. Refer to About Modes of Operation on page 1-8 for descriptions of the eight modes.

For use within ProcessLogix, the structure of the MODE_BLK parameter is expanded to add MODE to the existing Actual, Target, Permitted, and Normal modes as outlined in Table 2.C.

Table 2.C Mode Descriptions

ProcessLogix Mode Structure	Data Type	Description	FIM Action
MODE	Enumeration	Write Only ProcessLogix style mode enumeration MAN, AUTO, CAS, NORMAL, BCAS, NONE	The FIM captures all writes to MODE and maps valid changes to MODE.TARGET.
MODE.TARGET	Enumeration	Read/Write Target mode OOS, MAN, AUTO, CAS, RCAS, ROUT	If the value NORMAL is written to the MODE.TARGET, the FIM replaces it with the value from MODE.NORMAL.
MODE.ACTUAL	Enumeration	Read Only Actual Mode OOS, IMAN, LO, MAN, AUTO, CAS, RCAS, ROUT	
MODE.PERMITTED	Bitstring	Read/Write Permitted mode MAN, AUTO, CAS, RCAS, ROUT OOS is always permitted	If a new MODE.NORMAL value is entered, it is validated against the MODE.PERMITTED values.
MODE.NORMAL	Enumeration	Read/Write Normal mode MAN, AUTO, CAS, RCAS, ROUT OOS is not Normal	If a new MODE.NORMAL value is entered, it is validated against the MODE.PERMITTED values.

Table 2.D shows how ProcessLogix modes are mapped to fieldbus ones.

Table 2.D Mapping ProcessLogix Modes to Fieldbus

ProcessLogix Mode	Fieldbus Mode	Comment
MAN	Man	
AUTO	Auto	
CAS	Cas	
NORMAL	Normal	When setting as target mode, read MODE.NORMAL value and write to MODE.TARGET.
BCAS	Error	Not used in fieldbus blocks. Attempt to set to target is illegal.
NONE	Error	Not used in fieldbus blocks. Attempt to set to target is illegal.

Control Mode Priorities and Indications

Table 2.E shows the 2-character and 4-character mode indications to be used in operating displays and lists the mode priorities based on several interpretations. The Priority Order interpretation is based on the Out-of-Service mode being serviced over all others. The Control Order interpretation is based on the traditional control engineer's concept that Cascade is a higher mode of operation than Automatic, Automatic is a higher mode of operation than Manual, and so on. The Dominance Order interpretation is based on Foundation Fieldbus special rules for modes dominating one another. For example, Out-of-Service dominates over Manual, Manual dominates over Remote Out, and Remote Out dominates over Remote Cascade. This is relevant, if multiple mode bits are set in the target (or normal) mode bitstrings.

A block uses the concept of priority to compute an actual mode that is different than the target mode, and to determine if the particular actual mode allows write access.

Table 2.E 2-character and 4-character mode indications

Mode	Mode Abbreviation		Priority Interpretation		
	2-Character	4-Character	Priority Order (8=highest)	Control Order (8=highest)	Dominance Order (6=Highest)
Out-of-Service	OS	OOS	8	1	6
Initialization Manual	IM	IMan	7	2	—
Local Override	LO	LO	6	3	—
Manual	M	Man	5	4	5
Auto	A	Auto	4	5	1
Cascade	C	Cas	3	6	2
Remote Cascade	RC	RCas	2	7	3
Remote Output	RO	ROut	1	8	4

TIP

The ProcessLogix software installation wizard for Server includes a dialog box for choosing the desired mode acronyms. Select the Fieldbus acronyms radio button to use the mode abbreviations listed above in the Station displays.

While every block type or block instance does not need to support all eight modes, all eight indicator bits are present in the database. The mode bit assignments are listed in Table 2.F.

Table 2.F Mode Bit Assignments

Bit	=	Mode
0 (LSB)	=	Remote Output (ROut)
1	=	Remote Cascade (RCas)
2	=	Cascade (Cas)
3	=	Automatic (Auto)
4	=	Manual (Man)
5	=	Local Override (LO)
6	=	Initialization Manual (IMan)
7 (MSB)	=	Out of Service (OOS)

Rotary Switch Model versus Toggle Switch Model

The Fieldbus Foundation supports both the Rotary Switch and the Toggle Switch models of mode operation. The Rotary Switch model supports only one mode request at a time. For example, an operator can request OOS, Man, Auto, Cas, RCas, or ROut. It has no memory of previous target modes.

The Toggle Switch model supports more than one mode request at a time. For example, an operator can request Manual override of Cascade, Manual override of Remote Cascade, and so on.

ProcessLogix supports the Rotary Switch model as well as the following two instances of the Toggle Switch model.

- An operator may request the Cas mode at the same time the RCas mode is requested.
- An operator may request the Cas mode at the same time the ROut mode is requested.

ProcessLogix also ignores the following illegal mode combinations as defined by the Fieldbus Foundation.

- If ROut is set, RCas may not be set. If it is set, it will be ignored.
- The Auto and Man bits must always be of opposite states. If neither Auto nor Man or both are set, and the ROut, RCas, or Cas mode is set, Auto mode will be assumed with Man cleared. Likewise, If neither Auto nor Man or both are set, and neither ROut, RCas, nor Cas mode is set, Man mode will be assumed with Auto cleared. For the OOS mode, the Man bit should be set unless it is not permitted. If Man is not permitted, the Auto bit should be set unless it is not permitted. If neither Auto nor Man is permitted, the OOS bit should be set.

IMPORTANT An operator needs an access level of ENGR or higher to invoke the OOS mode or to return a block to an in-service mode.

ProcessLogix adheres to the following additional rules for setting fieldbus target mode bits in Table 2.G for its MODE supported subset of combinations.

Table 2.G Additional rules for setting fieldbus target mode bits

Fieldbus Mode	Rule
OOS	When setting as the target mode, obtain the target mode, preserve the Auto and Man bits, set the OOS bit, and optionally reset all the other bits. Reject the request, if the access level is not ENGR or higher.
IMan	This is a Read Only parameter and can not be set as the target mode. Never set the IMan as the target mode.
LO	This is a Read Only parameter and can not be set as the target mode. Never set the LO as the target mode.
Man	When setting as the target mode, set the Man bit and reset all the other bits. Reject the request, if the current mode is OOS and the access level is not ENGR or higher.
Auto	When setting as the target mode, set the Auto bit and reset all the other bits. Reject the request, if the current mode is OOS and the access level is not ENGR or higher.
Cas	When setting as the target mode, set both Cas and Auto bits and reset all the other bits. Reject the request, if the current target mode is OOS and the access level is not ENGR or higher.
RCas	When setting as the target mode, set both RCas and Auto bits and reset all the other bits. Reject the request, if the current target mode is OOS and the access level is not ENGR or higher.
ROut	When setting as the target mode, set both ROut and Auto bits and reset all the other bits. Reject the request, if the current target mode is OOS and the access level is not ENGR or higher.
Normal	When setting as the target mode, read the MODE.NORMAL value and write to the MODE.TARGET. Reject the request, if the current target mode is OOS and the access level is not ENGR or higher.

Display indications and mode calculation

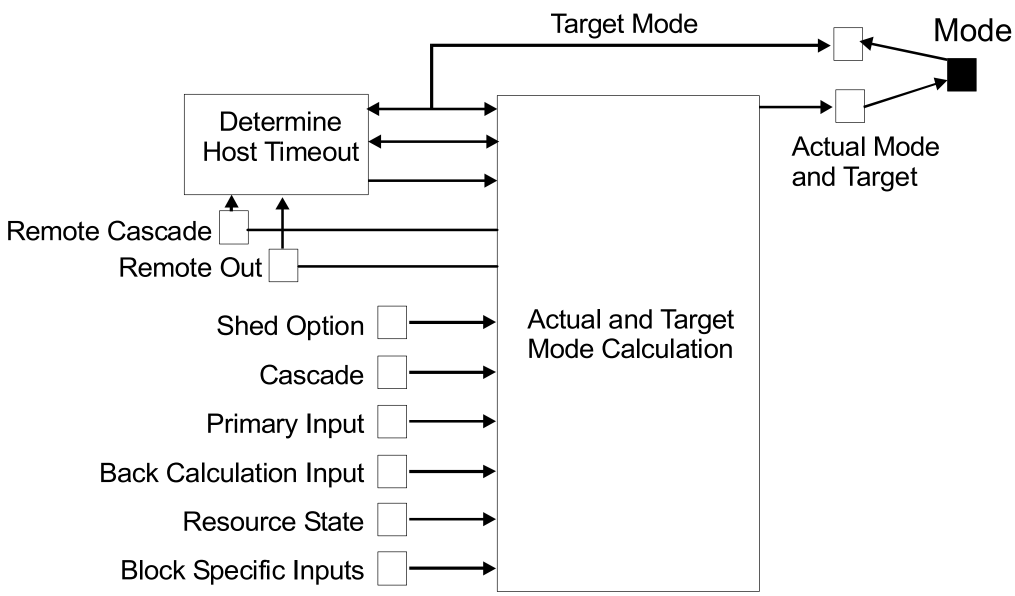
The fieldbus mode indications for actual mode and composite actual/target modes will appear in the following formats on Station displays as shown in Table 2.H.

Table 2.H Fieldbus mode indications

Format	Description	Examples
a	Satisfied in mode a; actual same as target.	OOS, MAN, AUTO, CAS, RCAS, ROUT
a (t)	In mode a; not satisfied in higher target mode t.	MAN (A), CAS (RC), IM (A), LO (CAS), AUTO (M), CAS (M)

The block mode calculation of actual mode considers the input parameter status attributes, input values, and resource state as represented graphically in Figure 2.8.

Figure 2.8 Block mode calculation summary



See Appendix C for list of conditions, which will change the mode in order of priority with Good (Non-Cascade) status on input parameter as the lowest priority.

Link and Block Schedules

Link Active Scheduler (LAS)

All links must have a Link Active Scheduler (LAS). The LAS operates at the data link layer as the bus arbiter for the link. It dynamically provides the following functions.

- Recognizes and adds new devices to the link.
- Removes non-responsive devices from the link.
- Distributes Data Link and Link Scheduling time on the link. The data link layer synchronizes the network-wide Data Link Time. Link scheduling time is a link specific time represented as an offset from Data Link Time. It is used to indicate when the LAS on each link begins and repeats its schedule. System Management uses it to synchronize function block execution with the data transfers scheduled by the LAS.
- Polls devices for buffered data at scheduled transmission times.
- Distributes a priority-driven token to devices between scheduled transmissions.

Any device on the link may become the LAS as long as it is capable. The devices that are capable of becoming the LAS are called Link Master devices. All other devices are referred to as Basic devices.

The FIM is Link Master capable and supports both a primary and a backup link schedules. It is designated as the primary Link Master.

Upon startup or failure of the existing LAS, the Link Master devices on the link bid to become the LAS. The Link Master that wins the bid begins operating as the LAS immediately upon completion of the bidding process. Link Masters that do not become the LAS act as basic devices when viewed by the LAS. They also act as LAS backups by monitoring the link for failure of the LAS, and by bidding to become the LAS when a LAS failure is detected.

ATTENTION

If a LAS is too large to fit in the active Link Master capable device, the user must reconfigure the device to become a Basic one through Control Builder, and restart the device to initiate the change.

Link Schedule

The Link Schedule is the overall schedule for the link. It includes both the link data transfer and the device function block execution schedules. An independent Link Schedule is provided for the FIM interface port for each link. A backup Link Schedule is provided for all Link Master capable devices on the link.

The link data transfer schedule is derived from the portion of the link schedule that deals with publication of parameters. The Control Builder (CB) provides a default link schedule of publications and function block execution phasing based on the function block connections in the user configured control strategy. The basis for the link schedule is this link's content from all currently loaded Control Modules (CM). Execution phasing is based solely on function block existence in the CM. Order of execution is based on the order in CM (ORDERINCM) parameter for each block. Publications are based on inter-device function block connections and device to ProcessLogix or ProcessLogix to device function block connections. The following publication rules apply.

- Function block publications appear in the link data transfer schedule in the order specified by their ORDERINCM parameters. (Duplicate values of ORDERINCM may produce indeterminate ordering of those blocks involved.)
- If the user changes the sequence of execution order for function blocks in a schedule, the ORDERINCM parameters of the involved function blocks are appropriately adjusted.
- Publication of each output is scheduled immediately after execution of the function block that produces the value, considering inter-publication delays and potential conflicts.
- Blocks publish, if their output is connected to an input in another device or the FIM.
- No unneeded time delay is allowed in the default link data transfer schedule.
- The macrocycle is the least common multiple of the execution periods of all the CMs involved in the link data transfer schedule.

Function block execution schedule

The function block execution schedule is derived from the portion of the link schedule that deals with starting the execution of each function block or FB_START indications. The link schedule provides only those entries that pertain to the blocks residing in a given fieldbus device. While device function blocks may be synchronized to the link schedule, it is not a Foundation Fieldbus mandated feature. They may run asynchronously.

The block execution time can be broken into these three phases.

1. Preprocessing — Snap-shot of parameter values
2. Execution — Function block outputs are determined
3. Postprocessing — Block output values, alarm and associated trend parameters are updated.

Since input parameter values used by a function block must not change during execution, a copy of the input parameter values is captured or snapped at the beginning of execution. Also, since block outputs to other blocks must be time coincident, the output values are only updated at the completion of the function block execution. The block algorithm execution phase is always executed in the following ordered sequence as shown in Figure 2.9.

1. Determine the actual mode attribute of the mode parameter. This calculation is based on the target mode and the status attributes of input parameters.

2. Calculate the set point, if the Set Point parameter is defined for the function block.

   The calculation of working set point is based on the actual mode, set point input parameters such as cascade and remote cascade, and any backward path input status. Also, the value of the controlled parameter, process variable, may be used for set point tracking. The resulting set point is shown in the set point parameter.

3. Execute the control or calculation algorithm to determine the value and status of output parameters in the forward path.

   The conditions that determine the status attribute of output parameters. The value attributes of the block's input parameters and contained parameters, the actual mode and the working set point are used in this algorithm. Also, where defined by the block profile, some blocks may use the status of selected inputs.

In general, the calculation of actual mode and the use of actual mode in the algorithm accounts for the status of critical inputs.

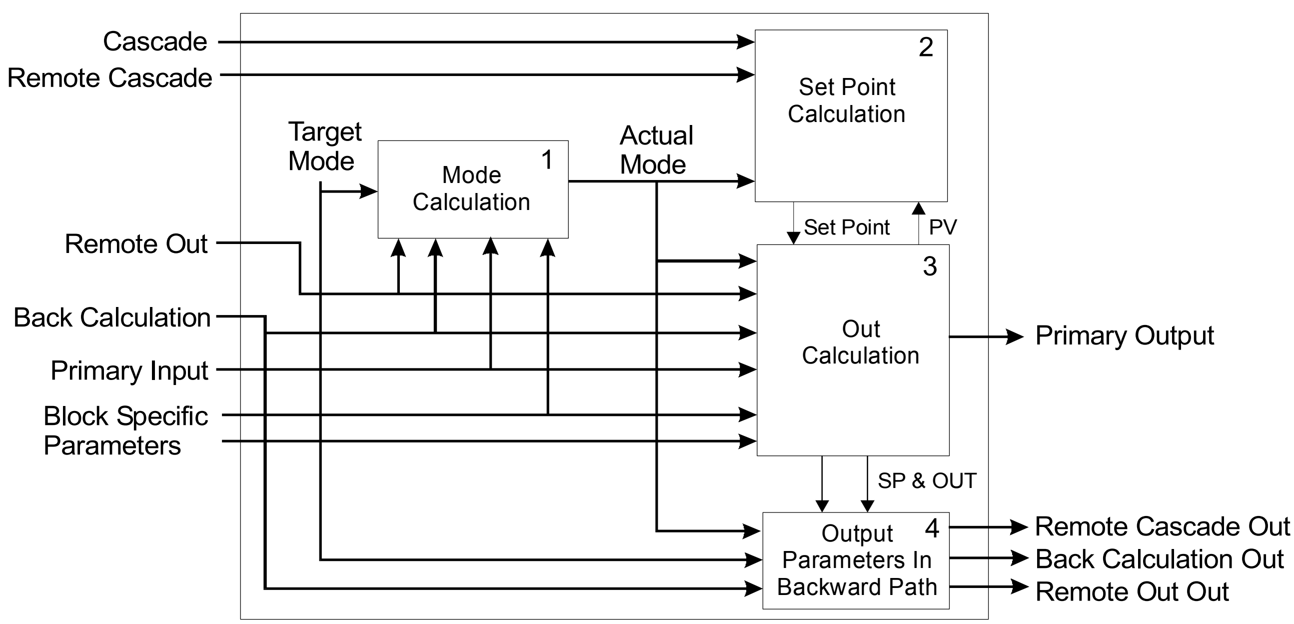
4. Calculate output parameters in the backward path.

   This phase applies only to output blocks and calculation blocks designed for use in a cascade path.

TIP

A fieldbus device whose period of function block execution is an integer factor of the macrocycle of the link will have a function block execution schedule prepared that has the optimal shorter cycle. For example, if the control strategy includes a CM with a 10 second period for a temperature loop, a second CM with a 1 second period for a pressure loop, and a third CM with a 250 millisecond period for a flow loop, a 1 second macrocycle can be downloaded to the device that contains functions blocks used in the 1 second and 250 ms CMs.

Figure 2.9 Algorithm execution phase sequence



Tags, Addresses, and Live List

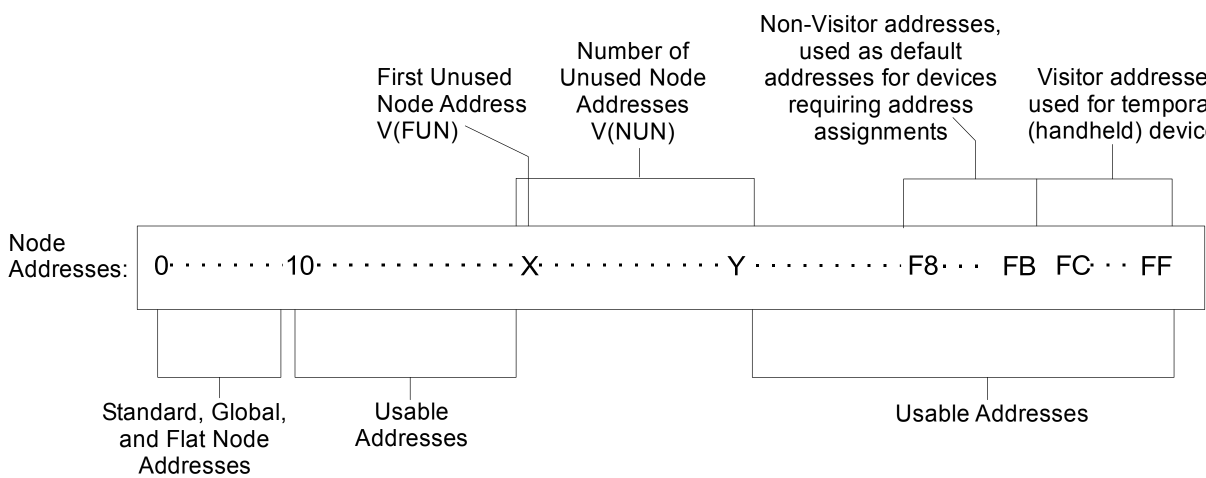
Tag and address assignments

Before a fieldbus device can actively join a network it must be assigned a name and data link address. Device names are system specific identifiers called physical device tags (PD_TAG).

The PD_TAGs may be assigned by the vendor or through the System Management Kernel (SMK), normally in an off-line configuration environment so devices without tags are kept off the operational network.

The SMK for devices without tags are set to the Uncommissioned state and connected to the bus at one of four default device addresses. The Data Link Layer specifies these default addresses as non-visitor node addresses. The following figure shows the general allocation of data link layer addresses to field devices.

Figure 2.10 Summary of address allocations for fieldbus devices



ATTENTION

Temporary devices such as handheld interfaces are not assigned tags or addresses. They join the network through one of four data link visitor addresses reserved for them in the data link layer protocol.

Live List and Uncommissioned Devices

FOUNDATION Fieldbus defines a live list as a 32-byte bitstring (256 bits) where each bit represents an address of the fieldbus network. A set bit at a particular bit number means that a device is present at that address. The LAS of the network owns the live list and maintains it as part of its operation.

The FIM constantly monitors the live list for each fieldbus link or device connected to it. When the LAS for the link recognizes a new device at a default address, it adds it to its live list according to the data link layer procedures. The FIM detects the change in the live list and makes a connection to the new uncommissioned device. It gathers the following information from the device to be passed to Control Builder.

Table 2.I Gathered information from device passed to Control Builder

Name	Description	Data Type	Access
PdTag	Physical Device Tag	32-byte string	Read/Write
Address	Device Address	Unsigned8	Read/Write
DevID	Globally unique Device Identifier	32-byte string	Read Only
Vendor	Vendor name string	32-byte string	Read Only
ModelName	Model Name string	32-byte string	Read Only
Rev	Application Revision	32-byte string	Read Only
ManufID	Manufacturer Identifier	Unsigned32	Read Only
DevType	Device Type code	Unsigned16	Read Only
DevRev	Device Revision	Unsigned8	Read Only
DdRev	DD Revision	Unsigned8	Read Only

Control Builder uses the device information to create an item in its Monitoring tree to represent the new uncommissioned device on the given link. Users can now view and configure pertinent information for the uncommissioned device through appropriate Link block and device block configuration forms in Control Builder.

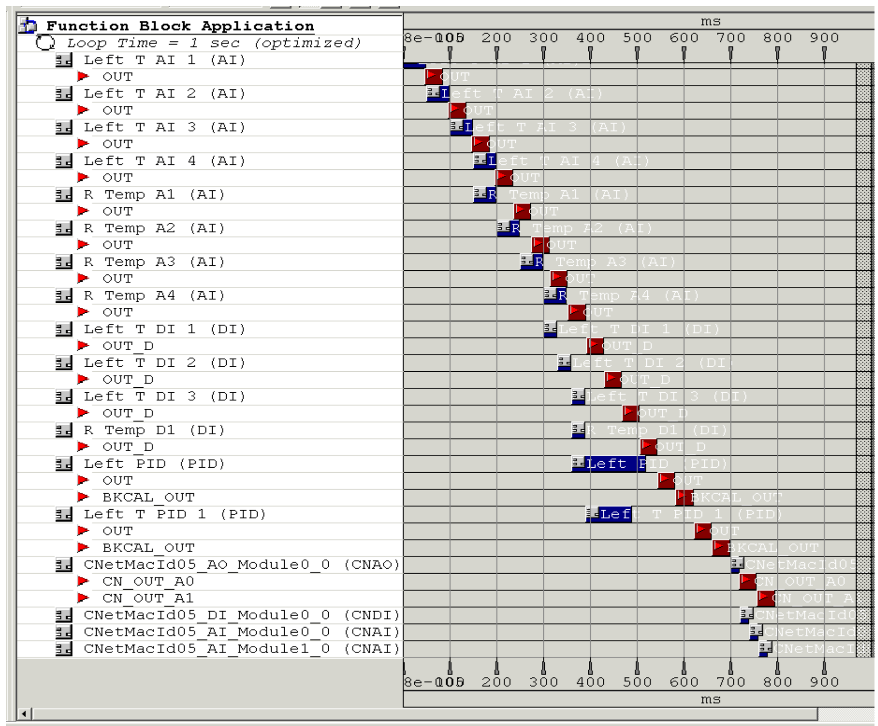
TIP

The FIM must be configured and loaded through Control Builder before you can view the module. You can view the module's links and devices through the Monitoring tab of Control Builder.

Foundation Fieldbus Performance

Foundation Fieldbus, FF, is a very powerful network providing both communication and distributed control capability. However, fast response is not one of its great capabilities. The screen capture below reflects the time allocated for 18 function blocks to publish their outputs on Fieldbus. The average time is about 40 ms. per published value.

Therefore, in the application reflected in this schedule, you should plan on a loop closure time of on the order of 1 second if you want new data from all devices each time you run the loop calculation.



Part of the reason that Fieldbus is slow is that Fieldbus devices operate on very small amounts of current.

- 10 to 20 ma. per device is typical.
- Translates into slow computations in the transmitters.
- Typically takes 100 ms for a fieldbus transmitter to make a new measurement of an input with all the associated calculations completed.
- Therefore, when attempting to determine the performance of a fieldbus system, please recognize these facts.

Performance calculation considerations:

- Some pressure transmitters will read their transducers and create a new floating point digital readings of the PV every 100 ms.
- That value can only be read every 40 to 50 milliseconds because of the Fieldbus data rate, and of the Fieldbus protocol.
- The data rate is 31.25 Kbps, or 31 bits per millisecond. Very very slow by comparison with ControlNet or Ethernet.
- A minimum Fieldbus message uses 99 bits. A minimum response uses 150 bits. Just to put those messages on the wire takes 8 ms.

The protocol says that you must allow time for each device to send nonscheduled messages, in addition to the Publishing of the Precess Variables, that are scheduled. The protocol also says that you must allow significant time for a Fieldbus device to respond to a request for data or information. The result of the slow data rate and the protocol dictate that Fieldbus configuration tools allow 40 to 50 milliseconds for the transmission of data from each Function Block. Also, many pressure transmitters measures both the pressure and the temperature.

If the application dictates that both values must be used, then 80 to 100 milliseconds will be allocated to communicating with those two function blocks, in that one pressure transmitter. Both the Pressure and the Temperature interface with other Fieldbus devices through independent function blocks, so each require their own 40 to 50 milliseconds.

The CN2FF operates on the Fieldbus side at the max. speed of the Fieldbus, and at the ControlNet rate on the ControlNet side. Therefore, the CN2FF is not a limiting factor in a Fieldbus systems performance. When a CN2FF operates, the Fieldbus side and the CN side run asynchronously. When the CN2FF receives data, it is stored in the CN2FF and is Produced on CN at the NUT rate. Therefore, in a typical CN2FF Fieldbus system, the controller will be receiving a lot of redundant data.

In a PLX system, with a FIM fieldbus interface, the Fieldbus side operates at the Fieldbus data rate, and the controller side operates at the backplane rate, so again, it is not a restriction on the performance of a fieldbus system.

Notification Scheme

Fieldbus versus ProcessLogix Alarm Priorities

The Fieldbus alarms are closely integrated with the existing ProcessLogix notification system. The ProcessLogix Server handles FIM alarms in the same way it handles Control Processor ones. But, the fieldbus devices themselves own their alarm data and generate the alarms, clears, and events.

Fieldbus devices use 0 to 15 as numeric priorities for alarm reporting. ProcessLogix alarms use Journal, Low, High and Urgent as priorities with a sub-priority of 0 to 255. Table 2.J shows how fieldbus priorities are mapped to ProcessLogix priorities and severities.

Table 2.J Mapping Fieldbus Priorities to ProcessLogix

Fieldbus Alarm Priority	ProcessLogix Alarm Priority	ProcessLogix Alarm Severity
0	(Can never be seen by FIM or above)	(Can never be seen by FIM or above)
1	(Can never be seen by FIM or above)	(Can never be seen by FIM or above)
2 BLOCK_ERR bit 14 (power-up) BLOCK_ERR bit 15 (Out-of-Service)	Journal (Event System Only)	2
3 All other bitstring indications: (BLOCK_ERR bits 0-13, XD_ERROR bits 16-25)	System Level Diagnostic (High)	2
2 (User selected)	Journal	2
3	Low	3
4	Low	4
5	Low	5
6	Low	6
7	Low	7
8	High	8
9	High	9
10	High	10
11	High	11
12	Urgent	12
13	Urgent	13
14	Urgent	14
15	Urgent	15

Fieldbus Alarm Conditions

Fieldbus devices provide both process and device related alarms. The fieldbus devices themselves own their alarm data; generates and clears the alarms and events. The process alarms are associated with process variable conditions and they are reported as process alarms into ProcessLogix.

The device alarms are associated with actual device conditions or processes within the block as indicated by BLOCK_ERR and XD_ERROR bitstring alarms. These alarms are reported as device or system alarms into the ProcessLogix notification system. Table 2.K summarizes the possible fieldbus alarm enumerations and lists the alarm/event type identification to be used in the alarm summary and event summary displays in Station.

TIP

Fieldbus alarm functions do not support rate of change (ROC) alarms. ROC alarms can only be generated in applications that use ProcessLogix data acquisition blocks for input signal conditioning.

Table 2.K Fieldbus alarm enumerations and alarm/event type identification

Enumeration	Description	Alarm/Event Type
UNDEF	Undefended Alarm	No Action
LO	Low Limit Alarm	PVLO
HI	High Limit Alarm	PVHI
LO LO	Critical Low Limit Alarm	PVLOLO
HI HI	Critical High Limit Alarm	PVHIHI
DV LO	Deviation Low Alarm	DEVLO
DV HI	Deviation High Alarm	DEVHI
DISC	Standard Discrete Alarm	OFFNORM
DISC	Standard Discrete Alarm	CHNGOFST
DISC	DevCtl Fail Alarm	FBDCFAIL
DISC	DevCtl Accept Alarm	FBDCACC
DISC	DevCtl Ignore Alarm	FBDCIGN
BLOCK	BLOCK_ERR: 0 (Other (LSB)	FFOTHER
BLOCK	BLOCK_ERR: 1 (Block Configuration Error)	FFBLKCFG
BLOCK	BLOCK_ERR: 2 (Link Configuration Error)	FFLNKCFG
BLOCK	BLOCK_ERR: 3 (Simulate Active)	FFSIMACT
BLOCK	BLOCK_ERR: 4 (Local Override)	FFLO

Table 2.K Fieldbus alarm enumerations and alarm/event type identification

Enumeration	Description	Alarm/Event Type
BLOCK	BLOCK_ERR: 5 (Dev Fault State Set)	FFFLSAFE
BLOCK	BLOCK_ERR: 6 (Dev Needs Maintenance Soon)	FFDEVNMS
BLOCK	BLOCK_ERR: 7 (I/P Failure or PV BAD Status)	FFINFL
BLOCK	BLOCK_ERR: 8 (O/P Failure)	FFOUTFL
BLOCK	BLOCK_ERR: 9 (Memory Failure)	FFMEMFL
BLOCK	BLOCK_ERR: 10 (Lost Static Data)	FFLSTDTA
BLOCK	BLOCK_ERR: 11 (Lost NV Data)	FFLNVDTA
BLOCK	BLOCK_ERR: 12 (Readback Check Failed)	FFRBCKFL
BLOCK	BLOCK_ERR: 13 (Dev Needs Maintenance Soon)	FFDEVNMS
BLOCK	BLOCK_ERR: 14 (Power Up)	FFPWRUP
BLOCK	BLOCK_ERR: 15 (Out-Of-Service)	FFOOS
BLOCK	XD_ERROR: 16 (Unspecified Error)	TBUNSPEC
BLOCK	XD_ERROR: 17 (General Error)	TBGENRAL
BLOCK	XD_ERROR: 18 (Calibration Error)	TBCALERR
BLOCK	XD_ERROR: 19 (Configuration Error)	TBCFGERR
BLOCK	XD_ERROR: 20 (Electronics Failure)	TBELECFL
BLOCK	XD_ERROR: 21 (Mechanical Failure)	TBMECHFL
BLOCK	XD_ERROR: 22 (I/O Failure)	TBIOFL
BLOCK	XD_ERROR: 23 (Data Integrity Error)	TBDTAERR
BLOCK	XD_ERROR: 24 (Software Error)	TBSWERR
BLOCK	XD_ERROR: 25 (Algorithm Error)	TBALGERR
UPDATE	TB Static Data Update Event	TBSTCHNG
UPDATE	FB Static Data Update Event	FBSTCHNG
WRITE	Write Protect Change Alarm	RBWPCHNG
UPDATE	Link Object Update Event	FBLOCHNG
UPDATE	Trend Object Update Event	No Action

ATTENTION

When using the 1788-CN2FF, no alarms come from Fieldbus devices when used with ProcessLogix. Data with status will be produced.

Alert Object Formal Model

The alert object allows block alarms and events to be reported to a device responsible for alarm management.

Class: Alert
Subclass of: Root
Attributes:

1. (m) (r) DD Member Id
2. 2. (m) (Key) Index
3. 3. (m) (r) Data Type
  3.1 (m) (r) Meta Type = RECORD
  3.2 (m) (r) Type Name = Alert
4. (m) (r) Sub-index
  4.1 (m) (r) Block Index - Unsigned16
  4.2 (m) (r) Alert Key - Unsigned8
  4.3 (m) (r) Standard Type - Unsigned8
  4.4 (m) (r) Mfr Type - Unsigned8
  4.5 (m) (r) Message Type - Unsigned8
  4.6 (m) (r) Priority - Unsigned8
  4.7 (m) (r) Time Stamp - Time Value
5. (m) (r) Data Length
6. (m) (r) Units = “ “
7. (m) (r) Usage = CONTAINED
8. (m) (r) Storage = DYNAMIC
9. (m) (r) List of Valid Values

Standard type enumerations 0-12 are defined. Message type enumeration’s 0-3 are defined - see attribute definitions.

10. (m) (r) Initial Value
11. (m) (r) DD Item Id

Services:

1. (m) FB_Alert_Notify
2. (m) FB_Alert_Ack

Table 2.L Attribute Definitions

Attribute	Definition
DD Member Id	A unique number which identifies the alert. This number will be assigned as part of the development of the Device Description (DD). A DD member Id is assigned if an object is defined as part of a structure. A value of zero (0000) will be used for the DD member Id if the object is not part of a structure.
Index	The location of the alert in the OD.
Meta Type	Identifies the alert as a record (multiple values of different types).
Type Name	Identifies the data format as a data structure associated with the Meta type.
Sub-index	Attributes of an object which may be individually accessed through the FB_Read and FB_Write service by using the sub-index number with the object index number. Sub-index numbers are assigned based on Meta type.
Data Length	The number of bytes required to represent the data type.
Units	The engineering units in which the value is represented.
Usage	Indication of whether the alert may be linked to a block parameter.
Storage	Specification that alert must be stored in dynamic (D) memory.

List of Valid Values

Standard type will have the following enumerated values.

Table 2.M Standard type valid values

Valid Values	Meaning
0	Undefined
1	LO - Low limit
2	HI - High limit
3	LO LO - Critical low limit
4	HI HI - Critical high limit
5	DV LO - Deviation low
6	DV HI - Deviation high
7	DISC - Discrete
8	BLOCK - Block Alarm
9	UPDATE - Static data update
10	WRITE - Write protect changed

Table 2.M Standard type valid values

Valid Values	Meaning
11	UPDATE - Link associated with function block
12	UPDATE - Trend associated with block
Message type will be enumerated in the following manner: 0 = 1 = Event Notification 2 = Alarm Clear 3 = Alarm Occur	

The alert object contains information from an alarm or update event object, which is to be sent in the notification message. The alert object will be invoked by the alert notification task. If multiple alarms or event parameters are unreported, then the one with the highest priority or is the oldest of equal priority will be selected by the alert notification task.

The selected alert object is sent in a message at the first opportunity - less than the alert confirm time. If a confirmation from an interface device is not received by the alarm notification routine in the field device within a time determined by the resource block confirm time parameter, then the alert will be considered unreported so it may be considered for selection.

1757-FIM Planning Considerations

Reference Publications

Please refer to the following Rockwell Automation publications for general planning details and installation considerations for the ProcessLogix system in general.

Table 3.A Publication References

Publication Name	Publication Number
ProcessLogix R400.0 Installation and Upgrade Guide	1757-IN040B-EN-P
ProcessLogix R400.0 Selection Guide	1757-SG001B-EN-P
1757-FIM Installation Instructions	1757-IN913A-EN-P
1757-RPT Installation Instructions	1757-IN915A-EN-P
1788-CN2FF Installation Instructions	1757-IN051B-EN-P
NI-FBUS Configurator User Manual	1788-6.5.2
1757-PLX52 ProcessLogix Controller Module Installation Instructions	1757-IN901C-EN-P
Other Manuals Available	
ProcessLogix Theory Manual	1757-RM805A-EN-P
ProcessLogix Function Block Reference	1757-RM810A-EN-P
ProcessLogix Error Codes and Troubleshooting	1757-TG001A-EN-P
ProcessLogix Function Block Parameters	1757-RM811A-EN-P

If this is a new ProcessLogix system installation, we recommend that you familiarize yourself with the contents of these publications before you install any ProcessLogix system equipment. Visit us at: http:\\www.theautomationbookstore.com or contact your local sales office to obtain these manuals.

Installation declaration

ATTENTION

Environment and Enclosure

This equipment is intended for use in a Pollution Degree 2 industrial environment, in overvoltage Category II applications (as defined in IEC publication 60664-1), at altitudes up to 2000 meters without derating.

This equipment is considered Group 1, Class A industrial equipment according to IEC/CISPR Publication 11. Without appropriate precautions, there may be potential difficulties ensuring electromagnetic compatibility in other environments due to conducted as well as radiated disturbance.

This equipment is supplied as “open type” equipment. It must be mounted within an enclosure that is suitably designed for those specific environmental conditions that will be present and appropriately designed to prevent personal injury resulting from accessibility to live parts. The interior of the enclosure must be accessible only by the use of a tool. Subsequent sections of this publication may contain additional information regarding specific enclosure type ratings that are required to comply with certain product safety certifications.

See NEMA Standards publication 250 and IEC publication 60529, as applicable, for explanations of the degrees of protection provided by different types of enclosure. Also, see the appropriate sections in this publication, as well as the Allen-Bradley publication 1770-4.1 (“Industrial Automation Wiring and Grounding Guidelines”), for additional installation requirements pertaining to this equipment.

FIM and I/O module allowance

Be sure your ProcessLogix System Fieldbus and I/O requirements do not exceed the capacities listed in the following table. In terms of processing allocations, the FIM is the equivalent of three I/O modules.

Table 3.B

Component	Total Per Controller	Total Per Server
Maximum number of FIMs plus I/O modules divided by three (including local and remote chassis I/O and rail I/O).	21	100
Maximum number of H1 links (independent LAS)	42	200
Maximum number of fieldbus devices[1]	672	3,000

(1) Each H1 link is capable of supporting (a practical limit) at least 16 fieldbus devices. This number may vary depending on the dynamics of the link.

Fieldbus network references

Please refer to the following publications for guidance in designing and implementing the fieldbus network to be interfaced to the ProcessLogix system through the FIM and its companion Remote Termination Panel (RTP).

Table 3.C

Publication Number/Title	Scope	Source
AG-140 / Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium Application Guide	Overview of what you need to know to wire, power, and layout network components	Fieldbus Foundation 9390 Research Blvd. Suite II-250 Austin Texas 78759-9780 www.fieldbus.org
AG-165 / Fieldbus Installation and Planning Guide	Outlines things to consider before installing a fieldbus network	Fieldbus Foundation 9390 Research Blvd. Suite II-250 Austin Texas 78759-9780 www.fieldbus.org
Relcom Inc.	Provides fieldbus wiring products, wiring design and installation data. Offer a free Fieldbus Wiring Design and Installation Guide you can download.	Visit the Relcom Inc. website www.relcominc.com

Fieldbus wiring selection and calculation

The preferred cable for connecting fieldbus devices is #18 AWG (0.8mm^2) shielded, twisted pair wire. It is important to calculate how the planned topology for your fieldbus segment, selected wiring, supplied power and intended mix of fieldbus devices may impact the overall performance of a fieldbus network.

The original Fieldbus specification allows using twisted pair wiring, which is commonly used for 4-20 ma transmitters.

TIP

We can't emphasize enough the use of high quality network (wire) installation. The higher quality of installation materials the better performance you will achieve in your network application

Universal manufacturers sell "Fieldbus cable" which meets all the specifications required for "Fieldbus cable".

Windows based, Fieldbus Segment Calculator tools are available in the market place that can assist you in calculating the performance characteristics of a planned fieldbus segment.

See Appendix D for a condensed overview of fieldbus wiring considerations provided for convenient reference. This information does overlap some information that is found in other data references as well.

Installing 1757-FIM Fieldbus Interface Module

Refer to the 1757-FIM Installation Instructions, publication 1757-IN0913A-EN-P.

Installing 1757-RTP Remote Terminator

Refer to the 1757-RTP Installation Instructions, publication 1757-IN915A-EN-P.

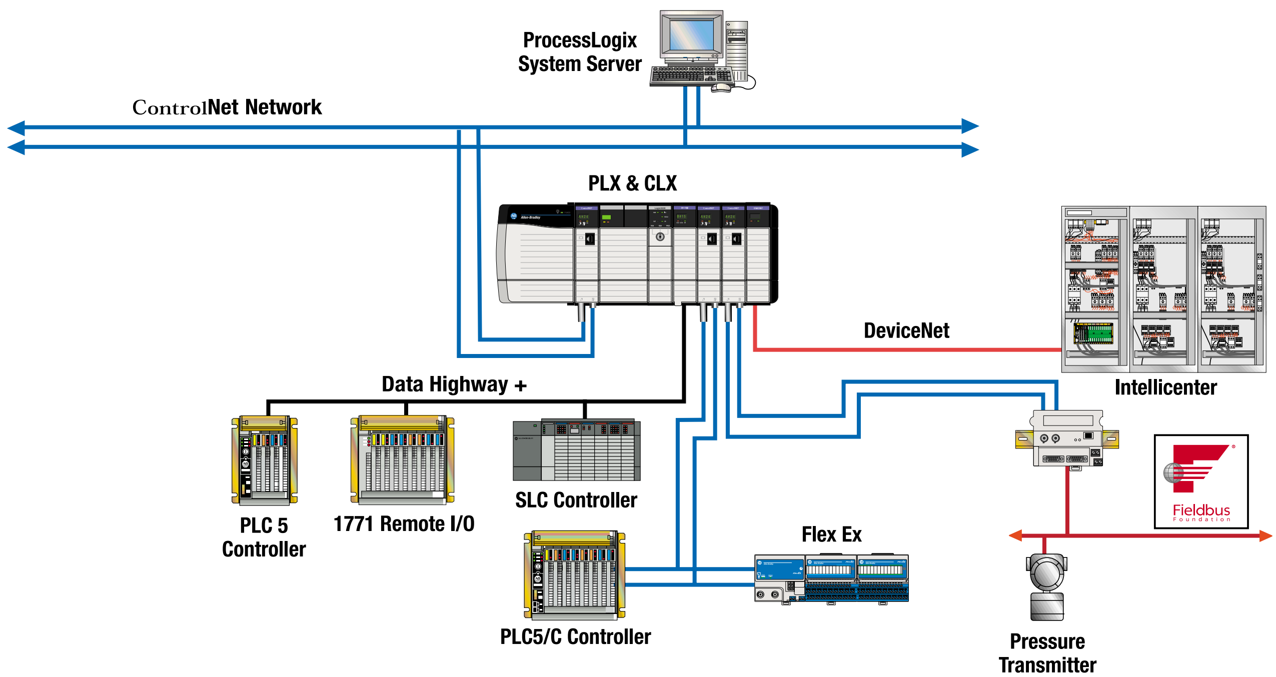
Configurating the 1757-FIM

Before You Start

Table 4.A Where do you begin?

Question:	If your answer is:	If your answer is
What do you know about Control Builder?	Nothing. Read the *Functional Overview* section in the Control Building Guide or locate the topic in Knowledge Builder. This section shows you how to launch the application and complete the Server login.	Yes, you can skip this section.
Do you know how to configure a Control Processor Module?	No. Read the *Creating a Control Processor Module* section in the Control Building Guide or find the topic in Knowledge Builder. This section shows you how to create a Control Processor Module (CPM) and its associated Control Execution Environment (CEE).	Yes, you can skip this section.
Can you configure a Control Module?	No. Read at least the *Creating and Saving a Control Module* section in Knowledge Builder. This section shows you how to create a Control Module (CM) and insert and connect function blocks.	Yes, you can skip this section.
Are you familiar with your system architecture?	To complete the configuration data for certain components, you must know the planned or current location of the associated hardware components in your ProcessLogix system architecture. This includes the chassis slot location for any given CPM and FIM. We suggest that you create a simple diagram that outlines the location of components in your system showing slot locations and communication addresses for reference during configuration.	
Are you ready?	Once you have addressed all the questions in this section, you are ready to move on to the next section *Configuring Fieldbus Components*. At this point, you should have at least a working knowledge of the Control Builder application.	

Figure 4.1 Example Rockwell Fieldbus Configuration



Configuring Fieldbus Components In a Control Strategy

About ProcessLogix control strategy configuration

You use ProcessLogix's Control Builder application to configure a process Control Strategy using predefined function blocks. Since Fieldbus Foundation had been functionally integrated with the ProcessLogix system, the Control Builder enables the inclusion of fieldbus related Function Blocks for easy integration of fieldbus functions within the overall Control Strategy.

ProcessLogix R400.0 Control Builder includes a separate utility called the Fieldbus Library Manager application. The Fieldbus Library Manager provides the capability to create templates for fieldbus devices based on the vendor supplied Device Description (DD). This means each fieldbus device has an associated template for viewing and defining the configurable attributes of its fieldbus function blocks. These attributes include naming and identifying the component's location within the network as well as setting device and channel specific parameters, as applicable.

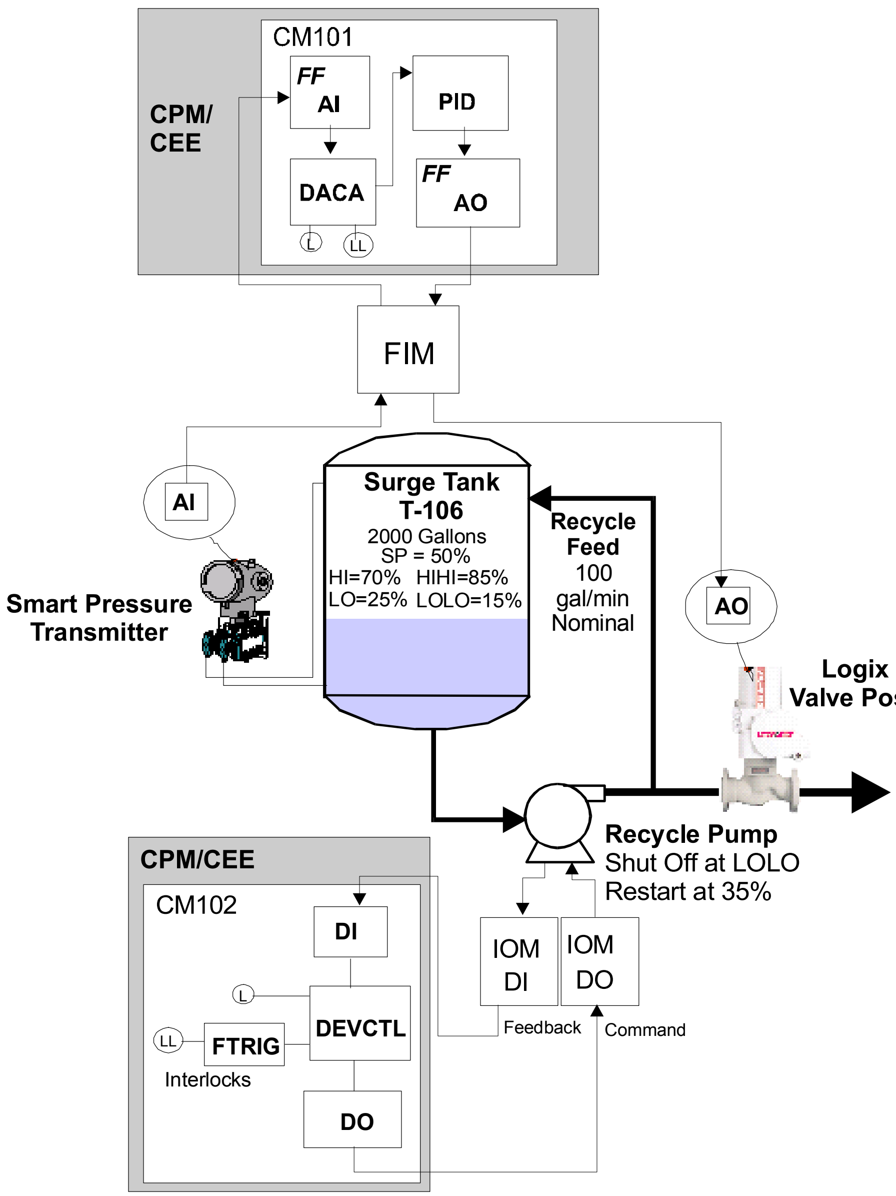
ATTENTION

The following information is only intended as a supplement to the Control Building Guide and does not repeat the basic functionality details for calling up, navigating, and interacting with the application.

Example Application and Control Strategy for Procedural Reference

Figure 4.2 shows a process feed and recycle line application being controlled through a ProcessLogix control strategy loaded in a Control Process Module (CPM) and associated Control Execution Environment (CEE). This sample application and control strategy will be used for reference to illustrate the applicability of functions in the following procedures

Figure 4.2 Sample Application and Control Strategy Integrating Fieldbus Devices with a ProcessLogix System as the Supervisory Control.



The application involves controlling the level of a 2000 gallon surge tank with a steady-state 100 gallon per minute (gal/min) process feed and recycle line. A fieldbus approved smart pressure transmitter is being used to monitor the level in the surge tank. A fieldbus approved valve positioner is being used to regulate the control valve in the process feed line.

The following are some pertinent characteristics about this application and the corresponding ProcessLogix control strategy for reference.

- The goal is to recycle the process fluid back to the process with a minimum swing in the recycle feed rate.
- The tank level set point (SP) is 50 percent.
- The tank level low (LO) alarm is 25 percent and the low-low (LOLO) alarm is 15 percent.
- The tank level high (HI) alarm is 70 percent and the high-high (HIHI) alarm is 85 percent.
- The ProcessLogix control strategy includes a tank level Control Module (CM) named CM101 and a pump control CM named CM102.
- The CM101 includes a ProcessLogix Proportional, Integral, Derivative (PID) function block configured to operate as a Proportional and Integral (PI) two-mode controller.
- The tuning for the PI controller is "loose" to allow some swing in the level of fluid in the tank.
- The CM101 includes a ProcessLogix Data Acquisition (DATAACQ) function block to provide the alarm flags for the LO, LOLO, HI, and HIHI tank level alarms.
- The CM101 includes a fieldbus Analog Input (AI) function block to integrate the tank level indicating signal from the ST3000FF smart pressure transmitter with the control strategy. It includes a fieldbus Analog Output (AO) function block to integrate the Logics 1400 Valve Positioner with the PI controller output from the control strategy.
- The CM102 includes a ProcessLogix Device Control (DEVCTL) function block to control the recycle pump through corresponding Discrete Input and Discrete Output Input/Output Module function blocks. It also includes links to the LO and LOLO alarm flags for the DATAACQ block in CM101 for device safety override and output override interlocks, respectively.
- The recycle pump is to be restarted at 35 percent.

