Honeywell

Experion

Series C Fieldbus Interface Module

User's Guide

EP-DCX454
R301.1
11/06

Notices and Trademarks

Honeywell International

Process Solutions

2500 West Union Hills

Phoenix, AZ 85027

1-800 343-0228

About This Document

This document provides information for planning, installing, configuring, operating, maintaining and troubleshooting hardware components and software function blocks associated with the Sereis C Fieldbus Interface Module (FIM4). It also includes some basic **FOUNDATION™** Fieldbus technology data for reference.

Release Information

Document Name	Document ID	Release Number	Publication Date
Series C Fieldbus Interface Module User's Guide - cfimg	EP-DCX454	301.1	11/06

References

The following list identifies all documents that may be sources of reference for material discussed in this publication.

Document Title

Contacts

World Wide Web

The following Honeywell web sites may be of interest to Process Solutions customers.

Honeywell Organization	WWW Address (URL)
Corporate	*http://www.honeywell.com*
Honeywell Process Solutions	*http://hpsweb.honeywell.com*

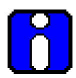
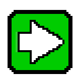
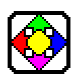
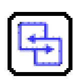
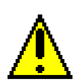
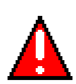
Telephone

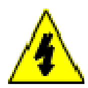
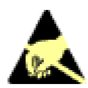
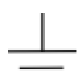
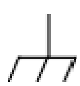
Contact us by telephone at the numbers listed below.

Location	Organization	Phone
United States and Canada	Honeywell IAC Solution Support Center	1-800-822-7673
Europe	Honeywell TAC-EMEA	+32-2-728-2704
Pacific	Honeywell Global TAC - Pacific	1300-300-4822 (toll free within Australia) +61-8-9362-9559 (outside Australia)
India	Honeywell Global TAC - India	+91-20-2682-2458
Korea	Honeywell Global TAC - Korea	+82-2-799-6317
People's Republic of China	Honeywell Global TAC - China	+86-10-8458-3280 ext. 361
Singapore	Honeywell Global TAC - South East Asia	+65-6580-3500
Taiwan	Honeywell Global TAC - Taiwan	+886-7-323-5900
Japan	Honeywell Global TAC - Japan	+81-3-5440-1303
Elsewhere	Call your nearest Honeywell office.	

Symbol Definitions

The following table lists those symbols used in this document to denote certain conditions.

Symbol	Definition
	**ATTENTION:** Identifies information that requires special consideration.
	**TIP:** Identifies advice or hints for the user, often in terms of performing a task.
	**REFERENCE -EXTERNAL:** Identifies an additional source of information outside of the bookset.
	**REFERENCE - INTERNAL:** Identifies an additional source of information within the bookset.
CAUTION	Indicates a situation which, if not avoided, may result in equipment or work (data) on the system being damaged or lost, or may result in the inability to properly operate the process.
	**CAUTION**: Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices. **CAUTION** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.
	**WARNING**: Indicates a potentially hazardous situation, which, if not avoided, could result in serious injury or death. **WARNING** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.

Symbol	Definition
	**WARNING, Risk of electrical shock**: Potential shock hazard where HAZARDOUS LIVE voltages greater than 30 Vrms, 42.4 Vpeak, or 60 VDC may be accessible.
	**ESD HAZARD:** Danger of an electro-static discharge to which equipment may be sensitive. Observe precautions for handling electrostatic sensitive devices.
	**Protective Earth (PE) terminal**: Provided for connection of the protective earth (green or green/yellow) supply system conductor.
	**Functional earth terminal**: Used for non-safety purposes such as noise immunity improvement. NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national local electrical code requirements.
	**Earth Ground**: **Functional earth connection.** NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.
	**Chassis Ground**: Identifies a connection to the chassis or frame of the equipment shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.

Contents

Tables

Figures

Introduction

About this guide

Intended audience

Personnel who are responsible for integrating fieldbus devices with the Experion system.

Prerequisite skills

- Familiar with working in a Windows operating environment.
- Familiar with using these Experion applications:
  - Configuration Studio
  - Control Builder
  - Series C Platform Analyzer
- Familiar with FOUNDATION Fieldbus technology.

How to use this guide

Choose a topic associated with the task you want to complete on the tree view in Knowledge Builder or click one of the topic references listed in the *What Task Do You Want to Complete* section. This guide covers tasks for planning your installation through troubleshooting considerations.

Related Documents

For more information about using Control Builder, see:

- *Control Building Guide*

Series C FIM Purpose

The Series C Fieldbus Interface Module (FIM4) serves as the bridge between the Experion control environment and the fieldbus control functions. 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 Experion system controls downstream function blocks, but it does not allow itself to be controlled by an upstream function block that resides in a fieldbus device.

Series C FIM versus CIOM-A FIM

The Series C Fieldbus Interface Module (FIM4) is designed to complement the C300 Controller and support Fault Tolerant Ethernet (FTE) communications within Experion R300 systems or later. While the Series C FIM is functionally equivalent to the Chassis I/O-Series A (CIOM-A) FIM, its physical design, including mounting and wiring requirements, is drastically different as shown in the following figure. The Series C FIM supports up to four H1 links while the CIOMA FIM only supports up to two. Please refer to the *Series A Fieldbus Interface Module User's Guide* for information that is specific to integrating fieldbus through a CIOM-A FIM.

Figure 1 Series C FIM Mounted on non-redundant Input/Output Termination Assembly (IOTA).

Series C FIM Functional Overview

Like the CIOM-A FIM, the Series C FIM features a multi-level function block model based upon the hardware hierarchy and the physical fieldbus network. The Experion Control Builder application supports integral creation and configuration of fieldbus function blocks with Experion system function blocks to incorporate fieldbus devices in a unified Experion Control Strategy. This means Experion 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. The following figure shows how icons are used to identify Series C FIM, H1 Links, and fieldbus devices in the Control Builder **Project** tab.

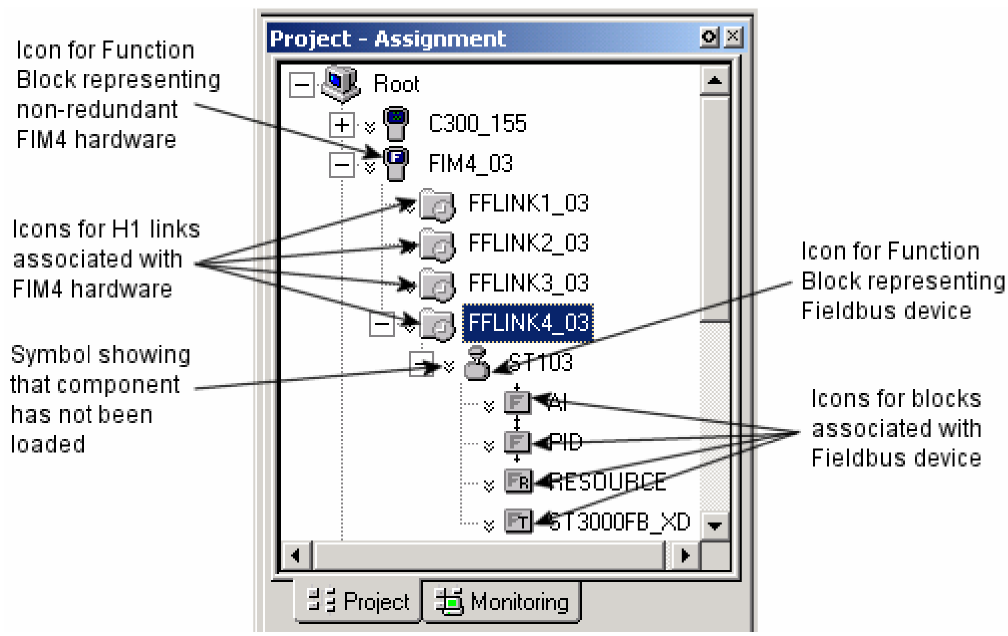


Figure 2 Typical Series C FIM Hardware/Function Block Hierarchy in Project Tab.

Getting Started

This section provides a quick reference to topics in this book for a given task related to using the Series C FIM to integrate fieldbus devices with the Experion system.

What Task Do You Want to Complete

The following table lists some of the tasks covered in this document that are related to implementing a Series C FIM in your Experion R300 system or later. If you are viewing this document online, just click the reference to jump to the topic in this document.

If You Want to . . .	Then, See This Topic . . .
Plan and Design a Series C FIM installation.	*Series C FIM Planning and Design*
Install Series C FIM hardware.	*Series C FIM Installation and Upgrades*
Wire Series C FIM hardware.	*Series C FIM Installation and Upgrades*
Upgrade Series C FIM firmware.	*Series C FIM Installation and Upgrades*
Add FIM4 block with Links to **Project**.	*Adding FIM4 Block to Project*
Check FFLink configuration.	*Checking FFLink configuration*
Make a fieldbus device **Type** from vendor DD.	*Making Fieldbus Device Type from Vendor DD*
Edit device block parameters.	*Editing Device Block Parameters*
Add a fieldbus device to **Project**.	*Adding a Fieldbus Device to Project*
Assign a device to a FFLink in Project.	*Assigning a Device to a FFLink in Project*
Check a device's configuration.	*Checking device configuration*
Make fieldbus block template and assign the block to a device.	*Making fieldbus block template and assigning function block to device*
Use an instantiable block.	*Block Instantiation Support*
Load blocks to FIM4 components.	*Loading FIM4 Components Online*
Review a typical scenario for commissioning a fieldbus device.	*Fieldbus Device Commissioning*
Use a redundant Series C FIM pair.	*Series C FIM Redundancy Functionality*
Power up a Series C FIM.	*Series C FIM Startup*
Change the functional class of a fieldbus	*Checking/Changing Fieldbus Device*