System Management Timers

T1, T2, and T3 are the System Management Timers. The units are 1/32000 of a second, so 96000 gets 3 seconds.

T1

T1 specifies how long the 1788-CN2FF waits for an answer to a System Management message, such as Set PD Tag.

The time needs to include the time to acquire the Token for Unscheduled Transmission, transmit the message, remote node to process the message, remote node to acquire Token, transmit reply. Depending on the Function Block execution and Publishing schedule, this might take as little as 100 mS, but 3 seconds is recommended by the FF, and it doesn't hurt to allow extra time for slow nodes or slow commands.

T2

T2 specifies how long a Remote Node, such as a Pressure Transmitter waits for the next System Management message in a series of System Management messages until it concludes that the System Manager has failed. This is used for SET_ADDRESS sequence as shown in the following diagram.

As you can see, to set a node's address requires several messages to be sent from the System Manager (the CN2FF in our case) to a node. If the System Manager fails before completing all the steps, the Remote Node must give up and resume operation at its previous settings. The time for the remote node to wait after each step is called T2.

T3

Specifies how long after a node is changed to a new address it might take for it to begin communication at the new address.

This time includes stuff like writing the new information to flash memory and re-initializing some stuff, but is primarily the time it takes for the Link Active Scheduler (LAS), the CN2FF in this case, to Probe the Node at its new address.

Since the CN2FF only occasionally polls the addresses that have shown no previous activity (called the Slow Poll List), it takes relatively longer to detect a node at a new address than to pass the Token to a known node at an active address. The sequence is sort of like:

Pass Token 17
Pass Token 18
Pass Token 19
Pass Token 20
Probe Node 32
<no answer>
Pass Token 17
Pass Token 18
Pass Token 19
Pass Token 20
Probe Node 33
<no answer>
etc. up to
Probe Node 255
<no answer>
Pass Token 17
Pass Token 18
Pass Token 19
Pass Token 20
Probe Node 21
<answer from our new Node 21>
Pass Token 17
Pass Token 18
Pass Token 19
Pass Token 20
Pass Token 21 (this one is in the Live List now)
Probe Node 22
etc.

You can see that the token is passed around the ring perhaps 200 times between Probes of inactive addresses to see if there is someone new.

Software Example:

```
Forever:
    For each address not in the Live List
        For each node in the Live List
            Pass Token
            <node transmits if it has something to say>
            <node returns token>
        end For
        Probe Address to see if a Node is there now.
        if Node answers, add to Live List
    end For
End Forever
```

ACSYNCINTR

This is the period of time between Application Clock synchronization messages. Application Clock synch messages are used to coordinate the 'application clock' among the various nodes. The Application Clock is used by each Node to begin execution of its Function Blocks at the Scheduled Time.

This is important so that the Function Block will be Done Executing at the time it is Scheduled to Publish the answers (outputs) of the Function Block to other nodes on the Fieldbus. On H1, the LAS will tell the node (through a Compel Data message) when to Publish, but it is up to the node to schedule block execution at the right time.

If the crystals of the clocks of all the Fieldbus Nodes were exactly the same, this message would not be necessary, but since a man with 2 watches never knows what time it is, the time must be re-synchronized.

The clock synchronization method facilitates synchronization of clocks which run at different speeds very well. It does not do so well for clocks that speed up and slow down. i.e. consistently slow clocks work ok, but sporadically slow clocks are difficult.

So the question is how long does it take until the clocks drift far enough apart for anyone to care?

If it is set to 1 ms, then you waste all your fieldbus messages updating the clock and never get anything done. If it is set to say 5 hours, then a clock can drift a long way from the LAS's clock causing “Stale Data” because the function block did not execute prior to the node being Compelled to Publish its data (there are other causes of Stale Data also).

5 seconds is a comprimise - reigning in deviant clocks before they get too far out of hand, yet not wasting much network bandwidth. You might increase it if you are in a real pinch for network speed, but there won’t be a measurable improvement (by removing 1 ms of traffic out of 5 seconds or 1/5000).

You can decrease it if you think that the clocks are drifting apart prior to 5 seconds, but most of the cause of clock drift is because of the jitter in delivery of time sync messages (theory, don't repeat it), so if anything, spreading the transmission error out over a longer time would help the synchronization.

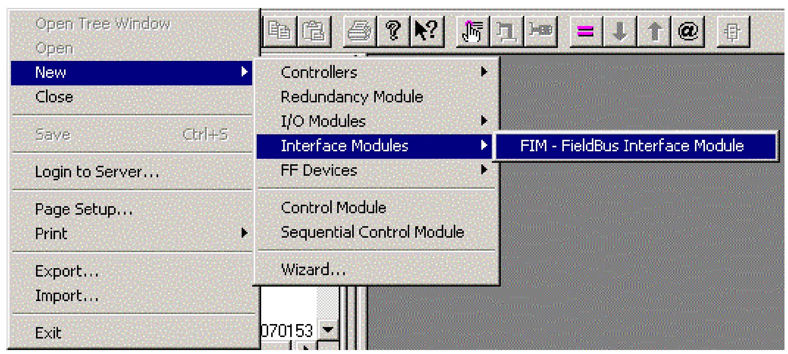
Adding Fieldbus Interface Module to Project

Use the following procedure to add a Fieldbus Interface Module block to the Project tab in Control Builder. This also adds two Link blocks for the two H1 fieldbus links that can be associated with this FIM.

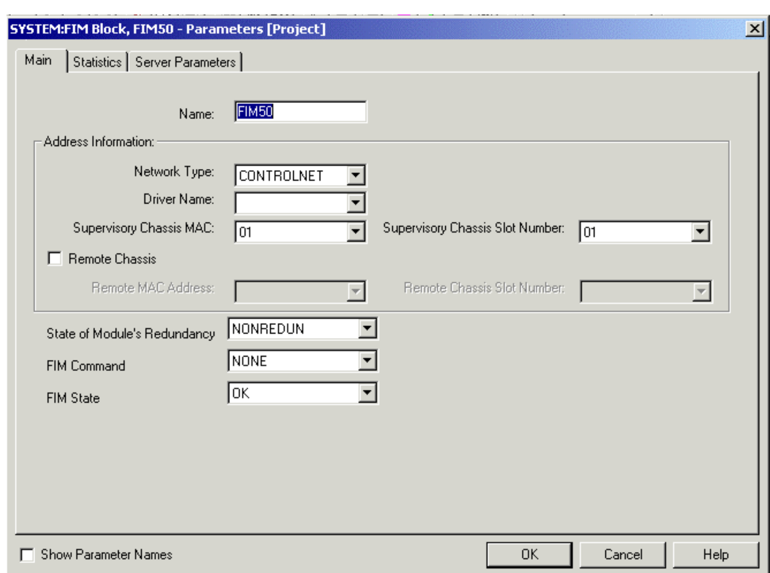
TIP

You can configure a FIM block in the Control Builder Project tab without the FIM hardware installed. However, it is good idea to have the communications driver and hardware that is going to be used for the system installed and configured. The FIM needs the name of the communications driver specified on its configuration form to complete its configuration data. Like the CPM, the FIM represents a hardware module and the block configuration specifies the communication path to the hardware.

3. With Control Builder running, click **New** ⇒ **Interface Modules** ⇒ **FIM - Fieldbus Interface Module**.



The FIM Block Parameters window opens.



4. Leave the CB assigned Name FIMxx, where xx equals the next unique sequential number. Or, enter a unique name of up to 16 characters.

5. In the Network field, select the communications medium your ProcessLogix system uses. Ethernet or ControlNet, ControlNet is the default selection.

6. In the Driver Name field, select the correct communications driver.

TIP

The communications driver must be installed and configured for it to be included in the dropdown list.

7. If the FIM is located in a remote chassis, go to step 9.

   In the Supervisory Chassis MAC field, select the MAC address assigned to the ControlNet Network module connected to the Supervisory network also known as the “uplink”.

8. In the Supervisory Chassis Slot Number field, select the slot number where the FIM is installed. Go to step 12.

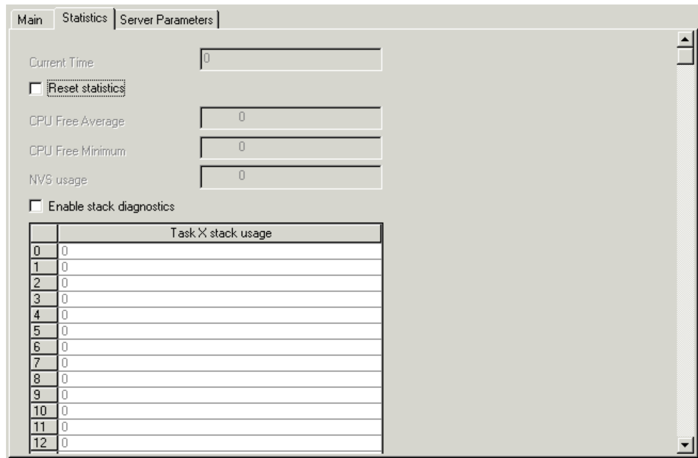
9. If the FIM is located in a remote chassis, select the check box next to the Remote Chassis Field.

10. In the Remote MAC Address field, select the address of the ControlNet Network module in the Remote I/O Chassis.

11. In the Remote Chassis Slot Number field, select the slot number where the FIM is installed.

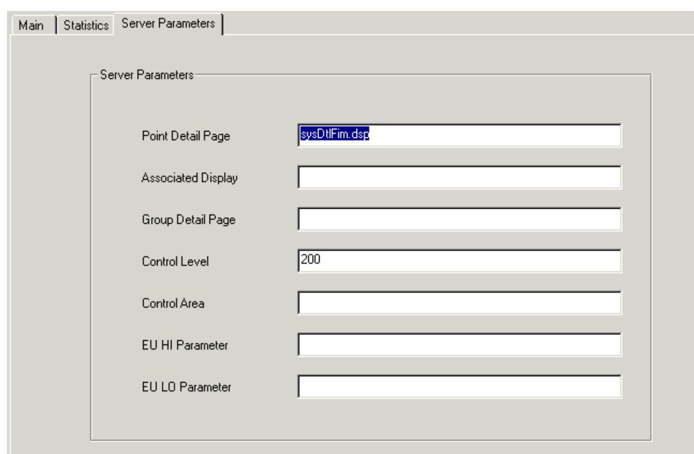
12. Leave all other fields on the Main tab at their defaults, as these are the only valid values at this point.

    Click the **Statistics** tab.



    Data is only present in these fields when the FIM/LINK is loaded and communicating with the system.

13. Click the **Server Parameters** Tab.

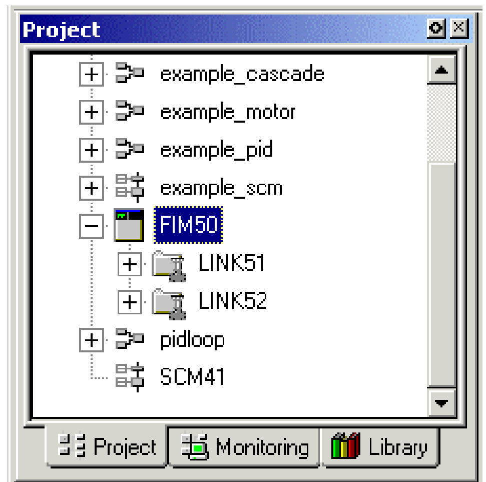


14. Leave the Point Detail Page and Control Level fields at their default.

    The Associated Display and Group Detail Page are not required to complete the configuration, but can be entered if known.

15. Click **OK**.

The FIM icon is added to the Project tab. The FIM also includes icons for the two H1 fieldbus links that it supports.



TIP

Refer to the 1757-FIM as a controller because the module can function independently, without a 1757-PLX52.

Checking link configuration

Use the following steps to check the link configuration of the links associated with a given FIM block. This procedure assumes that you have configured a FIM block in the Project tab of Control Builder.

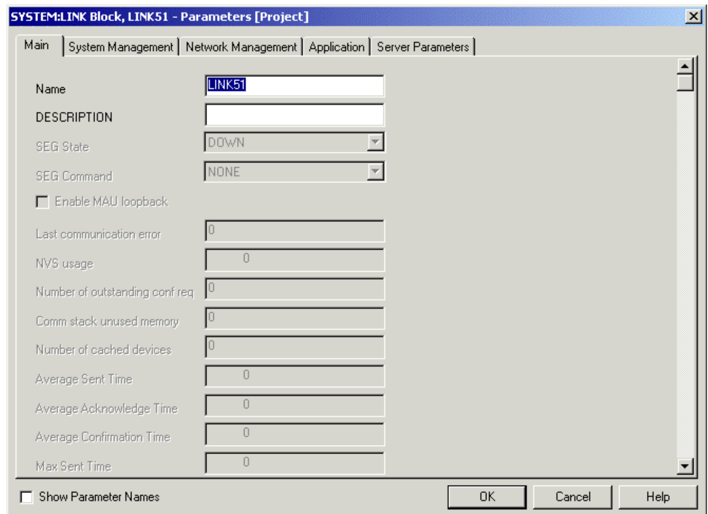
TIP

You can configure a Link through the Project tab of Control Builder without having the link installed. However, some parameters on the Link configuration form can only be viewed through the Monitoring tab with the FIM and Link installed and communicating with the system.

Be sure to click the plus sign in front of the FIM icon to open its directory tree and expose the link icons.

1. Double-click the link icon .

The Link Block Parameters window opens.

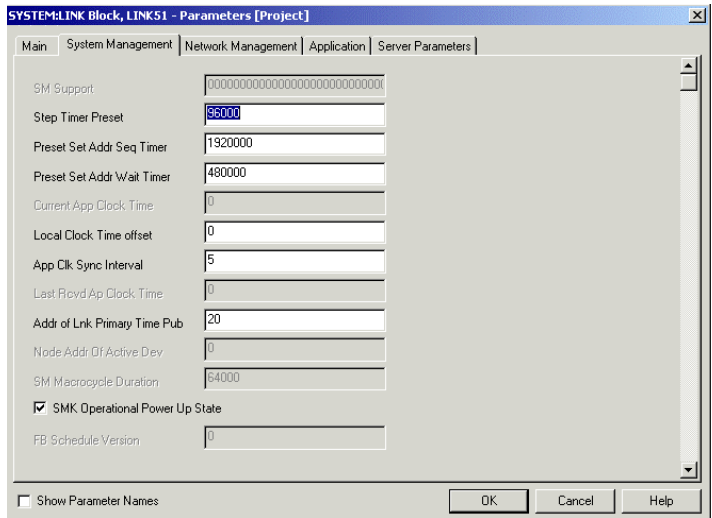


2. Leave the CB assigned name LINKxx, where xx equals the next unique sequential number assignment. Or, enter a unique number of up to 16 characters.

3. In the Description field, enter a description of up to 24 characters. This text appears in applicable detail and group displays associated with this block.

The other parameters can not be configured because they are only active in the Monitoring tab after the FIM/LINK is loaded and communicating with the system.

4. Click the **System Management** tab.



ATTENTION

Do not change the default value settings for the active parameters in this window unless you are familiar with tuning the performance of fieldbus links.

5. In the Step Time Preset (T1) field, either leave the default value of 96000, or enter a new value.

This is the preset value for the System Management step timer in 1/32 millisecond increments.

6. In the Preset Set Addr Seq Timer (T2) field, either leave the default value of 1920000, or enter a new value.

This is the preset value for the System Management set address sequence timer in 1/32 millisecond increments.

7. In the Preset Set Addr Seq Timer (T3) field, either leave the default value of 480000, or enter a new value.

This is the preset value for the System Management set address wait timer in 1/32 millisecond increments.

8. In the Local Clock Time Offset (LOCTIMDIFF) field, either leave the default value of 0, or enter a new value.

   This value is used to calculate the local time from the Curr App Clock Time (CURTIME) in the number of +/- 1/32 millisecond increments to add to the clock to obtain local time.

9. In the App Clk Sync Interval (ACSYNCINTR) field, either leave the default value of 5, or enter a new value.

   This value is the interval in seconds between time messages on the link. System management sets it during node address assignment. Set it 5 because this is a good compromise for more information see page 4-6.

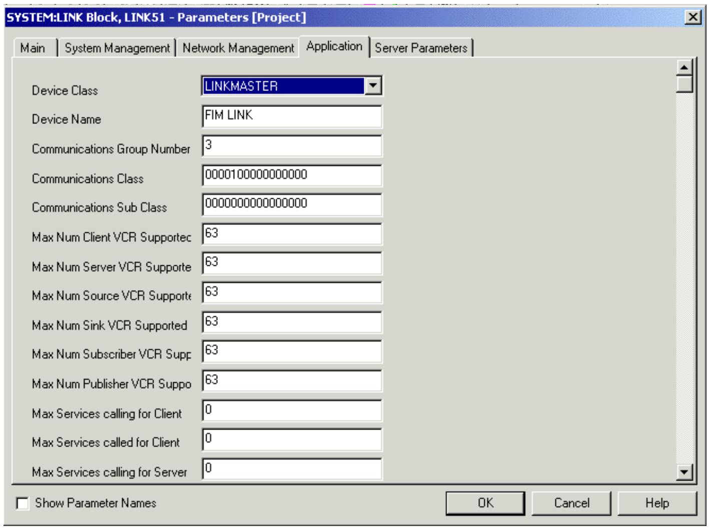
10. In the Addr of Lnk Primary Time Pub (PTIMEPUB) field, either leave the default value of 20, or enter a new value.

    This value is the node address of the primary time publisher for the local link. System Management sets it during node address assignment. It is a configuration error if the SM Support (SM_SUPPORT) parameter of the device specified by this node address does not have the Application Clock Synchronization feature bit set.

11. To enable the SMK Operational Power Up state (OPERPWRUP), click the check box next to the field.

- If the device is in the SM_OPERATIONAL state or fully commissioned and operating when power is lost, this parameter controls which state the SMK will enter after powerup.
- If OPERPWRUP is enabled, the SMK will enter the SM_OPERATIONAL state on powerup.
- If OPERPWRUP is disabled, the SMK will enter the INITIALIZED state.

12. Click the **Application** tab.

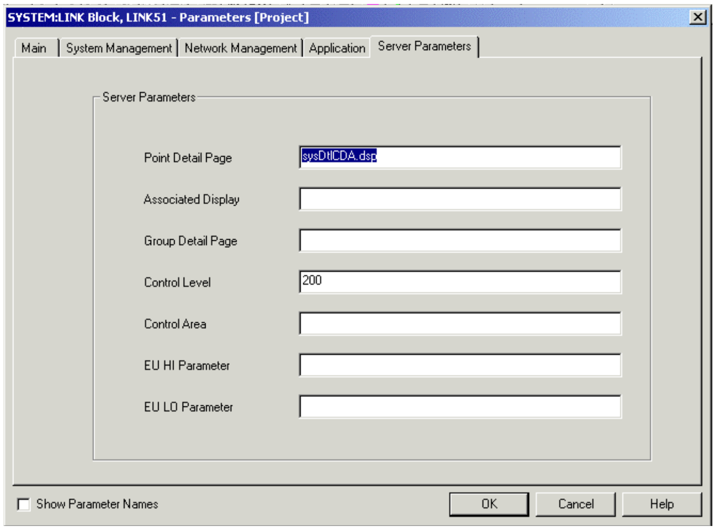


13. In the Device Name (DEVNAME) field, either leave the default name, or enter a new name of up to 33 characters.

TIP The remaining parameters on this tab define the Application Relationships that determine how Application Processes communicate with each other. Leave the default values for these parameters, then revisit them in the Monitoring tab when the FIM/LINK is communicating with the system.

14. The parameters on the Network Management tab are only accessible in the Monitoring tab with the FIM/LINK communicating with the system.

Click the **Server Parameters** tab.



15. Leave all other fields with their default displays and click **OK**.

16. Repeat these steps for another Link, as required.

Making a Fieldbus Device Template from a Vendor's DD

Use the following steps to add a template to the Control Builder Library using the Fieldbus Library Manager utility included with ProcessLogix R400.0 Engineering Tools and vendor supplied Device Description (DD) files, version 4.01 or later.

TIP

You must have the DD files for the fieldbus device either on a floppy diskette or the Foundation Fieldbus Compact Disc supplied by the manufacturer. This means you can make a device template while the system is offline. You can skip this procedure, if a template for the given device type already exists in the Control Builder Library database.

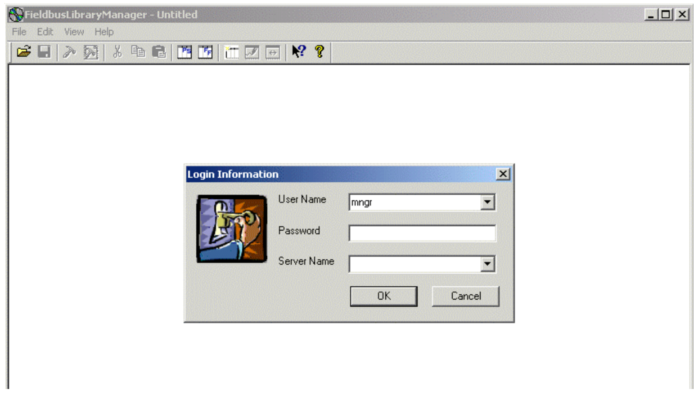
Rockwell Automation has tested and included some devices as part of the base product for you convenience. See Appendix E for general information about using the Fieldbus Library Manager utility and lists of available templates.

ATTENTION

Exit the Control Builder application before launching the Fieldbus Library Manager utility.

1. Click **Start** ⇒**Programs** ⇒**ProcessLogix Engineering Tools** ⇒**Fieldbus Library Manager**.

   The Fieldbus Library Manager (FLM) login window opens.



2. In the **Server Name** field, select the Server where you want the device template stored.

   The device template is stored in the Engineering Repository Database (ERDB) on the Server.

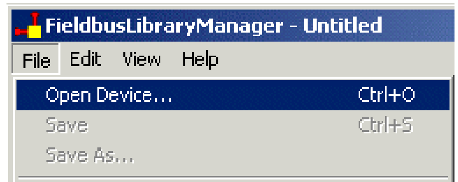
3. Enter your login password in the **Password** field.

   You need an access level of at least Engineer to make a template. See your system administrator to get your assigned password, if required. The password is tied to the operator security for Station.

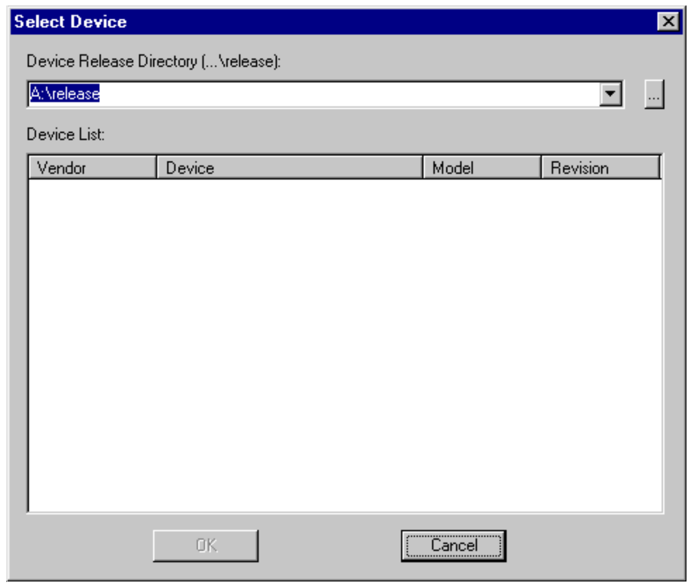
4. Click **OK**.

5. Insert the DD floppy diskette into your computer's floppy drive.

6. In the FLM window, click **File ⇒Open Device**.



The Select Driver window opens.

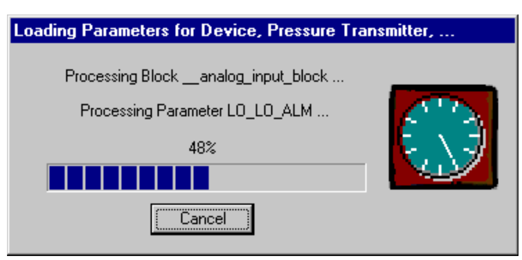


7. In the Device Release Directory field, select the device you want to make a template for in the Device List box.

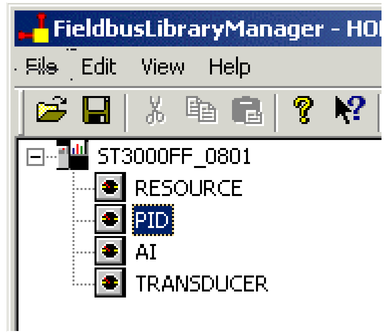
   You can use either the drop-down menu or Browse to select the device.

8. Click **OK**.

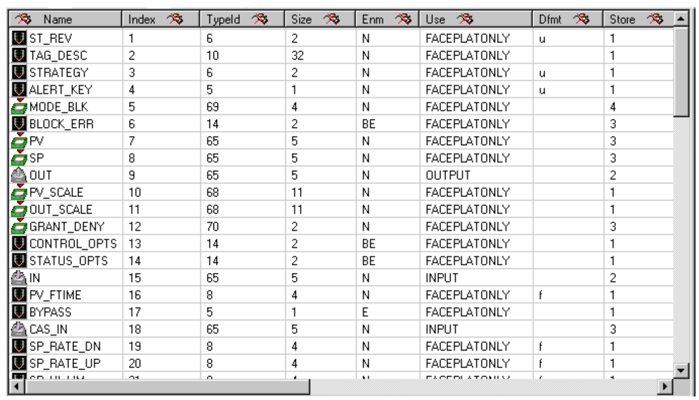
   The device data begins to load to the FLM.



9. After loading is complete, double-click the function block you want to view or edit under the device directory on the left side of the FLM window.



Block parameters are displayed in table format on the right side of the FLM window.



Name	Index	TypeId	Size	Enm	Use	Dfmt	Store
ST_REV	1	6	2	N	FACEPLATONLY	u	1
TAG_DESC	2	10	32	N	FACEPLATONLY		1
STRATEGY	3	6	2	N	FACEPLATONLY	u	1
ALERT_KEY	4	5	1	N	FACEPLATONLY	u	1
MODE_BLK	5	69	4	N	FACEPLATONLY		4
BLOCK_ERR	6	14	2	BE	FACEPLATONLY		3
PV	7	65	5	N	FACEPLATONLY		3
SP	8	65	5	N	FACEPLATONLY		3
OUT	9	65	5	N	OUTPUT		2
PV_SCALE	10	68	11	N	FACEPLATONLY		1
OUT_SCALE	11	68	11	N	FACEPLATONLY		1
GRANT_DENY	12	70	2	N	FACEPLATONLY		3
CONTROL_OPTS	13	14	2	BE	FACEPLATONLY		1
STATUS_OPTS	14	14	2	BE	FACEPLATONLY		1
IN	15	65	5	N	INPUT		2
PV_FTIME	16	8	4	N	FACEPLATONLY	f	1
BYPASS	17	5	1	E	FACEPLATONLY		1
CAS_IN	18	65	5	N	INPUT		3
SP_RATE_DN	19	8	4	N	FACEPLATONLY	f	1
SP_RATE_UP	20	8	4	N	FACEPLATONLY	f	1

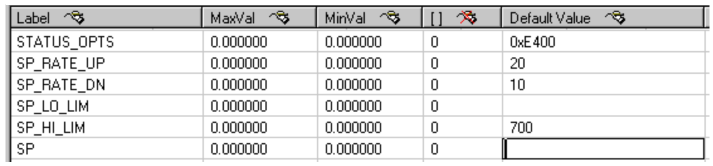
TIP

By keeping default values, you may use the same template for like vendor devices used in multiple locations in your application. You can make adjustments to selected device parameters through Control Builder configuration access.

You can find definitions for many of the standard fieldbus function blocks in Appendix A in this document. Vendors also supply a text file that includes definitions for each function block used in their device.

10. To edit a given parameter value, double-click in the desired cell, or right-click the field and select edit.

The edit field is activated.

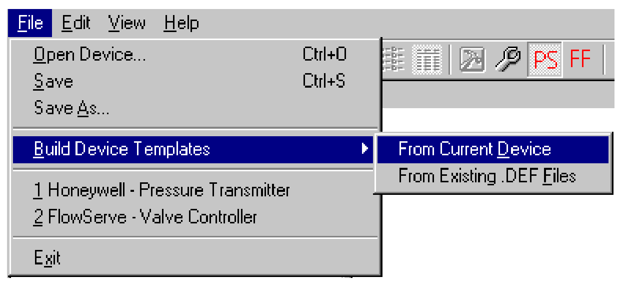


Label	MaxVal	MinVal	[ ]	Default Value
STATUS_OPTS	0.000000	0.000000	0	0xE400
SP_RATE_UP	0.000000	0.000000	0	20
SP_RATE_DN	0.000000	0.000000	0	10
SP_LO_LIM	0.000000	0.000000	0	
SP_HI_LIM	0.000000	0.000000	0	700
SP	0.000000	0.000000	0	

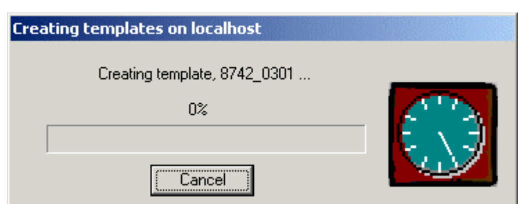
11. Enter the desired value and press **Enter**.

12. Repeat Steps 9, 10, and 11 to edit parameters for your application, as required.

13. Click **File ⇒Build Device Template ⇒From Current Device**.



The ProcessLogix device template is built from current data in the ERDB and a device directory (name pre-defined by the vendor) is added to the Library tab in Control Builder. If the device directory already exists, the device is added to the directory.



TIP

If your ProcessLogix system architecture includes multiple clients, the newly created device template resides only in the database of the Server designated during FLM login. You must be logged onto the given Server to access device templates stored in its ERDB through Control Builder.

14. Click **OK** to acknowledge the build complete message.

15. Repeat Steps 6 to 14 to make other device templates.

16. (Optional) Launch Control Builder to confirm that the device template is listed in the Rockwell Automation directory in the Library tab.

Making a fieldbus device template from existing definition (.DEF) files

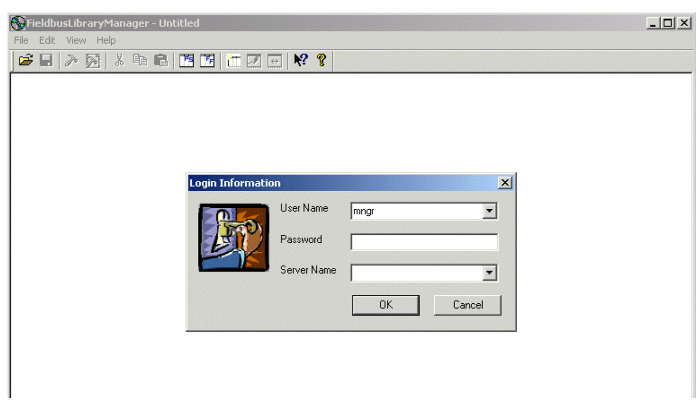
Use the following procedure to make a device template from definition files that have been previously saved to a folder on the Server's hard drive. The default folder location is: ***C:\Honeywell\tps50\system\bin\er\ffdevices***.

ATTENTION

Exit the Control Builder application before launching the Fieldbus Library Manager utility.

1. Click **Start** ⇒**Programs** ⇒**ProcessLogix Engineering Tools** ⇒**Fieldbus Library Manager**.

   The Fieldbus Library Manager (FLM) login window opens.



2. In the **Server Name** field, select the Server where you want the device template stored.

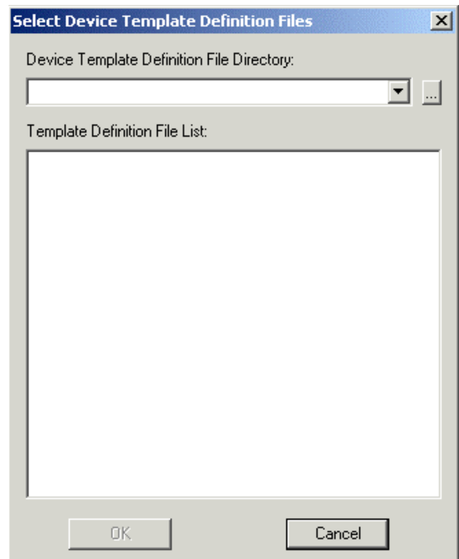
   The device template is stored in the Engineering Repository Database (ERDB) on the Server.

3. Enter your login password in the **Password** field.

   You need an access level of at least Engineer to make a template. See your system administrator to get your assigned password, if required. The password is tied to the operator security for Station.

4. Click **OK**.

5. Click **File ⇒Build Device Template ⇒From Existing .DEF Files**.
   The Select Device Template Definition Files dialog box opens.



6. Click **Browse** to navigate to the directory containing the saved definition (.DEF) files. The default directory is: C:\Honeywell\tps50\system\bin\er\ffdevices.

7. Click **OK**.

8. Click **OK** to initiate the template build from selected .DEF files.

9. Click **OK** to confirm.

   The device directory (name pre-defined by the vendor) is added to the Library tab in Control Builder. If the device directory already exists, the device is added to the directory.

TIP

If your ProcessLogix system architecture includes multiple clients, the newly created device template resides only in the database of the Server designated during FLM login. You must be logged onto the given Server to access device templates stored in its ERDB through Control Builder.

10. Click **OK** to acknowledge the build complete message.

11. Repeat Steps 5 to 10 to make other device templates.

12. (Optional) Launch Control Builder to confirm that the device template is listed in the Rockwell Automation directory in the Library tab.

Adding a Fieldbus Device to Project

Use either method that follows to add a fieldbus device from a Library tab directory to the Project tab. These procedures assumes that Control Builder is running and you have made templates for the fieldbus devices that will be used in your application, using the Fieldbus Library Manager utility.

TIP

If a FIM and its associated Links exist in the Project tab, you can drag and drop a fieldbus device from a directory in the Library tab directly to a Link. The following procedure adds a fieldbus device to the (offline) Project tab and then shows how to assign it to a Link through the Assignment dialog box in the next procedure. Once a device is assigned to a Link, it represents a matching physical device that is to reside on the given Link.

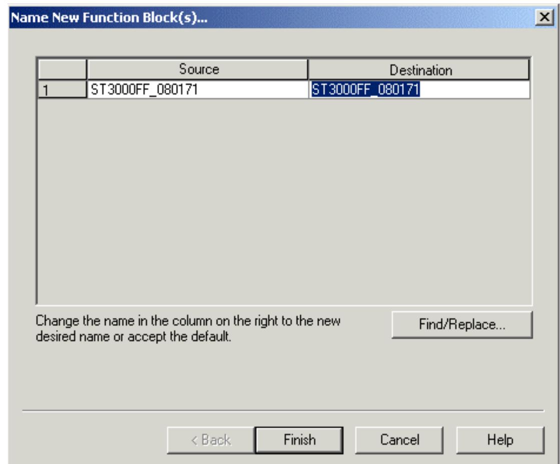
Method 1: Drag and Drop

1. Click the **+** in front of the applicable vendor template directory in the Library tab.

   The directory tree expands to show stored device templates.

2. Drag the device icon to a open area in the project tab. The cursor appears with a +, when positioned in a valid location.

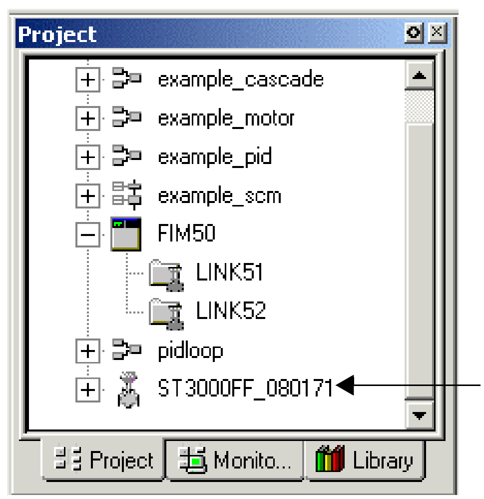
The Name New Function Block(s) dialog box opens.



3. Leave the default name that appears in the Destination column or enter a new name of up to 16 characters.

4. Click **Finish**.

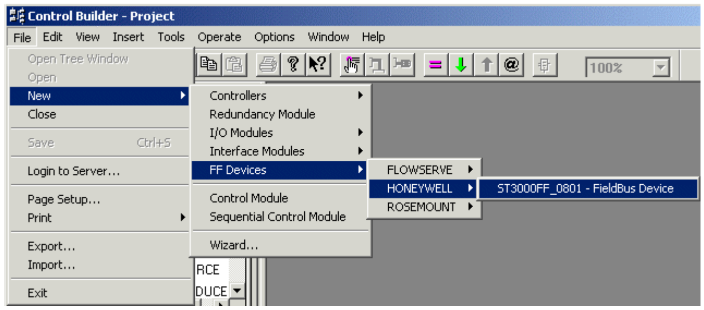
A device icon with the given name is created in the Project tab.



5. Repeat this method to add other devices.

Method 2: File Menu

1. Click **File** ⇒**New** ⇒**FFDevices** ⇒**Rockwell Automation** ⇒**(desired device name)-Fieldbus Device**.

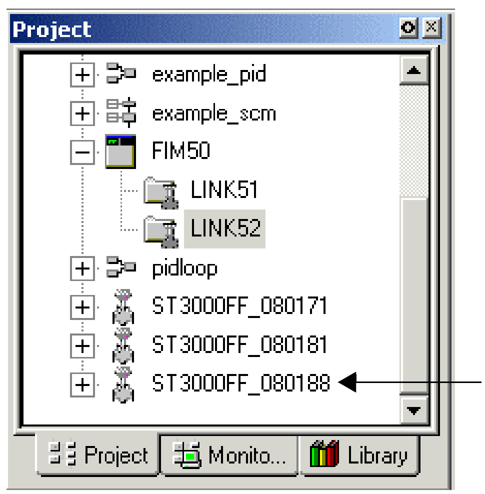


The Block Parameters window opens.

2. Leave the default Name or enter a desired name of up to 16 characters.

3. Click **OK**.

A device icon with the given name is created in the Project tab.



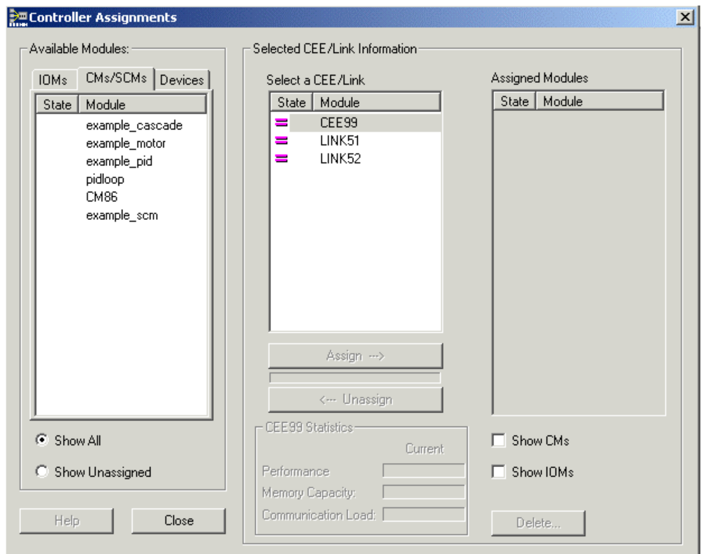
4. Repeat this method to add other devices.

Assigning a Device to a Link in Project

Use the following procedure to assign a device to a Link associated with the applicable FIM in the Project tab.

1. Click **Tools** ⇒**Assign**. Or, click the assign button in the toolbar.

   The Controller Assignments window opens.



2. Click the **Devices** tab.
3. Click the device you want to assign to a Link to highlight it.
4. Confirm that the desired Link is selected in the Select Link list box.
5. Click **Assign** to assign the selected device to the selected Link.

   The device is added to the Assigned Devices box with the assign state icon.

6. Click **Close**.

7. Repeat the procedure to assign other devices.

TIP

An alternate method to the following procedure is to drag and drop the device to the applicable Link in Project.

Checking Device Configuration

Use the following procedure to check the configuration of a selected fieldbus device in the Project tab. This procedure assumes that the device has been assigned to a Link.

TIP

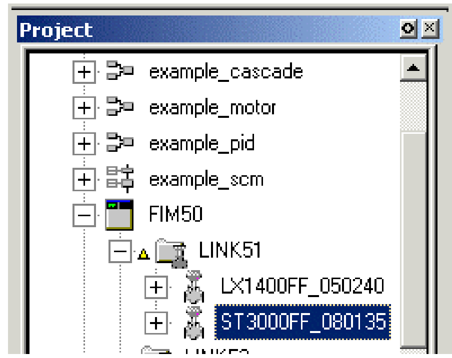
While the device has been assigned to a link, this is still "offline" configuration of a matching physical device that is to be connected to this Link.

Each device on the link must have a unique physical device tag (PD_TAG) that the fieldbus system relates to a node address. The physical device tag, node address, and the manufacturer device identifier (DEV_ID) are used to match a configured device to a physical device.

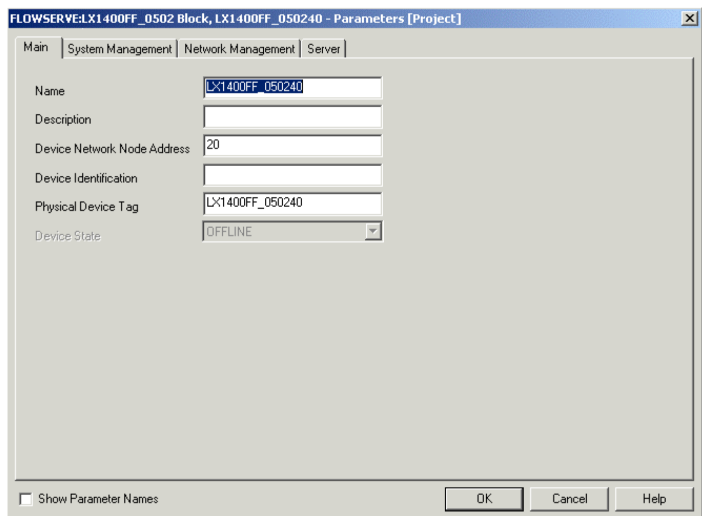
In ProcessLogix, each independent component in the control strategy must have a unique tag name (NAME). This is automatically enforced through the Control Builder application. For this reason, the unique name that was assigned to the device when it was created in project is also assigned as the device's physical device tag. This is done to assure that the names are unique within the system. This means a change in NAME results in an automatic change in PD_TAG to keep them the same. However, a change in PD_TAG does not result in an automatic change in the assigned NAME.

So, the name and physical device tag can be different. Since it is possible to change a physical device tag configuration using a handheld communicator, you **must** be sure the device name specified on this configuration form matches the device name configured in the device.

1. Double-click the device icon in the Project tab.



The Block Parameters window opens.



2. In the Name field, leave the default name or enter a unique name of up to 16 characters.

3. In the Description (DESC) field, enter the desired description of up to 59 characters, or leave it blank.

4. In the Device Network Node Address (ADDR) field, leave the default address.

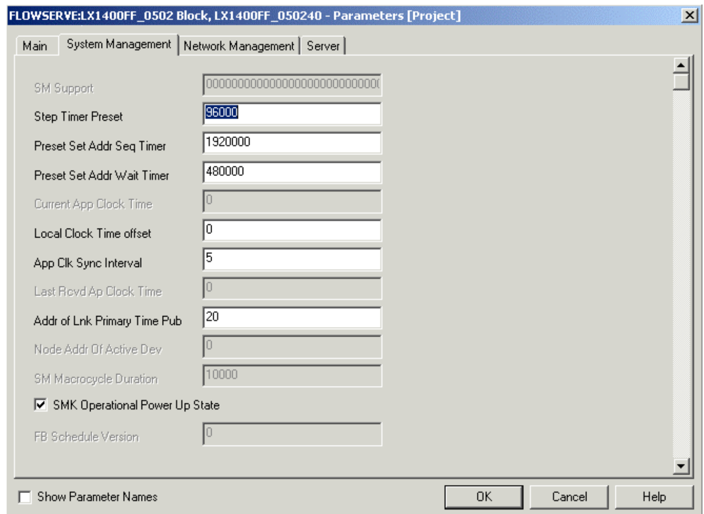
5. In the Device Identification (DEV_ID) field, enter the manufacturer's ID for the device, if known.

   The ID will be detected when the device is commissioned.

6. In the Physical Device Tag (PD_TAG) field, leave the assigned tag name, or enter the name that matches the one configured in the device, up to 16 characters.

7. The Device State (DEVSTATE) parameter is only active in the Monitoring tab after the FIM/LINK/DEVICE is loaded and communicating with the system.

8. Click the **System Management** tab.



ATTENTION

Do not change the default value settings for the active parameters in this window unless you are familiar with tuning the performance of fieldbus links.

In most cases, the parameter values will mirror those configured for the Link. The grayed out parameters are only accessible in the Monitoring tab with the FIM/LINK communicating with the system.

9. In the Step Time Preset (T1) field, either leave the default value of 96000, or enter a new value.

   This is the preset value for the System Management step timer in 1/32 millisecond increments.

10. In the Preset Set Addr Seq Timer (T2) field, either leave the default value of 1920000, or enter a new value.

    This is the preset value for the System Management set address sequence timer in 1/32 millisecond increments.

11. In the Preset Set Addr Wait Timer (T3) field, either leave the default value of 480000, or enter a new value.

This is the preset value for the System Management set address wait timer in 1/32 millisecond increments.

12. In the Local Clock Time Offset (LOCTIMDIFF) field, either leave the default value of 0, or enter a new value.

This value is used to calculate the local time from the Curr App Clock Time (CURTIME) in the number of +/- 1/32 millisecond increments to add to the clock to obtain local time.

13. In the App Clk Sync Interval (ACSYNCINTR) field, either leave the default value of 5, or enter a new value. See page 4-8 for more information.

This value is the interval in seconds between time messages on the link. System management sets it during node address assignment.

14. In the Addr of Lnk Primary Time Pub (PTIMEPUB) field, either leave the default value of 20, or enter a new value.

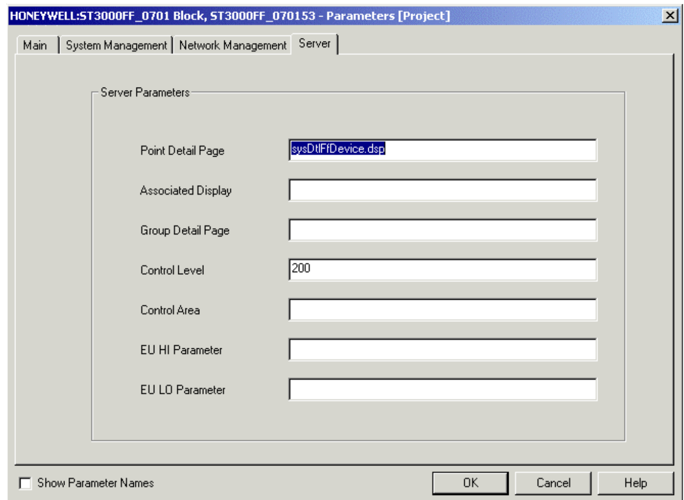
This value is the node address of the primary time publisher for the local link. System Management sets it during node address assignment. It is a configuration error if the SM Support (SM_SUPPORT) parameter of the device specified by this node address does not have the Application Clock Synchronization feature bit set.

15. To enable the SMK (System Management Kernal) Operational Power Up state (OPERPWRUP), click the check box next to the field.

If the device is in the SM_OPERATIONAL state or fully commissioned and operating when power is lost, this parameter controls which state the SMK will enter after powerup. If OPERPWRUP is enabled, the SMK will enter the SM_OPERATIONAL state on powerup. If OPERPWRUP is disabled, the SMK will enter the INITIALIZED state.

16. The parameters on the Network Management tab are only accessible in the Monitoring tab with the FIM/LINK communicating with the system.

    Click the **Server** tab.



IMPORTANT If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server (area code assignments are made through Station). If you do not have a DSA, you can skip this field, if Areas is not enabled through Station.

17. Leave all other fields with their default displays (unless you have created a custom graphic) and click **OK**.

18. Repeat this procedure for each device, as required to support your control strategy.

Creating Control Module for Sample PID Loop

The following procedures are optional tutorials on creating a CM to provide the PID control loop function for the example application shown in Figure 4.2. They provide a general reference for including fieldbus function blocks with ProcessLogix function blocks in a control strategy. This procedure assumes that Control Builder is running and that you have added the Control Processor (CPM)/Control Execution Environment (CEE) block to the Project tab. Also, the following fieldbus components have been created in the Project tab, using the previous procedures in this section.

- FIM named FIM_B1
- Associated Links named LINK_S101 and LINK_S102
- E & H Endress Hoisers pressure transmitter named ST101 and assigned to LINK_S101
- Flowserve valve controller named LX14_101 and assigned to LINK_S101

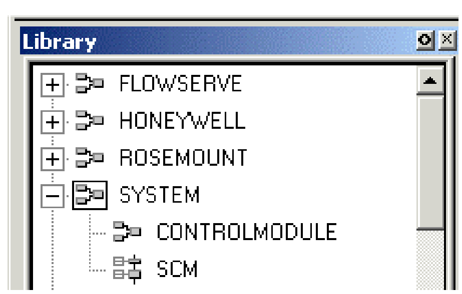
TIP

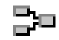
The following procedure is really intended for users who have never used Control Builder before. If you have used Control Builder to build ProcessLogix control strategies, you will find adding fieldbus function blocks is intuitive, since it is the same as adding ProcessLogix function blocks.

Also, the procedures are abbreviated in some cases in the interest of brevity. Once you get the "feel" of using Control Builder, it is more efficient to interact directly with the application and use the online help to find an answer to a question about a function or an entry.

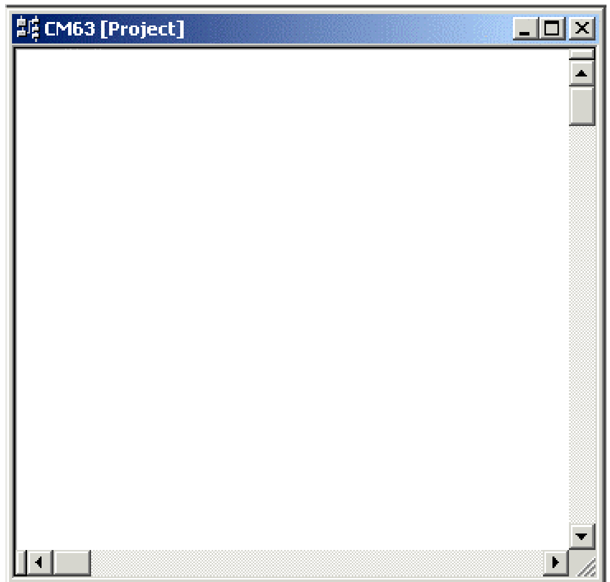
Creating Control Module in Project

1. In the Library tab, expand the System directory by clicking the  .



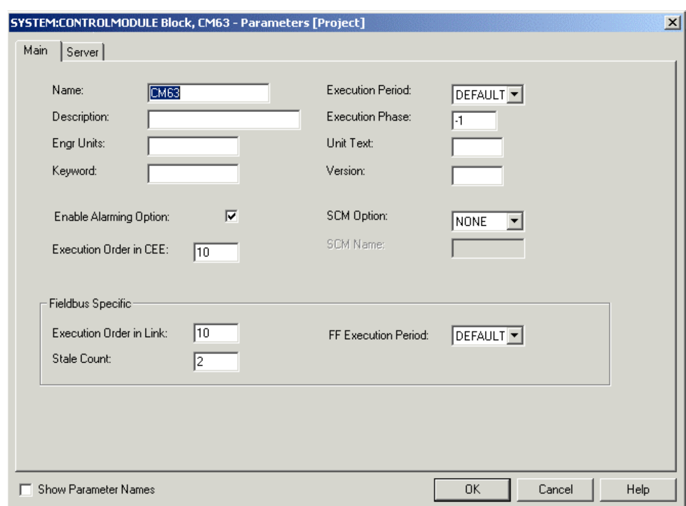
2. Double-click the CONTROLMODULE icon  .

   A new control module opens.



3. Click **Tools ⇒Configure Module Parameters**.

   The Control Module Block Parameters window opens.



4. In the **Name** field, enter **CM101**.

5. In the **Description** field, type **Surge Tank Control Loop**.

6. In the **Eng. Units** field, type **Percent**.

7. In the **Keyword** field, type **Recycle**.

8. Click the **Enable Alarming Option** check box, if it is not already selected.

9. In the **Execution Order in CEE** field, enter **30** (the smaller the number, the earlier the module executes in the cycle).

ATTENTION

The Execution Order in CEE parameter only applies to contained ProcessLogix function blocks in CM's that are assigned to a CEE. It has no affect on contained fieldbus blocks and it is disabled, if the CM is assigned to a Link.

10. Leave the **Execution Period** field at **DEFAULT**.

This equals1000ms for a 50ms CEE or 200ms for a 5ms CEE.

11. Leave the **Execution Order in Link** field at the default of **10** (the smaller the number the sooner the device is polled on the link).

ATTENTION

The Execution Order in Link parameter only applies to contained fieldbus blocks. It combines with the combines with the block's Execution Order in CM parameter to determine how the fieldbus block participates in the Link Active Schedule.

12. Leave the **FF Execution Period** field at **DEFAULT**.

ATTENTION

The FF Execution Period parameter only applies to contained fieldbus blocks. It determines how the fieldbus blocks participate in the Link Active Schedule.

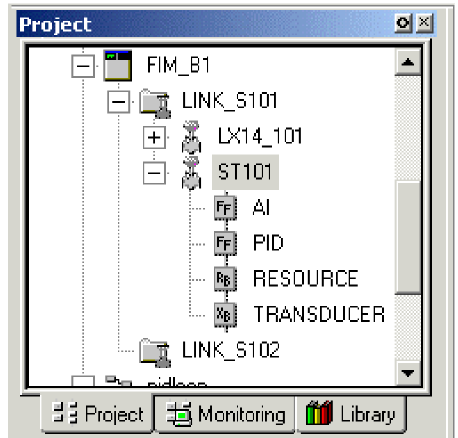
13. Leave the **Stale Count** field at the default of **2**.

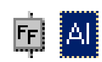
14. Leave all other parameters at their default values as they do not apply to this example.

Click **OK** to close the window and save the values.

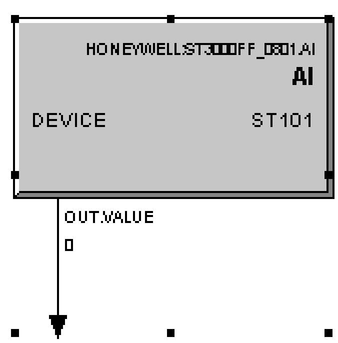
Adding blocks to CM for sample loop

1. In the Project tab, expand the FIM down to the device block level by clicking the  left of FIM, then LINK_S101, then ST101.

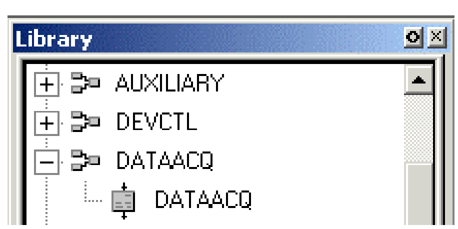


2. Click and drag the AI block  to CM101 chart.

   An FF AI block is added to the CM101 chart.

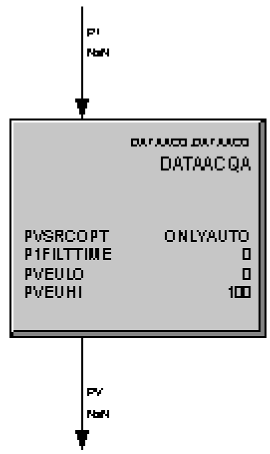


3. In the Library tab, expand the DATAACQdirectory by clicking the .

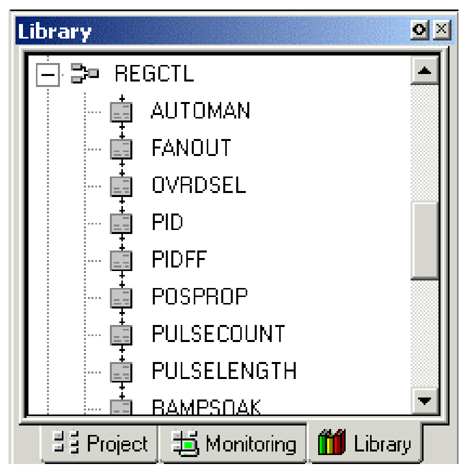


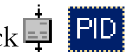
4. Click and drag the DATAACQ block  to the CM101 chart.

   A DATAACQ block is added to the CM101 chart.

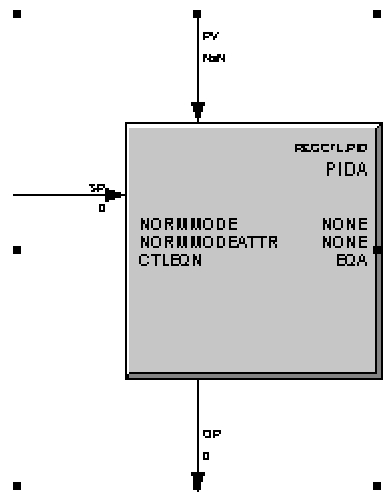


5. In the Library tab, expand the REGCTL directory by click the .

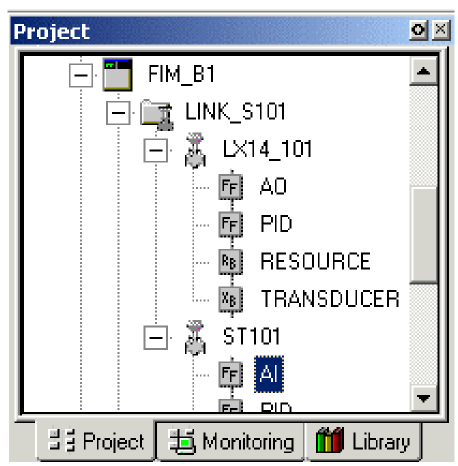


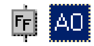
6. Click and drag the PID block  to the CM101chart.

The PID block is added to the CM101 chart.

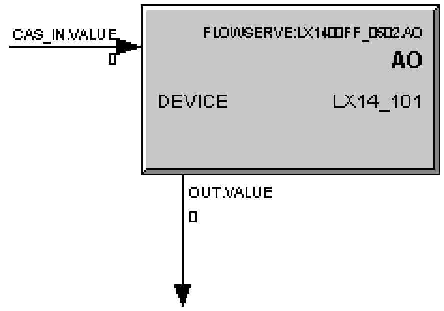


7. In the Project tab, expand the LX14_101 by clicking the .



8. Click and drag the AO block  to the CM101 chart.

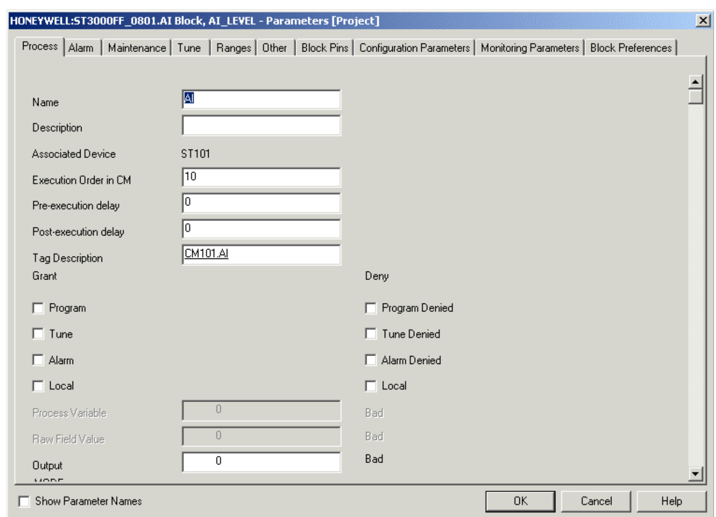
The AO block is added to the CM101 chart.



Configuring AI block for sample loop

1. In the CM101 chart, double-click the AI block.

The AI Block Parameters opens.



2. In the **Name** field, type **AI_LEVEL**.

3. In the **Description** field, type **Input for Tank 106 Level**.

4. Leave the **Execution Order in CM** at the default of **10** (the smaller the number the sooner the block executes within the CM cycle).

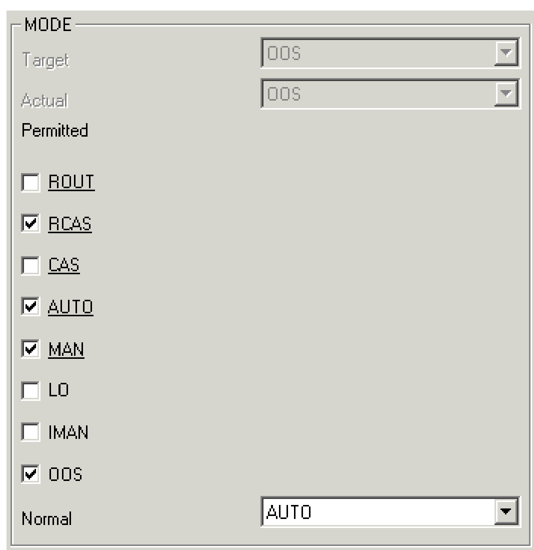
ATTENTION

The Execution Order in CM parameter defines the order of execution and publication for all blocks contained in the CM. For contained fieldbus blocks, this parameter combines with the CM's Execution Order in Link parameter to determine how the fieldbus block participates in the Link Active Schedule.

5. Leave the **Tag Description** field at the default of **CM101.AI_LEVEL** (CM name plus block name).

6. In the **Grant** and **Deny** fields, click the desired checkboxes to select the functions you want to Grant/Deny an operator.

7. Scroll down to the **Mode** section.



8. Click the desired checkboxes to select the modes you want to permit.

9. Click **OK** to clock the window and save the configuration.

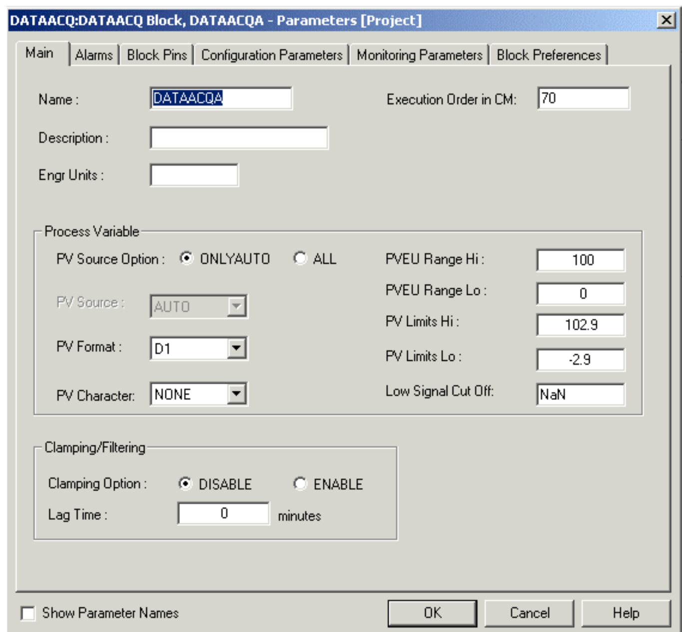
Configuring DATAACQ block for sample loop

1. In the CM101 chart, double-click the DATAACQ block.

TIP

The DATAACQ is a 1757-PLX52 function as well as a Foundation Fieldbus function, with ControlBuilder you can mix both in a control strategy.

The DATAACQ Block Parameters window opens.



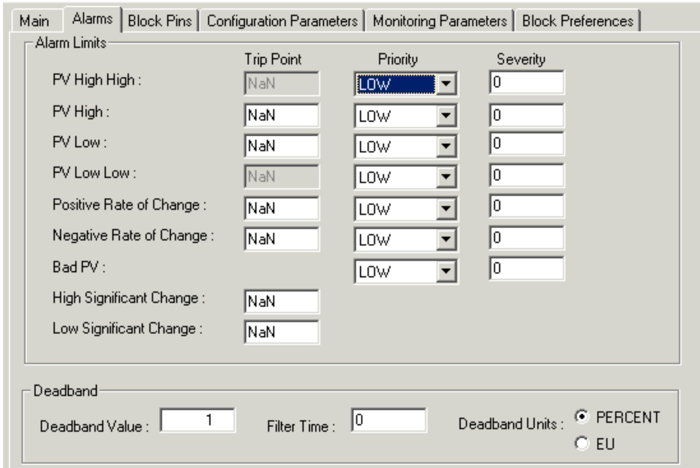
2. In the **Name** field, type **DATAACQ_101**.

3. In the **Description** field, type **Level Input Conditioning**.

4. In the **Eng. Units** field, type **Percent**.

5. Leave the **Execution Order in CM** field at the default of **20**.

6. Click the **Alarms** tab.

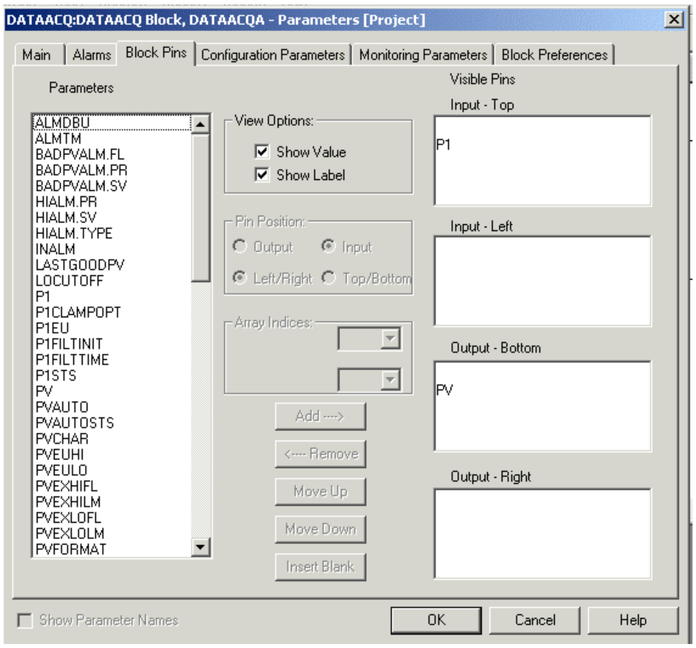


7. Enter or select the following values for these Alarm Limits:

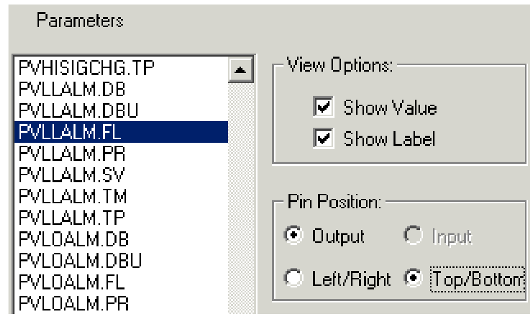
Alarm Limit	Trip Point	Priority	Severity
PV High High	85	URGENT	15
PV High	70	HIGH	8
PV Low	25	HIGH	8
PV Low Low	15	URGENT	15

Leave all other Alarm Limits at their default values.

8. Click the **Block Pins** tab.



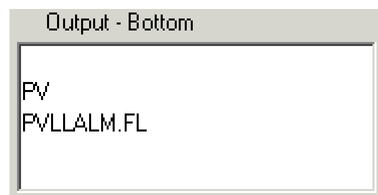
9. Select **PVLLALM.FL** in the Parameters list.



10. Verify that the Pin Position is Output and Top/Bottom.

11. Click **Add**.

PVLLALM.FL is added to the Output-Bottom list.



12. Select **PVLOALM.FL** in the Parameters list.

13. Verify that the Pin Position is Output and Top/Bottom.

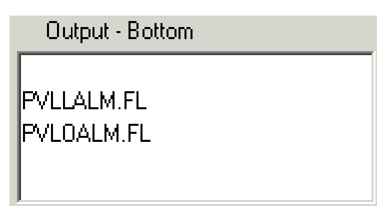
14. Click **Add**.

PVLOALM.FL is added to the Output -Bottom list.

15. Select **PV** in the Output-Bottom list.

16. Click **Remove**.

PV is removed from the Output-Bottom box.



17. Select **PV** in the Parameters list.

18. Select **Left/Right** in the Pin/Position box.

19. Click **Add**.

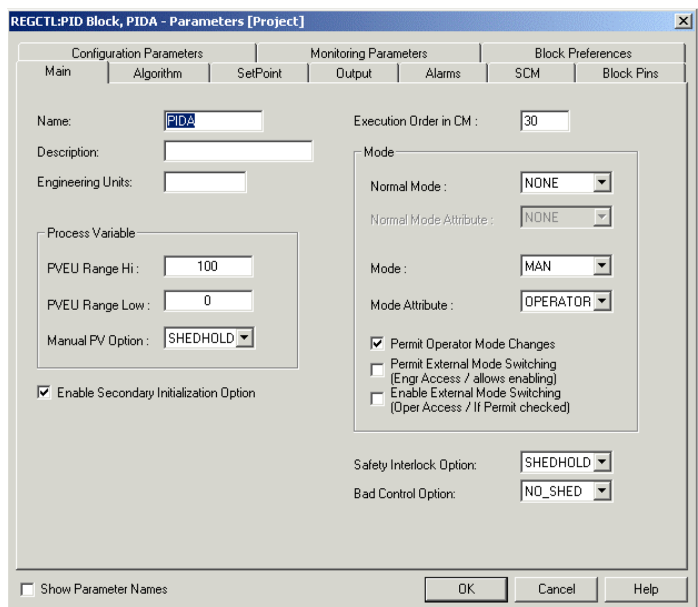
PV is added to the Output-Right box.

20. Click **OK** to close the window and save the configuration.

Configuring PID block for sample loop

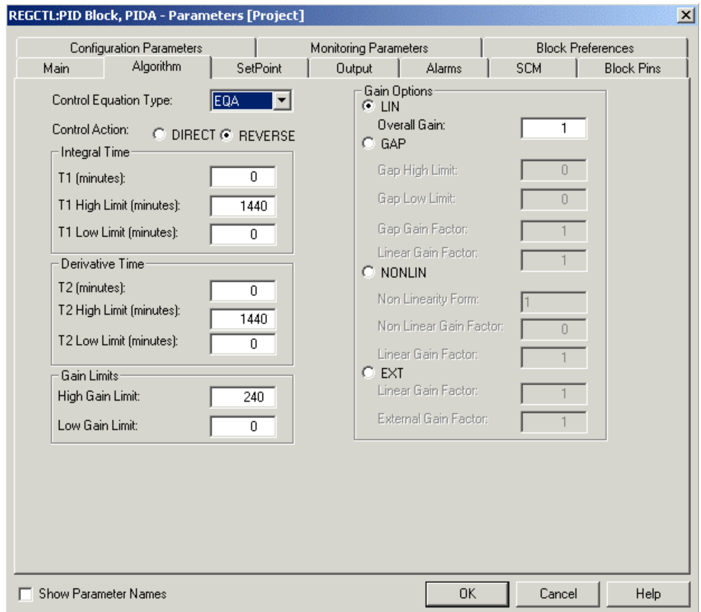
1. In the CM101 chart, double-click the PIDA block.

   The PID Block Parameters window opens.



2. In the **Name** field, enter **PID_101**.
3. In the **Description** field, enter **Recycle Loop Controller**.
4. In the **Engineering Units** field, enter **Percent**.
5. In the **Normal Mode** field, select **AUTO**.
6. In the **Normal Mode Attribute** field, select **NORMAL**.

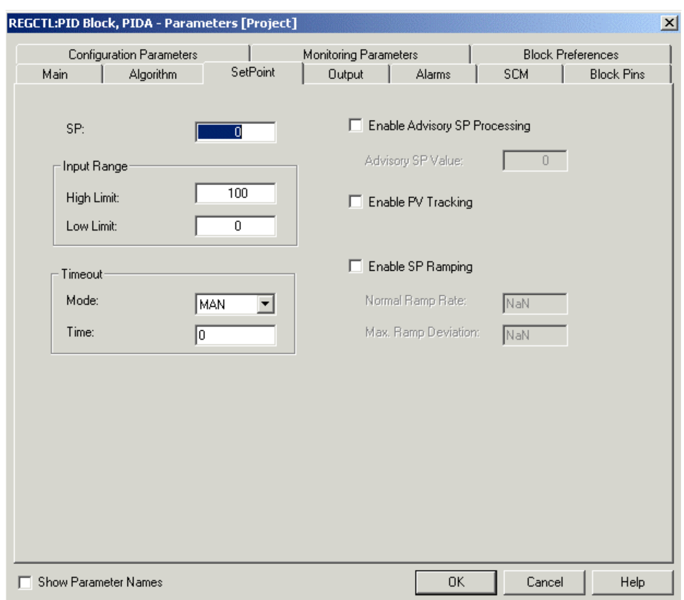
7. Click the **Algorithm** tab.



8. Leave the **Control Equation Type** field at the default of **EQA**.

9. In the **Control Action** field, select **DIRECT**.

10. Click the **Set Point** tab.



11. In the **SP** field, enter **50**.

12. Click **OK** to close the window and save the configuration.

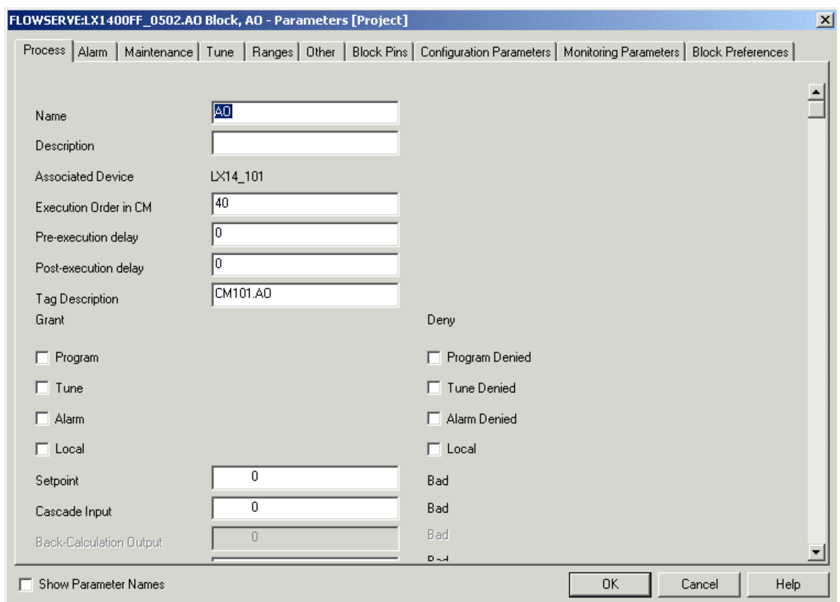
Configuring AO block for sample loop

TIP

This is a function block that was loaded using the Fieldbus Library Manager and the DD from the Flowserve Actuator.

1. In the CM101 chart, double-click the AO function block.

   The AO Block Parameters window opens.



2. In the **Name** field, enter **AO_FLOW**.

3. In the **Description** field, enter **Flow Control Output**.

4. Click **OK** to close the window and save the configuration.

Wiring blocks in CM101 for sample loop

1. In CM101 chart, double-click the OUT_VALUE pin on the AI_LEVEL block.

   The Pin is highlighted and cursor changes to cross-hairs +.

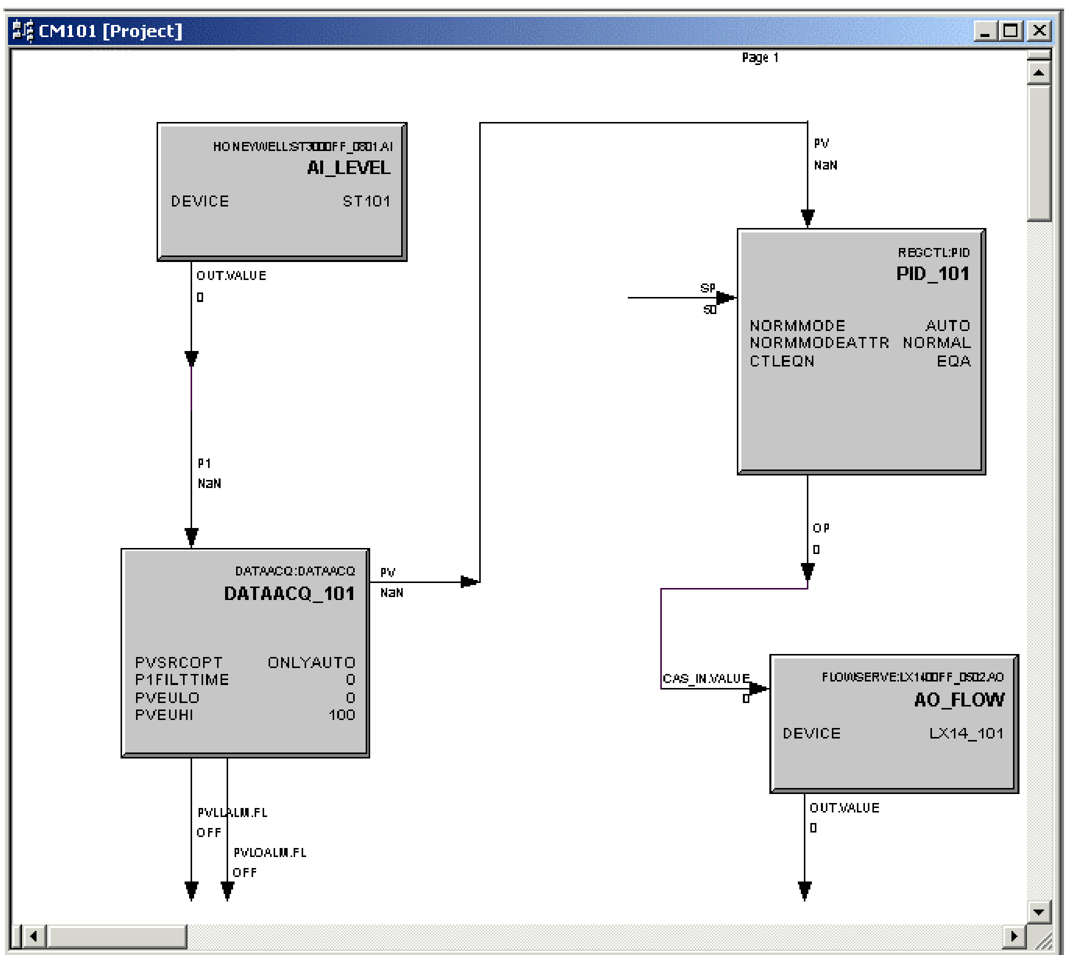
2. Move cursor over the P1 pin for the DATAACQ block and click.

   A wire is drawn between the pins, the P1 pin is highlighted, and the cursor reverts to its normal shape.

3. Repeat Steps 1 and 2 to wire the DATAACQ_101 PV pin to the PID_101 PV and the PID_101 OP to the AO_FLOW CAS_IN.VALUE. Be sure to add vertices where required by clicking in the desired path to the final pin.

   See Figure 4.3 for the completed CM101 with all blocks wired.

Figure 4.3 Completed CM101 for sample loop

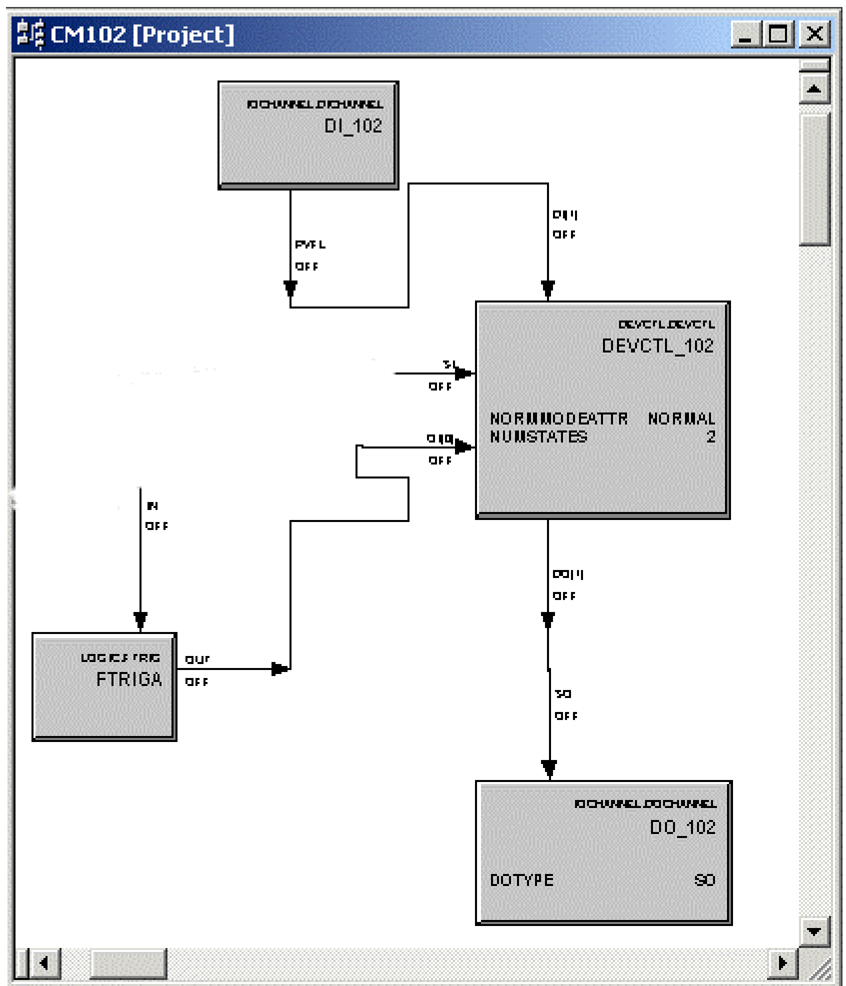


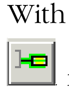
4. Close the CM101 chart and click **Yes** to save changes.

Adding parameter connectors for sample loop interlocks

This procedure assumes that a CM named CM102 will be created for the pump control loop in the example application. The CM102 will include a standard Device Control block and Discrete I/O Channel blocks to start and stop the pump as shown in Figure 4.4.

Figure 4.4 Sample CM with Device Control block for pump control in sample loop.



1. With CM102 chart open, click the parameter connector button  in the toolbar.

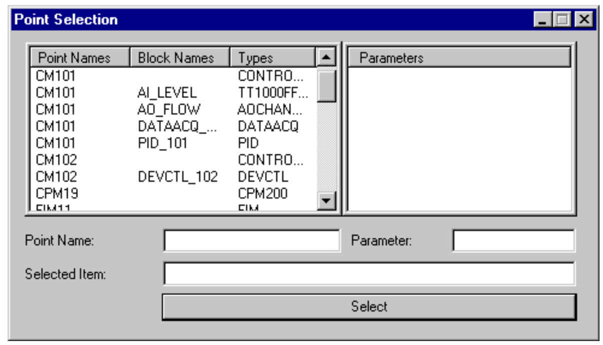
   The cursor changes to cross-hairs +.

2. Click SI pin on DEVCTL block. Double-click area adjacent to pin.

3. Parameter connection box appears.

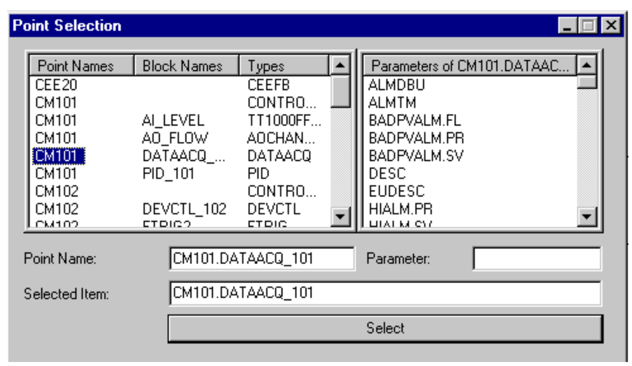
4. Click **Browse** .

The Point Selector window opens.



5. In the **Point Names** list, scroll to find **CM101 DATAACQ_101** and select it.

The Parameters list is populated with applicable parameters.

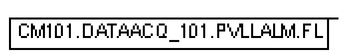


6. In the **Parameters** list, scroll to find **PVLLALM.FL** and select it.

PBLLALM.FL is inserted into both the Parameters field and the Selected Item field.

7. Click **Select**.

The parameter name is inserted into the parameter connection box.



8. Click the IN pin on the FTRIGA block. Double-click the area adjacent to the pin.

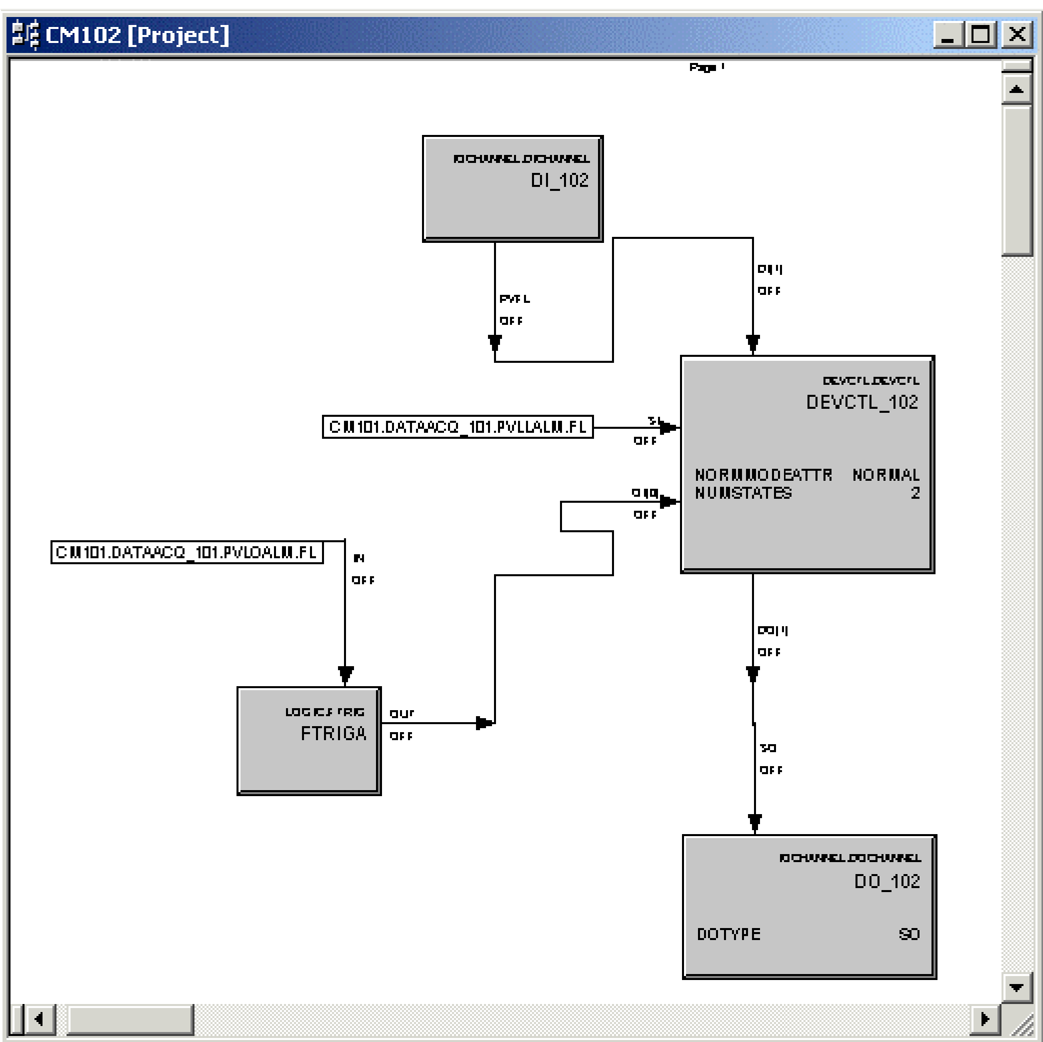
   The parameters connection box appears.

9. Repeat the previous steps to select DATAACQ_101.PVLOALM.FL as the parameter name.

10. Close the Point Selector dialog box.

    See Figure 4.5 for the completed CM102 with parameter connections.

Figure 4.5 Completed CM102 with parameter connections for sample loop interlocks.



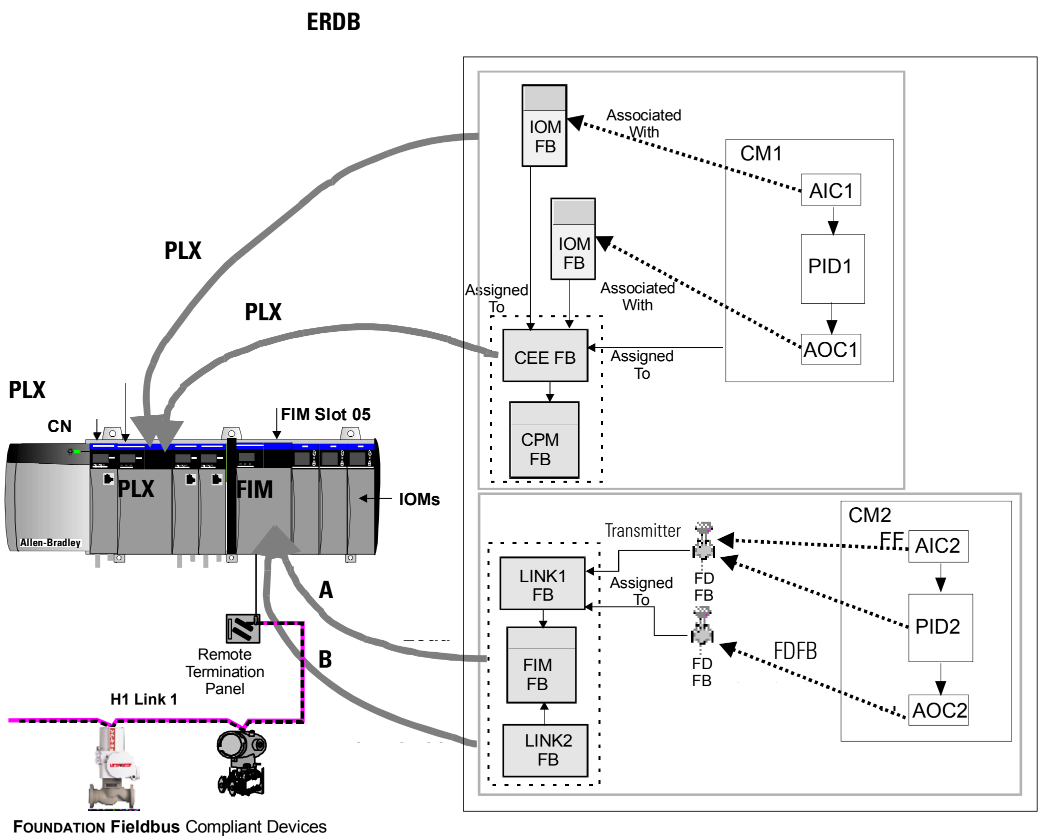
Loading Components Online

About load operations

The ProcessLogix system provides the ability to build Foundation Fieldbus control strategies offline, online or without being connected to the field components that will translate the strategy into the actual control operations. The process of transferring the Control Strategy to the “live” working components in the field is called the load operation.

The load operation functionally copies configuration data from the control strategy that is stored in the Engineering Repository Database (ERDB) to the assigned field component in the system architecture. In ProcessLogix R400.0, the load operation has been expanded to include fieldbus components. The load operation assures that the planned Foundation Fieldbus system matches the actual one. The communication addresses and physical location assignments specified for components through Control Builder configuration must match the actual addresses and locations of components in the system. Figure 4.6 is a simplified graphical representation of what happens during a load operation.

Figure 4.6 Overview of load operations used to initiate components online.



Notes:

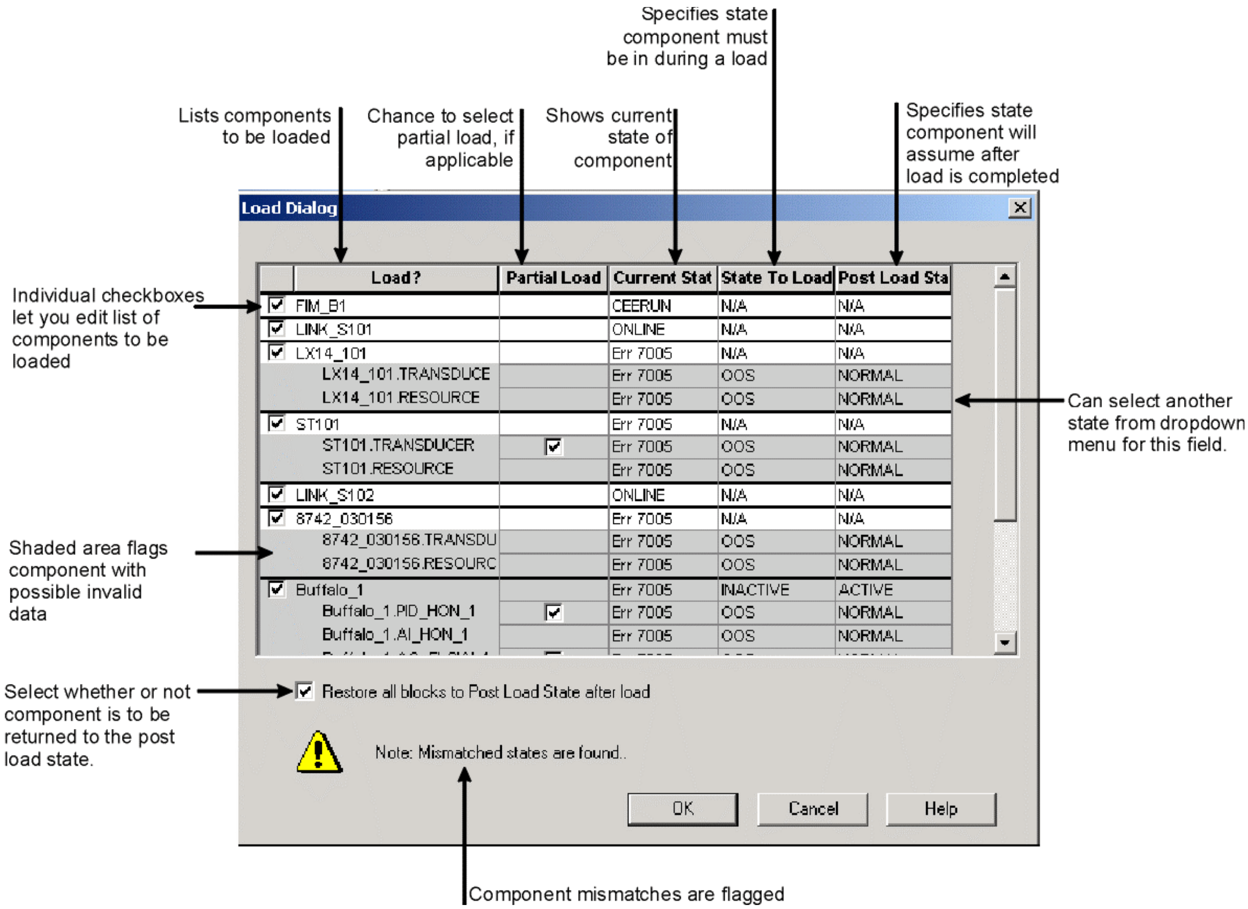
CM = Control Module
CN = ControlNet Module

FB = Function Block
FIM = Fieldbus Interface Module

About the new load dialog box

Figure 4.7 shows a sample Load Dialog box invoked for a load with contents operation for a FIM. It provides a brief description of the dialog box features for quick reference.

Figure 4.7 Descriptions of the FIM Dialog Box Features



General load considerations

The following are some general load considerations to keep in mind, when you are loading fieldbus components. In most cases, the load dialog box will quickly guide you through the load operations and will alert you to potential system problems.

Table 4.B

If you are loading...	Consider this...
A FIM.	Be sure ALIV does not appear in the LED display on the front of the FIM. If it does, you must first load the FIM's personality firmware using the NTOOLS utility supplied with ProcessLogix Engineering Tools.
A CM that contains only fieldbus function blocks.	We recommend that CMs containing only fieldbus function blocks be assigned and loaded only to the appropriate FIM LINK. They will only take up unnecessary memory and execution time in the CPM/CEE.
A CM that contains both ProcessLogix and fieldbus function blocks.	Be sure CMs that contain any ProcessLogix function blocks are assigned and loaded to a CEE. Control Builder enforces this and will also load the Foundation Fieldbus.
A fieldbus device through a FIM.	Be sure the device configuration in Project tab has the proper PD_TAG specified. While you can load a device to the link from Project, you can **not** load and commission a device until it is connected to the link and its PD_TAG and address agree with those specified in Project. Please see the next two sections *Fieldbus device states* and *Fieldbus device matching rules* for more information.

Fieldbus Device States

A fieldbus device is unaware of the steps being executed to configure a network. Its System Management Kernel (SMK) does sense the completeness of its configuration to determine what services it can provide. The following table shows the three major states a SMK in a field device must go through and the associated services for each state before a device can fully function on the network.

Table 4.C

SMK State	System Management Services
Un-initialized	SM_IDENTIFY SET_PD_TAG (Clear = False)
Initialized	SM_IDENTIFY SET_ADDRESS SET_PD_TAG (Clear = True)
Operational	SM_IDENTIFY CLEAR_ADDRES FIND_TAG_QUERY FIND_TAG_REPLY FB_START SMIB_Acess

Un-initialized State

In the **un-initialized** state, a fieldbus device has neither a physical device tag nor a node address assigned by a master configuration device. The only access to the device is through system management, which permits identifying the device and configuring the device with a physical device tag.

Initialized State

In the **initialized** state, a fieldbus device has a valid physical device tag, but no node address has been assigned. The device is ready to be attached to the network at a default system management node address. Only system management services for assigning a node address, clearing the physical device tag, and identifying the device are available.

Operational State

In the **operational** state, a fieldbus device has both a physical device tag and an assigned node address. Its application layer protocols are started to allow applications to communicate across the network. Additional network management configuration and application configuration may be needed for the device to become fully operational.

Fieldbus device matching rules

The FIM initiates the following matching rules depending upon whether or not the device identification (DEV_ID) is specified, when a device is loaded from Control Builder.

Table 4.D

If the Device ID is	And, the Device State Is	The matching rule is
Specified.	un-initialized	The device ID (DEV_ID) must match. If the desired physical device tag (PD_TAG) and node address (ADDR) are available, FIM will assign them to the device.
	Initialized	The device ID (DEV_ID) and physical device tag (PD_TAG) must match. If the desired address is not in use, FIM will assign it to the device.
	Operational	The device ID (DEV_ID), physical device tag (PD_TAG), and node address (ADDR) must match those specified through configuration in Control Builder. Any mismatch results in an error.
Not specified.	Initialized	The physical device tag (PD_TAG) must match. If the desired node address (ADDR) is not in use, FIM will assign it to the device.
	Operational	The physical device tag (PD_TAG) and node address (ADDR) must match those specified through Control Builder. Any mismatch results in an error.

Loading a FIM and its Links

The following procedure outlines the typical steps used to “load” a FIM through the Project tab in Control Builder. It assumes that the FIM and its associated RTP are installed and capable of communicating with the system.

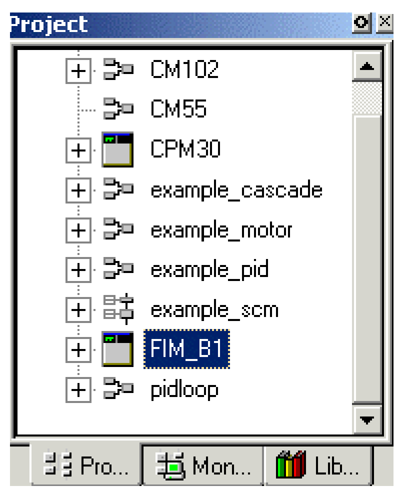
1. Verify that the OK LED on the front of the FIM is flashing and the numbers 1 and 2 are sequencing in the left-hand LED on the front panel display.

   This confirms that the designated FIM is operating and that the RTP and the H1 segment is OK.