If You Want to . . .	Then, See This Topic . . .
device.	*Functional Class*
Check the current link schedule.	*Viewing and Optimizing Link schedule configuration*
Use Fieldbus Methods Manager to interact with vendor supplied ***Methods*** for given device.	*Using Fieldbus Methods Manager*
Review maintenance related requirements.	*Series C FIM Maintenance*
Investigate the cause of a problem.	*Series C FIM Troubleshooting*
Know more about how fieldbus is integrated with Experion control.	*Fieldbus Integration with Experion Control Reference*
Know more about **FOUNDATION** Fieldbus technology.	*Fieldbus Technology Overview*
Check definitions for fieldbus standard function block parameters.	*Standard Function Block Parameters Reference*

Series C FIM Planning and Design

This section includes the following topics on information and tasks associated with planning and designing an Experion System to include Series C FIMs and fieldbus devices.

General Planning References

Please refer to the following Knowledge Builder publications for planning and design details for the Experion system in general and the Fault Tolerant Ethernet supervisory network. For the sake of brevity, this Guide does not repeat the applicable general guidelines, considerations, cautions, and so on that are covered in these other Guides.

- Control Hardware Planning Guide
- Server and Client Planning Guide
- Fault Tolerant Ethernet Overview and Implementation Guide

Fieldbus Network Wiring and Installation references

Please refer to the following publications for guidance in designing and implementing the fieldbus network to be interfaced to the Experion system through the Series C FIM H1 links.

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 9005 Mountain Ridge Drive Bowie Building — Suite 190

Publication Number/Title	Scope	Source
		Austin, Texas 78759-5316 USA http://www.fieldbus.org/
AG-163 / 31.25 kbit/s Intrinsically Safe Systems Application Guide	Complements the previous document, introduces you to the principles of intrinsic safety, and outlines how to apply approved devices in a hazardous area.	Fieldbus Foundation 9005 Mountain Ridge Drive Bowie Building — Suite 190 Austin, Texas 78759-5316 USA http://www.fieldbus.org/
AG-180 / Engineering Guide and Intrinsically Safe Systems Combo	A general comprehensive reference.	Fieldbus Foundation 9005 Mountain Ridge Drive Bowie Building — Suite 190 Austin, Texas 78759-5316 USA http://www.fieldbus.org/

Identifying Series C FIM Components

The following table identifies the Series C FIM components that will be needed to provide a non-redundant or redundant fieldbus interface with an Experion R300 or later system. The **CC** prefix in a model number.means the component's printed wiring boards are coated to provide additional protection from the environment and the **CU** prefix means the borads are uncoated.

Component	Description	Honeywell Model Number
Series C Fieldbus Interface Module (FIM4)	Module mounts on non-redundant or redundant Input/Output Termination Assembly. Serves as a "gateway" between the Experion system and H1 fieldbus links.	CC-PFB401 CU-PFB401
Non-Redundant Input/Output	Provides physical connection to Series C FIM, H1 Links, and FTE cables. Mounts	CC-TFB401

Component	Description	Honeywell Model Number
Termination Assembly (IOTA)	on carrier in Series C cabinet.	CU-TFB401
Non-Redundant Input/Output Termination Assembly (IOTA)	Same as row above - This model supersedes model CC/CC-TFB401.	CC-TFB402 CU-TFB402
Redundant Input/Output Termination Assembly (IOTA)	Provides physical connection to Primary and Secondary Series C FIMs, H1 Links, and FTE cables. Mounts on carrier in Series C cabinet.	CC-TFB411 CU-TFB411
Redundant Input/Output Termination Assembly (IOTA)	Same as row above - This model supersedes model CC/CU-TFB411.	CC-TFB412 CU-TFB412
Redundant Power Conditioner IOTA (Optional)	Provides physical connection to Series C FIM IOTA to provide the necessary field power to bus-powered devices on a given link. The models listed below are available from the vendor MTL-Relcom. (The Series C FIM does not provide power for the H1 network.)	
	Redundant fielbus power IOTA	F660A
	8-segment redundant fieldbus power IOTA	F860
Non-Redundant Power Conditioner IOTA (Optional)	Same as above but in a non-redundant version. The model listed below is available from the vendor MTL-Relcom.	
	Non-Redundant fieldbus power IOTA	F660A-NR
Power Conditioner Cable	Provides pre-fabricated cable with 16-pin connectors for power conditioner to Series C FIM IOTA connections. The cables listed below are available from the vendor MTL-Relcom.	
	30 cm (1 ft)	FCAB-05
	1 m (3 ft)	FCAB-06