TIP

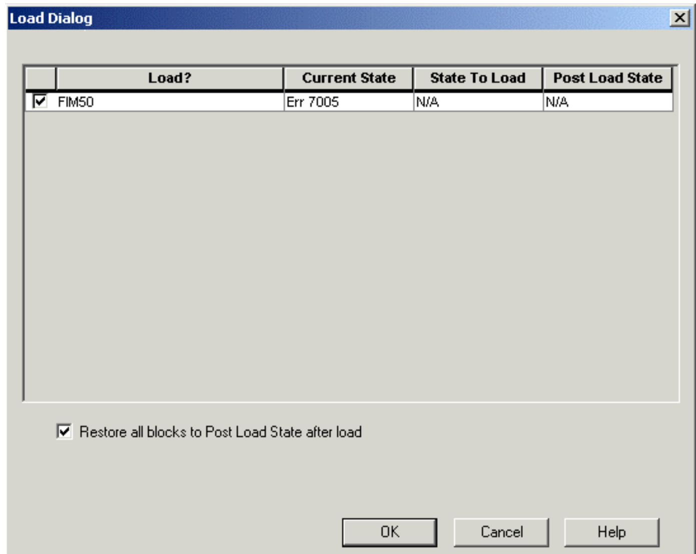
If ALIV appears in the FIM’s front panel display, you must load the FIM’s personality firmware before proceeding.

2. Click the FIM block icon in the Project tab.



3. Click **Tools ⇒Load**.

The Load Dialog window opens.



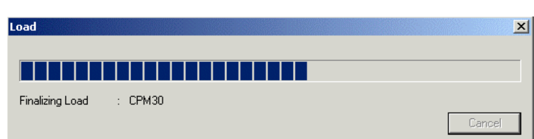
TIP The associated FIM Links are included with a FIM load even if the selected action is Load instead of Load with Contents. We suggest just loading the FIM without all of its contents first to be sure communications paths are working.

4. Verify that a check appears to the left of the FIM listed in the Load? column.

5. Click **OK**.

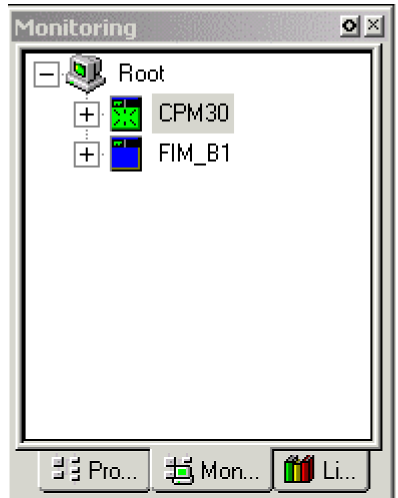
TIP Err7005 appears in the Current State column if no CPM/CEE blocks have been downloaded yet. This means that the parameter does not exist in the database. The State to Load and Post Load States are not applicable, since the FIM contains no data at this time.

The Load Progress window opens.



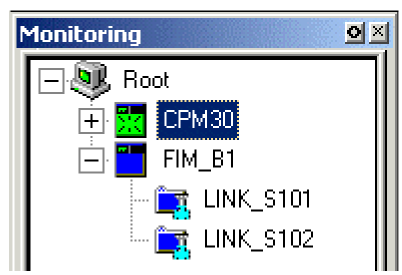
6. After the load completes, click the **Monitoring** tab.

   The Monitoring tab opens.



7. Verify that the FIM icon appears in Monitoring tab.

8. Click the [+] to expand the FIM.



9. Verify that the LINK icons appear under the FIM.

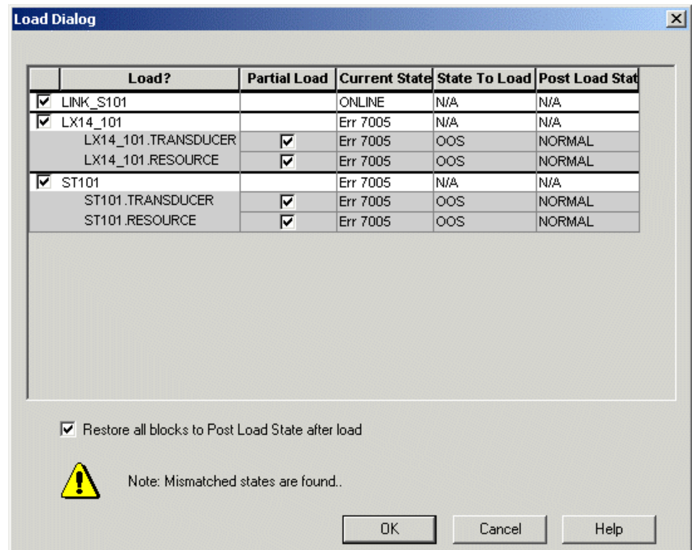
Loading Link contents or fieldbus device

Use the following procedure to load the Fieldbus link contents or fieldbus devices. It assumes that the FIM has been loaded and the fieldbus devices are installed and powered on the Links.

1. In Project tab, click the desired LINK icon.

2. Click **Tools ⇒Load With Contents**.

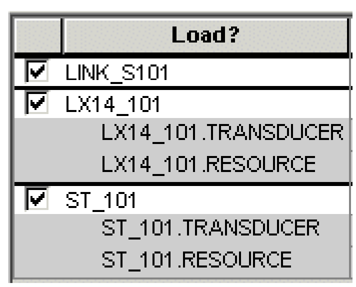
The Load window opens.



TIP

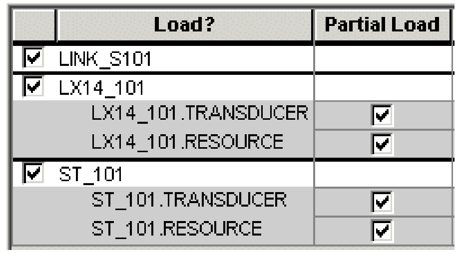
If you want to load just a Fieldbus device, click the device icon and click Tools->Load. Use LINK/Load With Contents so you can load more than one device at a time.

3. Verify that checkmarks appear in the LINK and Device checkboxes listed in the Load? column.



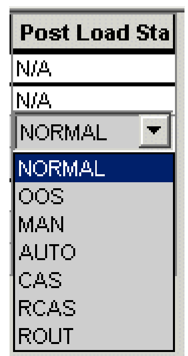
If you do not want to load a given device, just click it to remove the checkmark from its checkbox.

4. To load all, leave the Partial Load checkboxes selected for all blocks.



5. To change the Post Load State for selected components, click the appropriate Post Load State row.

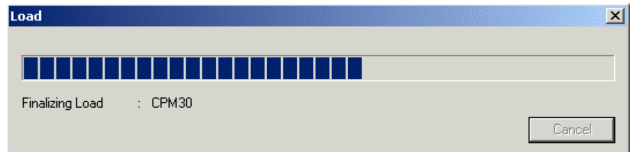
   The dropdown menu opens.



6. Select the appropriate state.

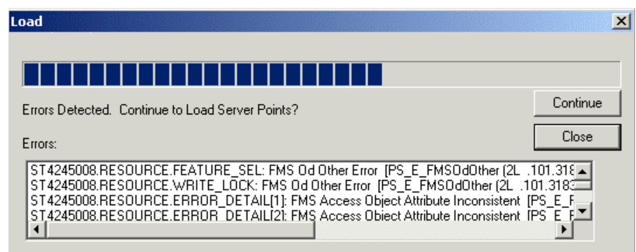
7. Click **OK**.

   The load is initiated, shown by the Load window.

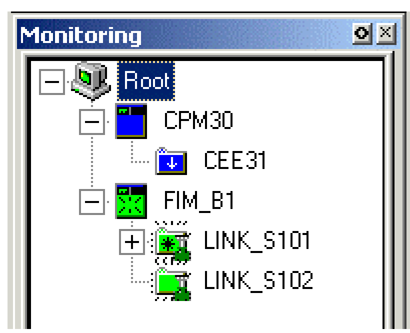


TIP

If errors are detected, they will be listed and you will prompted to select whether or not you want to continue the load with errors. It is a good idea to note the errors and abort the load (close), so you can go back and correct the errors before completing the load.



8. After the load is complete, click the **Monitoring** tab.



9. Verify that the Link is activated.

10. Repeat Steps 1 to 9 to load another Link or device.

Summary

The Control Builder for the R400 ProcessLogix system lets you add and configure fieldbus components as a natural and intuitive extension to the existing ProcessLogix components. This promotes a seamless integration of the fieldbus functions with existing ProcessLogix capabilities.

Operating the 1757-FIM

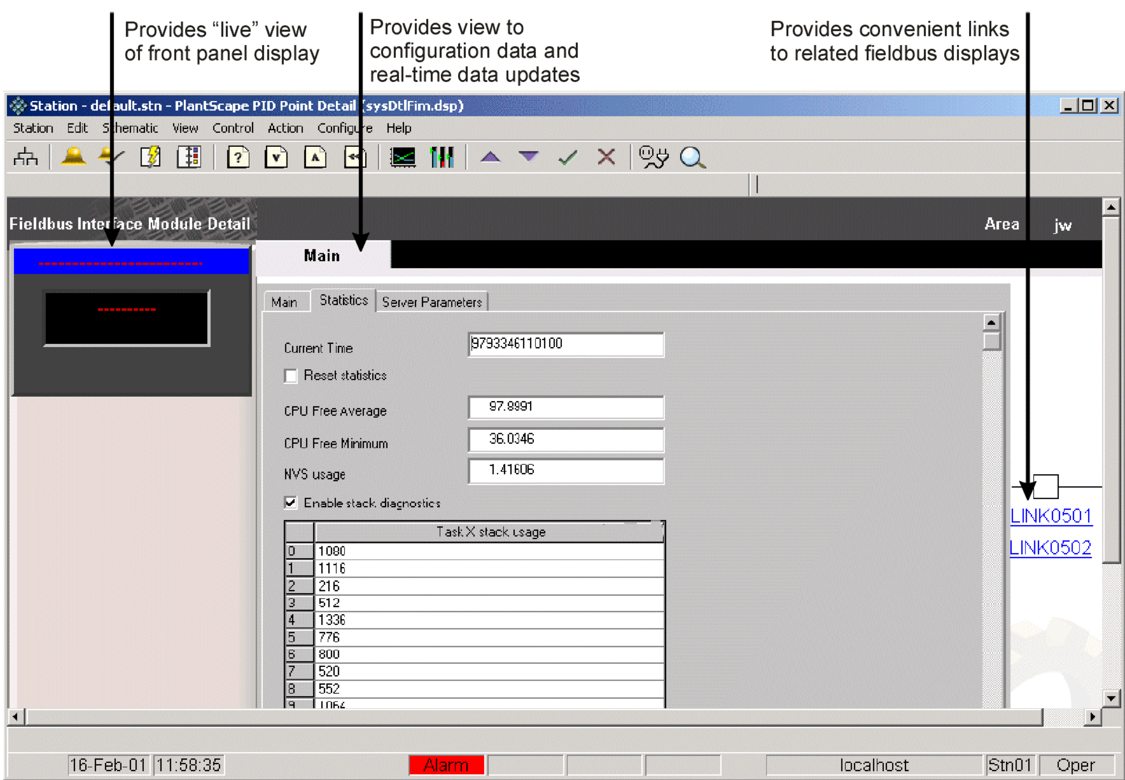
Monitoring Fieldbus Functions Through Station Displays

Using Station Detail displays

The ProcessLogix Server Station application includes pre-configured Detail displays for the FIM, Link, device and fieldbus function blocks. These displays are the default entries for the Point Detail Page parameter on the Server Parameters tab of the configuration form. Once you establish communications with a fieldbus H1 link you can begin monitoring the status of any component that has been loaded as part of a Control Strategy to a FIM with points registered in the ProcessLogix Server. The Detail displays let you quickly view the component's current state, fault status, and pertinent configuration data.

Figure 5.1 shows a sample FIM Detail display. The fieldbus Detail displays feature links to related fieldbus component displays.

Figure 5.1 Typical FIM Detail display in Station.

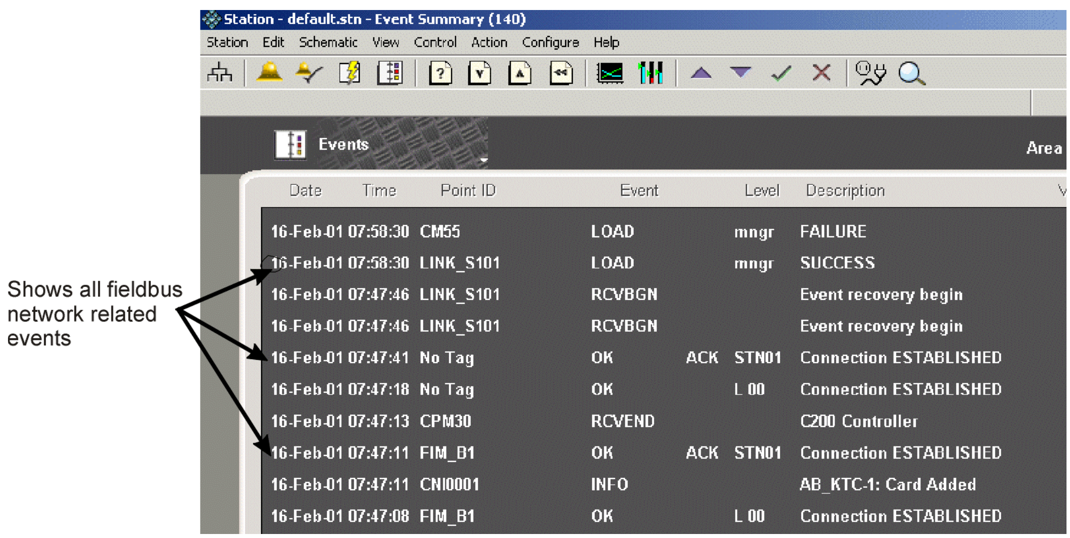


See the *Operator's Guide* in *Knowledge Builder* for detailed information about calling up, navigating, and viewing Station displays.

Using Station Event Summary display

Like the Detail displays, the Alarm and Event Summary displays support the integration of fieldbus generated notifications and events. It is integrated with ProcessLogix component data and is for the most part self-explanatory. Figure 5.2 shows a typical Event Summary display that includes FIM, Link, and Device indications. Use this display to get a quick review of recent actions that have been initiated within the system.

Figure 5.2 Event Summary display includes fieldbus related details.



Monitoring Fieldbus Functions Through Monitoring Tab

Inactivating/Activating a Link

Use the following procedure to inactivate/activate a fieldbus Link through the Monitoring tab in Control Builder.

ATTENTION

Inactivating a fieldbus Link essentially shuts downs the Link and the FIM, if both Links are inactivated, and interrupts the transfer of data to the ProcessLogix system. Be sure you system can tolerate the lost of live data, while the link is inactive.

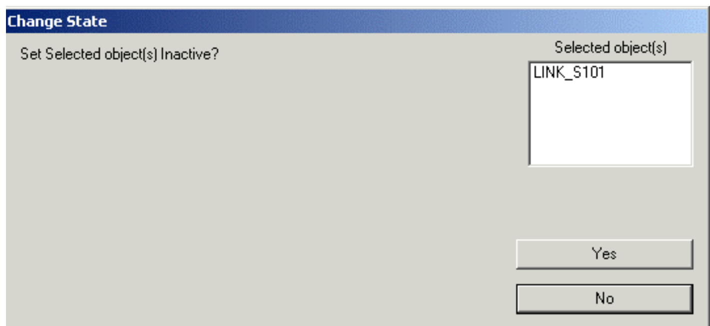
TIP

You can initiate this same function through the corresponding Detail display in Station.

1. In the Monitoring tab, click the LINK icon.

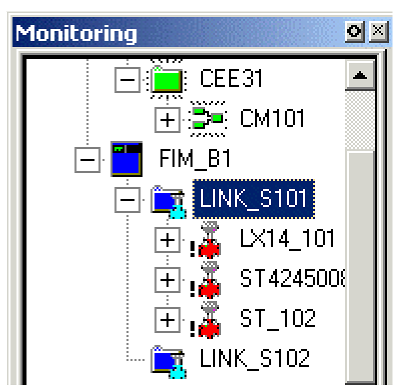
2. Click Toggle State  in the toolbar.

   The Change State window opens.



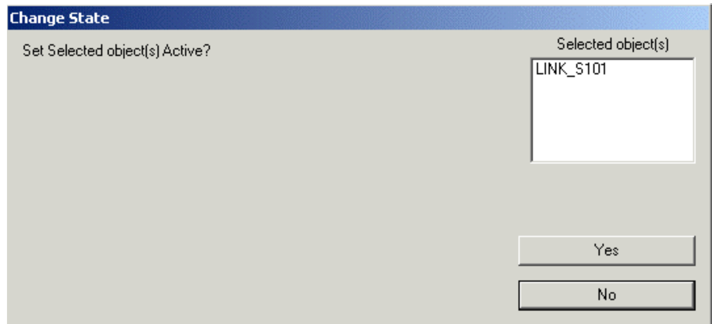
3. Click **Yes**.

   This initiates the state change. The icon changes from green to blue. The FIM is only inactivated when both Links are inactivated (shown below).



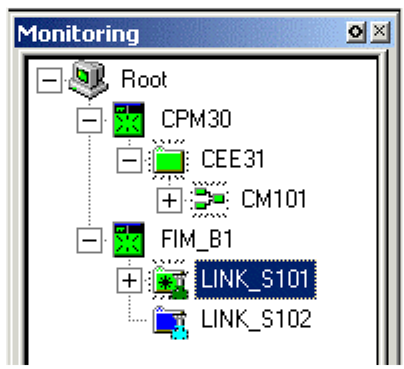
4. With LINK icon selected, click the Toggle State button  in the toolbar.

   The Change State window opens.



5. Click **Yes**.

   This initiates the state change. The icon changes from blue to green.



Monitoring/Interacting with given component/block

Once you download a FIM and its contents, you can use the Monitoring tab to interact with the components including the function blocks in the fieldbus devices.

1. Double-click the desired component/block icon in the Link/Device tree menus under the FIM icon to call up the associated Configuration form.

2. Click the given tab to view the current status of the related fieldbus data.

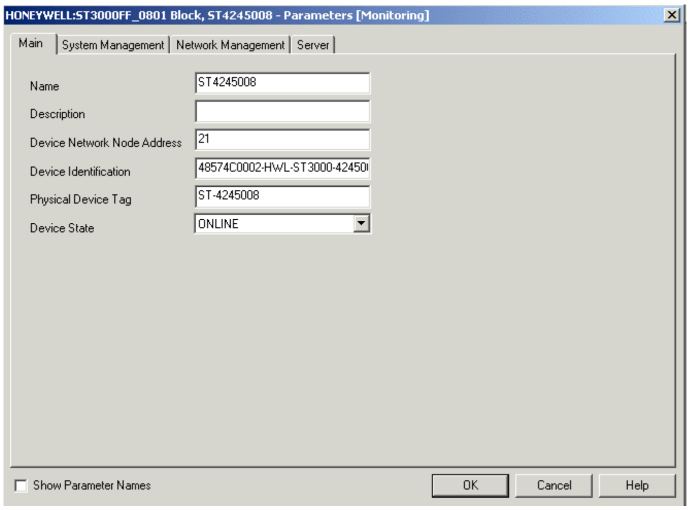
   Both the FIM and LINK function blocks must be active to view on-line data. The data can be viewed either by name or parameter reference by **not** checking or checking the Show Parameter Names checkbox at the bottom of each tab. See the *On-Line Monitoring Using Control Builder* section in Knowledge Builder.

Checking fieldbus device functional class

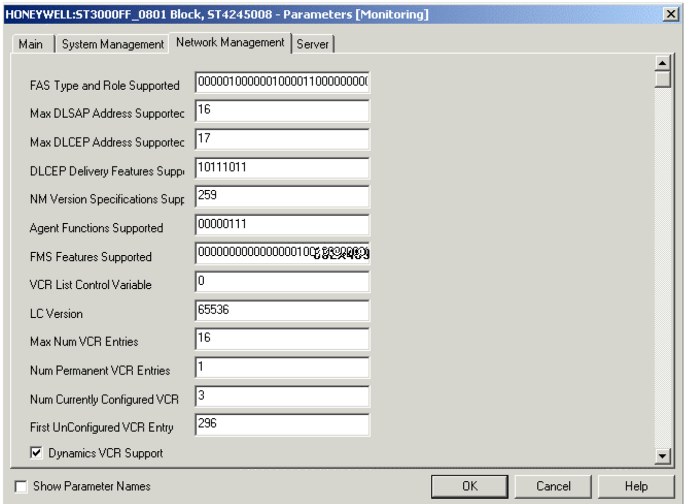
Use the following procedure to check and change, if necessary, the functional class of the fieldbus device.

1. Double-click the device icon.

   The Device Configuration Parameters window opens.



2. Click the **Network Management** tab.



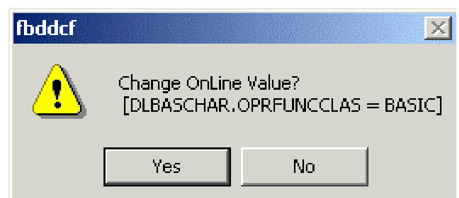
3. Scroll to find the **DL Operational Functional Class** field.

4. Verify that the functional class is appropriate for the device, Basic or Linkmaster.

TIP The FIM is the primary Linkmaster for both Links. If you designate a device as a "backup" Linkmaster, be sure it has the capacity to handle the Link Active Schedule. Otherwise, you may have to reset the device and restart it to restore operation if the LAS is too large for it to handle.

5. If the functional class is not correct, select the appropriate functional class.

   The Change Online Value? prompt opens.

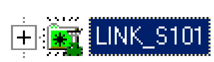


6. Click **Yes** to acknowledge the change.

7. Click **OK** to close the window.

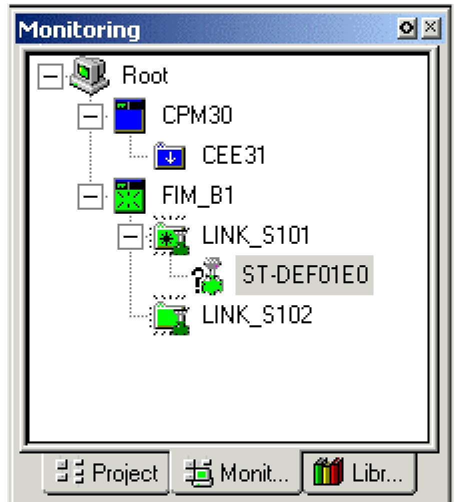
Checking live list and interacting with uncommissioned devices

Use the following procedure to check the link for devices that are added to the Live List as uncommissioned.

1. Verify that the Link icon has an asterisk 
.

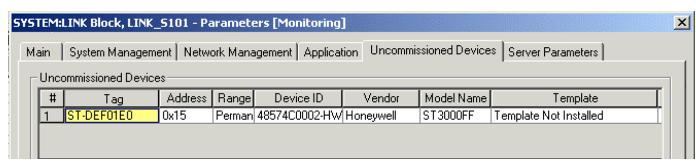
   The asterisk signals that an uncommissioned device has been added to the link.

2. Click the [+] to expand the Link.



3. Double-click the uncommissioned device icon.

The LINK Parameters window opens.



4. Review the device details.

TIP

The device manufacturer assigns tag and device IDs. It is a good idea to record the device's Tag name and Device ID for future reference.

5. If a template exists, go to the procedures in Chapter 4 to create, assign, configure, and load the device.

   If a template does not exist, create one using the vendor's DD files and the Fieldbus Library Manager, as described in the Chapter 4. A template must exist before a device can be integrated into the system.

TIP

It is possible that the uncommissioned device may have an earlier version of the vendor's software (Device Rev). In this case, the device would appear to have no matching template even if one exists for the same device but it is for a different software version. If the vendor supports flash upgrade of its device's firmware, you can upgrade the device's firmware through this tab. Please see the next Section *Maintenance, Checkout and Calibration* for more information.

A new Link Active Schedule (LAS) must be loaded to the FIM and other LinkMaster devices anytime there is a change that invalidates the active LAS, such as adding a device, removing a device, or making a configuration change.

6. Click **OK** to close the window.

Using the Tools Menu Functions

Many of the Tools menu functions designed for use with the Controller and CEE components also apply for the FIM and LINKs components. These include Upload, Update, and Snapshot functions. These functions are useful for correcting mismatches that may occur between the components and the database. Please refer to the Control Building Guide in Knowledge Builder for details about a given menu function.

1757-FIM General Maintenance, Checkout, and Calibration

Adding, Removing and Replacing Components

About Removal and Insertion Under Power

WARNING

When you insert or remove the module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Be sure that power is removed or the area is nonhazardous before proceeding.

IMPORTANT

This is a general document on how to use ProcessLogix tools and that full maintenance should use live analysts. This document is not intended to replace Foundation Fieldbus or the device Manufacturers installation and training documents.

Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

You can remove and insert the FIM without removing power. The removal of the FIM results in the loss of communications with both H1 fieldbus links.

The removal of an individual fieldbus device only breaks communications with that device.

General Procedure

ATTENTION

We recommend that you proceed with **extreme caution** whenever replacing any component in a control system. Be sure the system is offline or in a safe operating mode.

Component replacements may also require corresponding changes in the control strategy configuration through Control Builder, as well as downloading appropriate data to the replaced component.

The direct replacement of a FIM of the same kind is just a matter of disconnecting the RTP cable, removing the existing component, installing a new one in its place, and connecting the RTP cable. You may also have to load its “personality” image firmware before it can become fully operational.

If you are adding a fieldbus device, follow the installation instructions for the component and then configure it through Control Builder to integrate it with your control strategy.

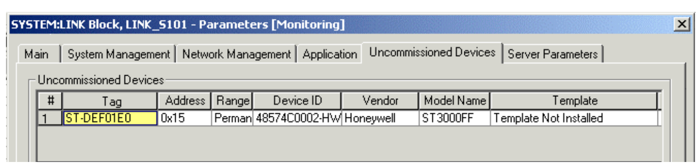
If you are removing and/or replacing a fieldbus device, proceed with **extreme caution**. You must delete, restore, and/or create all hardware connections and the control strategy database configuration through the Control Builder. You may also have to create a new template for the device and/or upgrade the device’s firmware.

Upgrading firmware in an uncommissioned device

Use the following procedure to upgrade the firmware in an uncommissioned device through Control Builder. This procedure assumes that you have the vendor supplied upgrade file for the device. It also assumes that you are monitoring the FIM/LINK through the Monitoring tab in Control Builder.

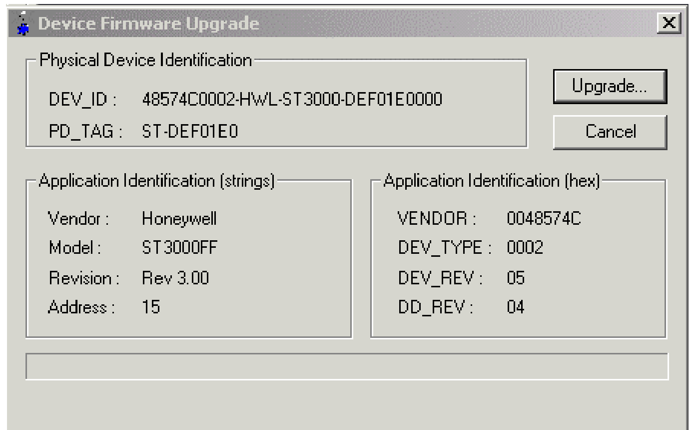
1. Double-click the uncommissioned device icon .

   The LINK Parameters window opens.



2. Click **Load Firmware**.

   The Device Firmware Upgrade window opens.



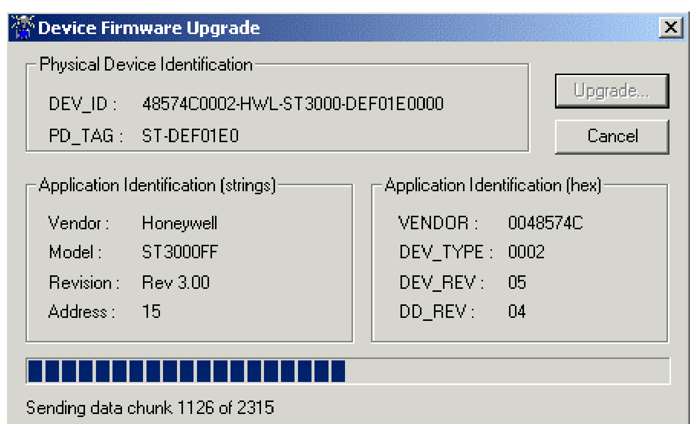
3. Click **Upgrade**.

   The Open window opens.

4. Locate and select the vendor's upgrade file.

5. Click **Open**.

   This initiates the firmware upgrade.

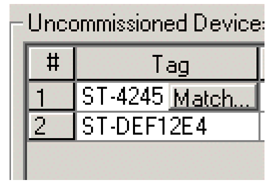


6. After the firmware upgrade is complete, wait for the device to rejoin the network. This may take up to 3 minutes.

7. When the device has rejoined the network, click the  close the window.

8. Verify that there is an existing template listed in the **Templates** column that matches the upgraded version of the device.

9. To automatically match the template, click in the Tag field to expose the Match button. Click it to initiate enhanced matching functions.



10. If a template exists, use The procedures in Chapter 4 to include the device in the Control Strategy.

    If a template does not exist, get the DD file for the device and use the Fieldbus Library Manager to create one. Refer to Making a Fieldbus Device Template from a Vendor's DD on page 4-17.

11. Include the fieldbus device in the control strategy and the Link Active Schedule.

    This completes the firmware upgrade.

12. Click **OK**.

    The LINK Parameters window closes.

Interpreting Component LED Indications

FIM LED indications

As shown in Figure 6.1, the FIM has one four-character display and three two-color LEDs on its front panel. From left to right, the LEDs provide Link 1 status, Link 2 status, and module health status, respectively. The following table summarizes some typical indications for reference.

Figure 6.1 FIM front panel indicators. Table 1 FIM LED Interpretations

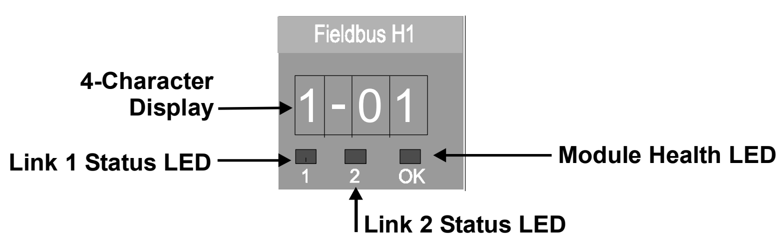


Table 6.A LED Definitions

If Module Health LED is:	And, 4-Character Display shows:	The FIM is:
Flashing Red/Green	TEST	Running its self-test.
Flashing Green/Off	BOOT	Initiating its startup or boot sequence.
	ALIV	in its ALIVE state and ready for its Personality Image load.
	LOAD	In the midst of a firmware load sequence.
	RDY	In its READY state and ready for its Boot image load.
Solid or Flashing Green	Alternating link states as noted in the following rows.	Operating normally. If solid green, there are ProcessLogix Control Data Access (CDA) connections to the FIM.
If Link 1 or Link 2 LED is:	**And, 4-Character Display shows:(1)**	**Then, LINK is:**
Off or Flashing Red	X- - -	Down/offline.
Flashing Green	X – FW	Loading firmware.
Flashing Red	X – FE	Having a firmware load error.
	X – CE	Having a communications initialization error.
	X – DE	Having a database initialization error.
Flashing Green or Off	X – CI	Initializing communications with fieldbus devices.
	X – DI	Initializing database.
	X – YY	Communicating with fieldbus devices.

(1) Display alternates between Link 1 and Link 2 in two second intervals. The X is either 1 or 2 for the respective Link and YY equals the number of fieldbus devices present on a given Link.

Checking Fieldbus Device Calibration

Please refer to the manufacturer's documentation for the fieldbus device to determine the recommended calibration schedules and procedures. The Other tab on the Parameters form for a device's transducer block provides pertinent calibration information, when accessed through the Monitoring tab in Control Builder.

Using the 1788-CN2FF, ControlNet-to-FOUNDATION Fieldbus H1 Linking Device

This chapter describes:

- the blocks in the linking device
- configuring the linking device to access the AI, AO, DI, DO function blocks on the Fieldbus network from ControlNet
- attributes of the created ControlNet objects, assembly objects, alarm handling
- ControlNet connection details

Blocks in the Linking Device

The linking device is similar to an I/O subsystem. An I/O subsystem typically contains several I/O modules. Each module has a number of channels. The channels perform either analog input, analog output, discrete input, or discrete output functions. The linking device models the I/O modules in software. The linking device has four types of function blocks:

- Multiple Analog Input (MAI)
- Multiple Analog Output (MAO)
- Multiple Discrete Input (MDI)
- Multiple Discrete Output (MDO)

Each of these function blocks has eight channels. Each channel is a combination of a value and Fieldbus status.

You can connect an analog input function block on the Fieldbus network to a channel on the MAI block, just as you would wire a 4-20ma analog input to an analog channel in your I/O subsystem. The main difference is that the Fieldbus function blocks do considerable processing. The channel value represents a scaled value in engineering units. Each value has an associated status. The status is more than a boolean of good or bad. It has four major status values of Good or Bad, with 16 sub-status values for each major status, and four limits.

Analog Inputs

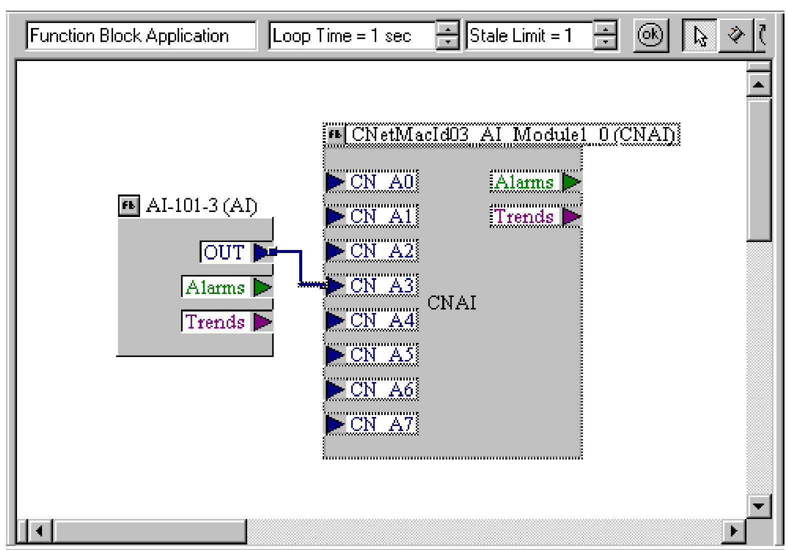
This section describes configuration of the linking device to access any analog value (and status) in a Fieldbus device. It also describes the attributes of the created ControlNet analog input object.

The linking device contains two of MAI block instances on each Fieldbus channels. Each instance of the MAI block is the software equivalent of an analog input module in an I/O subsystem. Each MAI block has eight channels, with each channel containing a float and a status pair. The linking device assigns a tag to each MAI block in the form CNetMacId*xx*_AI_Module*i-j,* where *xx* is the ControlNet network address, *i* is the Fieldbus channel number, and *j* is the module or instance number.

Configuration of Analog Inputs

When the NI-FBUS Configurator is started, and continuously thereafter, it lists all the devices and the function blocks in each device in its browse window. This includes the AI function blocks in the Fieldbus devices and the MAI function blocks in the linking device. You must connect the Fieldbus AI function blocks that will be accessed by ControlNet controllers (or devices) to the MAI channels, as shown in Figure 7.1.

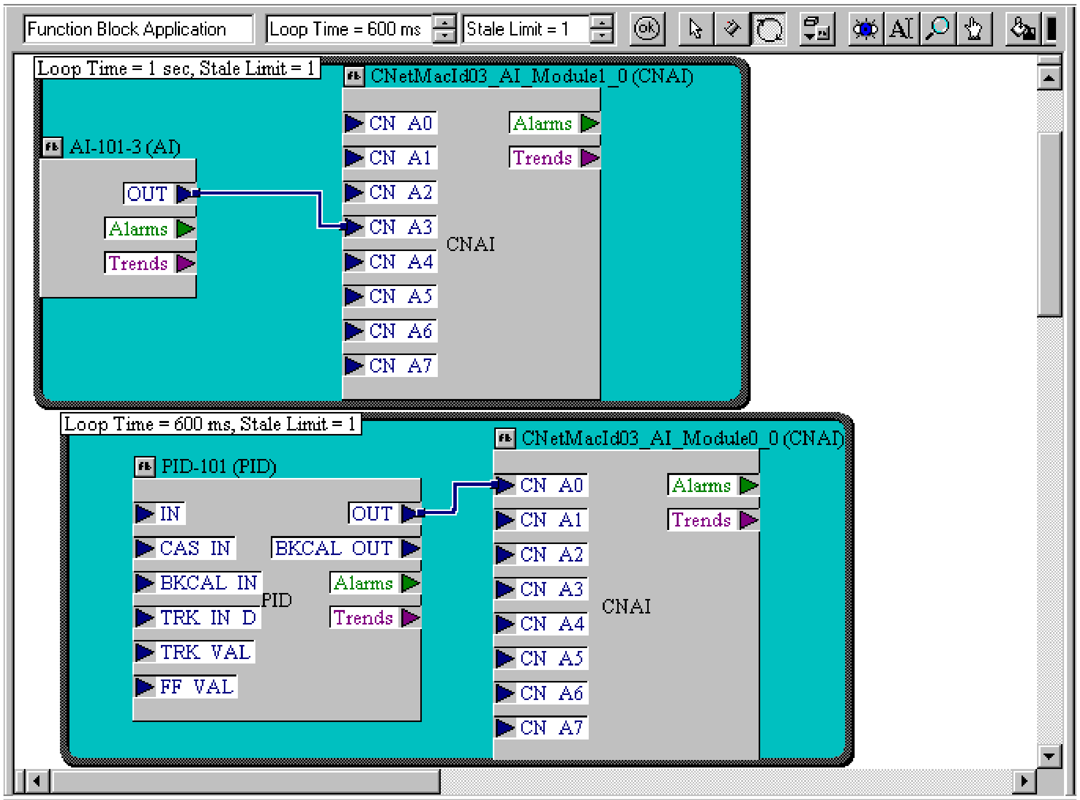
Figure 7.1 Sample Single Macrocycle MAI Configuration



The NI-FBUS Configurator sets up the analog input devices to publish their data on Fieldbus. The linking device MAI block subscribes to these values and produces them on ControlNet. Each MAI block has its own schedule of subscribing on Fieldbus and producing on ControlNet. For different rates (macrocycle) of production on

ControlNet, you must choose different MAI blocks as shown in the Figure 7.2.

Figure 7.2 Sample Multiple Macrocycle MAI Configuration



You do not have to connect to all the channels in an MAI block instance before using another instance. You do not have to use the channels in order. That is, you may use channels CN_A0 and CN_A5 and not use any of the others.

As shown in the lower loop of Figure 7.2, you can connect any parameter that is a float value and status combination to the MAI block. In other words, you can bring values from any function block to the PLC processor. The MAI block is not limited to interfacing with AI function blocks.

ControlNet Analog Input Objects

The linking device creates an instance of a ControlNet AI object for every wired channel in the MAI block instances. Each instance of the ControlNet AI object has the attributes and access rules shown in Table 7.A.

Table 7.A ControlNet Analog Input Object Attributes

Attribute Number	Name	Type	Access	Initial Value	Remarks
3	Value	Float	Get	0	
4	CNStatus	Boolean	Get	Bad	ControlNet status. 0-Good; 1-Bad.
107	LoFlag	uint8	Get/Set	0	See Alarm Handling for Analog Inputs
108	HiFlag	uint8	Get/Set	0	See Alarm Handling for Analog Inputs
109	LoLoFlag	uint8	Get/Set	0	See Alarm Handling for Analog Inputs
110	HiHiFlag	uint8	Get/Set	0	See Alarm Handling for Analog Inputs
150	FFstatus	uint8	Get	Bad:Out of Service	Fieldbus status of the value; obtained from the Fieldbus device.
151	TagDesc	String	Get	As configured	Tag of the Fieldbus function block that is represented by this object instance.

All attributes in Table 7.A are created if a standard Fieldbus AI function block is connected to a channel of an MAI block. Otherwise, only Value, CNStatus, TagDesc, and FFstatus are created.

Alarm Handling for Analog Inputs

You can configure Fieldbus AI function blocks to detect and report alarms.

Four process alarms (HI_HI_ALM, HI_ALM, LO_LO_ALM, and LO_ALM) are exposed through the ControlNet AI object. When you connect an AI function block to an MAI block, the NI-FBUS Configurator configures the field device to send the four alarms to the linking device.

In Fieldbus terminology, confirming an alarm means that the alarm has been received by an operator, and acknowledging an alarm means that the operator has taken the necessary action.

When an alarm condition is detected by a Fieldbus function block, the alarm is said to be ACTIVE. An active alarm should be Acknowledged and Confirmed after the condition causing the alarm is corrected. When the function block detects the alarm condition is no longer present, the alarm is said to be CLEARED. The CLEARED state must be Confirmed when it is detected. Users of the LD use the alarm attributes to process Fieldbus alarms.

Each attribute corresponding to a process alarm has the three alarm-related bits as shown in Figure 7.3.

Figure 7.3 Alarm Attribute Definition

Bit Position	7	6	5	4	3	2	1	0
Value	X	X	X	X	X	Acknowledge	CLEARED	ACTIVE

Bits 7-3, DON'T CARE, are undefined when read and should be ignored when performing a GET. The DON'T CARE bits should be written as 0 when performing a SET. Bit 2, Acknowledge, is undefined when read and should be ignored when performing a GET. When a user wants to Acknowledge an alarm, this bit should be written as a 1 when performing a SET. Each time the LD detects a 1 in the Acknowledge position, a Fieldbus Acknowledge will be generated. Therefore, users must be careful not to set the Acknowledge bit more than once per alarm. Bits 1 and 0, CLEARED and ACTIVE respectively, are read/write bits.

When an alarm condition is ACTIVE or CLEARED, these bits are read as a 1 when performing a GET. The user must Confirm each of these states by writing them to a 0 by a SET. Once an alarm has occurred (become ACTIVE), alarm processing is NOT complete until the user intervenes to remove the alarm condition, Acknowledges the alarm (writing a 1 in the Acknowledge bit position), Confirms the alarm (writing a 0 to the ACTIVE bit position once it is read as a 1), and Confirms the alarm condition being CLEARED (writing a 0 to the CLEARED bit position once it is read as a 1)

Analog Outputs

This section describes configuration of the linking device to control any analog value (and status) in a Fieldbus device, such as in a Analog Output (AO) function block. It also describes the attributes of the created ControlNet analog output object.

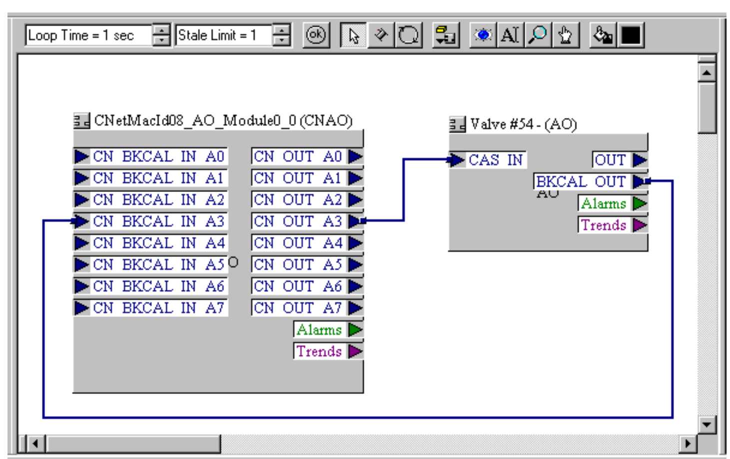
The linking device contains two of MAO block instances on each Fieldbus channels. Each instance of the MAO block is the software equivalent of an analog output module in an I/O subsystem. Each MAO block also has eight channels or outputs, with each channel containing a float and a status pair. Each MAO block has eight inputs to provide for the BKCAL or readback from the analog output function blocks. The linking device assigns a tag to each MAO block in the form CNetMacId*xx*_AO_Module*i-j*, where *xx* is the ControlNet network address, *i* is the Fieldbus channel number, and *j* is the module or instance number.

Configuration of Analog Outputs

The NI-FBUS Configurator lists all the devices and the function blocks in each device in its browse window. This includes the AO function blocks in the Fieldbus devices and the MAO function blocks in the linking device. You must connect the Fieldbus AO function blocks that will be controlled by the ControlNet controllers (or devices) to the MAO channels.

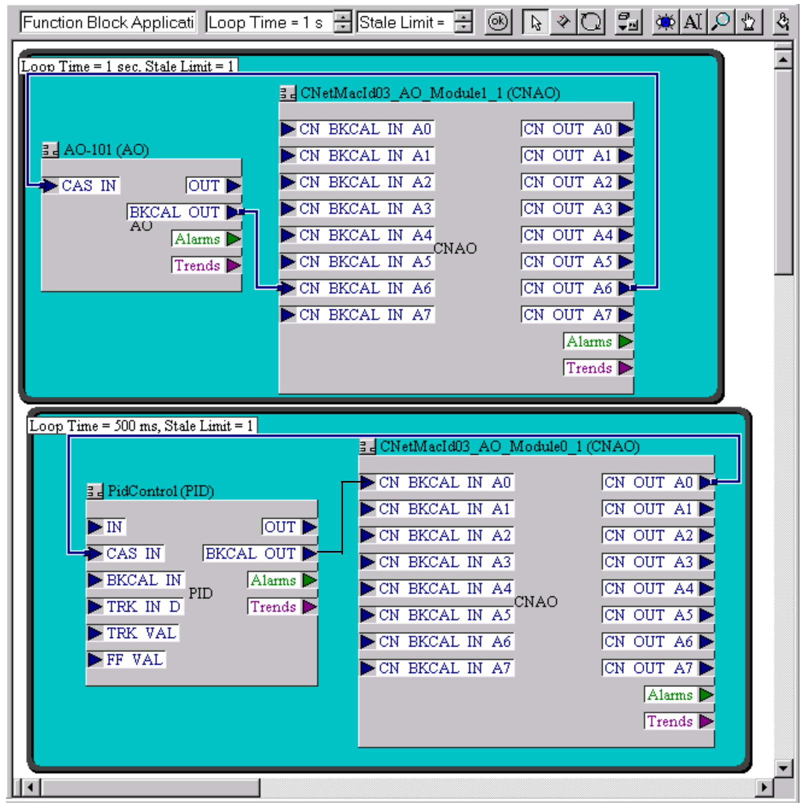
The controller provides the necessary mode handshake for cascade initialization of the AO block. In this case, you must connect CN_OUT_A*x* of the MAO block to CAS_IN of the AO block, and BKCAL_OUT of the AO block to BKCAL_IN_A*x* of the MAO block, as shown in Figure 7.4. Note that there is a strict ordering relationship between CN_OUT_A*x* and CN_BKCAL_IN_A*x* parameters of the MAO block. That is, CN_BKCAL_IN_A0 is associated with CN_OUT_A0, CN_BKCAL_IN_A1 with CN_OUT_A1, and so on.

Figure 7.4 Sample Cascaded AO Configuration



The linking device consumes analog values produced by the controller on ControlNet and publishes these values on Fieldbus. The NI-FBUS Configurator sets up the analog output devices to subscribe to their data on Fieldbus. Each MAO block has its own schedule to consume data on ControlNet and publish it on Fieldbus. For different rates of publishing on Fieldbus, you must choose different MAO blocks as shown in Figure 7.5.

Figure 7.5 Sample Multiple Macrocycle AO Configuration



You do not have to connect to all the channels in an MAO block instance before using another instance. You do not have to use the channels in order. That is, you may use channels CN_OUT_A3 and CN_OUT_A5 and not use any of the others.

As shown in the lower loop of Figure 7.5, you can connect the MAO block to any parameter that is a float value and status combination. In other words, you can bring values from a PLC processor to any function block. The MAO block is not limited to interfacing with AO function blocks.

ControlNet Analog Output Objects

The linking device creates an instance of a ControlNet AO object for every wired channel in the MAO block instances. Each instance of the

ControlNet AO object has the attributes and access rules shown in Table 7.B.

Table 7.B ControlNet Analog Output Object Attributes

Attribute Number	Name	Type	Access	Initial Value	Remarks
3	Value	Float	Set	0	This is written from the ControlNet side.
4	CNStatus	Boolean	Set	Bad	ControlNet status. 0-Good; 1-Bad.
11	FailStateValue	Float	Set	0	The linking device writes the value to the AO block when this attribute is changed by the controller.
151	TagDesc	String	Get	As configured	Tag of the Fieldbus function block that this object instance represents.
152	FailstateTime	Float	Set	0	The linking device writes the value to the AO block when this attribute is changed by the controller.
153	ReadBack	Float	Get	0	This represents the BKCAL_OUT value from the AO block.
154	BkCalStatus	uint8	Get	Bad:Not-Connected	This represents the status of the Readback attribute.
155	casInstatus	uint8	Set	Bad:Not-connected	This represents the status of the CAS_IN to the AO block.

The BkCalStatus is created only when the controller is capable of participating in the CAScade initialization handshake and when the BKCAL_OUT of the AO block is wired to a BKCAL_IN*x* parameter of the MAO block.

Only Value, CNStatus, casInstatus, ReadBack, and TagDesc are created when an MAO is connected to something other than an AO function block CAS_IN parameter.

Discrete Inputs

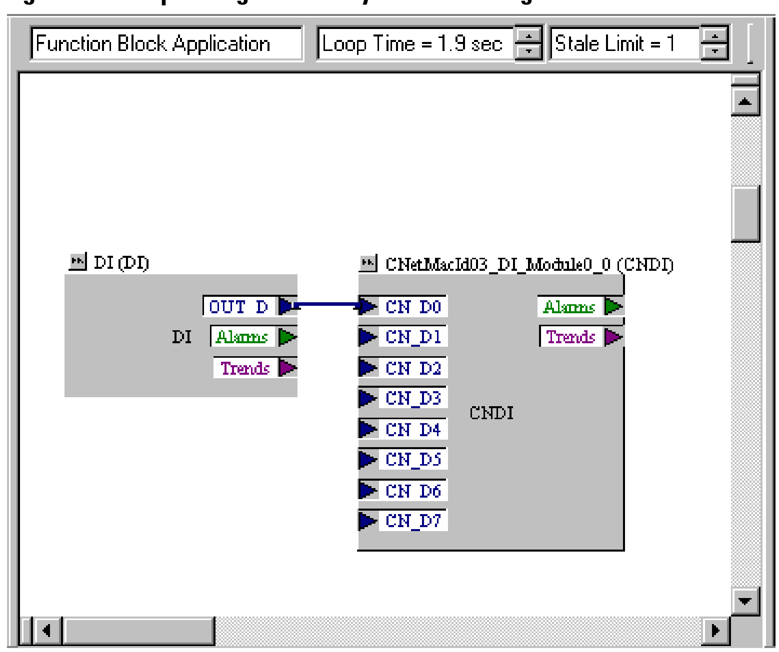
This section describes configuration of the linking device to access any discrete value (and status) in a Fieldbus device. It also describes the attributes of the created ControlNet discrete input object.

The linking device contains one MDI block instances on each of the two channels. Each instance of the MDI block is the software equivalent of a discrete input module in a I/O subsystem. Each MDI block has eight channels, with each channel containing a byte and a status pair. The linking device assigns a tag to each MDI block in the form CNetMacId*xx*_DI_Module*i-j,* where *xx* is the ControlNet network address, *i* is the Fieldbus channel number, and *j* is the module or instance number.

Configuration of Discrete Inputs

The NI-FBUS Configurator lists all the devices and the function blocks in each device in its browse window. This includes the DI function blocks in the Fieldbus devices and the MDI function blocks in the linking device. You must connect the Fieldbus DI function blocks that will be accessed by the ControlNet controllers (or devices) to the MDI channels as shown in Figure 7.6. The remaining configuration is similar to that for the analog inputs.

Figure 7.6 Sample Single Macrocycle MDI Configuration



ControlNet Discrete Input Objects

The linking device creates an instance of a ControlNet DI object for every wired channel in the MDI block instances. Each instance of the ControlNet DI object has the attributes and access rules shown in Table 7.C.

Table 7.C ControlNet Discrete Input Object Attributes

Attribute Number	Name	Type	Access	Initial Value	Remarks
3	Value	Boolean	Get	0	
4	CNStatus	Boolean	Get	Bad	ControlNet status. 0-Good; 1-Bad.
150	Ffstatus	uint8	Get	Bad:Out of Service	Fieldbus status of the value; obtained from the Fieldbus device.
151	TagDesc	String	Get	As configured	Tag of the Fieldbus function block that is represented by this object instance.
156	DiscAlmFlag	uint8	Get/Set	0	See Alarm Handling for Discrete Inputs

All attributes in Table 7.C are created if a standard Fieldbus DI function block is connected to a channel on an MDI block. Otherwise, DiscAlmFlag is not created.

Alarm Handling for Discrete Inputs

You can configure Fieldbus DI function blocks to detect and report alarms.

The process alarm (DISC_ALM) are exposed through the ControlNet DI object. When you connect a DI function block to an MDI block, the NI-FBUS Configurator configures the field device to send the alarm to the linking device.

In Fieldbus terminology, confirming an alarm means that the alarm has been received by an operator, and acknowledging an alarm means that the operator has taken the necessary action.

When an alarm condition is detected by a Fieldbus function block, the alarm is said to be ACTIVE. An active alarm should be Acknowledged and Confirmed after the condition causing the alarm is corrected. When the function block detects the alarm condition is no longer present, the alarm is said to be CLEARED. The CLEARED state must be Confirmed when it is detected. Users of the LD use the alarm attributes to process Fieldbus alarms.

Each attribute corresponding to a process alarm has the three alarm-related bits as shown in Figure 7.7.

Figure 7.7 Alarm Attribute Definition

Bit Position	7	6	5	4	3	2	1	0
Value	X	X	X	X	X	Acknowledge	CLEARED	ACTIVE

Bits 7-3, DON'T CARE, are undefined when read and should be ignored when performing a GET. The DON'T CARE bits should be written as 0 when performing a SET. Bit 2, Acknowledge, is undefined when read and should be ignored when performing a GET. When a user wants to Acknowledge an alarm, this bit should be written as a 1 when performing a SET. Each time the LD detects a 1 in the Acknowledge position, a Fieldbus Acknowledge will be generated. Therefore, users must be careful not to set the Acknowledge bit more than once per alarm. Bits 1 and 0, CLEARED and ACTIVE respectively, are read/write bits.

When an alarm condition is ACTIVE or CLEARED, these bits are read as a 1 when performing a GET. The user must Confirm each of these states by writing them to a 0 by a SET. Once an alarm has occurred (become ACTIVE), alarm processing is NOT complete until the user intervenes to remove the alarm condition, Acknowledges the alarm (writing a 1 in the Acknowledge bit position), Confirms the alarm (writing a 0 to the ACTIVE bit position once it is read as a 1), and Confirms the alarm condition being CLEARED (writing a 0 to the CLEARED bit position once it is read as a 1).

Discrete Outputs

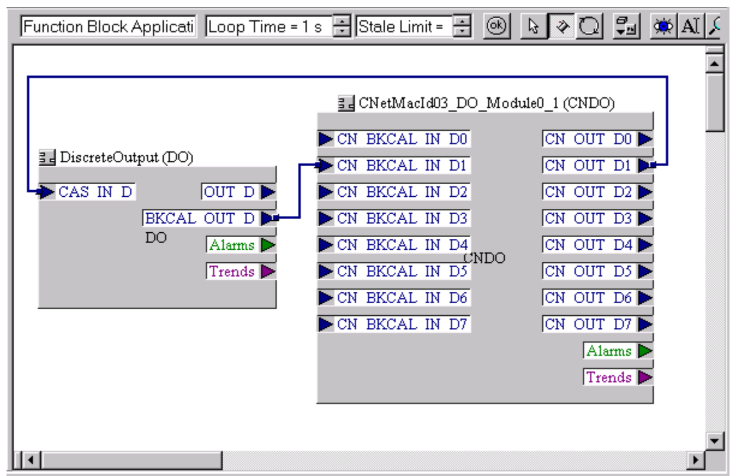
This section describes configuration of the linking device to control any discrete value (and status) in a Fieldbus device, such as in a Discrete Output (DO) function block. It also describes the attributes of the created ControlNet discrete output object.

The linking device contains one MDI block instances on each Fieldbus channels. Each instance of the MDO block is the software equivalent of a discrete output module in a I/O subsystem. Each MDO block also has eight channels or outputs, with each channel containing a byte and a status pair. Each MDO block has eight inputs to provide for the BKCAL or readback from the discrete output function blocks. The linking device assigns a tag to each MDO block in the form CNetMacId*xx*_DO_Module*i-j,* where *xx* is the ControlNet network address, *i* is the Fieldbus channel number, and *j* is the module or instance number.

Configuration of Discrete Outputs

The NI-FBUS Configurator lists all the devices and the function blocks in each device in its browse window. This includes the DO function blocks in the Fieldbus devices and the MDO function blocks in the linking device. You must connect the Fieldbus DO function blocks that will be controlled by the ControlNet controllers (or devices) to the MDO channels.

Figure 7.8 Sample Cascade DO Configuration



The configuration shown in Figure 7.8, the controller provides the necessary mode handshake for cascade initialization of the DO block. In this case, you connect CN_OUT_D*x* of the MDO block to the CAS_IN_D parameters of the DO block, and the BKCAL_OUT_D parameter of the DO block to BKCAL_IN*n* of the MDO block. Note that there is a strict ordering relationship between the CN_OUT_D*x* and CN_BKCAL_IN_D*x* parameters of the MDO block, that is, CN_BKCAL_IN_D0 is associated with CN_OUT_D0, CN_BKCAL_IN_D1 with CN_OUT_D1, and so on.

You can connect the MDO block to any parameter that is a byte value and status combination. In other words, you can bring values from a PLC processor to any function block. The MDO block is not limited to interfacing with DO function blocks.

ControlNet Discrete Output Objects

The linking device creates an instance of a ControlNet DO object for every wired channel in the MDO block instances. Each instance of the ControlNet DO object has the attributes shown in Table 7.D.

Table 7.D ControlNet Discrete Output Object Attributes

Attribute Number	Name	Type	Access	Initial Value	Remarks
3	Value	Boolean	Set	0	This is written from the ControlNet side.
4	CNStatus	Boolean	Set	Bad	ControlNet status. 0-Good; 1-Bad.
6	FailStateValue	uint8	Set	0	The linking device writes the value to the DO block when this attribute is changed by the controller.
151	TagDesc	String	Get	As configured	Tag of the Fieldbus function block that this object instance represents.
152	FailstateTime	Float	Set	0	The linking device writes the value to the DO block when this attribute is changed by the controller.
153	ReadBack	uint8	Get	0	This represents the BKCAL_OUT _D value from the DO block.
154	BkCalStatus	uint8	Get	Bad:Not-Connected	This represents the status of the Readback attribute.
155	casInstatus	uint8	Set	Bad:Not-connected	This represents the status of the CAS_IN_D to the DO block.

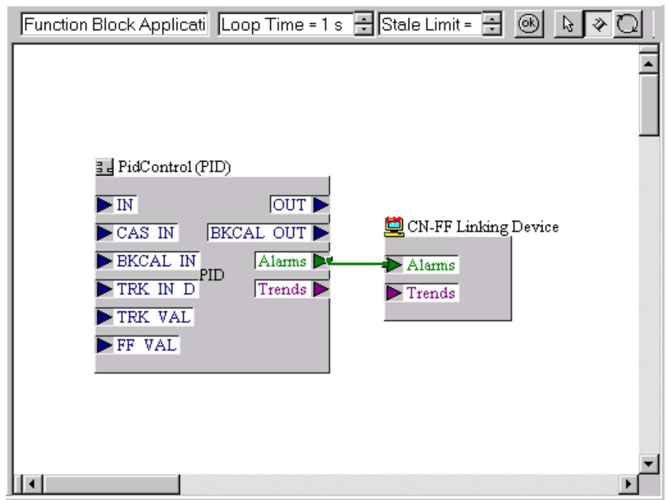
The BkCalStatus is created only when the controller is capable of participating in the CAScade initialization handshake, that is, when the BKCAL_OUT_D of the DO block is wired to a BKCAL_IN*x* parameter of the MDO block.

Only Value, CNStatus, casInstatus, ReadBack, and TagDesc are created when an MDO channel is connected to something other than a DO function block CAS_IN_D parameter.

Alarm Handling by the HMI

The linking device makes the process alarms from AI and DI function blocks visible as attributes of the created ControlNet objects, as discussed in the previous sections. Fieldbus devices generate other types of alarms; for example, they send out an alarm whenever their static configuration changes. Other function block types also generate alarms. In general all alarms, except the AI and DI process alarms, must be handled by a Fieldbus HMI. You must configure the linking device to receive such alarms and forward them to a PC-based HMI on ControlNet. You set up the linking device as shown in Figure 7.9.

Figure 7.9 Setting Up an Alarm



If you connect the AI and DI alarms to the linking device, the linking device forwards all alarms from these to the HMI. The process alarms are also visible to your PLC processor through the ControlNet objects. You must ensure, through your application design, that either the HMI or the PLC processor acknowledges the process alarms.

Assembly Objects

For each channel that you connect on the MAI, MAO, MDI, and MDO blocks, the linking device creates an instance of a ControlNet object corresponding to the type of channel. These individual instances are useful for unscheduled querying and setting object information. For scheduled communications on ControlNet, the linking device takes specific attributes from the corresponding object instances and combines them into input and output assembly objects.

The linking device uses input assembly objects to produce data on ControlNet and output assembly objects to consume data from ControlNet. The definition of assembly objects is determined after you configure the MAI, MAO, MDI and MDO blocks on the linking device. Assembly offsets are 0 based and increment by the size of data placed into them.

Table 7.E describes the attributes placed into the input and output assembly for each object type.

Table 7.E Attributes in Input and Output Assemblies

Object Type	Attributes in Input Assembly	Attributes in Output Assembly
AI	3, 4, 150	
AO (CAScade initialization)	153, 154	3, 4, 155
AO (NO CAScade initialization)	153	3, 4, 155
DI	3, 4, 150	
DO (CAScade initialization)	153, 154	3, 4, 155
DO (NO CAScade initialization)	153	3, 4, 155

MAI Blocks

For each MAI block configured in the linking device beginning with the lowest numbered module, each channel that is connected to a Fieldbus function block has attributes 3, 4, and 150 placed into the required input assembly object. Each MAI channel requires 6 bytes in the input assembly object.

MAO Blocks

For each MAO block configured in the linking device beginning with the lowest numbered module, each channel that is connected to a Fieldbus function block has attributes 3, 4, and 155 placed into the required output assembly object. Additionally, if your controller participates in cascade initialization for a specific channel (wiring BKCAL_OUT from AO), attribute 154 is placed into the required input assembly. Each MAO channel requires 6 bytes in the output assembly object and 6 bytes in the input assembly, if cascade initialization is performed—4 bytes in the input assembly if cascade initialization is *not* performed.

MDI Blocks

For each MDI block configured in the linking device beginning with the lowest numbered module, each channel that is connected to a Fieldbus function block has attributes 3, 4, and 150 placed into the required input assembly object. Each MDI requires 4 bytes in the input assembly object.

MDO Blocks

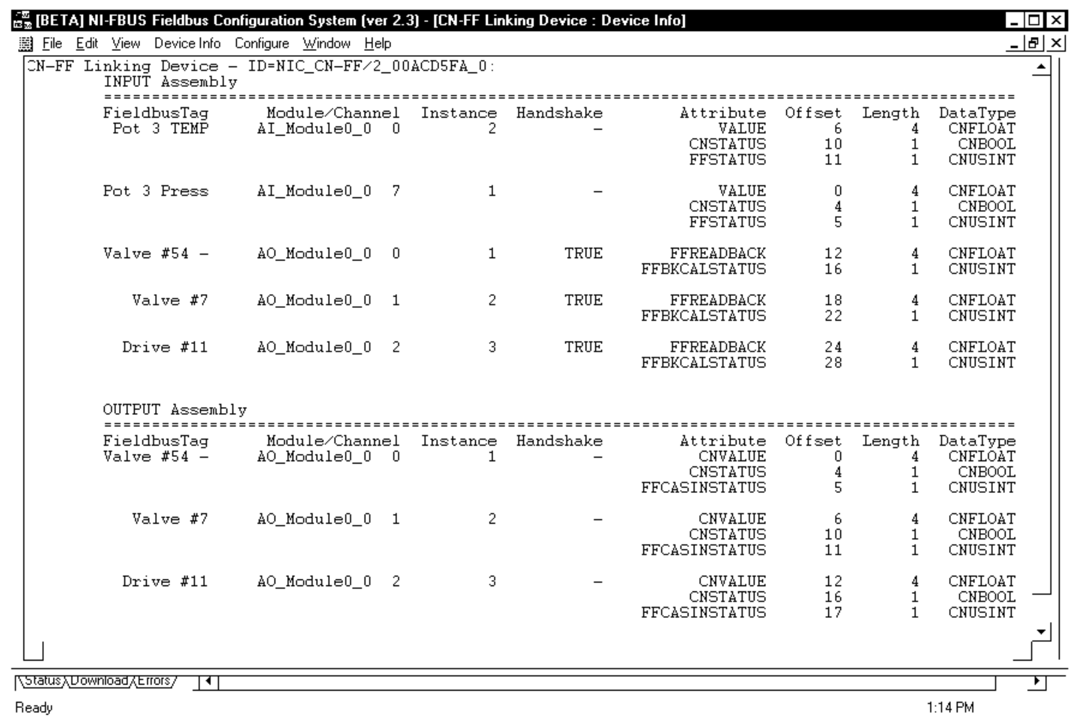
For each MDO block configured in the linking device beginning with the lowest numbered module, each channel that is connected to a Fieldbus function block has attributes 3, 4, and 155 placed into the required output assembly object. Additionally, if your controller participates in cascade initialization for a specific channel (wiring BKCAL_OUT_D from DO), attribute 154 is placed into the required input assembly. Each MDO channel requires 2 bytes in the input assembly object, and 4 bytes in the output assembly object, if cascade initialization is performed—2 bytes if cascade initialization is *not* performed.

Viewing Object Information in the NI-FBUS Configurator

The NI-FBUS Configurator can display information about the ControlNet objects created inside the linking device. This information contains the layouts and instance numbers of the input and output assembly objects as well as instance numbers of the base ControlNet objects created. To display this information, view the Device Info for the linking device in question. You can view the Device information in the NI-FBUS Configurator from the browse window. A sample Device Info display is shown in Figure 7.10.

To print the Device Info, put the cursor on the display of the Device Info that is shown in Figure 7.10, and click on Print, under the File menu. Note, if the cursor is not on the Device Info display, when you click on Print, you will be given an extensive menu of other items that you may print, but no listing for the Device Info.

Figure 7.10 Sample NI-FBUS Configurator View of ControlNet Object Information



Changing the Linking Device Configuration

Once a configuration is created in the linking device and a PLC processor or PC is using the offsets previously defined, an addition to the configuration should not require you to change functioning code in your application. For this reason, once a configuration has been stored into the linking device, if the user wires more Fieldbus devices with NI-FBUS, additional object instances are created and additional offsets are *appended* to the assembly objects; the existin.

The individual instances increment from the last one used in the category. The offsets within the assembly objects are assigned in the same order as described previously, but are placed after existing offsets. For example, assume you have two MAI channels and one MDO channel with cascade initialization. The instances and offsets are created. Sometime later, you need to add an additional MAI channel and two MAO channels to your configuration. These additions are placed *after* the final MDO from the previous configuration.

Because existing programs may be using a configuration, the deletion of a linkage from a Fieldbus network does not result in a renumbering of ControlNet objects or a reordering of the assembly object offsets. If the Fieldbus device is no longer present, Fieldbus and ControlNet status indicates an error condition, and the object and assembly offset resources continue to be in use. In order to remove existing objects from a configuration, you must remove the existing configuration. To accomplish this, select Download Configuration in the NI-FBUS Configurator, enable the **Clear Devices** checkbox in the dialog box that appears, and download a new configuration.

Trends and Alarms

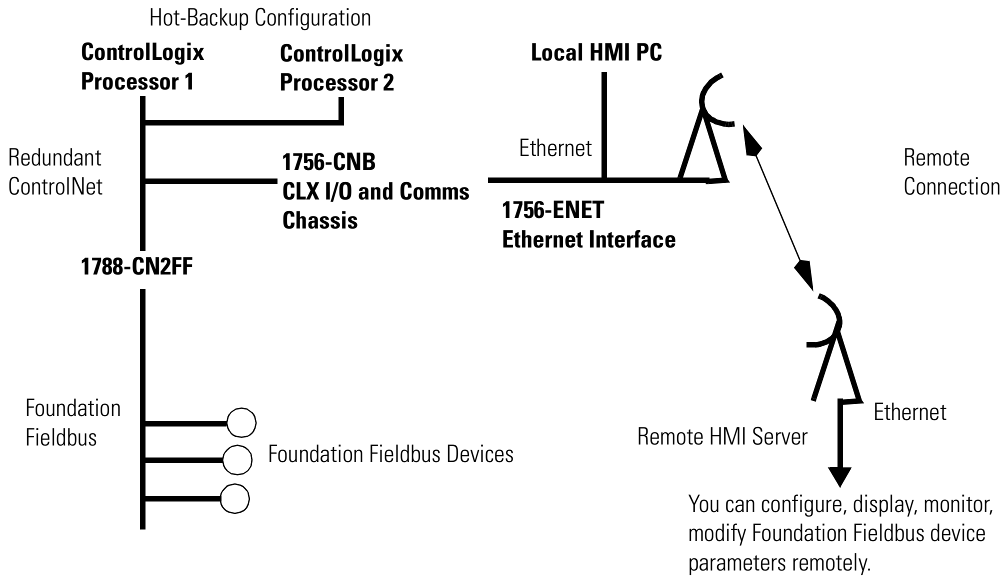
The linking device can receive trends and alarms from connected Fieldbus devices.

Use NI-FBUS configurator to connect trends and alarms to the linking device. Within the function block application, drag the linking device icon from the browse window. Connect trends and alarms as necessary from Fieldbus devices to the trends or alarms inputs. Refer to the NI-FBUS Configurator User Manual, publication 1788-6.5.2, for additional information.

Tips for Connecting to a 1756-ENET Controller

Here are some general tips that might help you if you are connecting to a Fieldbus network via a 1756-ENET controller. Rockwell has demonstrated the bridging capability of Ethernet to ControlNet for years. Accessing a 1756-CN2FF that sits on ControlNet is not a problem. In RSLinx, use the Ethernet driver, TCP. You can bridge to ControlNet (through a ControlLogix Gateway) and then to Foundation Fieldbus (through a 1788-CN2FF).

Figure 7.11 Remote Configuration Example



Use RSLinx to create a TCP-xx path from the computer that is running the 1788-FFCT. The path lists the Ethernet IP address of the remote 1756-ENET ethernet interface. Be sure that you have the correct IP address. Use an Internet Browser and rather than typing //htpp:www.... address, type the IP address (ex. 130. 151. 133. 48) of the 1756-ENET. If you have the correct address, you can look at the devices in the remote ControlLogix backplane. Refer to Remote Configuration of a Fieldbus Network via the 1788-CN2FF on page F-49 for more information.

Notes:

Standard Function Block Parameters[1]

Axxx Blocks

Table A.1 ACK_OPTION

Classification	Simple Variable
Description	Selects whether alarms associated with the block will be automatically acknowledged.
FF Data Type	Bit String
Range	1: Unacknowledge
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Table A.2 ALARM_HYS

Classification	Simple Variable
Description	Defines the amount of change a PV value must attain within the alarm limits before the alarm condition clears.
FF Data Type	Float
Range	0 to 50 percent of PV span
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0.5 percent

(1) This reference information is provided by **FOUNDATION**™ Fieldbus.

Want more Foundation Fieldbus information?

Visit the Fieldbus Foundation web site at www.fieldbus.org, or the following address, for more information:

9390 Research Blvd.
Suite II-250
Austin, TX 78759-9780

Table A.3 ALARM_SUM

Classification	Record
Description	Detects the current alert status, unacknowledged states, and disable states of the alarms associated with the block
FF Data Type	DS-74
Usage	C/Alarm Summary
Length	8
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	The data type summarizes 16 alerts using the following 4 elements. • 1 Current • 2 Unacknowledged • 3 Unreported • 4 Disabled

Table A.4 ALERT_KEY

Classification	Simple Variable
Description	The identification number of the plant unit. This data may be used in the host for sorting alarms.
FF Data Type	Unsigned 8
Range	1 to 255
Usage	C/Alert Key
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Bxxx Blocks

Table A.5 BAL_TIME

Classification	Simple Variable
Description	Specifies the time in seconds for the internal working value of Bias or Ratio to return to the operator set value. In PID block, specifies the time constant to be used to move the integral term to obtain balance, when the output is limited and the mode is Auto, Cas, or RCas.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4

Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Table A.6 BIAS

Classification	Simple Variable
Description	Specifies the Bias value in engineering units to be used in computing the function block output
FF Data Type	Float
Range	OUT_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Normally, the operator has permission to write these values, but PROGRAM or LOCAL remove the permission and grant it to a supervisory computer or a local control panel.

Table A.7 BKCAL_HYS

Classification	Simple Variable
Description	Defines the amount of change an output value must attain from the limit before the limit status is turned OFF.
FF Data Type	Float
Range	0 to 50 percent of output span
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0.5 percent

Table A.8 BKCAL_IN

Classification	Record
Description	The value and status from a lower block's BKCAL_OUT that is used to prevent reset windup and to initialize the control loop.
FF Data Type	DS-65
Usage	I/Back-Calculation Input
Length	5

Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.9 BKCAL_OUT

Classification	Record
Description	The value and status required by an upper block's BKCAL_IN so the upper block may prevent reset windup and provide bumpless transfer to closed loop control.
FF Data Type	DS-65
Usage	O/Back Calculation Output
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.10 BKCAL_OUT_D

Classification	Record
Description	The output value and status provided to an upstream discrete block that is used to provide bumpless transfer for closed loop control.
FF Data Type	DS-66
Usage	O/Back Calculation Output
Length	2
Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.11 BKCAL_SEL_1

Classification	Record
Description	The selector output value and status associated with SEL_1 input that is provided to BKCAL_IN of the block connected to SEL_1 to prevent reset windup.
FF Data Type	DS-65
Usage	O/Back Calculation Output
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.12 BKCAL_SEL_2

Classification	Record
Description	The selector output value and status associated with SEL_2 input that is provided to BKCAL_IN of the block connected to SEL_2 to prevent reset windup.
FF Data Type	DS-65
Usage	O/Back Calculation Output
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.13 BKCAL_SEL_3

Classification	Record
Description	The selector output value and status associated with SEL_3 input that is provided to BKCAL_IN of the block connected to SEL_3 to prevent reset windup.
FF Data Type	DS-65
Usage	O/Back Calculation Output
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.14 BLOCK_ALM

Classification	Record
Description	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alarm is entered in the sub-code field. The first alarm to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alarm reporting task, another block alarm may be reported without clearing the Active status, if the sub-code has changed.
FF Data Type	DS-72
Usage	C/Alarm
Length	13
Storage	Dynamic
Remarks	The data type describes discrete alarms using the following five elements: • 1 Unacknowledged • 2 Alarm State • 3 Time Stamp • 4 Subcode • 5 Value

Table A.15 BLOCK_ERR

Classification	Simple Variable
Description	Reflects the error status associated with the hardware or software components associated with a block. It is a bit string that can show multiple errors.
FF Data Type	Bit String
Usage	C/Block Error
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic

Table A.16 BYPASS

Classification	Simple Variable
Description	Provides the means to bypass the normal control algorithm. When BYPASS is On, the setpoint value is directly transferred to the output. To prevent a bump upon BYPASS switching, the setpoint automatically initializes to the output value or process variable and sets the path broken flag for one execution.
FF Data Type	Unsigned 8
Range	1: Off 2: On
Usage	C/Contained
Length	1

Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is 0.

Cxxx Blocks

Table A.17 CAS_IN

Classification	Record
Description	Represents the remote setpoint value that must come from another fieldbus block or a distributed control system (DCS) block through a defined link.
FF Data Type	DS-65
Usage	I/Cascade Input
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.18 CAS_IN_D

Classification	Record
Description	Represents the remote setpoint value for a discrete block that must come from another fieldbus block or a distributed control system (DCS) block through a defined link.
FF Data Type	DS-66
Usage	I/Cascade Input
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.19 CHANNEL

Classification	Simple Variable
Description	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.
FF Data Type	Unsigned 16
Range	1 to Manufacturer Limit

Usage	C/Channel
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	The initial value is 0.

Table A.20 CLR_FSAFE

Classification	Simple Variable
Description	Serves as a switch to reset/clear the device failsafe state after the fault condition is cleared.
FF Data Type	Unsigned 8
Range	1: Off 2: Clear
Usage	C/Contained
Length	1
Storage	Dynamic
Remarks	The operator can control PROGRAM or LOCAL access to these values.

Table A.21 CONFIRM_TIME

Classification	Simple Variable
Description	Defines the time between retries of alert reports.
FF Data Type	Unsigned 32
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	The initial value is 32000 milliseconds.

Table A.22 CONTROL_OPTS

Classification	Simple Variable
Description	Represents bit string for control options to alter the calculations done in an applicable function block.
FF Data Type	Bit String
Usage	C/Contained
Length	2

Valid Views	VIEW_4
Storage	Static
Remarks	See the following for a list of the control options by bit and applicable function block.

Table A.23 CONTROL_OPTS Bit Selections

Bit	Meaning	Function Block					
		BG	**CS**	**ML**	**PD**	**PID**	**RA**
0	Bypass Enable (LSB)				X	X	
1	SP-PV Track in Man				X	X	X
2	SP-PV Track in ROut				X	X	
3	SP-PV Track in LO or IMan				X	X	X
4	SP Track retained target	X			X	X	X
5	Direct Acting				X	X	
6	Balance Ramp	X			X		X
7	Track Enable	X		X	X	X	X
8	Track in Manual	X		X	X	X	X
9	Use PV for BKCAL_OUT				X	X	X
10	Act on IR	X			X		X
11	Use percent for IN_1	X					X
12	Obey SP limits if Cas or RCas	X			X	X	X
13	No OUT limits in Manual	X	X	X	X	X	X
14	Reserved						
15	Reserved						

Table A.24 CYCLE_SEL

Classification	Simple Variable
Description	A bit string to identify the block execution method selected for this resource.
FF Data Type	Bit String
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static
Remarks	Changing this parameter may be fatal to communication.

Table A.25 CYCLE_TYPE

Classification	Simple Variable
Description	A bit string to identify the block execution methods available for this resource.
FF Data Type	Bit String
Range	Set by Manufacturer
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Dxxx Blocks

Table A.26 DEV_REV

Classification	Simple Variable
Description	Identifies the manufacturer revision number associated with the resource. An interface device uses it to locate the DD file for the resource.
FF Data Type	Unsigned 8
Range	Set by manufacturer
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.27 DEV_TYPE

Classification	Simple Variable
Description	Identifies the manufacturer's model number associated with the resource. An interface device uses it to locate the DD file for the resource.
FF Data Type	Unsigned 16
Range	Set by manufacturer
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.28 DD_RESOURCE

Classification	Simple Variable
Description	Identifies the tag of the resource that contains the Device Description for this resource.
FF Data Type	Visible String
Usage	C/DD Resource
Length	32
Storage	Static
Remarks	Read Only

Table A.29 DD_REV

Classification	Simple Variable
Description	Identifies the revision of the Device Description associated with the resource so an interface device can locate the DD file for the resource.
FF Data Type	Unsigned 8
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.30 DISC_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the discrete alarm.
FF Data Type	DS-72
Usage	C/Alarm
Length	13
Storage	Dynamic
Remarks	Read Only The data type consists of data that describes discrete alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.31 DISC_LIM

Classification	Simple Variable
Description	Identifies state of discrete input that will generate an alarm.
FF Data Type	Unsigned 8
Range	PV state
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static

Table A.32 DISC_PRI

Classification	Simple Variable
Description	Identifies the priority of the discrete alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static

Table A.33 DV_HI_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the high deviation alarm.
FF Data Type	DS-71
Usage	C/Alarm
Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.34 DV_HI_LIM

Classification	Simple Variable
Description	Defines the high deviation alarm limit setting in engineering units.
FF Data Type	Float
Range	0 to PV Span, + infinity
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is + infinity

Table A.35 DV_HI_PRI

Classification	Simple Variable
Description	Defines priority of the high deviation alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Table A.36 DV_LO_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the low deviation alarm.
FF Data Type	DS-71
Usage	C/Alarm
Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.37 DV_LO_LIM

Classification	Simple Variable
Description	Defines the low deviation alarm limit setting in engineering units.
FF Data Type	Float
Range	– infinity, - PV Span to 0,
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is – infinity

Table A.38 DV_LO_PRI

Classification	Simple Variable
Description	Defines priority of the low deviation alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Exxx Blocks

No E parameters

Fxxx Blocks

Table A.39 FAIL_SAFE

Classification	Simple Variable
Description	Condition set by lost of communication to an output block, failure promoted to an output block or a physical contact. When the failsafe action is active, the output function blocks will perform their FSAFE action
FF Data Type	Unsigned 8
Range	1: Clear 2: Active
Usage	C/Contained
Length	1

Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only

Table A.40 FEATURES

Classification	Simple Variable
Description	Bit string that identifies the supported resource block options.
FF Data Type	Bit String
Range	Set by manufacturer
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.41 FEATURE_SEL

Classification	Simple Variable
Description	Bit string that identifies the selected resource block options.
FF Data Type	Bit String
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static

Table A.42 FF_GAIN

Classification	Simple Variable
Description	Defines the gain value used to multiply the feed-forward signal before it is added to the calculated control output.
FF Data Type	Float
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.43 FF_SCALE

Classification	Record
Description	Defines the feed-forward input high and low scale values, engineering units code, and number of digits to the right of the decimal.
FF Data Type	DS-68
Range	0-100 percent
Usage	C/Scaling
Length	11
Valid Views	VIEW_4
Storage	Static
Remarks	The Data type consists of data that describes floating point values for display purposes. It uses the following four elements. • Engineering Units at 100 percent • Engineering Units at 0 percent • Units Index • Decimal Point

Table A.44 FF_VAL

Classification	Record
Description	Represents the feed-forward value.
FF Data Type	DS-65
Usage	I/Input
Length	5
Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	Read Only The Data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.45 FIELD_VAL

Classification	Record
Description	Represents the raw value from the field device in percent of transducer span, with a status reflecting the transducer condition, before signal characterization (L_TYPE) or filtering (PV_FTIME).
FF Data Type	DS-65
Usage	C/Contained
Length	5

Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The Data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.46 FIELD_VAL_D

Classification	Record
Description	Represents the raw value of a field device discrete input with a status reflecting the transducer condition.
FF Data Type	DS-66
Usage	C/Contained
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.47 FREE_SPACE

Classification	Simple Variable
Description	Identifies the percent of memory available for further configuration. Zero in a preconfigured resource.
FF Data Type	Float
Range	0 - 100 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Dynamic
Remarks	Read Only

Table A.48 FREE_TIME

Classification	Simple Variable
Description	Identifies the percent of block processing time that is free to process additional blocks.
FF Data Type	Float
Range	0 - 100 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only

Table A.49 FSAFE_TIME

Classification	Simple Variable
Description	Represents the reaction time in seconds from the detection of a failure at the output block remote setpoint to the output block action, if the condition still exists.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.50 FSAFE_VAL

Classification	Simple Variable
Description	Defines the preset analog setpoint to use when a failure occurs. Value is ignored, if the IO_OPTS Failsafe to value option is false.
FF Data Type	Float
Range	PV_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.51 FSAFE_VAL_D

Classification	Simple Variable
Description	Defines the preset discrete setpoint to use when a failure occurs. Value is ignored, if the IO_OPTS Failsafe to value option is false.
FF Data Type	Unsigned 8
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static

Gxxx Blocks

Table A.52 GAIN

Classification	Simple Variable
Description	Represents dimensonless gain used by several different algorithms.
FF Data Type	Float
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.53 GRANT_DENY

Classification	Record
Description	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm parameters.
FF Data Type	DS-70
Usage	C/Access Permission
Length	2
Valid Views	VIEW_2
Storage	Non-Volatile
Remarks	The data type consists of access control flags for access to block parameters. It uses the following two elements. • Grant • Deny

Hxxx Blocks

Table A.54 HARD_TYPES

Classification	Simple Variable
Description	Identifies types of hardware that are available as channel numbers on this resource.
FF Data Type	Bit String
Range	Set by manufacturer
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.55 HI_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the high alarm.
FF Data Type	DS-71
Usage	C/Alarm
Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.56 HI_HI_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the high high alarm.
FF Data Type	DS-71
Usage	C/Alarm

Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • 1 Unacknowledged • 2 Alarm State • 3 Time Stamp • 4 Subcode • 5 Value

Table A.57 HI_HI_LIM

Classification	Simple Variable
Description	Defines the high high alarm limit setting in engineering units.
FF Data Type	Float
Range	PV_SCALE, + infinity
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is + infinity

Table A.58 HI_HI_PRI

Classification	Simple Variable
Description	Defines priority of the high high alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Table A.59 HI_LIM

Classification	Simple Variable
Description	Defines the high alarm limit setting in engineering units.
FF Data Type	Float
Range	PV_SCALE, + infinity

Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is + infinity

Table A.60 HI_PRI

Classification	Simple Variable
Description	Defines priority of the high alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0

Ixxx Blocks

Table A.61 IO_OPTS

Classification	Simple Variable
Description	Identifies user-selectable options for altering the Input and Output block processing.
FF Data Type	Bit String
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	See the following for a list of the control options by bit and applicable function block.

Table A.62 IO_OPTS Bit Selections

Bit	Meaning	Function Block			
		AI	DI	AO	DO
0	Invert		X		X
1	SP-PV Track in Man			X	X
2	Reserved				
3	SP-PV Track in LO or IMan			X	X

4	SP Track retained target			X	X
5	Increase to close			X	
6	Failsafe to value			X	X
7	Use Failsafe value to restart			X	X
8	Target to Man if failsafe activated			X	X
9	Use PV for BKCAL_OUT			X	X
10	Low Cutoff	X			
11	Reserved				
12	Reserved				
13	Reserved				
14	Reserved				
15	Reserved				

Table A.63 IN

Classification	Record
Description	Represents the primary input value of the block. Blocks that filter the input to get the PV require this parameter.
FF Data Type	DS-65
Usage	I/Primary Input
Length	5
Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	Read Only The Data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.64 IN_1

Classification	Record
Description	Represents the auxiliary input value to the block. It is used for values other than the PV.
FF Data Type	DS-65
Usage	I/Input
Length	5

Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	Read Only The Data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Jxxx Blocks

No J parameters

Kxxx Blocks

No K parameters

Lxxx Blocks

Table A.65 LIM_Notify

Classification	Simple Variable
Description	Defines the maximum number of unconfirmed alert notify messages allowed.
FF Data Type	Unsigned 8
Range	0 to MAX_NOTIFY
Usage	C/Contained
Length	1
Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is MAX_NOTIFY

Table A.66 L_TYPE

Classification	Simple Variable
Description	Determines whether the values passed by the Transducer block to the Analog Input block may be used directly (DIRECT) or, if the value is in different units, must be converted linearly (indirectly); or with square root (Ind Sqr Root), using the input range defined by the transducer and associated output range.
FF Data Type	Unsigned 8
Range	1: Direct 2: Indirect 3: Ind Sqr Root
Usage	C/Contained

Length	1
Valid Views	VIEW_4
Storage	Static

Table A.67 LO_ALM

Classification	Record
Description	Represents the status of the low alarm and its associated time stamp.
FF Data Type	DS-71
Usage	C/Alarm
Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.68 LO_LIM

Classification	Simple Variable
Description	Defines the setting for the low alarm in engineering units.
FF Data Type	Float
Range	- Infinity, PV_SCALE
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is - Infinity

Table A.69 LO_LO_ALM

Classification	Record
Description	Represents the status of the low low alarm and its associated time stamp.
FF Data Type	DS-71
Usage	C/Alarm

Length	16
Storage	Dynamic
Remarks	Read Only The Data type consists of data that describes floating point alarms. It uses the following five elements. • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.70 LO_LO_LIM

Classification	Simple Variable
Description	Defines the setting for the low low alarm in engineering units.
FF Data Type	Float
Range	- Infinity, PV_SCALE
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is - Infinity

Table A.71 LO_LO_PRI

Classification	Simple Variable
Description	Represents the priority of the low low alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority
Length	1
Valid Views	VIEW_4
Storage	Static

Table A.72 LO_PRI

Classification	Simple Variable
Description	Represents the priority of the low alarm.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority

Length	1
Valid Views	VIEW_4
Storage	Static

Table A.73 LOW_CUT

Classification	Simple Variable
Description	Represents the limit used for the flow sensor input processing by the Analog Input block, if the Low Cutoff selection is choosen in IO_OPTS. If the calculated PV falls below this limit, the PV value is set to zero (0).
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	This function may be used to eliminate noise near zero from a flow sensor.

Mxxx Blocks

Table A.74 MANUFAC_ID

Classification	Simple Variable
Description	Defines the manufacturer's identification number. This number is used by an interface device to locate the DD file for the resource.
FF Data Type	Unsigned 32
Range	Set by manufacturer
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.75 MAX_NOTIFY

Classification	Simple Variable
Description	Defines the maximum number of unconfirmed notify messages possible.
FF Data Type	Unsigned 8
Range	Set by manufacturer
Usage	C/Contained
Length	1

Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.76 MEMORY_SIZE

Classification	Simple Variable
Description	Represents the available configuration memory in the empty resource.
FF Data Type	Unsigned 16
Range	Set by manufacturer
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.77 MIN_CYCLE_T

Classification	Simple Variable
Description	Defines the time duration of the shortest cycle interval that the resource can support.
FF Data Type	Unsigned 32
Range	Set by manufacturer
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

Table A.78 MODE_BLK

Classification	Record
Description	Represents the mode record of the block. Contains the Actual, Target, Permitted, and Normal modes.
FF Data Type	DS-69
Usage	C/Mode
Length	4

Valid Views	VIEW_1, VIEW_3
Storage	mix
Remarks	Normally, the operator has permission to write these values, but PROGRAM or LOCAL remove that permission and grant it to a supervisory computer or a local control panel. This block has a mixture of storage types. Static for modes Normal and Permitted, Non-Volatile for Target mode, and Dynamic for Actual mode. The data type consists of bit strings for Actual, Target Permitted and Normal modes. It uses the following four elements. • Target • Actual • Permitted • Normal

Nxxx Blocks

Table A.79 NV_CYCLE_T

Classification	Simple Variable
Description	Defines interval between writing copies of Non-Volatile (NV) parameters to NV memory. Zero means never.
FF Data Type	Unsigned 32
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	Ready Only

Oxxx Blocks

Table A.80 OUT

Classification	Record
Description	Represents the primary analog value calculated as a result of executing the function.
FF Data Type	DS-65
Range	OUT_SCALE +/- 10 percent
Usage	O/Primary Output
Length	5

Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Normally, the operator has permission to write this value, but PROGRAM or LOCAL remove that permission and grant it to a supervisory computer or a local control panel. The Data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.81 OUT_D

Classification	Record
Description	Represents the primary discrete value calculated as a result of executing the function.
FF Data Type	DS-66
Range	OUT_STATE
Usage	O/Primary Output
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Normally, the operator has permission to write this value, but PROGRAM or LOCAL remove that permission and grant it to a supervisory computer or a local control panel. The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.82 OUT_HI_LIM

Classification	Simple Variable
Description	Defines the maximum output value limit in all modes, unless the CONTROL_OPTS selection No Out limits in Manual is chosen.
FF Data Type	Float
Range	OUT_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is 100.

Table A.83 OUT_LO_LIM

Classification	Simple Variable
Description	Defines the minimum output value limit in all modes, unless the CONTROL_OPTS selection No Out limits in Manual is chosen.
FF Data Type	Float
Range	OUT_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is 0.

Table A.84 OUT_SCALE

Classification	Record
Description	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters that have the same scaling as OUT.
FF Data Type	DS-68
Usage	C/Scaling
Length	11
Valid Views	VIEW_2
Storage	Static
Remarks	The Data type consists of data that describes floating point values for display purposes. It uses the following four elements. • Engineering Units at 100 percent • Engineering Units at 0 percent • Units Index • Decimal Point

Table A.85 OUT_STATE

Classification	Simple Variable
Description	Represents the index to the text describing the states of a discrete output.
FF Data Type	Unsigned 16
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static

Pxxx Blocks

Table A.86 PV

Classification	Record
Description	Represents either the primary analog value for use in executing the function, or a process value associated with it. It may also be calculated from the READBACK value of an Analog Output block.
FF Data Type	DS-65
Usage	C/Process Variable
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.87 PV_D

Classification	Record
Description	Represents either the primary discrete value for use in executing the function, or a process value associated with it. It may also be calculated from the READBACK_D value of a Discrete Output block.
FF Data Type	DS-66
Usage	C/Process Variable
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.88 PV_FTIME

Classification	Simple Variable
Description	Defines the time constant of a single expotential filter for the Process Variable in seconds.
FF Data Type	Float
Range	Positive
Usage	C/Contained

Length	4
Valid Views	VIEW_4
Storage	Static

Table A.89 PV_SCALE

Classification	Record
Description	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters that have the same scaling as PV.
FF Data Type	DS-68
Usage	C/Scaling
Length	11
Valid Views	VIEW_2
Storage	Static
Remarks	The Data type consists of data that describes floating point values for display purposes. It uses the following four elements. • Engineering Units at 100 percent • Engineering Units at 0 percent • Units Index • Decimal Point

Table A.90 PV_STATE

Classification	Simple Variable
Description	Defines the index to the text describing the states of a discrete PV.
FF Data Type	Unsigned 16
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static

Qxxx Blocks

No Q parameters

Rxxx Blocks

Table A.91 RA_FTIME

Classification	Simple Variable
Description	Defines the time constant of a single expotential filter for the value to be ratioed in seconds.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.92 RATE

Classification	Simple Variable
Description	Defines the derivative time constant in seconds.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.93 RCAS_IN

Classification	Record
Description	Represents target setpoint and status provided by a supervisory host to the analog control or output block.
FF Data Type	DS-65
Usage	C/Remote-Cascade In
Length	5
Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.94 RCAS_IN_D

Classification	Record
Description	Represents target setpoint and status provided by a supervisory host to the analog control or output block.
FF Data Type	DS-66
Usage	C/Remote-Cascade In
Length	2
Valid Views	VIEW_3
Storage	Non-Volatile
Remarks	The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.95 RCAS_OUT

Classification	Record
Description	Represents block setpoint and status after ramping. It serves as input to a supervisory host for back calculation that allows action to be taken under limiting conditions or mode change.
FF Data Type	DS-65
Usage	C/Remote-Cascade Out
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.96 RCAS_OUT_D

Classification	Record
Description	Represents block setpoint and status. It serves as input to a supervisory host for back calculation that allows action to be taken under limiting conditions or mode change.
FF Data Type	DS-66
Usage	C/Remote-Cascade Out
Length	2

Valid Views	VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.97 READBACK

Classification	Record
Description	Represents the “readback” of the actual continuous valve or other actuator position in transducer units.
FF Data Type	DS-65
Usage	C/Contained
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.98 READBACK_D

Classification	Record
Description	Represents the "readback" of the actual discrete valve or other actuator position in the transducer state.
FF Data Type	DS-66
Usage	C/Contained
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.99 RESET

Classification	Simple Variable
Description	Represents the Integral time constant in seconds. It is the inverse of repeats per minute.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static

Table A.100 RESTART

Classification	Simple Variable
Description	Allows a manual restart to be initiated. The following degrees of restart are possible. • Run • Restart Resource • Restart with Defaults • Restart Processor
FF Data Type	Unsigned 8
Range	1: Run 2: Restart Resource 3: Restart with Defaults 4: Restart Processor
Usage	C/Contained
Length	1
Storage	Dynamic
Remarks	Changing this parameter, may be fatal to communication.

Table A.101 ROUT_IN

Classification	Record
Description	Represents target output and status provided by a host to a control block for use as the block's output in ROUT mode.
FF Data Type	DS-65
Usage	C/Remote-Output In
Length	5

Valid Views	VIEW_3
Storage	Dynamic
Remarks	The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.102 ROUT_OUT

Classification	Record
Description	Represents block output and status. It serves as input to a host for back calculation in ROut mode that allows action to be taken under limited conditions or mode change
FF Data Type	DS-65
Usage	C/Remote-Output Out
Length	5
Valid Views	VIEW_3
Storage	Dynamic
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.103 RS_STATE

Classification	Simple Variable
Description	Defines the state of the function block application state machine.
FF Data Type	Unsigned 8
Range	1: Start/Restart 2: Initialization 3: On-Line Linking 4: On-Line 5: Standby 6: Failure
Usage	C/Resource State
Length	1
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	Read Only

Sxxx Blocks

Table A.104 SEL_1

Classification	Record
Description	Represents first input value to the selector.
FF Data Type	DS-65
Usage	I/Cascade Input
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.105 SEL_2

Classification	Record
Description	Represents second input value to the selector.
FF Data Type	DS-65
Usage	I/Cascade Input
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.106 SEL_3

Classification	Record
Description	Represents third input value to the selector.
FF Data Type	DS-65
Usage	I/Cascade Input
Length	5

Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.107 SEL_TYPE

Classification	Simple Variable
Description	Defines the type of selector action as High, Medium, or Low.
FF Data Type	Unsigned 8
Range	1: High 2: Low 3: Medium
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static

Table A.108 SET_FSAFE

Classification	Simple Variable
Description	Allows the failsafe condition to be manually initiated by selecting Set.
FF Data Type	Unsigned 8
Range	1: Off 2: Set
Usage	C/Contained
Length	1
Storage	Dynamic
Remarks	The operator can control PROGRAM or LOCAL access to this value.

Table A.109 SHED_OPT

Classification	Simple Variable
Description	Defines action to be taken on remote control device timeout.
FF Data Type	Unsigned 8

Range	1: Normal Shed, normal return -See Note 1 in Remarks 2: Normal Shed, no return - See Note 2 in Remarks 3: Shed to Auto, normal return 4: Shed to Auto, no return - See Note 3 in Remarks 5: Shed to Manual, normal return 6: Shed to Manual, no return - See Note 4 in Remarks 7: Shed to Retained target, normal return 8: Shed to Retained target, no return
Usage	C/Shed Option
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Note 1: Actual mode changes to the next lowest priority non-remote mode permitted but returns to the target remote mode, when the remote computer completes the initialization handshake. Note 2: Target mode changes to the next lowest priority non-remote mode permitted. The target remote mode is lost, so there is no return to it. Note 3: Target mode changes to Auto on detection of a shed condition. Note 4: Target mode changes to Man on detection of a shed condition.

Table A.110 SHED_RCAS

Classification	Simple Variable
Description	Defines timeout in milliseconds for computer writes to function block RCas locations.
FF Data Type	Unsigned 32
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static

Table A.111 SHED_ROUT

Classification	Simple Variable
Description	Defines timeout in milliseconds for computer writes to function block ROut locations.
FF Data Type	Unsigned 32
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static

Table A.112 SIMULATE

Classification	Record
Description	Allows the transducer analog input or output to the block to be manually supplied, when SIMULATE is enabled. When SIMULATE is disabled, the simulate value and status track the actual value and status.
FF Data Type	DS-82
Usage	C/Simulate
Length	11
Storage	Dynamic
Remarks	The data type consists of simulate and transducer floating point value and status and a simulate enable/disable discrete. It uses the following five elements. 1: Simulate Status 2:Simulate Value 3: Transducer Status 4: Transducer Value 5: Simulate Enable/Disable

Table A.113 SIMULATE_D

Classification	Record
Description	Allows the transducer discrete input or output to the block to be manually supplied, when SIMULATE is enabled. When SIMULATE is disabled, the simulate value and status track the actual value and status.
FF Data Type	DS-83
Usage	C/Simulate
Length	5
Storage	Dynamic
Remarks	The data type consists of a simulate and transducer discrete value and status and a simulate enable/disable discrete. It uses the following five elements. 1: Simulate Status 2:Simulate Value 3: Transducer Status 4: Transducer Value 5: Simulate Enable/Disable

Table A.114 SP

Classification	Record
Description	Defines the setpoint of any analog block.
FF Data Type	DS-65
Range	PV_SCALE +/- 10 percent
Usage	C/Setpoint
Length	5

Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Normally, the operator has permission to write this value, but PROGRAM or LOCAL remove the permission and grant it to a supervisory computer or a local control panel. The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Table A.115 SP_D

Classification	Record
Description	Defines the setpoint of any discrete block.
FF Data Type	DS-66
Range	PV_STATE
Usage	C/Setpoint
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Normally, the operator has permission to write this value, but PROGRAM or LOCAL remove the permission and grant it to a supervisory computer or a local control panel. The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.116 SP_HI_LIM

Classification	Simple Variable
Description	Defines the high limit for setpoint entry.
FF Data Type	Float
Range	PV_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is 100 percent

Table A.117 SP_LO_LIM

Classification	Simple Variable
Description	Defines the low limit for setpoint entry.
FF Data Type	Float
Range	PV_SCALE +/- 10 percent
Usage	C/Contained
Length	4
Valid Views	VIEW_2
Storage	Static
Remarks	Initial value is zero (0)

Table A.118 SP_RATE_DN

Classification	Simple Variable
Description	Defines the downward ramp rate in PV units per second for setpoint changes to invoke action in the Auto mode.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	If the ramp rate is set to zero or the block is not in Auto mode, the setpoint change is invoked immediately.