Component	Description	Honeywell Model Number
	2 m (6.5 ft)	FCAB-07
	4 m (13 ft)	FCAB-08
Control Firewall (9 ports)	Provides efficient distribution of FTE cables to multiple Series C FIMs.	CC-PCF901 CU-PCF901
Control Firewall IOTA	Provides physical connection for one uplink and up to eight separate FTE distribution cables Mounts on carrier in Series C cabinet.	CC-TCF901 CU-TCF901
FTE Cable	Provides pre-fabricated STP CAT5 Cable with RJ 45 connectors for FTE connections.	
	2 m (6.5 ft) (Y) = Yellow coded (G) = Green coded	51305482-102 (Y) 513054820202 (G)
	5 m (16 ft)	51305482-105 (Y) 513054820205 (G)
	10 m (33 ft)	51305482-110 (Y) 513054820210 (G)
	20 m (65.5 ft)	51305482-120 (Y) 51305482-220 (G)
Fieldbus Device Usage Licenses	Total number of FIMs per Server actually in use. (Purchase License model numbers in combinations that support the total number of FIMs required. Licenses must be purchased starting with the first FIM used.)	
	1	TC-FFLX01
	5	TC-FFLX05
	10	TC-FFLX10
	50	TC-FFLX50

Selecting and calculating fieldbus wiring requirements

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. It is beyond the scope of this document to address all possible fieldbus wiring scenarios.

You may also want to visit the Relcom Inc. website http://www.relcominc.com/ for more information on available fieldbus wiring products and more wiring design and installation data. They also offer a free *Fieldbus Wiring Guide* for downloading.

Reviewing intrinsically safe application considerations

The Series C FIM accepts up to three Intrinsically Safe (IS) mounted field devices on each H1 network it supports.

Be sure you take the reduced current limit of 83 mA into consideration when calculating performance characteristics for the planned IS fieldbus segment.

You are responsible for adhering to all applicable national codes and local ordinances for applying intrinsically safe equipment in a potentially explosive atmosphere.

Reviewing Series C FIM performance data

The following table lists some Series C FIM (FIM4) performance related data for quick reference. *This information is subject to change without notice*.

Number of Series C FIMs per C300 Controller	15 non-redundant or redundant (60 Links) (Regardless of redundancy)
Number of Series C FIMs per Experion Server	125 non-redundant or redundant pairs (In any combination of Series A FIMs and/or Series C FIMs)
Number of H1 Links per Series C FIM	Up to 4 (Each network defined as a **FOUNDATION Fieldbus** 31.25 kbps H1 network.)
Number of Devices per H1 Link	Up to 16 for non-intrinsically safe applications. Up to 3 for intrinsically safe applications. (See Note 1 below.)
Number of Devices per Experion Server	8000 maximum
Number of Blocks per Series C FIM	1200 maximum (See Note 2 below.)
Maximum Single-Variable Publications per Second on anH1 network	16
Available macrocycle periods	250, 500 milliseconds; 1, 2, 4, 8, 16, and 32 seconds
Number of Unique Block Types per Series C FIM	100 maximum across all devices connected to all links
Number of VCRs per H1 Link	82 maximum (See Note 3 below.)
Number of Agents per Link	100 (***Wires*** between fieldbus devices and Control Builder blocks.)
Number of Blocks per Device	Up to 30
Number of Peer connections per Series C FIM	Up to 5
Link Active Scheduler	Series C FIM supports two LAS domains.

Capabilities	2000 bytes maximum LAS schedule size. LAS schedule can contain up to four sub-schedules. Each sub-schedule can consist of up to 64 sequences. (See Note 4 below.)

Notes:

1. The maximum number of supportable devices per network is highly dependent on application, bandwidth, devices, available current, bus length and topology. An understanding of Fieldbus is crucial to system sizing.

2. Blocks are counted whether or not the fieldbus device's blocks are being used/loaded or not in the FIM strategy. Resources to support every device's block set must be reserved when the fieldbus device is configured on one of the FIM's H1 networks.

3. Each Series C FIM H1 network uses 2 VCRs, each device uses 2 VCRs, and each ***published*** connection to/from a Controller function block uses 1 VCR. Connections between devices do not use Series C FIM VCRs.

4. A 2 second macrocycle could, for example, have subschedule periods of 1 second, 500 milliseconds, and 250 milliseconds.

5. Do not use the Series C FIM with any control module executing more frequently than the fieldbus device.

Series C FIM Installation and Upgrades

This section includes the following topics on information and tasks associated with installing Series C FIM hardware and upgrading Series C FIM firmware as well as converting from non-redundant to redundant configuration or vice versa.

Installation Declarations

ATTENTION

This equipment shall be installed in accordance with the requirements of the National Electrical Code (NEC), ANSI/NFPA 70, or the Canadian Electrical Code (CEC), C22.1. It is intended to be mounted within an enclosure or suitable environment acceptable to the local *authority having jurisdiction*, as defined in the NEC, or *authorized person* as defined in the CEC.

ESD HAZARD

Electrostatic discharge can damage integrated circuits or semiconductors if you touch connector pins or tracks on a printed wiring board. Follow these guidelines when you handle any electronic component:

- Touch a grounded object to discharge static potential,
- Wear an approved wrist-strap grounding device,
- Do not touch the wire connector or connector pins,
- Do not touch circuit components inside a component,
- If available, use a static safe workstation,
- When not in use, keep the component in its static shield box or bag.

WARNING

Unless the location is known to be non-hazardous, do **not**:

- connect or disconnect cables,
- install or remove fuses, terminal blocks, and so on,

while the component is powered.

Installing Input/Output Termination Assembly CC- or CU-TFB401, CC- or CU-TFB411, CC- or CU-TFB402, or CC- or CU-TFB412

Prerequisites

- Carrier channel assembly for mounting IOTA is installed in a cabinet.
- Series C power supply is installed.
- Control Firewalls (CF9s) are installed.
- All wiring and pre-fabricated cables are available and labeled as applicable.
- You have the mounting hardware supplied with the components.

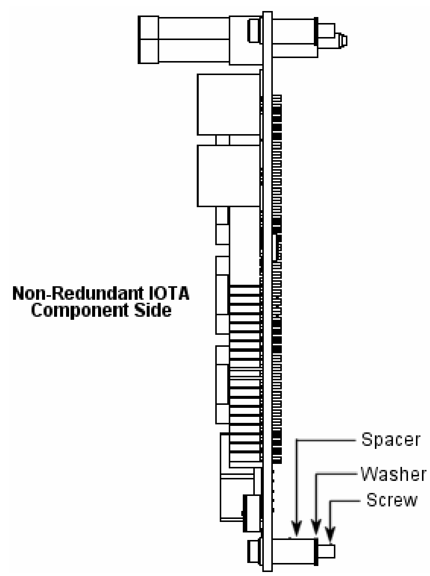
Considerations

- You can use a redundant IOTA to support a non-redundant Series C FIM application. Just be sure to install the non-redundant Series C FIM in the upper module slot on the IOTA.
- Users are responsible for supplying power, power conditioning and wiring for each fieldbus H1 link.
- Be sure the enclosure is connected to a protective earth ground using #8 AWG solid copper wire. There should be metal to metal contact between the grounding bus bar and the enclosure as well as the carrier.
- The Series C FIM supports connectivity of 350 mA for up to 16 non-intrinsically safe mounted devices or 83 mA for up to three intrinsically safe (IS) mounted devices per H1 link.
- You are responsible for observing all local ordinances governing the wiring and installation of electrical equipment.
- The Series C FIM is designed for removal and insertion under power. Be sure to observe general precautions when working with powered electrical equipment.
- The orientation of the **FTE A** and **FTE B** connectors may be reversed depending on the IOTA version that is installed. Be sure to always connect the **yellow** cable to the **FTE A** connector and the **green** cable to the **FTE B** connector.

TIP

When securing a redundant IOTA to the carrier channel assembly (CCA), install the mounting screws in all locations on one side of the board first and then install the screws in the locations on the other side.

Mounting IOTA

Step	Action
1	Select desired mounting location on carrier channel assembly (CCA) and align mounting holes in IOTA with screw hole locations on the carrier. See the following dimension drawings for details.
2	Be sure component side of IOTA is facing up. Secure IOTA to CCA using screws, washers and spacers provided. Insert spacers and washers between bottom of IOTA and top of CCA. 
3	This completes the procedure. Go to *Wiring CC/CU-TFB401 or CC/CU-TFB411 IOTA*, or *Wiring CC/CU-TFB402 or CC/CU-TFB412 IOTA* for connection details, as applicable.