Table A.119 SP_RATE-UP

Classification	Simple Variable
Description	Defines the upward ramp rate in PV units per second for setpoint changes to invoke action in the Auto mode.
FF Data Type	Float
Range	Positive
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	If the ramp rate is set to zero or the block is not in Auto mode, the setpoint change is invoked immediately.

Table A.120 ST_REV

Classification	Simple Variable
Description	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.
FF Data Type	Unsigned 16
Usage	C/Static Revision
Length	2
Valid Views	VIEW_1, VIEW_2, VIEW_3, VIEW_4
Storage	Static
Remarks	Read Only

Table A.121 STATUS_OPTS

Classification	Simple Variable
Description	Defines user-selectable options for the block processing of status.
FF Data Type	Bit String
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	See the following Table 3 for a list of the control options by bit and applicable function block.

Table A.122 STATUS_OPTS Bit Selections

Bit	**Meaning**	**Function Block**									
		AI	**DI**	**AO**	**DO**	**ML**	**BG**	**CS**	**PD**	**PID**	**RA**
0	IFS if BAD IN					X	X	X	X	X	X
1	IFS if BAD CAS_IN			X	X		X		X	X	X
2	Use Uncertain as Good					X	X	X	X	X	X
3	Propogate Failure Forward	X	X								
4	Propogate Failure Backward			X	X		X				X
5	Target to Manual if BAD IN								X	X	
6	Uncertain if Limited	X									
7	BAD if Limited	X									
8	Uncertain if Man Mode	X	X					X			X
9	Do not select if not Auto mode	X	X						X	X	
10	Do not select if not Cas mode								X	X	
11	Reserved										
12	Reserved										

13	Reserved										
14	Reserved										
15	Reserved										

Table A.123 STRATEGY

Classification	Simple Variable
Description	Assists in grouping blocks. This data is not checked or processed by the block.
FF Data Type	Unsigned 16
Usage	C/Strategy
Length	2
Valid Views	VIEW_4
Storage	Static

Txxx Blocks

Table A.124 TAG_DESC

Classification	Simple Variable
Description	Serves as user defined description of the block.
FF Data Type	Octet String
Usage	C/Tag Description
Length	32
Storage	Static
Remarks	Initial value is 32 space characters

Table A.125 TEST_RW

Classification	Record
Description	Defines read/write test parameter.
FF Data Type	DS-85
Usage	C/Test

Length	112
Storage	Dynamic
Remarks	The data type consists of function block test read/write data. It uses the following 15 elements. • Value 1 (Boolean) • Value 2 (Integer 8) • Value 3 (Integer 16) • Value 4 (Integer 32) • Value 5 (Unsigned 8) • Value 6 (Unsigned 16) • Value 7 (Unsigned 32) • Value 8 (Floating Point) • Value 9 (Visible String) • Value 10 (Octet String) • Value 11 (Date) • Value 12 (Time of Day) • Value 13 (Time Difference) • Value 14 (Bit String) • Value 15 (Time Value)

Table A.126 TRK_IN_D

Classification	Record
Description	Represents the discrete input for initiation of the external tracking function.
FF Data Type	DS-66
Usage	I/Input
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only The data type consists of the value and status of discrete value parameters. It uses the following two elements. • Status • Value

Table A.127 TRK_SCALE

Classification	Record
Description	Defines the high and low scale values, engineering units code, and number of digits to the right of the decmial point associated with TRK_VAL.
FF Data Type	DS-68
Usage	C/Scaling
Length	11

Valid Views	VIEW_4
Storage	Static
Remarks	The Data type consists of data that describes floating point values for display purposes. It uses the following four elements. •1 Engineering Units at 100 percent •2 Engineering Units at 0 percent •3 Units Index •4 Decimal Point

Table A.128 TRK_VAL

Classification	Record
Description	Represents the input value for external tracking.
FF Data Type	DS-65
Usage	I/Input
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	Read Only The data type consists of the value and status of floating point parameters that are Inputs or Outputs. It uses the following two elements. • Status • Value

Uxxx Blocks

Table A.129 UPDATE_EVT

Classification	Record
Description	Represents an alert generated by any change to the static data.
FF Data Type	DS-73
Usage	C/Event Update
Length	1, 4
Storage	Dynamic
Remarks	Read Only The data type consists of data that describes a static revision alarm. It uses the following five elements. • Unacknowledged • Update State •Time Stamp • Static Revision • Relative Index

Vxxx Blocks

No V parameters

Wxxx Blocks

Table A.130 WRITE_ALM

Classification	Record
Description	Represents alert that is generated if the write lock is cleared.
FF Data Type	DS-72
Usage	C/Alarm
Length	1, 3
Storage	Dynamic
Remarks	Read Only The data type describes discrete alarms using the following five elements: • Unacknowledged • Alarm State • Time Stamp • Subcode • Value

Table A.131 WRITE_LOCK

Classification	Simple Variable
Description	Used to disallow writes from anywhere, except to unlock/clear this parameter, when it is locked/set. Inputs will continue to read.
FF Data Type	Unsigned 8
Range	1: Unlocked 2: Locked
Usage	C/Contained
Length	1
Valid Views	VIEW_2
Storage	Static
Remarks	The operator can control PROGRAM and LOCAL access to this value.

Table A.132 WRITE_PRI

Classification	Simple Variable
Description	Defines the priority of the alarm generated by clearing the WRITE_LOCK.
FF Data Type	Unsigned 8
Range	0 to 63
Usage	C/Alert Priority

Length	1
Valid Views	VIEW_4
Storage	Static

Xxxx Blocks

Table A.133 XD_SCALE

Classification	Record
Description	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point used with a specified channel value obtained from the Transducer.
FF Data Type	DS-68
Usage	C/Scaling
Length	11
Valid Views	VIEW_2
Storage	Static
Remarks	The Data type consists of data that describes floating point values for display purposes. It uses the following four elements. • 1 Engineering Units at 100 percent • 2 Engineering Units at 0 percent • 3 Units Index • 4 Decimal Point

Table A.134 XD_STATE

Classification	Simple Variable
Description	Represents index to the text describing the states of a discrete for the value obtained from the Transducer.
FF Data Type	Unsigned 16
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static

Yxxx Blocks

No Y parameters

Zxxx Blocks

No Z parameters

Fieldbus Status Display Indications

Table B.1

Fieldbus Status Byte			Suffix	Independent Status Field
Quality	**Substatus**	**Limits**		
BAD	Non-Specific	No Limits	B	Bad
BAD	Non-Specific	Low Limit	B	Bad,LowLimit
BAD	Non-Specific	High Limit	B	Bad,HiLimit
BAD	Non-Specific	Constant	B	Bad,Constant
BAD	Configuration Error	No Limits	B	Bad,Config
BAD	Configuration Error	Low Limit	B	Bad,Config,L
BAD	Configuration Error	High Limit	B	Bad,Config,H
BAD	Configuration Error	Constant	B	Bad,Config,C
BAD	Not Connected	No Limits	B	Bad,NotCon
BAD	Not Connected	Low Limit	B	Bad,NotCon,L
BAD	Not Connected	High Limit	B	Bad,NotCon,H
BAD	Not Connected	Constant	B	Bad, NotCon,C
BAD	Device Failure	No Limits	B	Bad,DevFail
BAD	Device Failure	Low Limit	B	Bad,DevFail,L
BAD	Device Failure	High Limit	B	Bad,DevFail,H
BAD	Device Failure	Constant	B	Bad,DevFail,C
BAD	Sensor Failure	No Limits	B	Bad,SensFail
BAD	Sensor Failure	Low Limit	B	Bad,SnFail,L
BAD	Sensor Failure	High Limit	B	Bad,SnFail,H
BAD	Sensor Failure	Constant	B	Bad,SnFail,C
BAD	No Comm, Last Usable	No Limits	B	Bad,LUValue
BAD	No Comm, Last Usable	Low Limit	B	Bad,LUV,L
BAD	No Comm, Last Usable	High Limit	B	Bad,LUV,H
BAD	No Comm, Last Usable	Constant	B	Bad,LUV,C
BAD	No Comm, No Last Usable	No Limits	B	Bad,NoComm
BAD	No Comm, No Last Usable	Low Limit	B	Bad,NoComm,L
BAD	No Comm, No Last Usable	High Limit	B	Bad,NoComm,H
BAD	No Comm, No Last Usable	Constant	B	Bad,NoComm,C
BAD	Out-Of-Service	No Limit	B	Bad,OutSvc

Table B.1

Fieldbus Status Byte			Suffix	Independent Status Field
Quality	**Substatus**	**Limits**		
BAD	Out-Of-Service	Low Limit	B	Bad,OutSvc,L
BAD	Out-Of-Service	High Limit	B	Bad,OutSvc,H
BAD	Out-Of Service	Constant	B	Bad,OutSvc,C
UNCERTAIN	Non-Specific	No Limits	U	Uncertain
UNCERTAIN	Non-Specific	Low Limit	U	Uncertain,L
UNCERTAIN	Non-Specific	High Limit	U	Uncertain,H
UNCERTAIN	Non-Specific	Constant	U	Uncertain,C
UNCERTAIN	Last Usable	No Limits	U	Unc,LastOK
UNCERTAIN	Last Usable	Low Limit	U	Unc,LastOK,L
UNCERTAIN	Last Usable	High Limit	U	Unc,LastOK,H
UNCERTAIN	Last Usable	Constant	U	Unc,LastOK,C
UNCERTAIN	Substitute	No Limits	U	Unc,Subst
UNCERTAIN	Substitute	Low Limit	U	Unc,Subst,L
UNCERTAIN	Substitute	High Limit	U	Unc,Subst,H
UNCERTAIN	Substitute	Constant	U	Unc,Subst,C
UNCERTAIN	Initial Value	No Limits	U	Unc,Initial
UNCERTAIN	Initial Value	Low Limits	U	Unc,Init,L
UNCERTAIN	Initial Value	High Limits	U	Unc,Init,H
UNCERTAIN	Initial Value	Constant	U	Unc,Init,C
UNCERTAIN	Sensor Conversion Not Accurate	No Limit	U	Unc,NotAcc
UNCERTAIN	Sensor Conversion Not Accurate	Low Limit	U	Unc,NotAcc,L
UNCERTAIN	Sensor Conversion Not Accurate	High Limit	U	Unc,NotAcc,H
UNCERTAIN	Sensor Conversion Not Accurate	Constant	U	Unc,NotAcc,C
UNCERTAIN	Engineering Unit Range Violation	No Limits	U	Unc,Range
UNCERTAIN	Engineering Unit Range Violation	Low Limit	U	Unc,Range,L
UNCERTAIN	Engineering Unit Range Violation	High Limit	U	Unc,Range,H
UNCERTAIN	Engineering Unit Range Violation	Constant	U	Unc,Range,C
UNCERTAIN	Sub-Normal	No Limits	U	Unc,SubNorm
UNCERTAIN	Sub-Normal	Low Limit	U	Unc,SubNrm,L
UNCERTAIN	Sub-Normal	High Limit	U	Unc,SubNrm,H
UNCERTAIN	Sub-Normal	Constant	U	Unc,SubNrm,C
GOOD (Non-Cascade)	Non-Specific	No Limits	none	none
GOOD (Non-Cascade)	Non-Specific	Low Limit	L	Low Limit
GOOD (Non-Cascade)	Non-Specific	High Limit	H	High Limit

Table B.1

Fieldbus Status Byte			Suffix	Independent Status Field
Quality	**Substatus**	**Limits**		
GOOD (Non-Cascade)	Non-Specific	Constant	C	Constant
GOOD (Non-Cascade)	Active Block Alarm	No Limits	A	Alarm-Block
GOOD (Non-Cascade)	Active Block Alarm	Low Limit	A	Alarm-Blck,L
GOOD (Non-Cascade)	Active Block Alarm	High Limit	A	Alarm-Blck,H
GOOD (Non-Cascade)	Active Block Alarm	Constant	A	Alarm-Blck,C
GOOD (Non-Cascade)	Active Advisory Alarm	No Limits	A	Alarm-Advis
GOOD (Non-Cascade)	Active Advisory Alarm	Low Limit	A	Alarm-Advis,L
GOOD (Non-Cascade)	Active Advisory Alarm	High Limit	A	Alarm-Advis,H
GOOD (Non-Cascade)	Active Advisory Alarm	Constant	A	Alarm-Advis,C
GOOD (Non-Cascade)	Active Critical Alarm	No Limits	A	Alarm-Crit
GOOD (Non-Cascade)	Active Critical Alarm	Low Limit	A	Alarm-Crit,L
GOOD (Non-Cascade)	Active Critical Alarm	High Limit	A	Alarm-Crit,H
GOOD (Non-Cascade)	Active Critical Alarm	Constant	A	Alarm-Crit,C
GOOD (Non-Cascade)	Unacknow-ledged Block Alarm	No Limits	A	Alm-U-Block
GOOD (Non-Cascade)	Unacknow-ledged Block Alarm	Low Limit	A	Alm-U-Blck,L
GOOD (Non-Cascade)	Unacknow-ledged Block Alarm	High Limit	A	Alm-U-Blck,H
GOOD (Non-Cascade)	Unacknow-ledged Block Alarm	Constant	A	Alm-U-Blck,C
GOOD (Non-Cascade)	Unacknow-ledged Advisory Alarm	No Limits	A	Alm-U-Advis
GOOD (Non-Cascade)	Unacknow-ledged Advisory Alarm	Low Limit	A	Alm-U-Advs,L
GOOD (Non-Cascade)	Unacknow-ledged Advisory Alarm	High Limit	A	Alm-U-Advs,H
GOOD (Non-Cascade)	Unacknow-ledged Advisory Alarm	Constant	A	Alm-U-Advs,C
GOOD (Non-Cascade)	Unacknow-ledged Critical Alarm	No Limits	A	Alm-U-Crit
GOOD (Non-Cascade)	Unacknow-ledged Critical Alarm	Low Limit	A	Alm-U-Crit,L
GOOD (Non-Cascade)	Unacknow-ledged Critical Alarm	High Limit	A	Alm-U-Crit,H
GOOD (Non-Cascade)	Unacknow-ledged Critical Alarm	Constant	A	Alm-U-Crit,C
GOOD (Cascade)	Non-Specific	No Limits	none	none
GOOD (Cascade)	Non-Specific	Low Limit	L	Low Limit
GOOD (Cascade)	Non-Specific	High Limit	H	High Limit
GOOD (Cascade)	Non-Specific	Constant	C	Constant
GOOD (Cascade)	Acknowledge Initialization (IA)	No Limit		InitAck
GOOD (Cascade)	Acknowledge Initialization (IA)	Low Limit		InitAck,L
GOOD (Cascade)	Acknowledge Initialization (IA)	High Limit		InitAck,H
GOOD (Cascade)	Acknowledge Initialization (IA)	Constant		InitAck,C
GOOD (Cascade)	Request Initialization (IR)	No Limits	R	ReqInit

Table B.1

Fieldbus Status Byte			Suffix	Independent Status Field
Quality	**Substatus**	**Limits**		
GOOD (Cascade)	Request Initialization (IR)	Low Limit	R	ReqInit,L
GOOD (Cascade)	Request Initialization (IR)	High Limit	R	ReqInit,H
GOOD (Cascade)	Request Initialization (IR)	Constant	R	ReqInit,C
GOOD (Cascade)	Not Invited (NI)	No Limits	N	NotInvited
GOOD (Cascade)	Not Invited (NI)	Low Limit	N	NotInvited,L
GOOD (Cascade)	Not Invited (NI)	High Limit	N	NotInvited,H
GOOD (Cascade)	Not Invited (NI)	Constant	N	NotInvited,C
GOOD (Cascade)	Not Selected (NS)	No Limits	N	NotSelected
GOOD (Cascade)	Not Selected (NS)	Low Limit	N	NotSelectd,L
GOOD (Cascade)	Not Selected (NS)	High Limit	N	NotSelectd,H
GOOD (Cascade)	Not Selected (NS)	Constant	N	NotSelectd,C
GOOD (Cascade)	Local Override (LO)	No Limits	O	OverrideLocl
GOOD (Cascade)	Local Override (LO)	Low Limit	O	OverrideLc,L
GOOD (Cascade)	Local Override (LO)	High Limit	O	OverrideLc,H
GOOD (Cascade)	Local Override (LO)	Constant	O	OverrideLc,C
GOOD (Cascade)	Fault-State Active (FSA)	No Limits	F	FaultState
GOOD (Cascade)	Fault-State Active (FSA)	Low Limit	F	FaultState,L
GOOD (Cascade)	Fault-State Active (FSA)	High Limit	F	FaultState,H
GOOD (Cascade)	Fault-State Active (FSA)	Constant	F	FaultState,C
GOOD (Cascade)	Initiate Fault-State (IFS)	No Limits	none	InitFaultSt
GOOD (Cascade)	Initiate Fault-State (IFS)	Low Limit	L	LoLim,InitFS
GOOD (Cascade)	Initiate Fault-State (IFS)	High Limit	H	HiLim,InitFS
GOOD (Cascade)	Initiate Fault-State (IFS)	Constant	C	Const,InitFS

Mode Change Conditions

Reference

Input Parameters Status Attributes, Inputs Values and Resource States	Mode Parameter - Target Attribute Value					
	Out of Service (OOS)	**Manual (MAN)**	**Automatic (Auto)**	**Cascade (CAS)**	**Remote Cascade (RCAS)**	**Remote Output (ROUT)**
All input parameters have Good Status or Uncertain status with option set to treat Uncertain as Good.	OOS	MAN	AUTO	CAS	RCAS	ROUT
Remote cascade in has BAD status or time-out is detected and the shed option parameter value is Normal.	OOS	MAN	AUTO	CAS	Shed to next permitted or target mode(1)	ROUT
Remote cascade in time-out is detected and the shed option parameter value is not set to Normal.	OOS	MAN	AUTO	CAS	Set target and actual mode as specified	ROUT
Remote out in has BAD status or time-out is detected and the shed option parameter value is normal.	OOS	MAN	AUTO	CAS	RCAS	Shed to next permitted or target mode*
Remote out in time-out is detected and the shed option parameter value is not set to Normal.	OOS	MAN	AUTO	CAS	RCAS	Set target and actual mode as specified
Cascade input has a status attribute of BAD	OOS	MAN	AUTO	Shed to next permitted mode	RCAS	ROUT
Target mode is RCAS and remote cascade in does not have status of Good - Initialization Acknowledge; and actual mode attribute last execution was not RCAS.	N/A	N/A	N/A	N/A	Actual mode from last execution or AUTO	N/AF
Target mode is ROUT and remote out in status is not Good - Initialization Acknowledge; and actual mode attribute last execution was not ROUT.	N/A	N/A	N/A	N/A	N/A	Actual mode from last execution or MAN

Input Parameters Status Attributes, Inputs Values and Resource States	Mode Parameter - Target Attribute Value					
	Out of Service (OOS)	**Manual (MAN)**	**Automatic (Auto)**	**Cascade (CAS)**	**Remote Cascade (RCAS)**	**Remote Output (ROUT)**
Target mode is CAS and cascade input status is not Good - Initialization Acknowledge and actual mode attribute last execution was not CAS	N/A	N/A	N/A	Actual mode from last execution or AUTO	N/A	N/A
Status attribute of primary input parameter is BAD or Uncertain with option to treat Uncertain as BAD and bypass not set	OOS	MAN	MAN	MAN	MAN	ROUT
Status attribute of back calculation input parameter is BAD	OOS	IMAN	IMAN	IMAN	IMAN	IMAN
Status attribute of back calculation input parameter Good - failsafe active, local override, not invited, or initialization request	OOS	IMAN	IMAN	IMAN	IMAN	IMAN
Tracking is enabled, track input active and track override manual is Enabled.	OOS	LO	LO	LO	LO	LO
Tracking is enabled, initiated and track override manual is Disabled.	OOS	MAN	LO	LO	LO	LO
Failsafe is active in an output function block	OOS	LO	LO	LO	LO	LO
If target mode has changed from OOS to another mode since the block was last executed.	-	MAN	MAN	MAN	MAN	MAN
Resource state as reflected in the resource block parameter resource state is Standby.	OOS	OOS	OOS	OOS	OOS	OOS

(1) Shed to the next lowest priority, non-remote mode which is permitted or, optionally, which is retained by the target mode attribute and for which the required inputs are available.

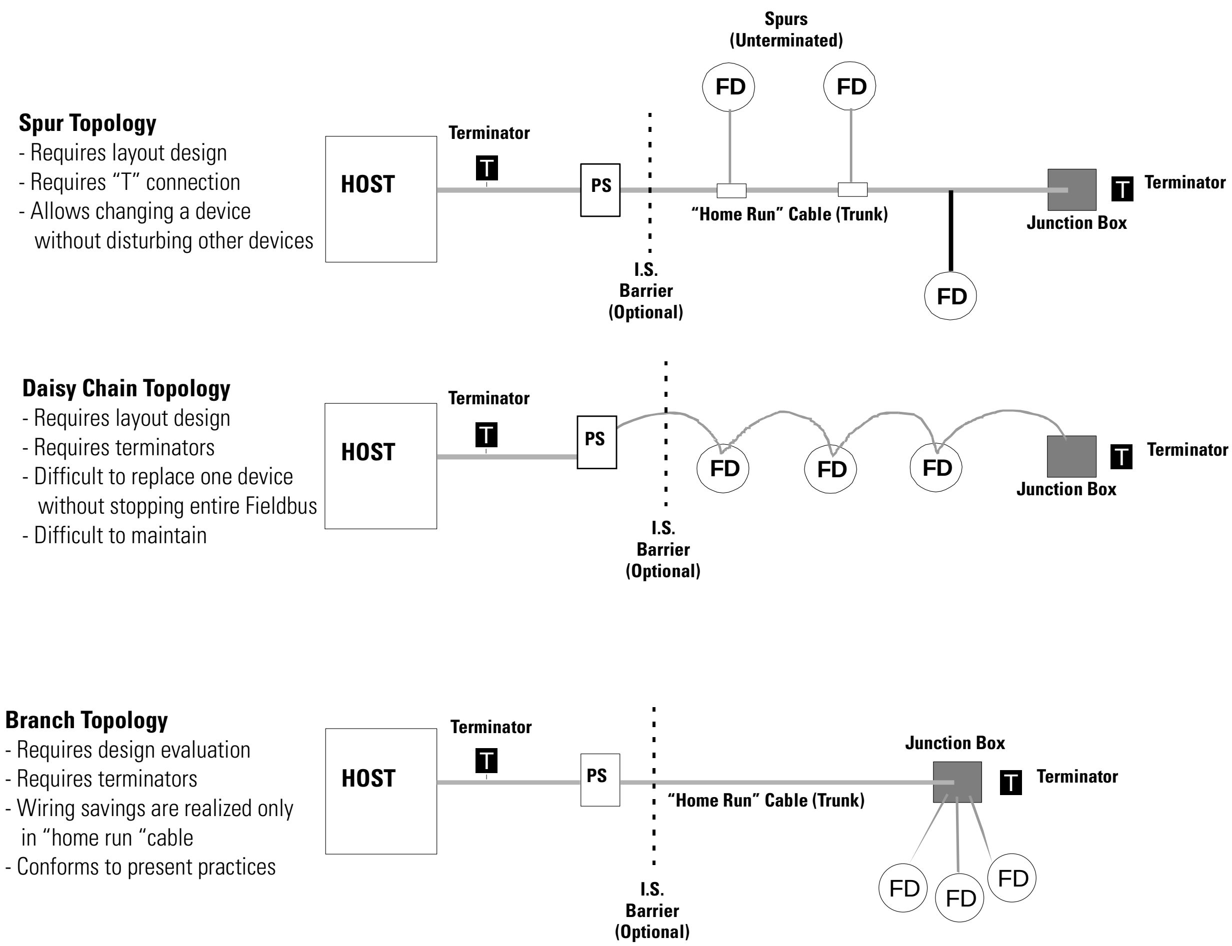
Fieldbus Wiring Considerations

The following wiring information is for general purposes only. Refer to each device's cooresponding wiring and installation instructions.

Fieldbus Topologies

Figure D.1 illustrates the Spur, Daisy Chain, and Tree type wiring topologies that can be used to connect fieldbus devices to one another and a host.

Figure D.1 Overview of fieldbus wiring topologies



TIP

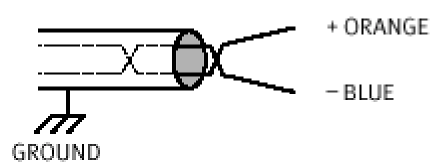
Maximum or acceptable cable distances may be significantly reduced by using Non-FF spec wire.

Power Conditioning

You must use a power conditioner between your Fieldbus power supply and the Fieldbus netwrok. You can use a power supply designed for Foundation Fieldbus operation which has the proper power conditioning elements. If you are using an ordinary power supply, a separate power conditioner must also be used. If an ordinary power supply is connected directly to the Fieldbus, the power supply would absorb signals on the cable because it would try to maintain a constant voltage level. The power conditioner puts an inductor between the power supply and the Fieldbus wiring. The inductor connects the DC power to the Fieldbus wiring but prevents signals from going into the power supply.

In practice, a real inductor is not used in the power conditioner but an electronic equivalent. The electronic inductor circuit has the added advantage of limiting the current provided to the network segment if the cable is shorted.

The voltage supplied to the Fieldbus cable can be as high as 32 V. The voltage at any device can be as low as 9 V for the device to operate correctly. A typical Fieldbus device takes about 20 mA of current from the cable. The Fieldbus is configured so that one of the wires has a (+) voltage, the other wire has a (-) voltage and the shield is grounded.



A cable with the orange wire as plus and the blue wire as minus is shown above. This type of cable is available from Fieldbus cable manufacturers. Other cables or existing plant wiring conventions may be different. Regardless of the color convention, keep the sense of Fieldbus polarity consistent throughout the plant." [1]

IMPORTANT We suggest that you not use White/Black or White/Red pairs since they may be mistaken for 115 volt power wiring.

(1) RELCOM inc., Fieldbus Wiring Design and Installation Guide (2221 Yew Street, Forest Grove, OR 97116) p. 5. Reprinted by permission.

Power Distribution

The design of a fieldbus network requires understanding of the electrical requirements of the devices (current/voltage) and the properties of the cable use (resistance). The number of devices on a fieldbus segment is limited by Ohm's law:

E = I x R

Where:

E = The voltage of the dc power supply

I = The amount of current in amperes drawn by each device (typically 20mA)

R = The resistance of the cable in ohms

EXAMPLE

Assume that the characteristics of the segment are a power supply output of 20Vdc, 18 AWG cable with a resistance of 22 ohms per kilometer (3281 feet), and a “home run” (trunk) cable that is 1 km (3,281 ft.) long. This results in a combined resistance of 44 ohms for both wires.

If each device at the branch draws 20mA and needs a minimum of 9Vdc, the cable can use up to 11Vdc (20 - 9 = 11) and the total current that can be supplied at the chickenfoot is 250mA (11Vdc/44 ohms = 250mA). This means the maximum number of bus powered devices that can be used at this example chickenfoot is 12 (250mA/20mA = 12 devices).

Be sure to check the fieldbus device specifications to determine the power requirements, since they can vary by device type and manufacturer. A physical test can (should) be performed on each H1 segment prior to power up to check for shorts and verify resistance. Test the power carrying capability of the fieldbus cable by connecting the wires together at one end of the cable and measuring the resistance across the wires at the other end with an ohmmeter.

Signal Degradation Limitations

One approach to minimizing signal distortion is to apply “rules of thumb” that suggest how long a fieldbus cable can be and still get adequate signal quality. This is based on published Fieldbus Foundation wiring guidelines.

The following table lists limits for some possible fieldbus cable types listed in the order of usage preference with 1 being preferred.

Table D.2 Fieldbus Cable Limits

Preference Type	Description	Distance (Meters/Feet)	Resistance (Ohms per km)	Attenuation (dB per km)
1	#18 AWG, shielded, twisted pair	1900/6233	22	3
2	#22 AWG, multiple-twisted, pairs with overall shield	1200/3937	56	5
3	#26 AWG, multiple-twisted, pairs without shield	400/1312	132	8
4	#16 AWG, multiple conductor, no twisted pairs with overall shield	200/656	20	8

Another rule of thumb based on Fieldbus Foundation published guidelines is for the length of spurs and how many devices can be on various lengths of cable as listed in the following table.

Table D.3 Number of devices on the fieldbus and maximum spur length

Number of Devices	Maximum Spur Length (Meter/Feet)
1 to 12	120/394 (one device per Spur)
13 to 14	90/295 (one device per Spur)
15 to 18	60/197 (one device per Spur)
19 to 24	30/98 (one device per Spur)
25 to 32	No spurs allowed

ATTENTION

The spur length limitations are for spurs with one device each. Please refer to the Fieldbus Foundation AG-140 Wiring and Installation Guide for rules on multiple devices per spur.

Cable Guidelines

A quick method to determine if the segment is within limits is to calculate a segment ratio. Total cable length is the sum of trunk and spur lengths. If different cable preference types are used, apportion each type against its limit and be sure the total is less than one (1).

EXAMPLE

For example, if the 500m (1640ft) trunk cable is preferred type 1 cable and the spurs are 100m (328ft) preferred type 2 cable, the total cable resistance equals 0.35 (500/1900 + 100/1200 = 0.35).

0.35 is less than 1.0, therefore segment resistance is good.

Cable Attenuation

Signals attenuate (or get smaller) as they travel though cables. Attenuation is measured in decibels (dB) based on the following formula:

dB = 20 log (V1/V2)

Where:

V1 = Amplitude of Signal Transmitted in Volts

V2 = Amplitude of Signal by device in Volts

- Cables have different attenuation ratings for a given frequency. The frequency of interest for fieldbus is 39 kHz.
- The preferred type 1 cable for fieldbus has an attenuation of 3 dB/km.
- A fieldbus device can transmit a signal as low as 0.75 volts peak-to-peak and detect a signal as small as 0.15 volts peak-to-peak. This means that the cable can attenuate the signal by 14 dB (20 log (0.75/0.15) = 14 dB).

Based on using the preferred type 1 cable a fieldbus cable run could be up to 4.6 km (15091 ft.) long, as determined from this formula:

14 dB / 3 dB/km = 4.6 km

A shorter cable will have proportionately less attenuation. For example, a preferred type 1 cable that is 500 m (1640 ft.) long would have an attenuation of 1.5 dB.

Signal Distortion vs Capacitance

Other factors such as varying characteristic impedance, spur connection reflections, and capacitive unbalance between the wires and shield can distort signals as they travel though the cable. This means an ideal signal transmission that is within fieldbus specifications can arrive at the other end of the cable as a distorted signal.

In addition to cable attenuation, calculations for determining the maximum fieldbus cable length must also use the following guidelines to calculate signal distortion due to spurs.

- Each spur is converted to an equivalent capacitance.
- All spurs are assumed to be at the extreme end of the trunk.
- Signal attenuation is calculated as 0.035 dB/nf.

For example, if the fieldbus topology includes five devices each on a 20 m (66 ft.) spur, the total cable length is 100 m (328 ft.), equivalent capacitance is 3 nf (100 m x 0.03 nf/m), and spur distortion is 0.105 dB (3 nf x 0.035 dB/nf = 0.105 dB).

Another source of distortion is the fieldbus device's equivalent capacitance. Use the following guidelines to calculate the distortion due to device capacitance.

- Determine each device's input capacitance.
- Assume all devices are at the extreme end of the trunk.
- Use 0.035 dB/nf as signal attenuation calculation.

EXAMPLE For example, if the fieldbus topology includes five devices each with a 3000 pF input capacitance, the equivalent capacitance is 15 nf (5 x 300 pF) and the resulting attenuation is 0.525 dB (15 nF x 0.035 dB/nF).

Calculating Attenuation

Take the sum of the following calculations to determine if the attenuation will exceed the allowed level of 14 dB.

- Calculate the cable attenuation.
- Calculate the attenuation due to spurs.
- Calculate the attenuation due to device capacitance.

EXAMPLE For example, the total attenuation for the fieldbus topology used in the previous examples in this section would be 2.13 dB (1.5 dB (cable attenuation) + 0.105 (spur attenuation) + 0.525 (device capacitance attenuation) = 2.13 dB).

Testing the Cable

Use a fieldbus tester and signal generator to determine if existing cable is suitable for fieldbus use or not. The tester and signal generator work together to determine the resistance of the wires and measure the signal quality of the transmission. This testing also determines if the wiring polarity is consistent.

Repeaters

Repeaters can be used in a Fielbus cable system. There can be a maximum of four repeaters between any two nodes on a Fieldbus. If repeaters are used, there must be terminators on each side of the repeaters. A Fieldbus with one repeater will use four terminators, you terminate each Fieldbus segment.

Notes:

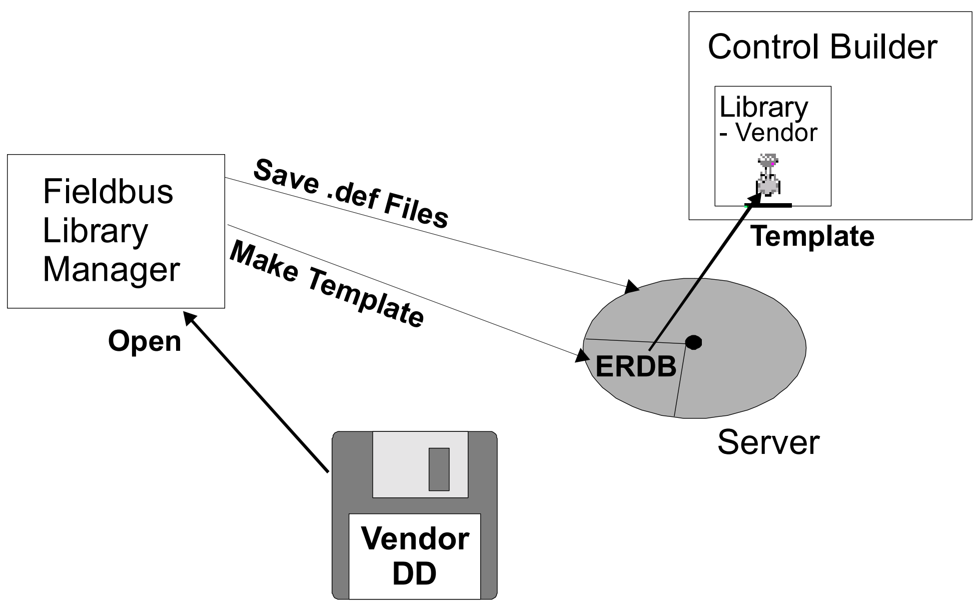
Fieldbus Library Manager

About Fieldbus Library Manager

The Fieldbus Library Manager (FLM) is an Engineering Tools utility for reading the vendor supplied Device Description (DD) files for fieldbus devices and creating device templates to be stored in the Engineering RepositoryDatabase(ERDB).Thestoredtemplatesareaccessiblethroughthe Library tab in Control Builder, where they are cataloged in vendor named directories. Figure E.1 shows a simplified graphical representation of the process.

It reads the DD binary files from a user supplied floppy diskette or a Foundation Fieldbus Compact Disc in an offline mode. It does not read data directly from an online fieldbus device. DD's can also be downloaded from www.fieldbus.org.

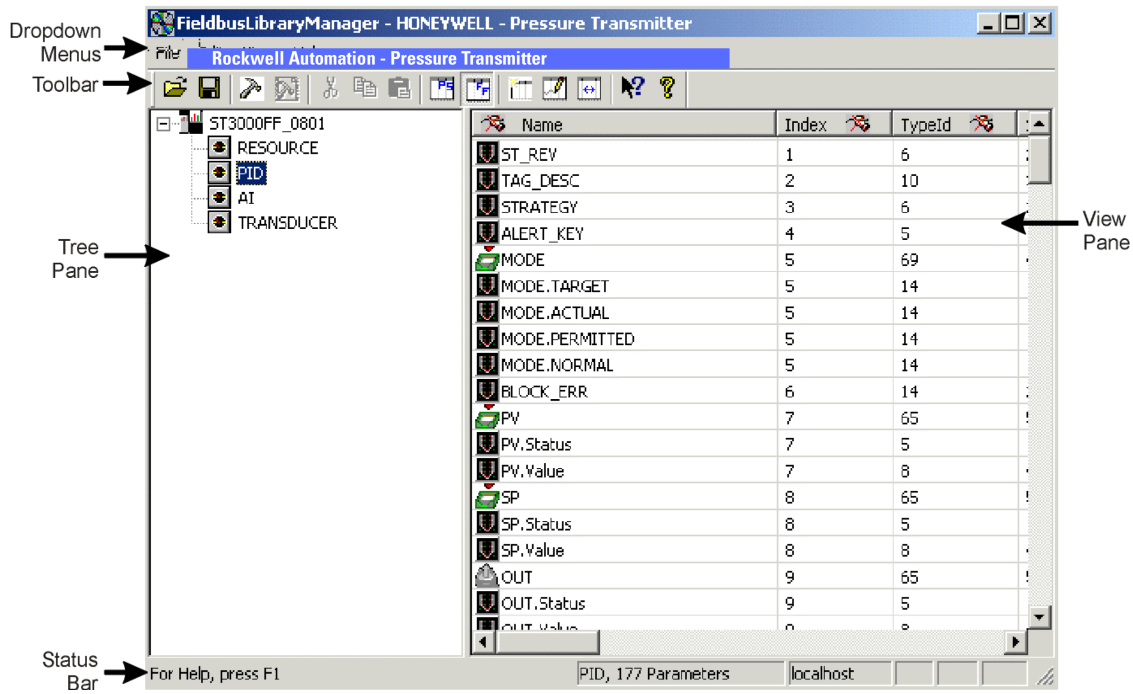
Figure E.1 Fieldbus Library Manager uses vendor DD file to create device template for Control Builder



Description

The FLM features a Windows type interface with drop-down menus, toolbar, directory tree pane, and view pane as shown in Figure E.2.

Figure E.2 Fieldbus Library Manager features common Windows type interface.



Menu and toolbar selections

Table E.1 The following table summarizes the functions you can initiate through a given menu selection or toolbar button for reference. If you are familiar with working in a Windows type environment, you should be able to intuitively interact with all the functions provided through FLM menu and toolbar selections.

Table E.2 Menu and toolbar selection summary

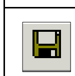
Click:	Or, follow this menu selection:	To perform this function:
	File->Open Device (Ctrl+O)	Opens Select Device dialog box so you can navigate to the \Release directory containing the DD file.
	File->Save (Ctrl+S)	Save the data to the default .DEF file directory: ***Rockwell Automation\tps50\system\er\ffdevices***
	File->Save As	Opens Select Directory dialog box so you can select another directory location where the .DEF file is to be saved.
	File->Build Device Template->From Current Device	Makes template from current open device file.

Table E.2 Menu and toolbar selection summary

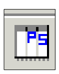
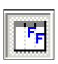
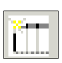
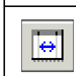
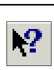
Click:	Or, follow this menu selection:	To perform this function:
	File->Build Device Template->From Existing .DEF Files	Opens dialog box so you can navigate to the directory containing the desired device .DEF files. Default directory is: ***Rockwell Automation\tps50\system\er\ffdevices***
	File->Login to server	Opens Login Information dialog box so you can login to another ProcessLogix Server in your system.
	Edit->Undo (Ctrl+Z)	Undo the last action.
	Edit->Cut (Ctrl+X)	Cut selected data from current location.
	Edit->Copy (Ctrl+C)	Copy selected data and store it on the clipboard.
	Edit->Paste (Ctrl+V)	Paste previous cut or copied data to selected location.
	View->Select View->ProcessLogix Display	Select ProcessLogix parameter data for viewing in the view pane.
	View->Select View-> FF Device Display	Select Fieldbus Foundation parameter data for viewing in the view pane.
	View->Select View->"Custom"	Select user customized view as named by user.
	View->Organize Views->Add View	Opens Add New View dialog box so you can enter name for a customized view you want to create. Customize the new view by selecting desired attributes through the Select View Attributes dialog box. The new custom view name is added to the Select View menu selections.
	View->Organize Views->Delete View	Opens Select View to Delete dialog box so you can delete the selected customized view.
	View->Organize Views->Modify View	Opens Select View to Modify dialog box so you can modify selected view through the Select View Attributes dialog box.
	View->Toolbar	Toggle Toolbar view On/Off.
	View->Status Bar	Toggle Status Bar view On/Off.
	View->Optimize View Columns	Automatically adjusts columns for optimum viewing.
	Help->Help Topics	Calls up online Help topics.

Table E.2 Menu and toolbar selection summary

Click:	Or, follow this menu selection:	To perform this function:
	Help->About Fieldbus Library Manager	Opens About Fieldbus Library Manager dialog box for general information about the utility.

1788-CN2FF Installation Example

Overview

This installation example provides a step by step approach to the installation of the 1788-CN2FF. For more information on the operation the 1788-CN2FF, and its companion software, the 1788-FFCT, Foundation Fieldbus Configuration Tool refer to Chapter 7, Using the 1788-CN2FF, ControlNet-to-FOUNDATION Fieldbus H1 Linking Device.

For more information on Foundation Fieldbus visit www.Fieldbus.org.

Table F.1 Topics Covered in the Installation Example

Required Hardware for Installation Example

❑ One 1788-CN2FF, ControlNet to Fieldbus Linking Device

❑ One 24 Power Supply for the CN2FF, such as 1794-PS1

1794-PS1 is a good power supply for use with the CN2FF, and is OK to use for a small demo with one or two Fieldbus devices, but it is too noisy for a real control application.

❑ FF convenience Relcom Connector Blocks.

Connector block should include built in power conditioner and two Fieldbus terminators.

❑ One Personal Computer, 233 MHz or faster, 128 Meg Memory

❑ With Windows – NT, Service Pack 3 or later.

You must use Windows NT/2000. The 1788-FFCT software is not compatible with Windows 95/98.

❑ High Resolution Monitor, 1200 x 1024 recommended.

Use at least 1024 x 800. The Fieldbus Configurator is much easier to use with a high resolution monitor. If you use a 640 X 480 screen, you will be unable to find some of the items on the toolbars.

❑ 1784-PCIC, 1784-KTCX15 or 1784-PCC card, for direct connection to ControlNet.

❑ PLC-5, or CLX, or PLX, or SLC with a ControlNet Interface.

❑ One Operating ControlNet Network supporting the Controller, the PLC-5, CLX or PLX, or SLC, plus a PC with KTCX15, 1784-PCC or a 1784-PCIC and one or more 1788-CN2FF's

❑ One or more Registered Fieldbus devices.

1788-FFCT software requires that the Fieldbus devices be on-line. Off-line configuration will be supported later.

Required Software

- ❑ RSLinx version 2.10.166 or later, OEM version or better.

IMPORTANT RSLinx Lite is not compatible.

- ❑ RSNetworks for ControlNet version 2.25 or later - - to schedule data to the controller.
- ❑ RSLogix 5, RSLogix 500, RSLogix 5000, or ProcessLogix ControlBuilder, to program the controller and to see the FF data in the file in the controller
- ❑ 1788-FFCT, the Foundation Fieldbus Configuration Tool, Version 2.3.6 or later.
- ❑ Device Descriptions, DD's, for the Linking Device, and for each of the Fieldbus Devices you plan to connect to the Fieldbus. DD's that match the FF devices should be supplied by the manufacturers of all Registered Fieldbus devices. The 1788-FFCT must have the DDs for each FF devices to configure that FF device.

Example Description

Imagine that you find yourself in a mill that has been controlled by an old DCS systems that has been updated using Fieldbus. You have been asked if there is any way for one of your systems, ProcessLogix, ControlLogix, or CompactLogix, to interface to the existing Fieldbus devices. Yes, Rockwell Automation offers ControlNet to Fieldbus linking device, called a 1788-CN2FF, which interfaces a ControlNet network with a Fieldbus network.

Connect Fieldbus Devices and Configure Linking Device to Gather Data

In this lab example, you will connect to Fieldbus devices on the Fieldbus network and then configure the 1788-CN2FF Linking device to gather data from the devices.

Use 1788-FFCT Software

Rockwell Automation uses the 1788-FFCT software to configure the Fieldbus part of the system. The software is manufactured by National Instruments.

Schedule Connections

The platform Rockwell uses to schedule the ControlNet connections to the controller is Windows NT with RSNetWorx.

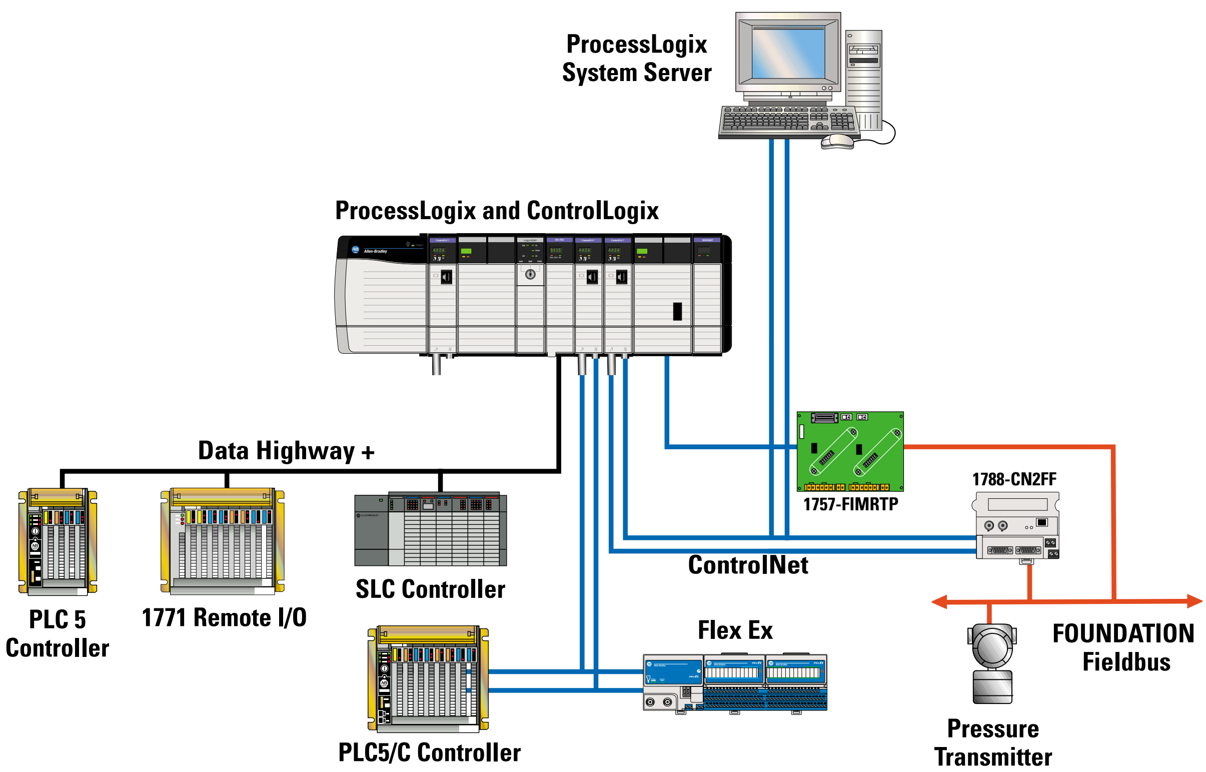
Utilize RSLinx

RSLinx OEM is needed for the 1788-FFCT software. The version of RSLinx that is bundled with many products is not RSLinx OEM. You must have RSLinx OEM or a more complete version of RSLinx for the 1788-FFCT to function.

Control the Process

Use ControlBuilder or RSLogix 5000 to control the process using the data from the existing Fieldbus Devices, connected via the CN2FF Linking Device.

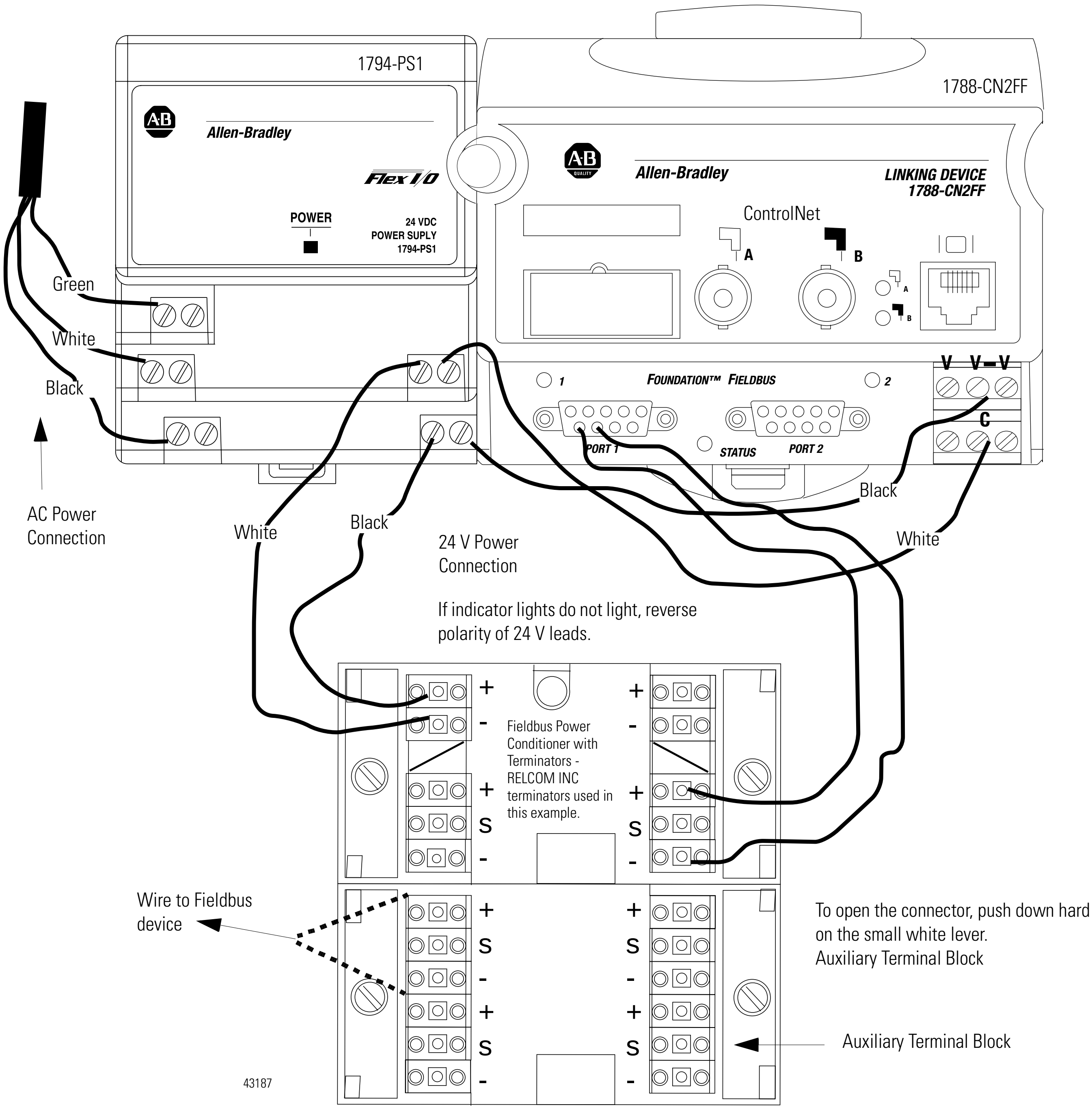
Figure F.1 Example Linking Device Configuration



Connecting the Hardware

1. Wire the 1784-PS1, the 1788-CN2FF, and the Terminal Block as shown in Figure F.2.

Figure F.2 Wiring the 1788-CN2FF



2. Connect the CN2FF and the other Fieldbus Devices to the Relcom terminal block.

The base terminal block supports either two or four device connections (depends on the type of block), plus the power connection. An auxiliary block supports four additional device connections.