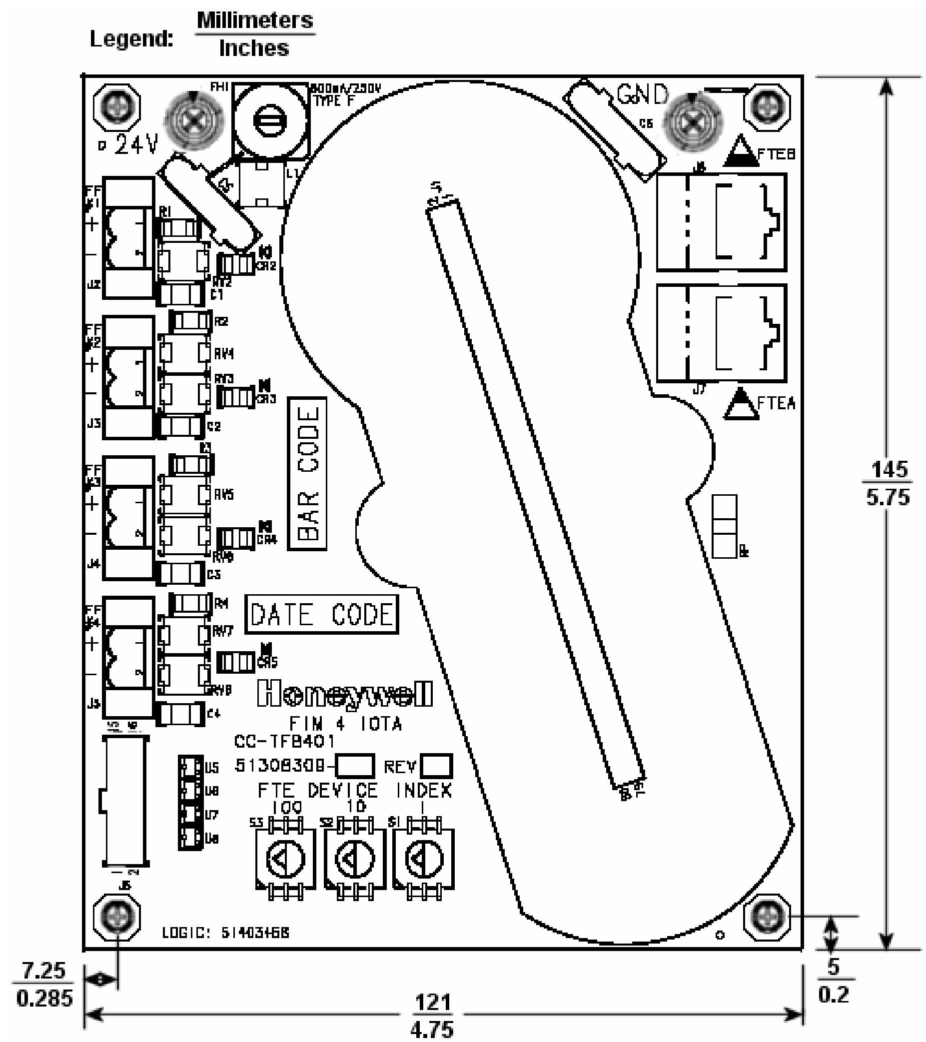


Figure 3 Mounting Dimensions for Non-Redundant IOTA CC- or CU-TFB401 (shown) or CC- or CU-TFB402

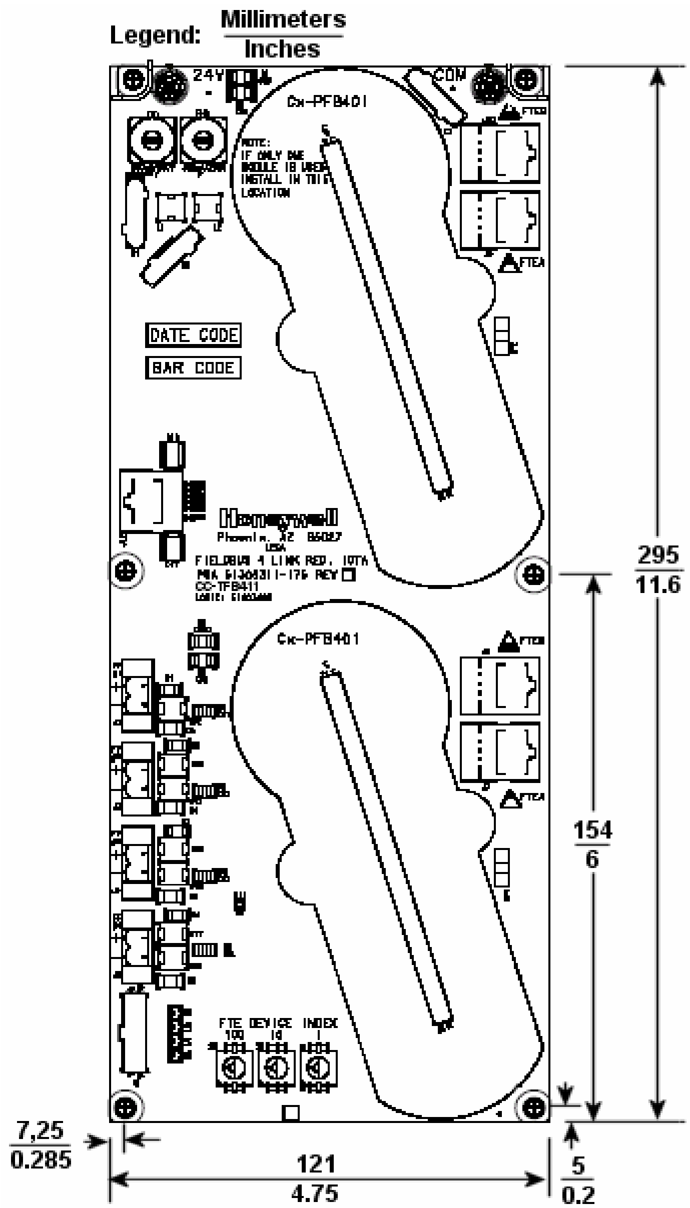
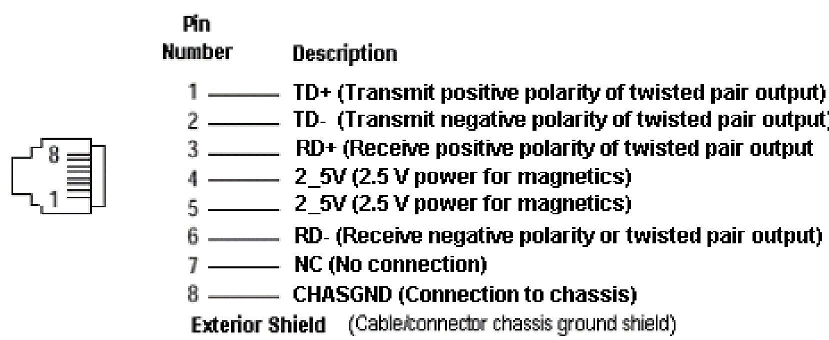
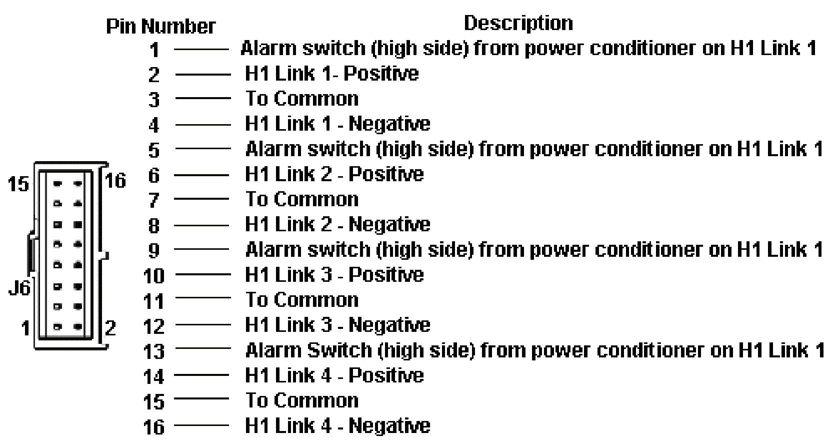


Figure 4 Mounting Dimensions for Redundant IOTA CC- or CU-TFB411 (shown) or CC- or CU-TFB412