IMPORTANT To provide power to a fieldbus device, use our ordinary 24V instrumentation power supply. You **must** also use a Fieldbus Power Conditioner (power isolator),

3. Select a Mac ID for the CN2FF.

4. Open the small door on the top of the 1788-CN2FF and set the rotary switches to an unused ControlNet Mac ID.

   In the examples that follow, Mac ID = 9 has been used. The units digit is on the right when you are facing the CN2FF. Cycle power on the CN2FF after you set the Mac ID.

Install the 1788-FFCT Software

1. Install the 1788-FFCT, Foundation Fieldbus Configuration Tool software.

TIP We suggest that you copy the FFCT files to your hard drive and install from there. Install the software in the default directory. This example is based on these default locations.

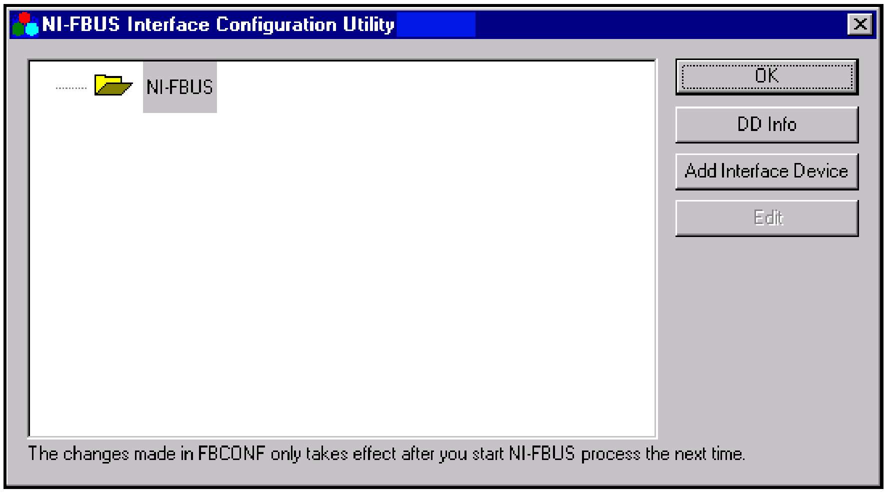
IMPORTANT If you have other National Instruments Fieldbus Monitor Software installed on your PC, the system may try to install this 1788-FFCT software in a directory called Monitor. Don't allow the software to install in that directory, make sure it installs in the NIFBUS directory.

2. Open Disk 1.

3. Click on the Setup with the monitor icon.

4. Accept the defaults during the installation.

When installation is complete, you see:



5. Click **OK**.

6. Follow instructions to restart the computer.

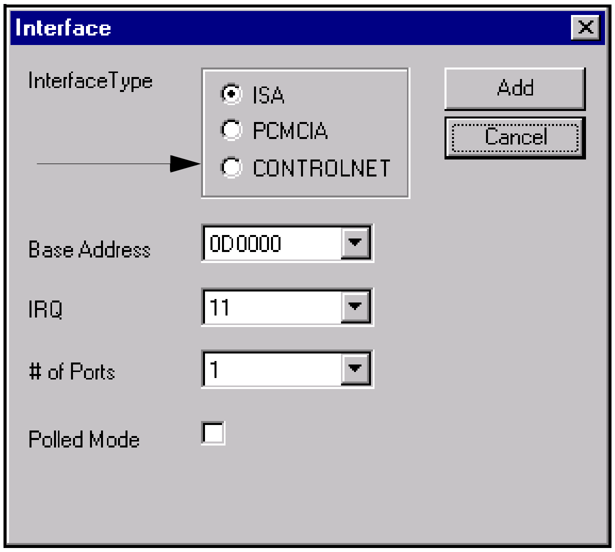
Adding an Interface Device

Whenever you want to add a new device, you will need to go through this procedure. You will do this procedure frequently.

1. Click on **Add Interface Device**.

You see:

Check CONTROLNET



2. For Interface Type check CONTROLNET.

   When you select ControlNet, the dialog box changes to reflect the default ControlNet parameters.

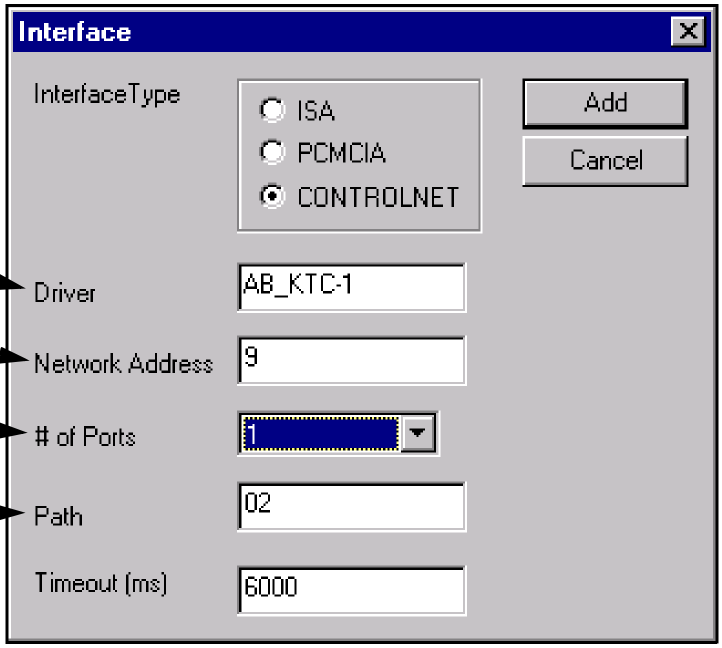
ATTENTION:
The driver name must be entered EXACTLY (case, punctuation, underscores) as it is listed in RSLinx. refer to Finding the Interface Driver Name on page F-9 for an example.

Enter the ControlNet driver name

Enter the ControlNet MAC-ID of the CN2FF.

Enter 1 or 2 depending on the number of Foundation Fieldbus channels attached to the 1788-CN2FF.

This is the default path from the PC to the 1788-CN2FF. Use this path with the PC on the same ControlNet as the 1788-CN2FF.



3. Enter the Driver name for you ControlNet interface, AB_PCIC-1, AB_KTC-1, or AB_PCC.

   If you do not know the name of your driver, follow the procedure in Finding the Interface Driver Name on page F-9.

4. In Network Address, type in the MAC ID of the 1788-CN2FF.

5. Select the number of ports you are using.

   For this example we are only using 1.

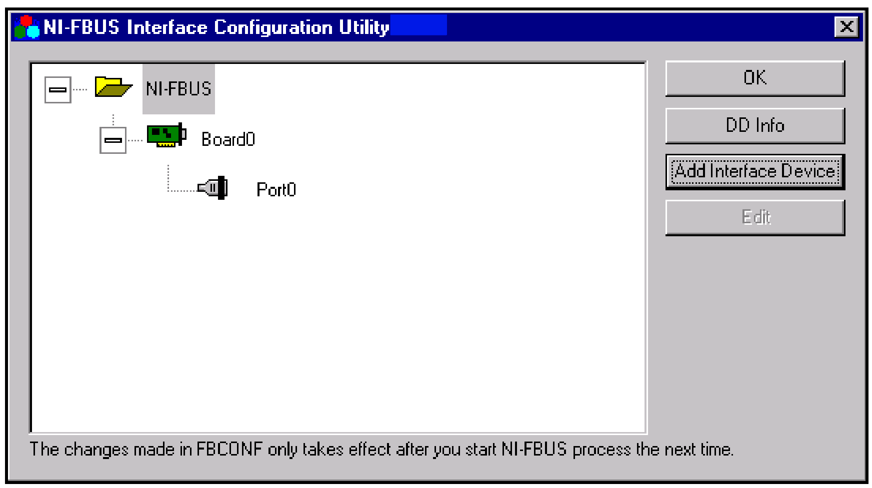
6. If your computer is on the same ControlNet and the 1788-CN2FF, set the Path to 02.

   If your computer is on another ControlNet, use the path suggested by RSLinx. A typical PLX path might be 2 6 1 6 2.

   2(1 space)6(1 space)1(1 space)6(1 space)2

7. Leave the Timeout as defaulted.

Click **Add**.You see:

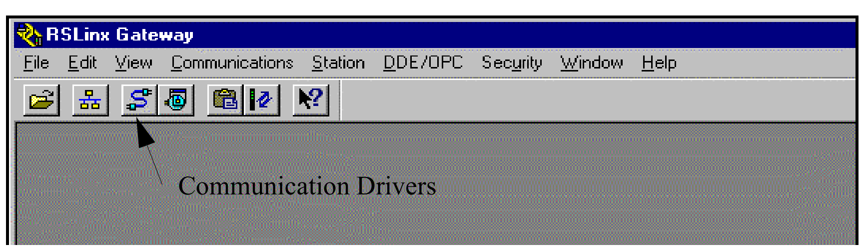


8. Go to Port Configuration on page F-12.

Finding the Interface Driver Name

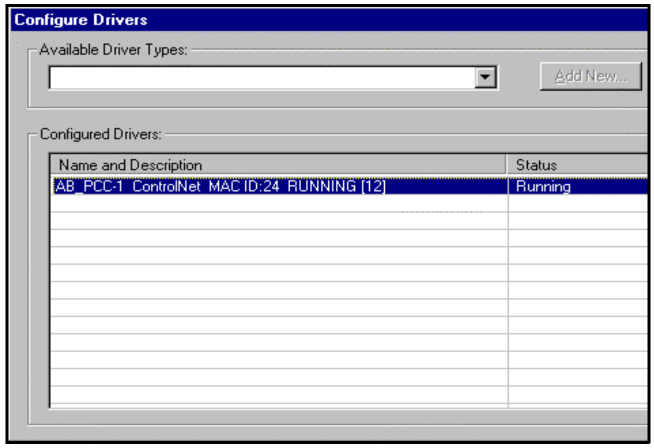
If you don't know the name of your device driver do the following:

1. In RSLinx, select Communication Drivers.:



You see:

As stated on page F-8, driver names must match exactly.



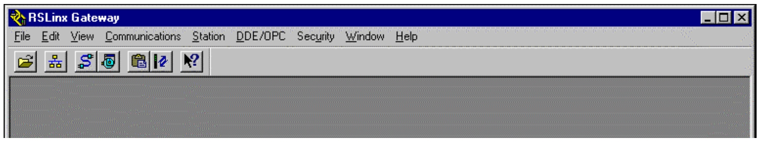
2. Make a note of the ControlNet Driver name and go back to Step 3 on page F-8 and type the name in the Driver Field at the Interface dialog box. The name here is AB_PCIC-1.

Assigning a Path to the 1788-CN2FF

Assigning a Path through which the 1788-FFCT software will communicate with the 1788-CN2FF. If you are familiar with the Path and you know it is correct, you can skip this section.

Use RSLinx to check the communication path for the 1788-CN2FF.

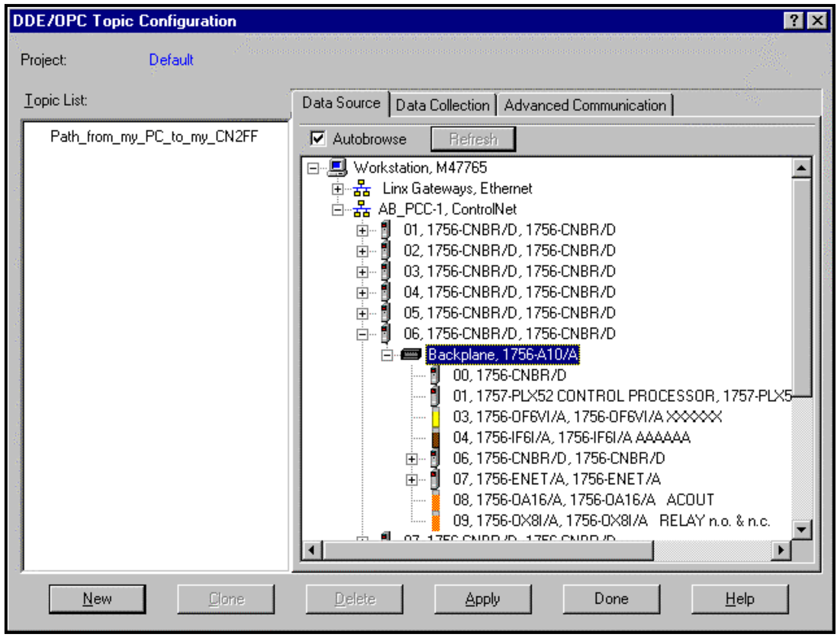
1. Start RSLinx.



2. Click on DDE/OPC.

3. Select Topic Configuration.

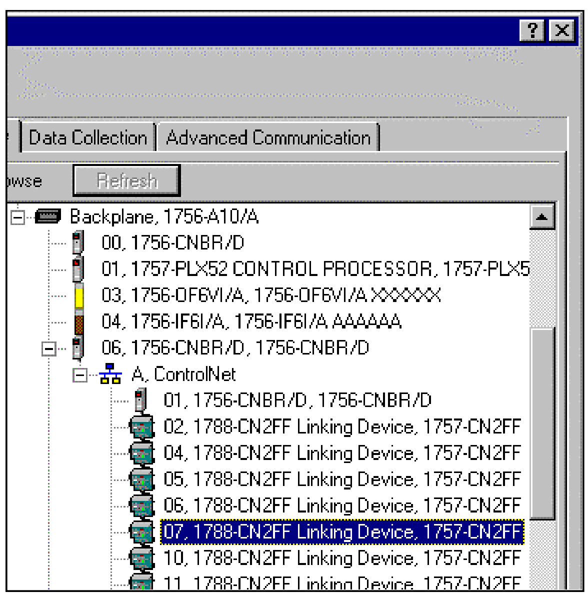
4. Click on New and type in a name.



5. Navigate to the 1788-CN2FF you want to configure.

6. Select the 1788-CN2FF.

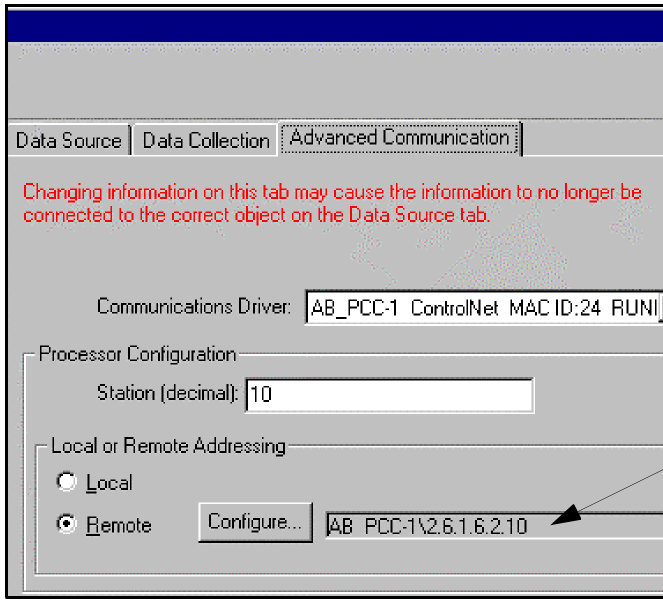
You see:



7. Click **Apply**.

8. Click **Advanced Communications**.

You see:

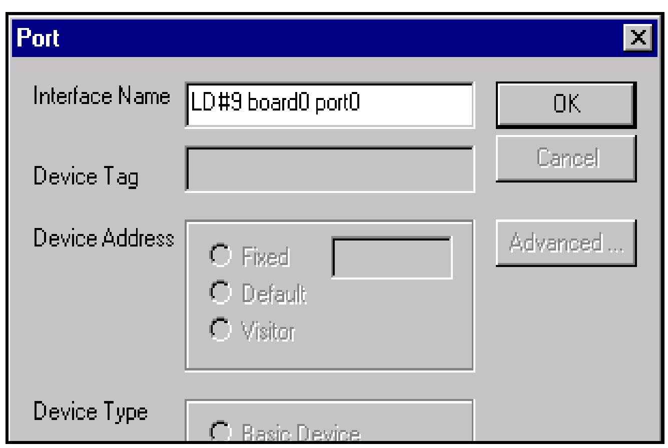


9. Copy the path from here to Step 6 on page F-9 in the Path field of the Interface dialog box.
   The path here is 2.6.1.6.2

   The 10 at the end of this string is the MAC ID.

Port Configuration

When you have finished with the Interface dialog box and click **Add**, you see:



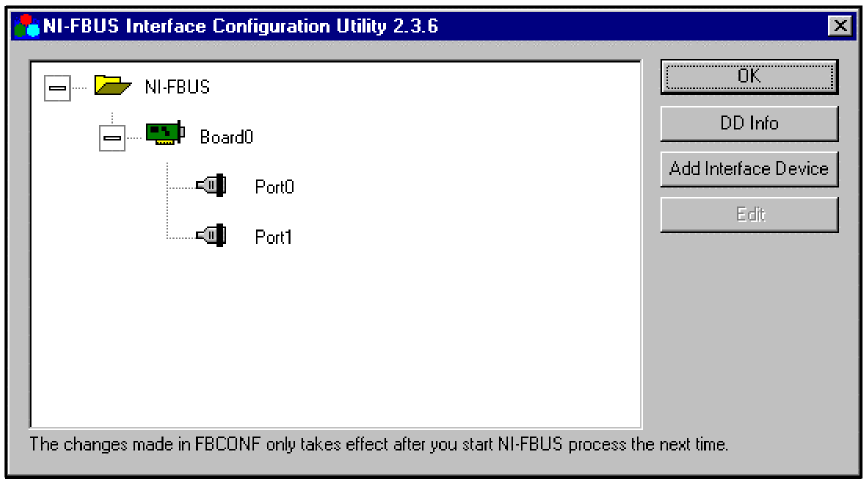
1. Type in a name that is meaningful to your project.

This is the name for the Fieldbus connection on port 0 of the linking device.

IMPORTANT The ports on the front of the 1788-CN2FF are labeled Port 1 and Port 2. However, the software labels them Port0 and Port1. We are using the first port on the linking device, which will appear as port 0 in the software, but is physically marked as port 1 on the linking device.

2. Click OK.

You see:



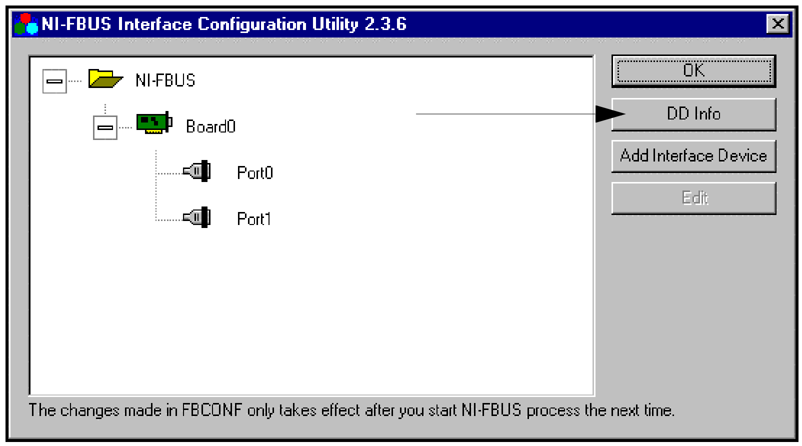
Now that you have defined the communication path to the Fieldbus network, the Device Descriptions (DD's) need to be loaded for the devices you are going to configure.

Installing Device Descriptions (DDs)

A DD is a file, which describes the FF device to the configuration software. DD's are like the EDS files we use on DeviceNet and ControlNet but DDs are typically 10 times larger.

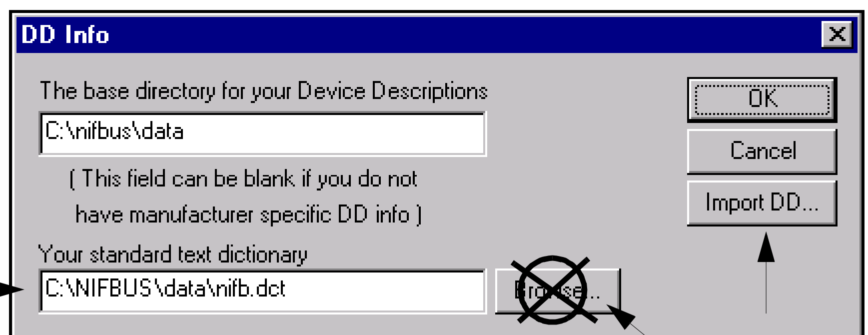
The Device Descriptions need to be installed for all the devices that you will connect to the Fieldbus, including the DD for the 1788-CN2FF.

1. Click DD Info.



You see:

The software program will install the Device Descriptions in the default directory C:\NIFBUS\DATA directory. If you change this directory, the software may not be able to find the DD's.



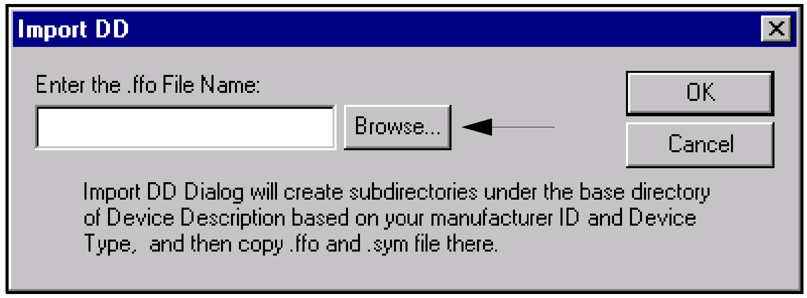
DO NOT CLICK THIS BROWSE BUTTON!

The program will automatically install the Device Descriptions in the default directory, C:\NIFBUS\DATA.

IMPORTANT If you accidently click Browse, go back and enter the name of the default directory: C:\nifbus\data.

2. Click on Import DD.

You see:



When you start, all the DD's will be located in another directory. Use Browse to find the location.

3. Insert the disk for the 1788-CN2FF in the A: drive.

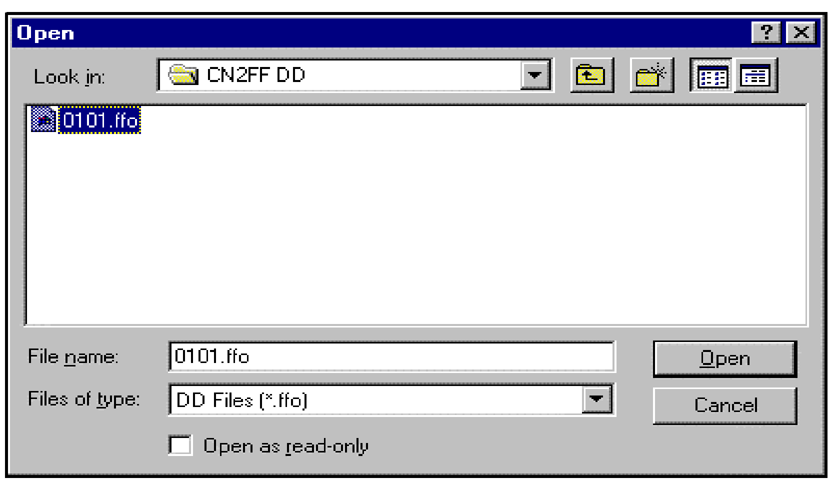
4. Click Browse.

You are looking for the DD Files. The DD for the CN2FF is on the disk that was shipped with the 1788-CN2FF. The DD's for the Fieldbus Devices should be on disks that ship with the device.

If you are not given the DD for a device, you can download most of them from the Fieldbus Foundation Web Site, www.Fieldbus.org. You can also go to the suppliers web site and download the DD files.

After you click Browse, you will need to navigate to the location of the DD's you are trying to install.

You see:



A DD file consists of two or three files with the extensions .ffo and .sym. The main DD file has an extension .ffo. When you select the .ffo file, the software automatically installs the other files.

5. Select the file with the extension .ffo and click **Open**.

You see:

6. Click **OK** to import the DD.

   You will receive a message telling you the copy succeeded.

TIP If you get a message that says: *Can Not Import that DD*, there is a good chance that the DD has already been installed on your system. Go on, and don't worry for now.

7. Repeat the process to import the DDs for all the other Foundation Fieldbus devices attached to the 1788-CN2FF.

IMPORTANT You must have a DD for every device or you will not be able to configure that device.

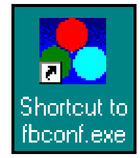
8. When you have copied all the DDs, click **OK** again to close the DD Info window.

9. Click **OK** to close the Interface Config utility.

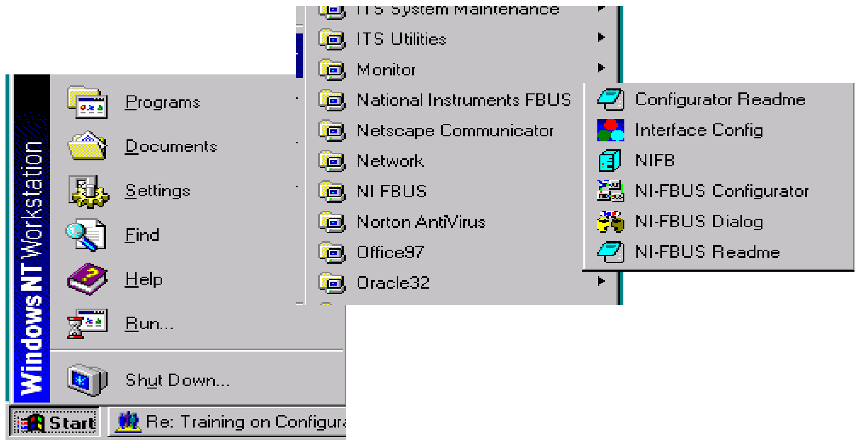
TIP If you have just installed the 1788-FFCT software, you will be asked if you want to restart your computer now.

10. Double-click the NIFB icon on the desktop.

TIP If the Icon is not on your Desktop, you can get it at Start\Programs\National Instruments\NIFB.exe.

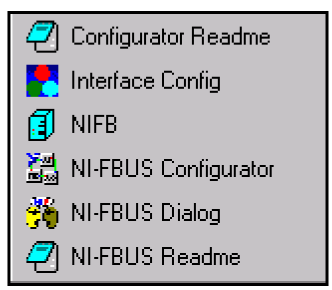


Starting NIFB

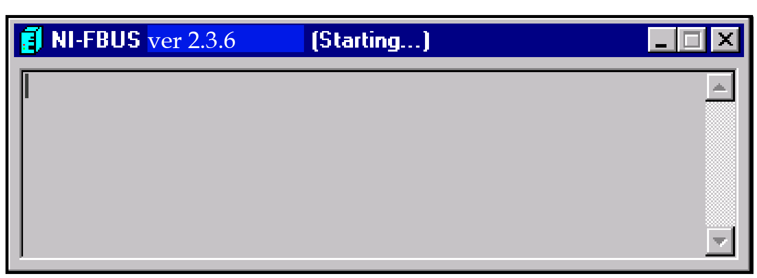


You are now ready for the next step toward configuring a Fieldbus.

1. Go to Start⇒Programs⇒National Instruments FBUS, and click on NIFB.



You see:



If you don’t see this screen, your interface is not configured properly, refer to Adding an Interface Device on page F-7 for more information. This window appears for a few seconds, then displays a quick message saying that all the interfaces are there, and then disappear.

This icon 

appears in the lower right corner of the toolbar indicating that the NIFB software is running.

Troubleshooting the Port Configuration

Use this section if NIFB does not start.

NIFB Software Install

If this window does not disappear, but says that NIFB can’t find Interface X on Board Y, You have a problem. Do the following:

1. First look at RSLinx to verify that you can see all the ControlNet devices on the network, including the CN2FF.

   If you see the Mac ID you’re trying to configure but you got the message, check to see that you entered the correct “Driver” on the “Interface” screen, refer to Adding an Interface Device on page F-7.

- The path was not entered correctly refer to Assigning a Path to the 1788-CN2FF, page F-10.

  The path will always start with 2 and end with 2, if your PC is on ControlNet. Don't put a space before the first 2 or after the last 2. Use only one space between each number in the path statement.

- The MAC ID of the CN2FF and the MAC ID entered do no agree, refer to step 9 on page F-12.

  After you edit any configuration attributes, you must close the NIFB software for the changes to take effect. The NIFB.exe program does not go back and check for changes.

TIP

Sometimes you may think the NIFB software is closed but it could still be running. To double check, look at Microsoft Task Manager and review the status of NIFB.exe. If it is still open it will be listed on the Task Manager, highlight NIFB and click End.

Once the NIFB software is running we are ready to start the configurator tool.

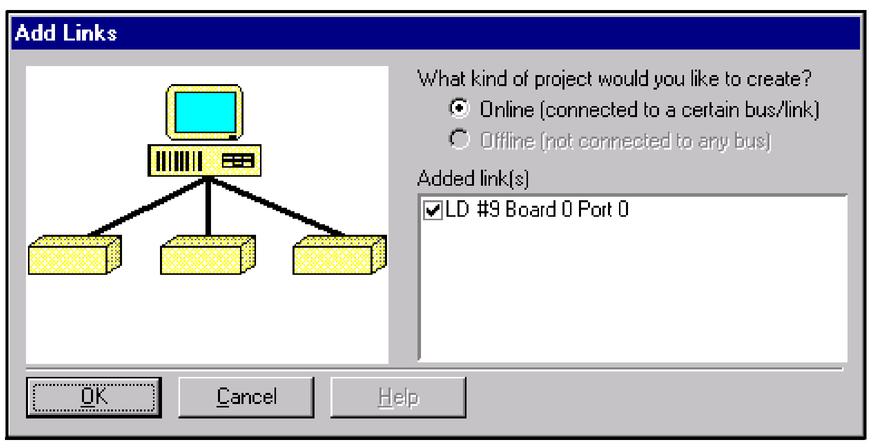
IMPORTANT You could go directly to the configuration tool without manually starting the NIFB software. The config tool, the FCS.exe, automatically starts the NIFB software. However, if there is a problem with the connection, the config tool may hang without providing an indication of the problem. Therefore, it is good practice to verify your connection by manually starting the NIFB.exe before starting the config tool.

Start FCS

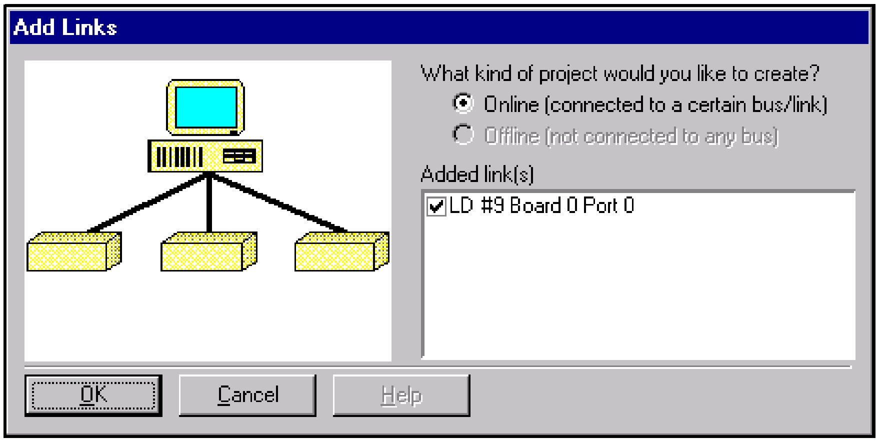
1. Double-click on the NI-FBUS configuration icon on the desktop.

   This software will again check to see that it can see all the Fieldbus interfaces that we listed in the Interface Config. Window, and it will ask which of the Fieldbus networks that you want to configure.

   The following window appears.



2. Select the Linking Device and associated Fieldbus you want to configure and click **OK**.



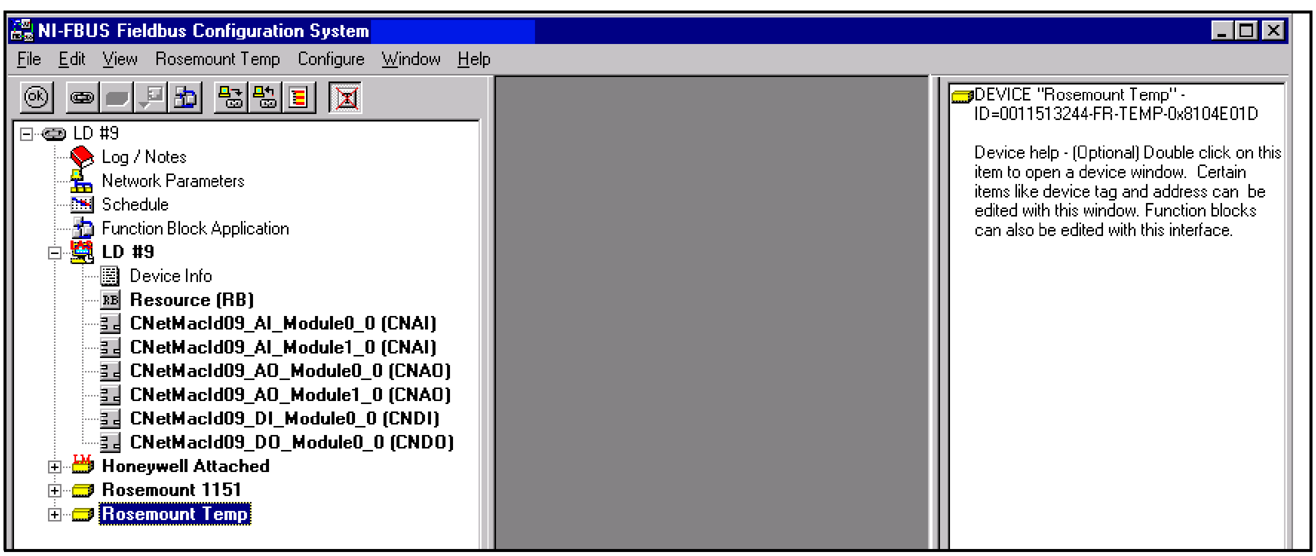
3. Uncheck one box so we only use one port on the CN2FF. Click **OK** to accept the selection.(we are only using one link).

TIP

The standard version of the FFCT only supports the configuration of 4 Fieldbus networks at one time. You can configure two Fieldbus networks connected to each of two CN2FFs or you can configure one fieldbus connected to each of 4 CN2FF.

If you must connect to more than 4 Fieldbus devices at one time, an additional license to support that can be arranged through Tech Support. We suggest that each CN2FF be configured and saved by itself.

You see a similar window:



IMPORTANT

If you are using both ports on a single CN2FF, you must:
1. Check both ports, now.
2. Configure both Fieldbus networks.
3. Perform two downloads.

The reason you must do both Fieldbus networks at the same time is that when you restart this Add Links function, you delete the data from the first download.

It will take several minutes for all the data to be gathered to complete this window. This is primarily a list of the Function Blocks that are in the attached Fieldbus devices. Until all the data is complete, you will see the "working" hour glass.

Wait until the software is finished gathering data. It will be finished when the working icon disappears. Although, you see nothing going on, there is a lot of Fieldbus communication going on.

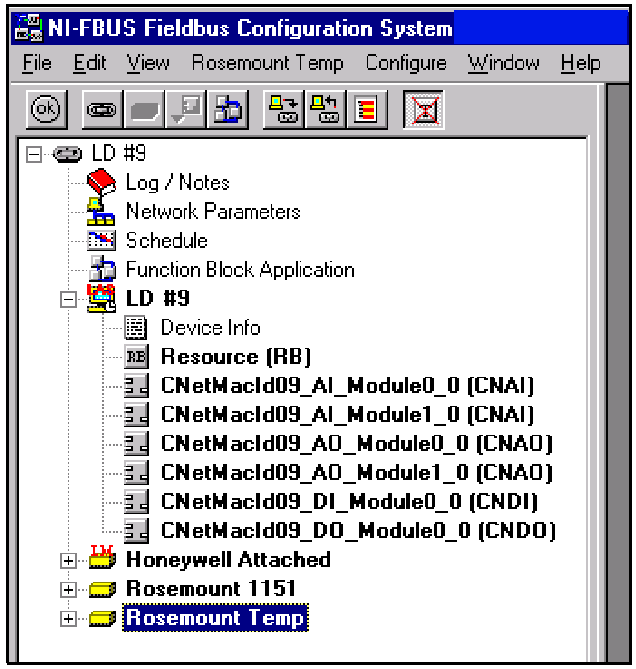
If you see:
- **yellow circles with an "!"** exclamation mark in them, that indicates that the FFCT configuration software doesn't understand that device. *This is a caution and needs to be addressed.*
- **UNK** to the right of a the name of a Function Block, that means again that the FFCT doesn't recognize that Function Block. These indicate that the DD's for those devices were not loaded, or did not match the data that the FFCT read from the device.

ATTENTION

You can't use a Function Block with a UNK note. The UNK is a show stopper. You need to get a more current DD from the device manufacturer.

4. When the working icon disappears, click **Network Parameters**.

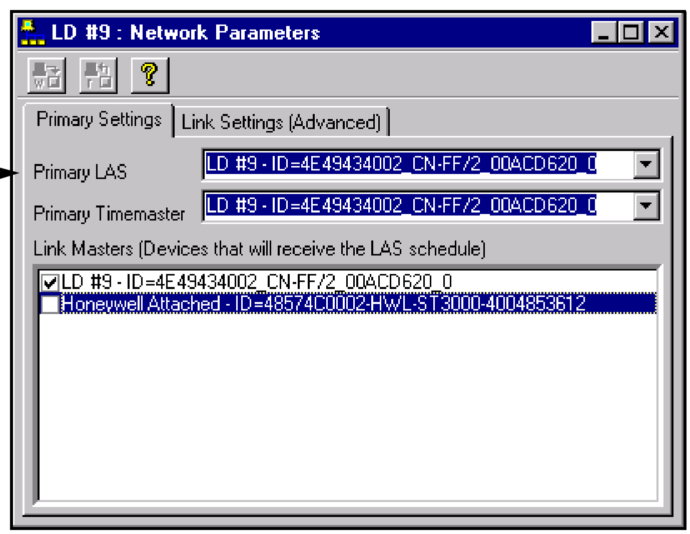
These are the devices on the FF network. Your list of devices and Function Blocks will be different. This is called the Function Block Library,



You see:

Primary LAS
LAS, Link Active Scheduler
LAS stands for Link Active Scheduler and identifies the device which will hold the network schedule and tell each node when to Publish its FF information.

Uncheck the download to Honeywell Attached. This will reduce the download time.



5. Select the Primary LAS, and the Primary Timemaster.

6. Use the two pull down menu's to select the Linking Device for both.

   Check only the Linking Device to receive the LAS schedule. In this example, you would remove the check by the "Honeywell

Attached." Doing this saves a little time since you will not load the LAS into the Honeywell device.

IMPORTANT	Some devices that are capable of being LAS can cause some conflict. For your first test, check only the 1788-CN2FF.

7. Use the Primary LAS and Primary Timemaster pull down lists to Select the 1788-CN2FF to be both the primary LAS and the Primary Timemaster. (the CN2FF is the second one listed above).

All the devices that have an LM overlaid on the icon to the left of the name of the FF Devices have the ability to be the LAS on that network.

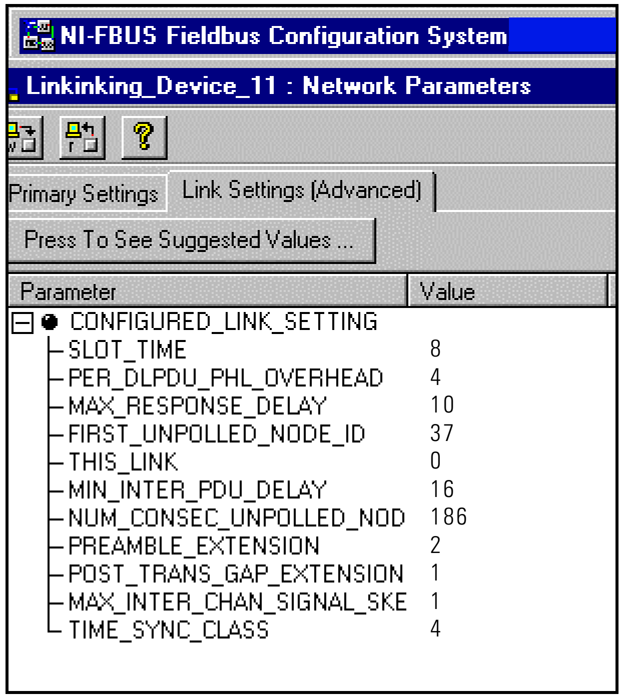
8. Click **Link Settings(Advanced)**.

The information on this screen relates to the timing of the network. You should be aware that if these parameters are set too low they could cause communications problems on the network. The default settings are those currently suggested by the Fieldbus Foundation.

To see if some of the attached devices are requesting more time:

a. Click **Press To See Suggested Values**.
b. Click **Apply Suggested Values**.
c. Click **Write Changes**, if there are any.

The best thing to do is use the values that the FF suggests: They are shown in the window.



Modifying Device and Function Block Names

1. Click **Write Changes** to save the changes.

FIRST_UNPOLLED_NODE_ID = 37

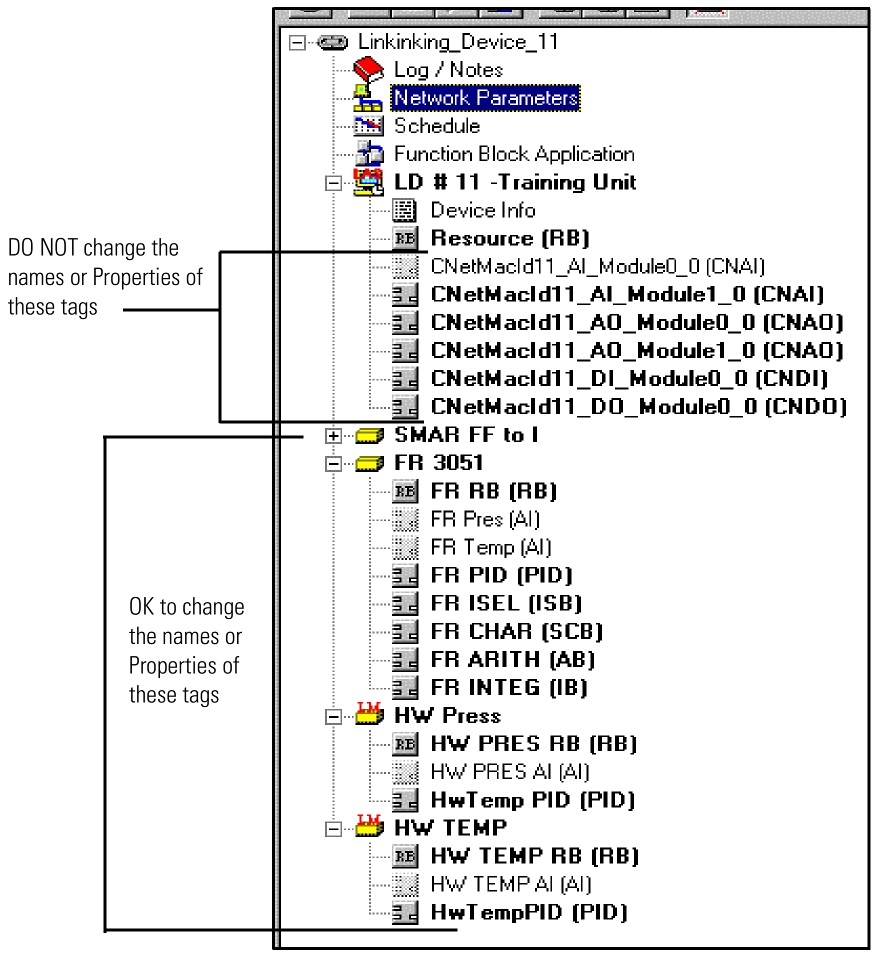
This setting defines 36 to be the last available address to be polled on this network. When addresses are assigned for this network, they will be assigned in the range of 16-36. (16 or 10 hex, is the first usable address on a Fieldbus network and it will be assigned to the CN2FF.).

2. Close the Network Parameters window.

3. When the hourglasses have disappeared you are ready to proceed.

The Library of Function Blocks in this example shows two pressure transmitters. You will give a tag name to each transmitter, check the FF network address of each transmitter, and then give names to each of the function blocks.

4. Move your mouse over the device name of your first device and click the right mouse button.



Changing a Tag Name

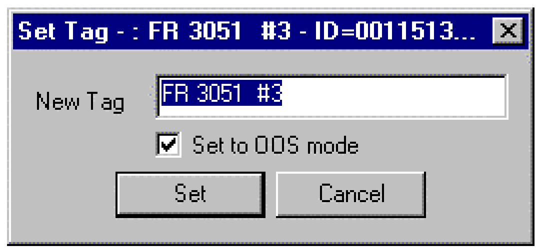
You can change the names of the Fieldbus Devices, and most of the function blocks so they have meaning in your application.

IMPORTANT Do not attempt to change the names of the Function Blocks in the 1788-CN2FF. If you change these names the 1788-CN2FF will not work properly.

1. Right click on a Function Block.

2. Select Set Tag.

3. Type any name that fits your application.



   Short names are easier to read in Device Info which you'll use later.

4. Check the Device addresses.

   The devices on your fieldbus should automatically be assigned a network address in the range of 16-36. This MAX address is based on the setting of the First_Unpolled_Node_ID.

   Check the address that was assigned to each device. Devices which have LM, Link Master, capability should have addresses of 23 or higher.

5. Right click on LM devices.

6. Click **Set Address**.

   If the address is less than 23, pick a new available address. The CN2FF should be at FF address 16. If not, change the address to 16.

7. Select **Set Tag**.

If you ever plan to change the name of a Function Block, do it now. After you do your download, you do not want to change any connections, or Tag names.

When you look at the **Device Info** window after you download, you will see the value of using very short names for the Device and the Function Blocks.

8. Highlight the RB function block in the Library of Function Blocks, and click the right mouse button.

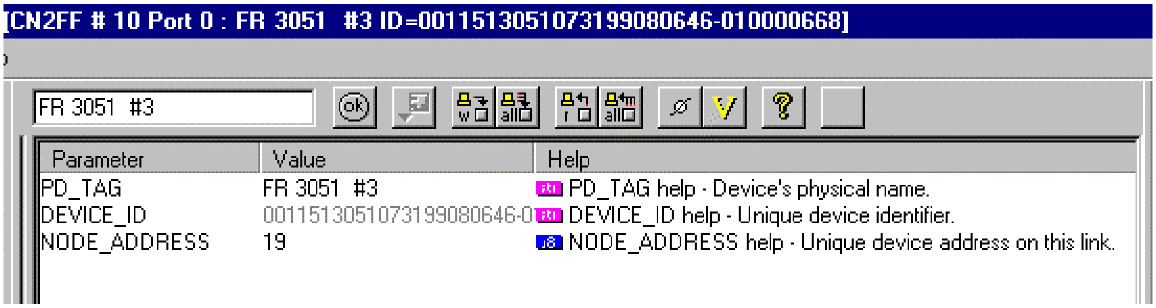
9. Select Set Tag.

10. Assign a name to the block that is appropriate for your application.

11. Continue this process for all the Fieldbus Devices.

    Each time you SET a TAG, and after a short delay, the Set Tag window will close, and you should get the following status messages on the bottom of the screen.

12. Double click on the device and you'll get a window like this.



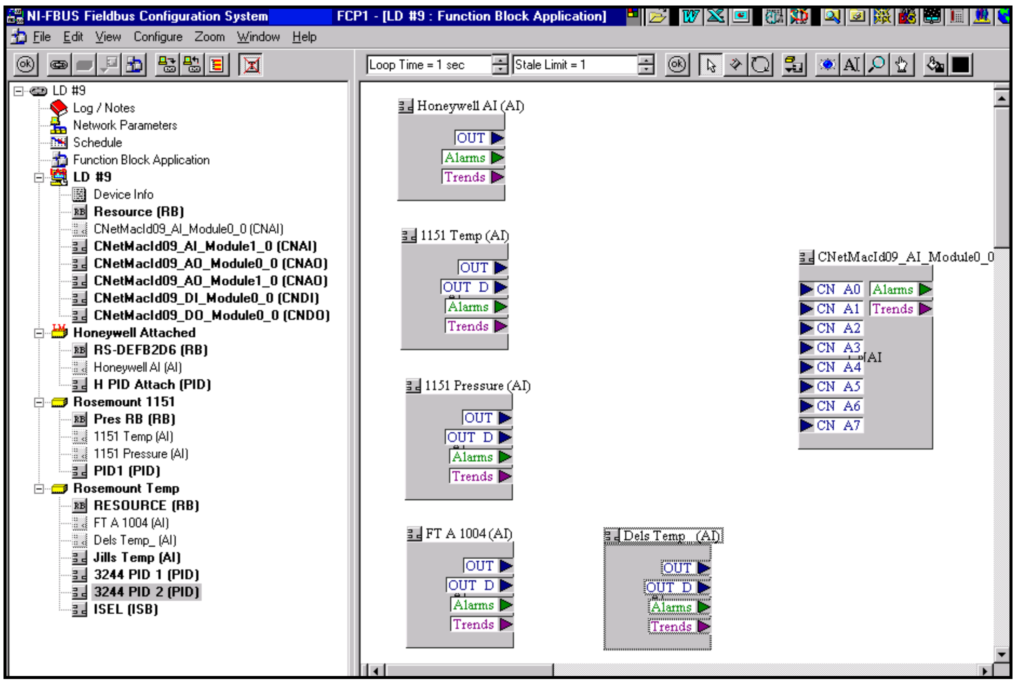
This screen shows the name and the node address assigned to the device, and lists a Device ID assigned by the manufacturer. It is a unique identification. You or the FF protocol may have assigned the Node Address. Because the Link Master device with the lowest node number becomes the Link Active Scheduler, you may want to adjust Node Numbers after they have been automatically assigned. You may also need to resolve Node Number conflicts if a new FF device was previously assigned a node number that another device is now using.

If necessary, you can change the FF address of a device by right clicking on the device and selecting Set Address. Then, enter the desired address in the window, and click **Set**.

Configuring the Fieldbus Device

Once you have finished assigning device names you need to program the Fieldbus devices to move data where we want it.

1. In the screen below, select all the Analog Input Function Blocks in the Fieldbus Transmitters in the column on the left, and drag them into the Function Block Application window.



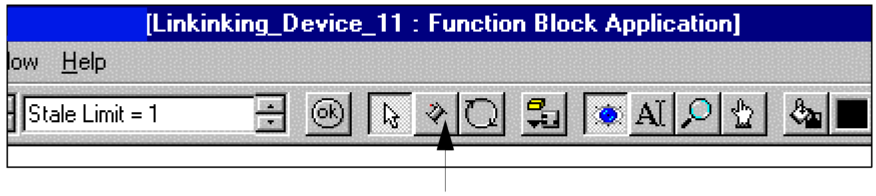
2. Drag one of the CN2FF Multiple Analog Input Function Blocks to the right side of the FB Application Window.

   All "AI" Blocks have AI in [ ] to the right of their Tag Name.

3. Consider the Loop Time which can be set in the window just above the Function Block Application.

   You will need 40 to 50 ms for the transmission of each data value you put on the Fieldbus. Therefore, if you connect 10 Function BLocks to the MAI, you need a loop time of 500ms or more.

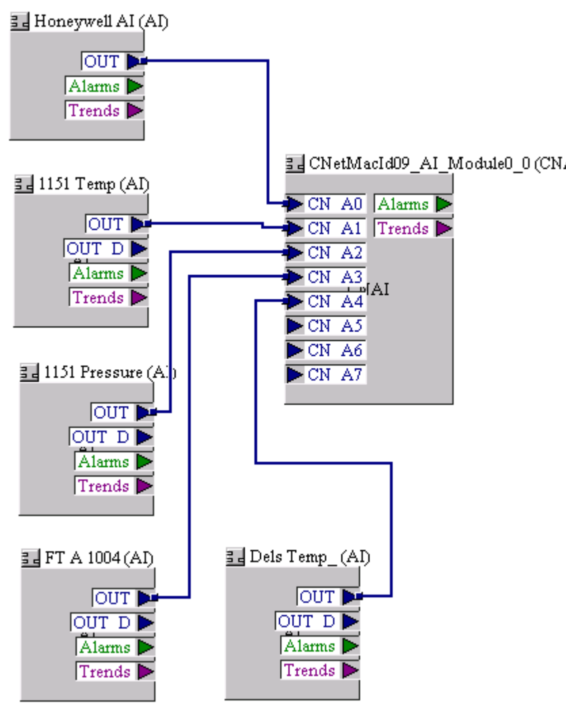
4. Click on the spool of wire in the top tool bar, and use the wire



to connect, or program the Function Blocks as your application requires.

Figure F.3 illustrates 5 Analog Input Function Blocks connected to one Multiple Analog Input Function Block, MAI FB.

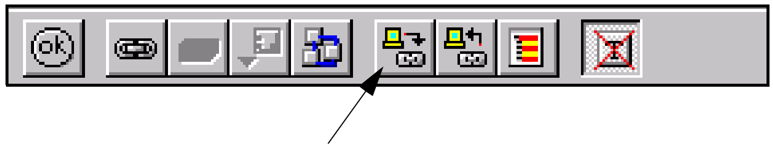
Figure F.3 5 Analog Input Function Blocks



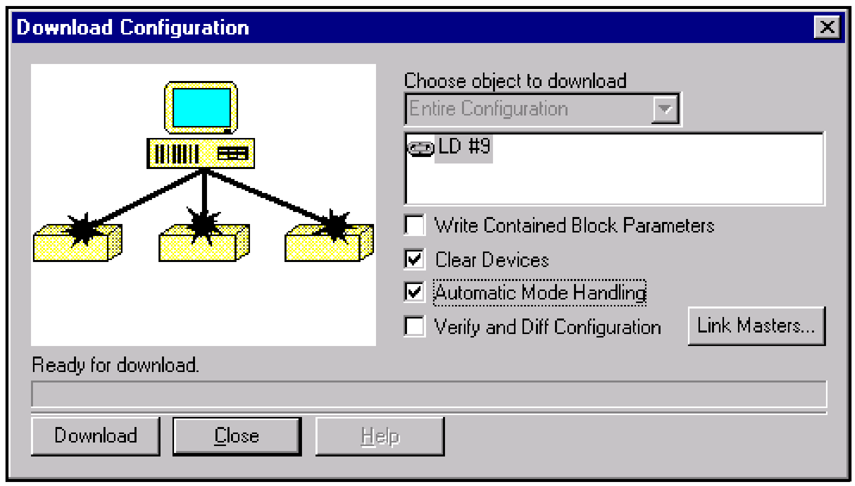
After creating this Function Block configuration (drag and drop), the 1788-FFCT software knows where you want connections, but the Fieldbus Devices do not. You must now download the configuration to the devices.

Download the Device Configuration

1. On the left side of the toolbar, Click on the Download Button on the top toolbar.



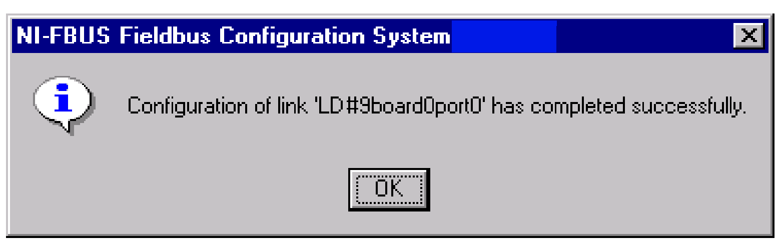
You see:



2. Check the boxes as shown:
   - Clear Devices
   - Automatic Mode Handling.

3. Click **Download**.

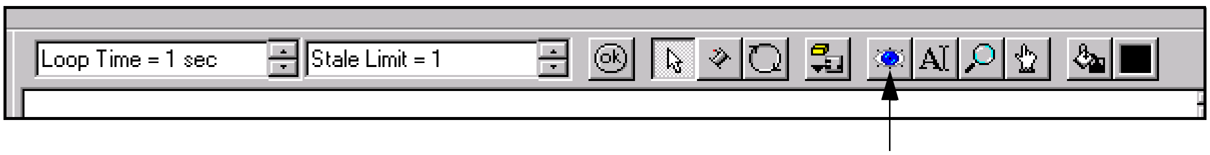
   This may take a few minutes. While waiting you can monitor the activity in the Download window at the bottom of the screen. When the download is complete, you will get a small new window to advise you. Click OK and click on Close in the Download window.



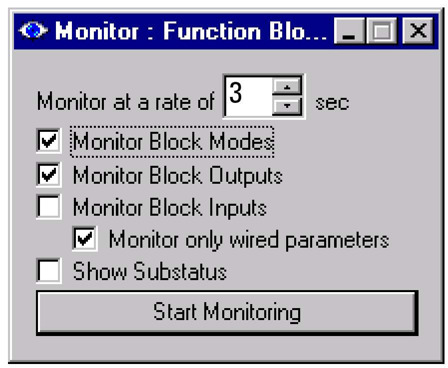
If the download does not complete, try it again. Some FF devices need to be told many times what they should do. If the download fails again, remove the FF device and configure again. Look at the download log and status for ideas.

If you continue to have download problems, refer to Troubleshooting an Application on page F-51.

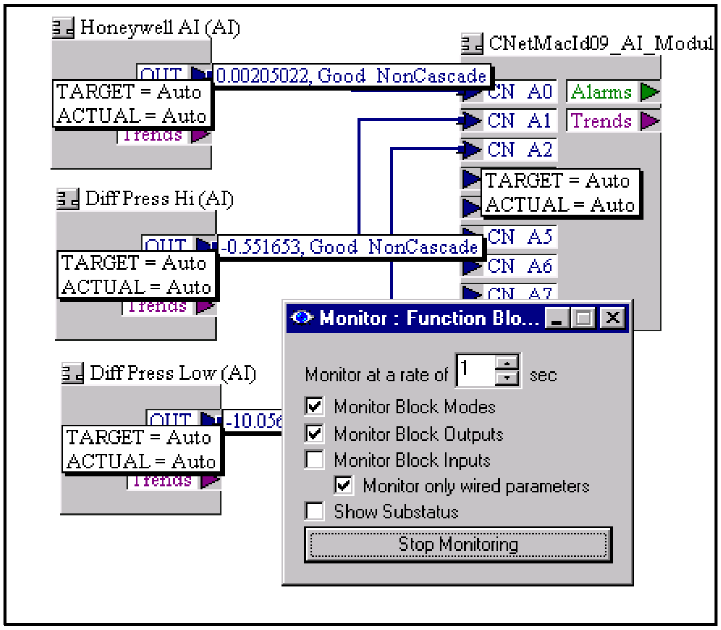
In the toolbar at the top of the screen, click on the blue eye to start the Monitor.



When you run the monitor, you get a display of the values being read from the attached transmitters.



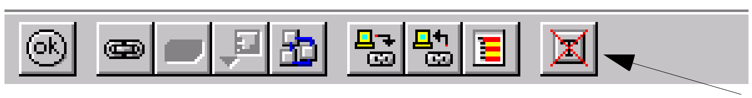
The Monitor window will appear as shown here. If you have only 3 or 4 function blocks attached to the CN2FF, set the monitor rate to 5 seconds. If you have 20 or so function blocks attached, set it for 10 seconds. The Monitor Data is acquired with messages to the FF devices and it adds a significant load to the FF network if you ask for many of readings.



4. Set the monitor to update the display every 5 seconds.

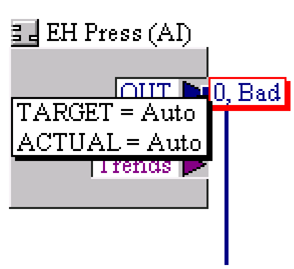
   The output values from the analog Function Blocks will be displayed along with the status of the Function Blocks. If you change the temperature or the pressure being measured by the Transmitter, you should see the new value on the screen

5. You may have an output that is displaying bad with a red border as shown below. If you see this, to refer to Troubleshooting an

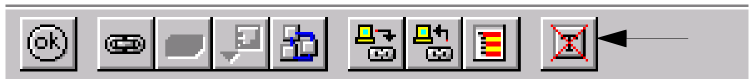


   Application on page F-51.

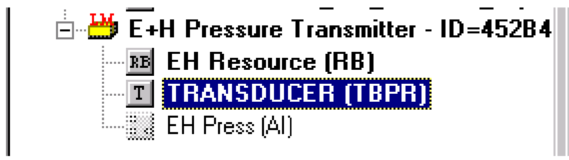
   Frequently the Hidden Transducer Block being set to OOS (out of service) causes this. If this happens, try the following:.



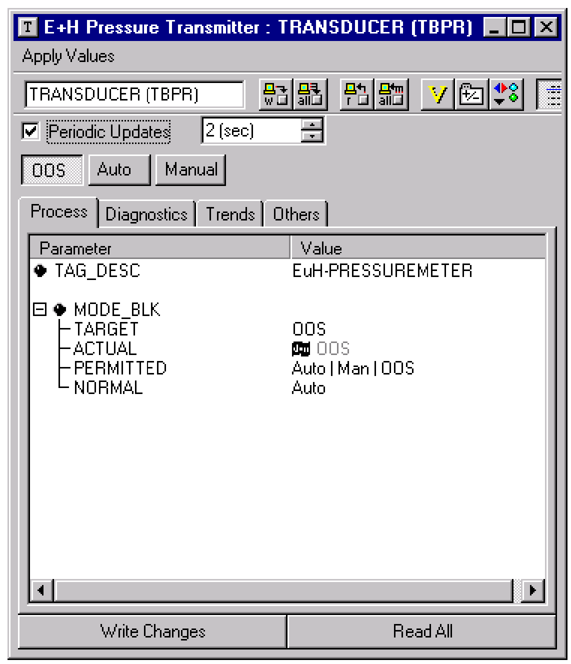
6. Select the **Show/Hide Transducer block** icon from the toolbar to display the hidden transducer block.



The hidden transducer blocks will now appear in the list of function blocks for the transmitters   The image below shows an E+H Transducer block.



7. Double-click on the transducer block. This screen appears:



8. To Change the block from OOS mode to Auto mode click on the AUTO buttons at the top of the window.

TIP

There is typically no reason for a Transducer Block to be in the OOS mode, other than the Configurator may not have been able to set it to AUTO. Therefore, manually set it to AUTO. If that is the only problem, the AI value will go to AUTO, the status will change to GOOD, and you will start to see data values from the transducer.

Also check the Resource Block. It should also be in Auto Mode. If it is not attempt to manually set it to Auto.

If the AI output does not go to AUTO, and the Status to Good, you will probably need help from the transmitter manufacturer.

9. Close the Transducer block window.

   The monitor window should now be displaying values with a status of **Good NonCascade**.

10. If the AI has not changed to "Auto" mode, check to see that the "Channel" and "L_type" have been initialized.

11. Double click on the offending AI block.

12. Under the Process Tab, look for Channel. This indicates which of the channels through the Foundation Fieldbus device is supported by this AI.

13. If the Channel is uninitalized, click on it and select one of the available choices.

14. Click Write.

15. Go to the Scaling Tab.

16. Look at L_type, or Lineariztion Type.

17. If it is uninitialized, set it to Direct, and write the value. If you still have problems with the device, call device manufacturer.

18. Squeeze the pressure ball on the transmitter or wait for a pressure cycle and verify you can see the pressure change on the monitor.

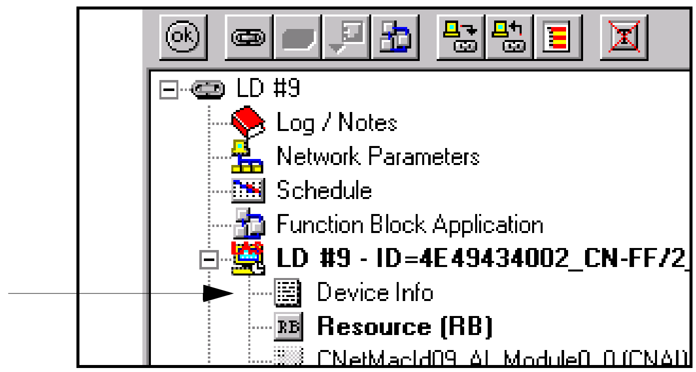
    Now that the Fieldbus side is set up and working, you are ready to configure the ControlNet side.

When the transmitter has been configured and the Monitor window is displaying process variables properly, you have completed the configuration. Congratulations!

Sending Data To the PLC-5, CLX, PLX or SLC

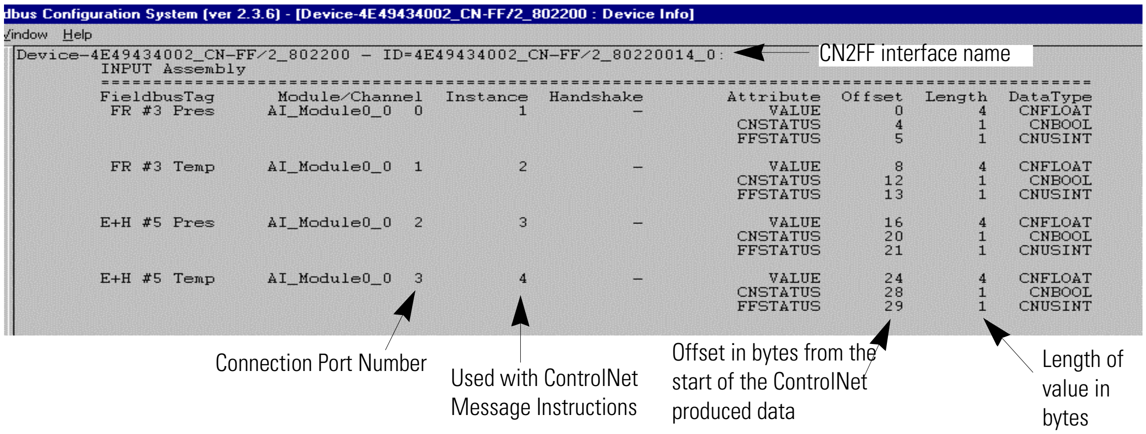
The following steps illustrate the sequence in which data will be sent from the Linking Device over ControlNet.

1. Double click on the "Device Info" icon.



A new screen displays all the data produced by the attached Fieldbus Devices by Function Block Tag name.

Figure F.4 Device Information Screen



2. Expand the window so you can read all the columns.

   This screen is the reason we recommend using a high resolution monitor and short Function Block names.

The Device Info screen describes the format of the data that is produced by the 1788-CN2FF on ControlNet.

The Offset column indicates the offset in bytes from the beginning of the transmission to each piece of data from a Fieldbus devise.

Schedule Data Transmission to Controllers with RSNetworx

You must use RSNetworx to schedule the transmission of data to the controller, RSLogix 5000, or Control Builder. Place the data in an integer data table, like N13, starting at location N13:0.

You may need to add some devices to the configuration so we recommend that you leave some empty words in the data table. Store the data from each of the linking devices at an easy to calculate location like N13:0, then store the data from the next CN2FF at N13:50, the data from the next CN2FF at N13:100, etc.

PLC-5 Data Manipulation

Figure F.4 on page F-37 shows that the Floating Point value of each Analog Input from the Fieldbus devices is a four byte ControlNet Float when it goes on the wire. When this four byte value arrives at the PLC-5, it is stored in an integer file in a PLC-5 word format. That process swaps the sequence of the data floating point which is the process variable.

To get the analog value into floating point format so you can use the data, you need to swap the sequence of the words, and then copy the 32 bit value to a Floating Point data table. When you do the copy, specify a length of 1.

When you are finished, you should be able to use the Monitor Function in the Fieldbus Configuration software to monitor the data coming from a Fieldbus device, and using a second window on the PC monitor, see the same value in the Floating Point Data Table in the controller. If you happen to have Fieldbus sensors that show the value that they are transmitting, all three readings should be the same.

If you are using a PLX system

The Offset Values must be set on the **IN** function blocks, which you drag to your Control Module from the Library. If you are receiving data from a Discrete IN, DI, function block, you will drag and set the offset in an IN_D function block.

If you are using a CLX processor

Tech Support at 440-646-5800 can provide a CLX program that includes a structure that spreads the FF data so it is understandable.

You can obtain the ControlLogix Example Code by doing the following:

1. Visit us at www.ab.com
2. Click on **Product Support.**
3. Click on **Knowbase**.
4. Search on Knowbase document number: **A5754787** .
5. Double click the file and download to you PC.

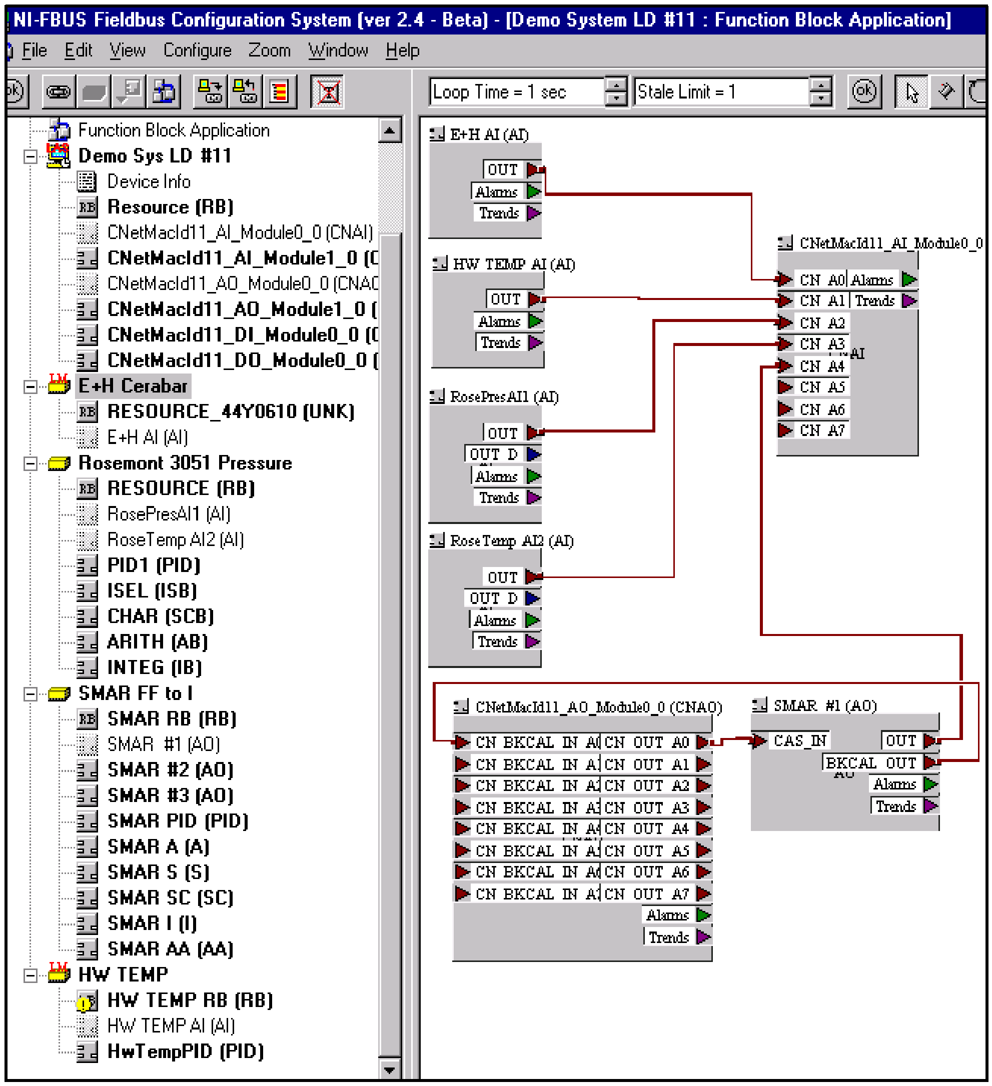
With the CLX, there are two pad bytes before the real data from the CN2FF starts to appear. Therefore, the CLX program from Tech

Support adds 2 to the listed Offset. (If you are using a PLC-5, the Offset values are correct.)

The 1788-FFCT and 1788-CN2FF are capable of producing data that can be used by all RA controllers that interface directly to ControlNet. This includes the PLX, CLX, PLC-5s and SLC.

If you have problems, call Rockwell Automation Technical Support at 440.646.5800. Also refer to Rockwell Automation Technical Support on page P-3.

PLC-5 and ControlLogix Applications



After the initial, setup the task is to get data to and from a controller. You can move Fieldbus data to and from SLCs, PLC-5's, ControlLogix 5550, and ProcessLogix. The first stage is to set up a working Fieldbus network that has connections to the linking device. Here is an example configuration.:

The AI function block, FB, in the E + H pressure transmitter, the AI FB in the Honeywell temperature transmitter, and both the Pressure AI and the Temperature AI in the Rosemount Pressure Transmitter are all connected to the CN2FF Multiple Analog Input, MAI, function block.

The AO FB in the Multiple Analog Output, MAO, of the CN2FF is connected to the CAS_IN pin of the SMAR #1 AO function block. The

OUT pin of the SMAR AO block is connected to the fifth input of the MAI module, on the MAI FB.

There is also the BackCalibration, or BKCAL connections shown from the SMAR. At present the FFCT software demands that there be a BKCAL connection to the MAO FB.

This combination of connections will create a ControlNet Object in the CN2FF that can be sent to controllers using RSNetworks or ControlBuilder, depending on which controller you are using.

IMPORTANT	If a Fieldbus Device becomes disconnected from the Fieldbus, or suddenly stops transmitting data, or if the transducer stopped functioning in a transmitter, and the device could not produce data, the CN2FF will still retain the last valid value that it received. To handle this, the Fieldbus Status Byte, and/or the byte listed as the ControlNet Status Byte will change. **It is absolutely necessary that the control application program monitor the Status of the PV**, and take whatever action is appropriate if a PV so no longer valid.

ControlLogix Application

We are now ready to use RSLogix 5000 and RSNetWorx software to set up and schedule a connection between the ControlLogix system and the Linking Device. Normally, all data exchanged between the ControlLogix controller and the CN2FF Linking Device is scheduled.

1. Open RSLogix 5000 software and create a project called CN2FF.

2. Add a 1756-CNB[R]/B to your project in the I/O configuration folder.

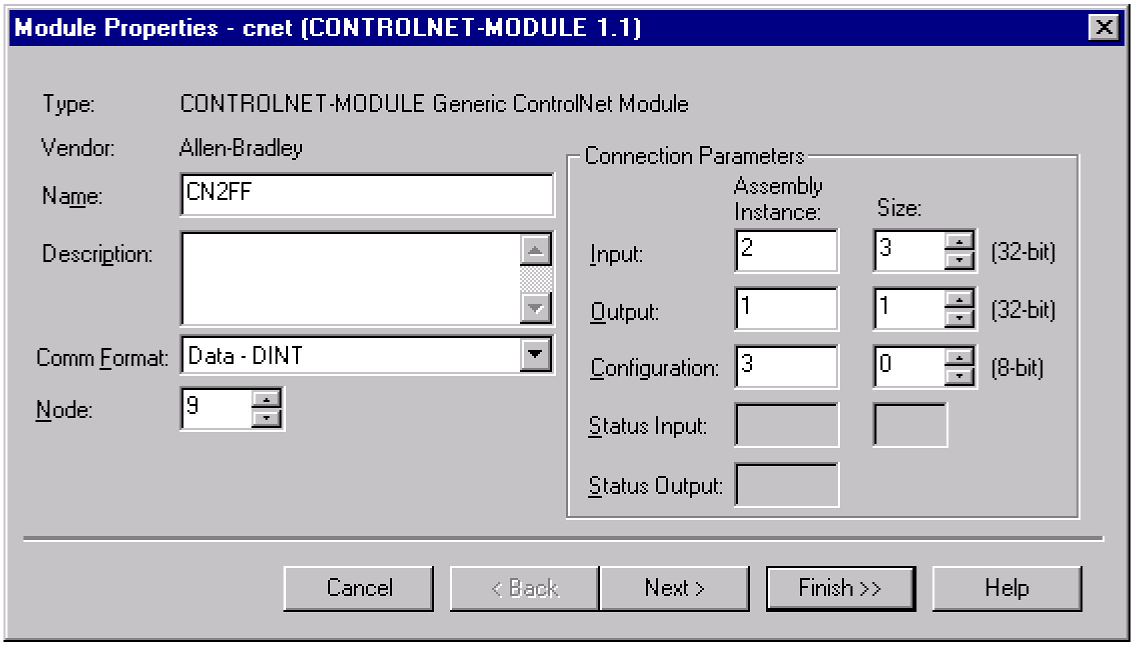
   The next step is to add the 1788-CN2FF module to the I/O configuration below the 1756-CNB[R]/B. However, the RSLogix 5000 software does not currently support the 1788-CN2FF, so we have to select the generic module connection.

3. Add a CONTROLNET-MODULE connection below the 1756-CNB[R]/B.

   This connection will be used for the1788-CN2FF module.

4. Fill in the parameters in the module properties window as shown in Figure F.5.

Figure F.5 ControlNet Module Properties



There are several parameters to note on this window:

Comm

The Comm Format is Data-DINT.

Assembly Instance

The Assembly Instance values must be set as shown in Figure F.5.

Input Size

Look back at the Device Info in the Configuration software. Notice that an Analog input repeats every 8 bytes. Therefore, you want an input area x times 8 where x is something greater than the number of Analog inputs that you expect may be connected to the CN2FF.

When you look at the data transmission that the CLX receives from the CN2FF, you'll see that there is nothing in the first 4 bytes. That is because there is four bytes of pad in front of the first data

Output Size

The output size is set to 1 since a generic connection requires at least one word of output. We are not using any outputs in this lab.

5. Select **Next** and set the RPI rate to 40 ms.

Given that the Fieldbus network operates much slower than the ControlNet network, and does not carry time critical information, it is a good idea to bump up the RPI for the Linking Device to preserve ControlNet bandwidth.

6. Select **Finish**.

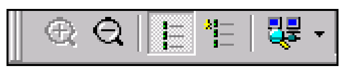
   The linking device is now configured for ControlNet communications.

7. Save your project and download it to the controller.

Now that the project is downloaded in the controller, you need to use RSNetWorx software to schedule the connection between the controller and the Linking device.

Schedule the Connection Between the Controller and the Linking Device

1. Open RSNetWorx.
2. Select **File** from the main menu.
3. Select **New**.
4. Select **ControlNet Configuration** and click **OK**.
5. Go online to the ControlNet network.
6. Select the **Edits Enabled** checkbox in the upper left of the screen.
7. Browse the network, click the Browse icon.

8. Save the network.

   RSNetWorx will schedule all the connections during the save operation.

View the Controller Tags

1. Open RSLogix 5000 and view the controller tags.
2. Expand the input tags for the linking device so that you can see the data values. Your screen should appear as shown in Figure F.6. Your values may be different.

Figure F.6 Data Values

Tag Name	Value	Force Mask	Style
+-CN2FF:C	{...}	{...}	
−-CN2FF:I	{...}	{...}	
−-CN2FF:I.Data	{...}	{...}	Decimal
+-CN2FF:I.Data[0]	0		Decimal
+-CN2FF:I.Data[1]	1026209150		Decimal
+-CN2FF:I.Data[2]	36864		Decimal
+-CN2FF:O	{...}	{...}	

The first double word is the pad. The pressure is contained in word 1. Remember, the pressure is returned as a floating-point value. You are currently viewing it in a DINT. Word 2 contains the 2 status bytes.

As you look at the values you realize that in their current representation the values are not of much use. Your first thought is to create a bunch of copy instructions to copy the values into their proper data type. Then you think that could end up being a lot of instructions if there were a lot of values.

A HA! You think, this is a great opportunity to take advantage of the User Defined Data Type feature of the ControlLogix processor. We can take this cryptic information coming from the Linking device and with a single copy instruction have it formatted into tags which we can easily read and use. Let's create a data type for the linking device.

3. Go offline with RSLogix 5000.

4. In the controller organizer, right mouse on **User Defined** under the **Data Types** folder and select **New Data Type** from the popup menu.

5. Set the name of the data type to: FF_Linking_Device

6. In the first line of the Name column for the members below, type: Pad

7. Tab over to the data type column and set the data type to DINT.

   We are using DINT for the first member since we know that the linking device sends 2 INT words of pad, which equals 1 DINT. We could have set the data type to INT and then set the size to 2.

8. On the second line type a name of: Pressure

9. Set the data type to REAL (REAL = floating point).

10. Click **Apply**.

    At this point we have the first two parts of the structure defined. Enter the remaining members as shown below.

11. Click **OK**.

    We now have a structure with which we can create tags just the same as Timers, Counters, etc. Let's create a tag which will hold the data coming from the Linking device.

12. Open the controller tags database and select the Edit tab on the bottom of the window.

13. Enter a tag name of: LD_Node_9

14. Tab over to the Data Type field.

15. Click the button with the 3 dots to bring up the choices for data type.

    The FF_Linking_Device is one of the data types listed.

16. Set the data type to FF_Linking_Device and click **OK**.

17. Press Enter to accept the new tag.

18. Open MainRoutine and enter the following rung.

This rung will copy the contents of the input data in the linking device tags to the tag **LD_Node_9**. The length is set to 1 since we are only copying to one element of LD_Node_9. The copy instruction will actually copy 12 bytes, since the copy instruction always bases the length on the size of the destination tag. Since the destination tag is actually a structure of 12 bytes, 1 element is 12 bytes.

19. Accept the rung and save your project.

20. Download the project to the controller.

    We are now ready to test the project.

Testing the Installation Example

1. Put the processor in RUN mode and view the individual elements under the tag structure LD_Node_9.

2. You should be able to read all the values directly now.

The actual pressure of about .04 psi (in this case) can now be read directly.

3. Squeeze the bulb (or place your thumb over the connector) on the transmitter.

4. Verify the pressure reading increases and follows the pressure displayed on the transmitter itself.

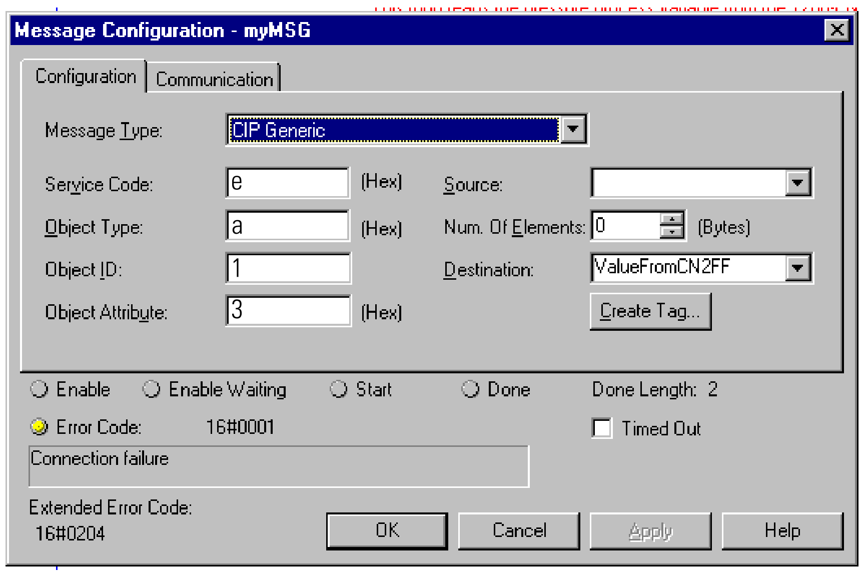
Messages to PLC-5s and CLX to Get Data from CN2FF

You can use a ControlNet message instruction to get any of the data that is stored in the different tables that list the various ControlNet objects.

TIP

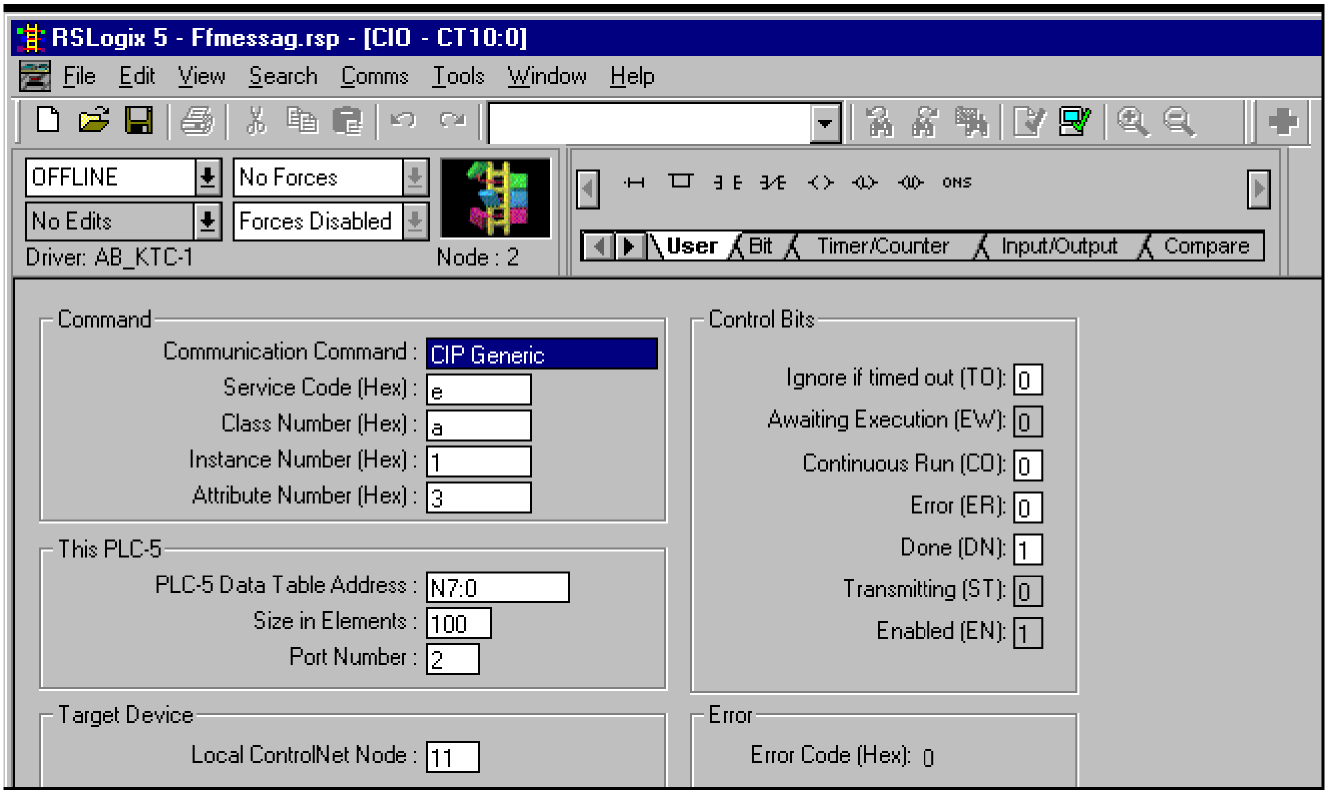
The best way to get data from a CN2FF into a controller is to have it produced by the CN2FF. This requires that you use RSNetworx to schedule the data. It is much more efficient than using message commands.

For a CLX to retrieve data, setup the message as shown below.



For a PLC-5 to retrieve data, setup the message as shown in Figure F.7.

Figure F.7 PLC-5 Message Setup



This example message will retrieve the Analog Process Variable at Instance, as shown in the Device Info Figure F.4 on page F-37, because the attribute is 3. If the attribute was 4, it would retrieve the ControlNet status. It would retrieve the FF status if the attribute was 150.

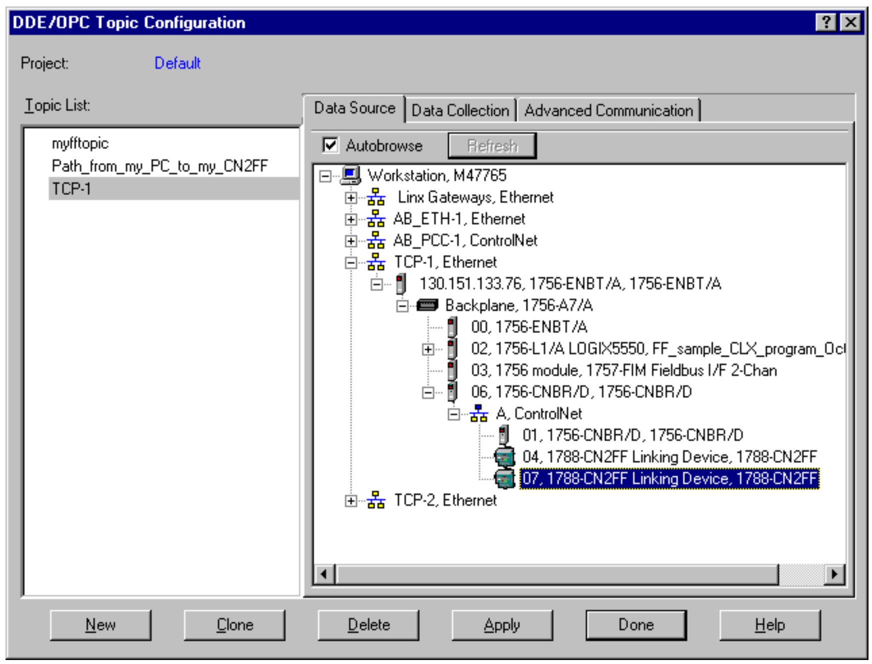
Remote Configuration of a Fieldbus Network via the 1788-CN2FF

The 1788-FFCT, the Foundation Fieldbus Configuration Tool, works through RSLinx, so the things that you expect to be able to do through RSLinx, you can in fact do with the FFCT, including configuration from a remote location.

In the section Assigning a Path to the 1788-CN2FF starting on page F-10, we determined the path to a CN2FF where multiple ControlNet networks were involved. The procedure for Ethernet is similar.

1. Set up an Ethernet Gateway Driver to connect to the 1756-ENET module in the remote rack where you will connect to ControlNet.

2. Open the DDE/OPC window.

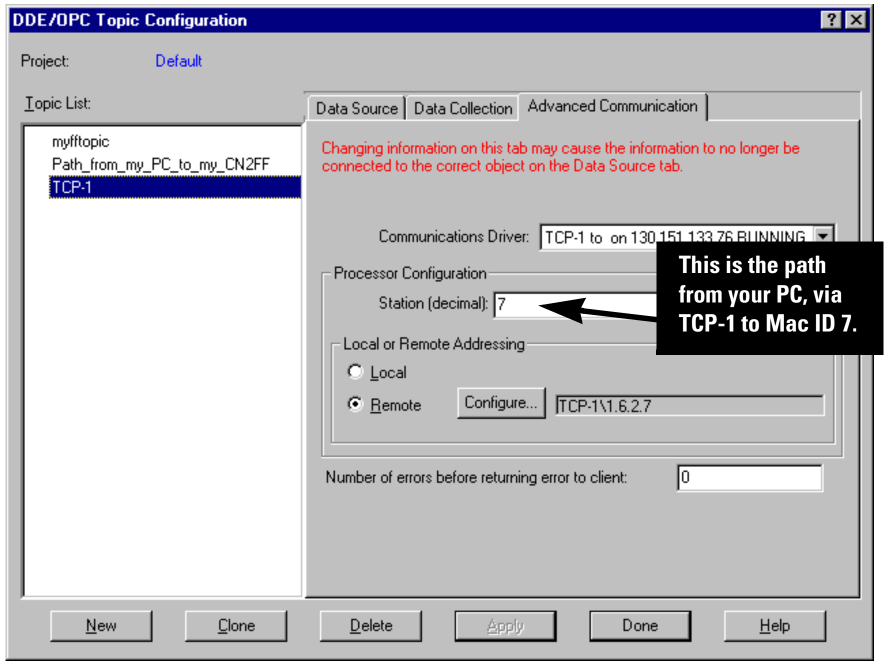


3. Create a New Topic, refer to page F-7.

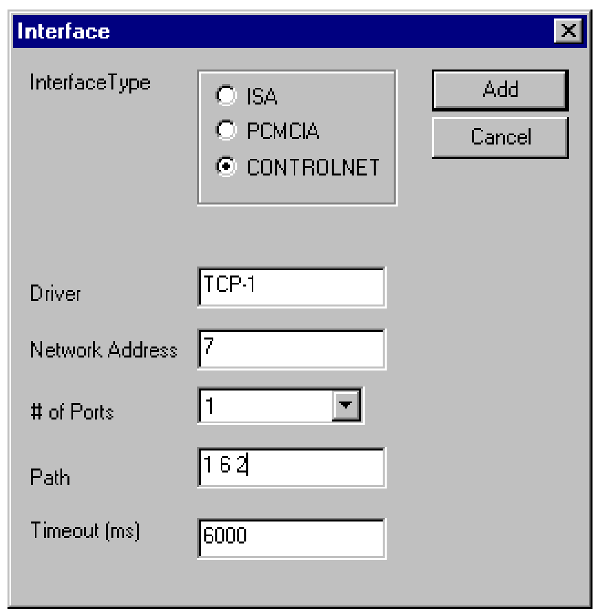
4. Start with a TCP-1 interface, navigate to and highlight the CN2FF.

5. Click **Apply**.

6. Select **Advanced Communications**.



7. Read the Path and copy it to the Path window of the Interface window, refer to page F-12.



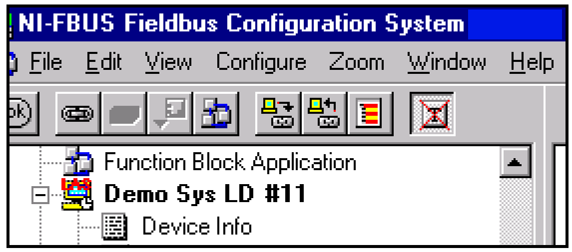
That's it. **Congratulations.**

You have told the Fieldbus Configuration software the path to communicate with the remote CN2FF.

Troubleshooting an Application

The Fieldbus AI Side: After you have created the FF configuration, click on the blue eyeball to see that you are getting data from all the AI FBs.

If you are getting "BAD Data" from a device, check first to see that the Resource Block and the Transducer Blocks in that device are in the "Auto" mode.



To see the Transducer Block, you must uncheck the "T" in the top toolbar that initially had a red X through it.

The Fieldbus AO Side: The AO FB in the output device must be in the Cascade Mode to accept inputs from the MAO FB.

If the data from the MAO module is listed as BAD, you are probably not providing the required FF status byte with your output PV. To show good data, the FF Status Byte needs to have a numeric value between -112 and -128. The best approach is to copy the FF status from the inputs that are supplying the PVs that were manipulated to produce the AO.

IMPORTANT At present, Rockwell Automation is not promoting the use of the distributed control capability of Fieldbus. That means that all the control is done in the controller. We are using the Fail-safe processing capability of the individual Fieldbus devices.

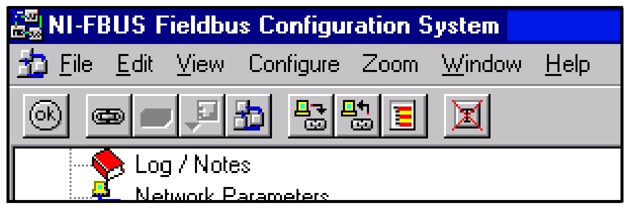
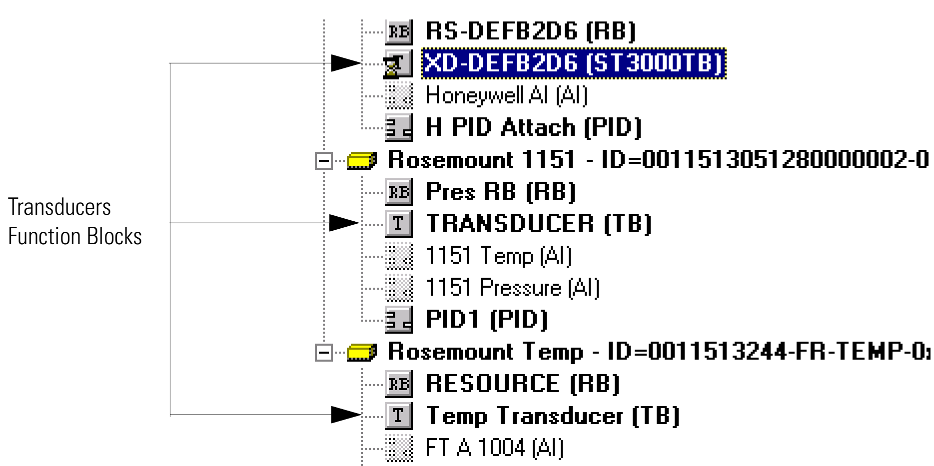
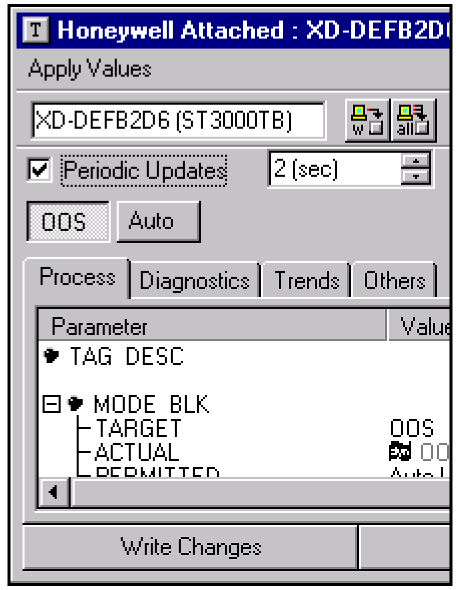


Table F.2 Troubleshooting

If you see:	Do this:
Bad Status, Bad Data Value	Make sure to see that all the Function Blocks and Transducers are in the proper operating mode.
a function block in OOS mode	OOS = Out Of Service Mode. In the FFCT window, click on the "T" with the red X through it. That will Unhide the Transducer Function Blocks.
a Transducer function block in OOS mode	Check to see that the Transducer FB is not in the OOS mode. If it shows OOS, change it. All the Function Blocks, including RB, should be in the Auto mode
Bad Data in the Monitor window	Be sure the RB Function Block is in Auto Mode.



The Transducer FB window opens.



If the OOS is active:

Table F.3

If:	Then:
the OOS is active	• Click on the "Auto" button, to put the Transducer Block in "Auto" mode.
the Transducer goes into Auto mode, and if that was your only problem	• The bad data indication will go away.
the Transducer FB doesn't go into Auto	• Check the mode of the Resource Block and the AI Block. • All should be in Auto Mode. • If the Transducer will not go into Auto Mode refer to the FF device manufacturer for help. • You also need to have the RB Resource Block in Auto Mode.

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future.
Thank you for taking the time to provide us feedback.

You can complete this form and mail it back to us, visit us online at www.ab.com/manuals, or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type Fieldbus Solutions for Rockwell Automation's Integrated Architecture

Cat. No. ProcessLogix, ControlLogix, and PLC5 Pub. No. 1757-UM006A-EN-P Pub. Date May 2002 Part No. 957603-37

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1 2 3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1 2 3	Can we add more information to help you? ☐ procedure/step ☐ illustration ☐ feature ☐ example ☐ guideline ☐ other ☐ explanation ☐ definition
Technical Accuracy (all provided information is correct)	1 2 3	Can we be more accurate? ☐ text ☐ illustration
Clarity (all provided information is easy to understand)	1 2 3	How can we make things clearer?
Other Comments		You can add additional comments on the back of this form.

Your Name ____________________ Location/Phone ____________________

Your Title/Function ____________________ Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at ____________________

___Yes, please contact me via ____________________

Return this form to: Allen-Bradley Marketing Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Phone: 440-646-3176 Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley

RELIANCE ELECTRIC

DODGE

ROCKWELL SOFTWARE

Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733

Publication 1757-UM006A-EN-P - May 2002 PN 957603-37

Allen-Bradley   ***Fieldbus Solutions for Rockwell Automation's Integrated Architecture***   ***User Manual***