Wiring CC/CU-TFB401 or CC/CU-TFB411 IOTA

Step	Action
1	• Connect yellow FTE cable from Control Firewall (CF9) to **J7 FTE A** connector on non-redundant IOTA. For Series C FIM mounted in the upper module slot on a redundant IOTA, connect yellow FTE cable from CF9 to **J9 FTE A** on the IOTA. • Connect green FTE cable from Control Firewall (CF9) to **J8 FTE B** connector on non-redundant IOTA. For Series C FIM mounted in the upper module slot on a redundant IOTA, connect green FTE cable from CF9 to **J10 FTE B** on the IOTA.  • See the following figures for typical connector locations on non-redundant model CC/CU-TFB401 and redundant model CC/CU-TFB411 IOTAs.
2	Repeat Step 1 to connect FTE yellow and green cables for partner Series C FIM in the lower module slot on a redundant IOTA to **J7 FTE A** and **J8 FTE B** connectors, respectively. Otherwise, go to Step 3.
3	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 1** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J2-1** and the negative (-) lead to **J2-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
4	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 2** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J3-**

Step	Action
	1 and the negative (-) lead to **J3-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
5	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 3** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J4-1** and the negative (-) lead to **J4-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
6	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 4** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J5-1** and the negative (-) lead to **J5-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
7	Tighten the screws in terminals **24 Vdc** + and **COM** (logic ground) to the vertical bus bar to connect to the cabinet resident 24 Vdc power supply.
8	If you have the optional redundant power conditioner IOTA, use the cable provided to connect to **J6** on non-redundant or redundant IOTA. 
9	This completes the procedure. Go to next section *Setting Unique Device Index for Series C FIM*.

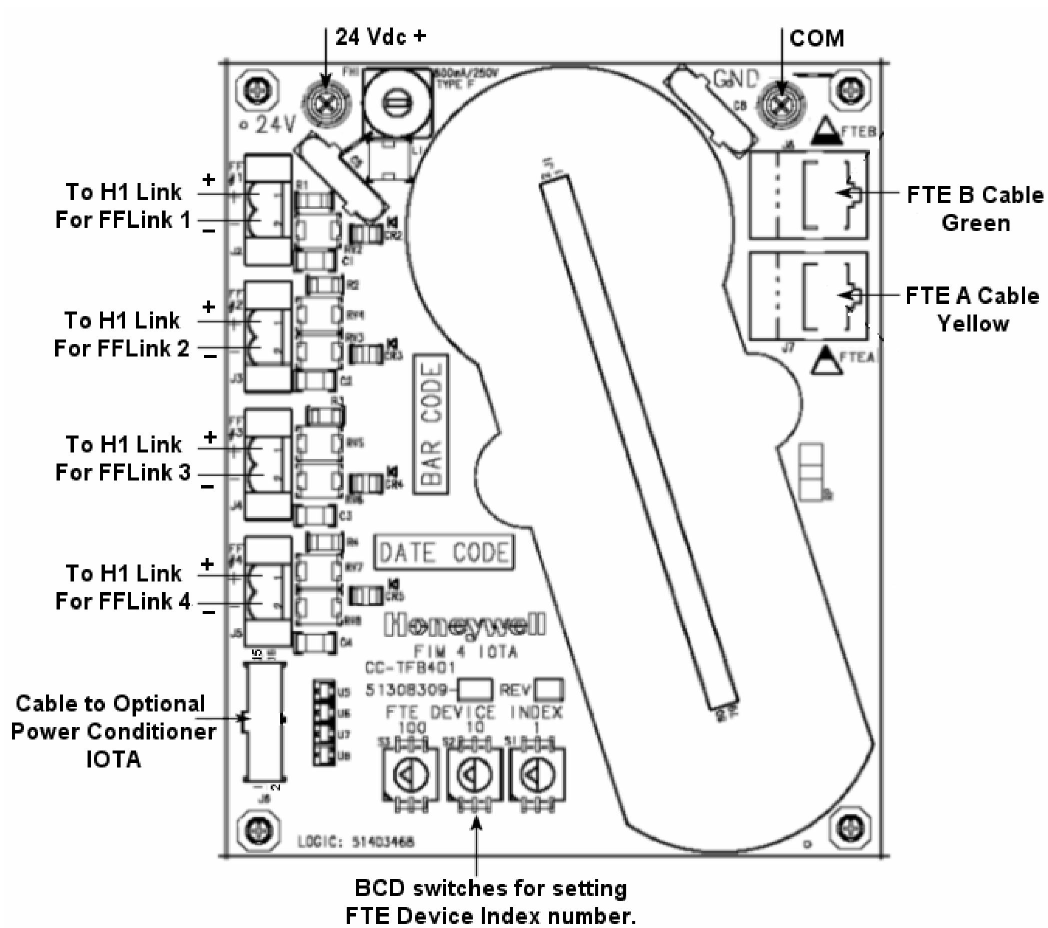


Figure 5 Typical Connector Locations on Non-Redundant Model CC/CU-TFB401 IOTA

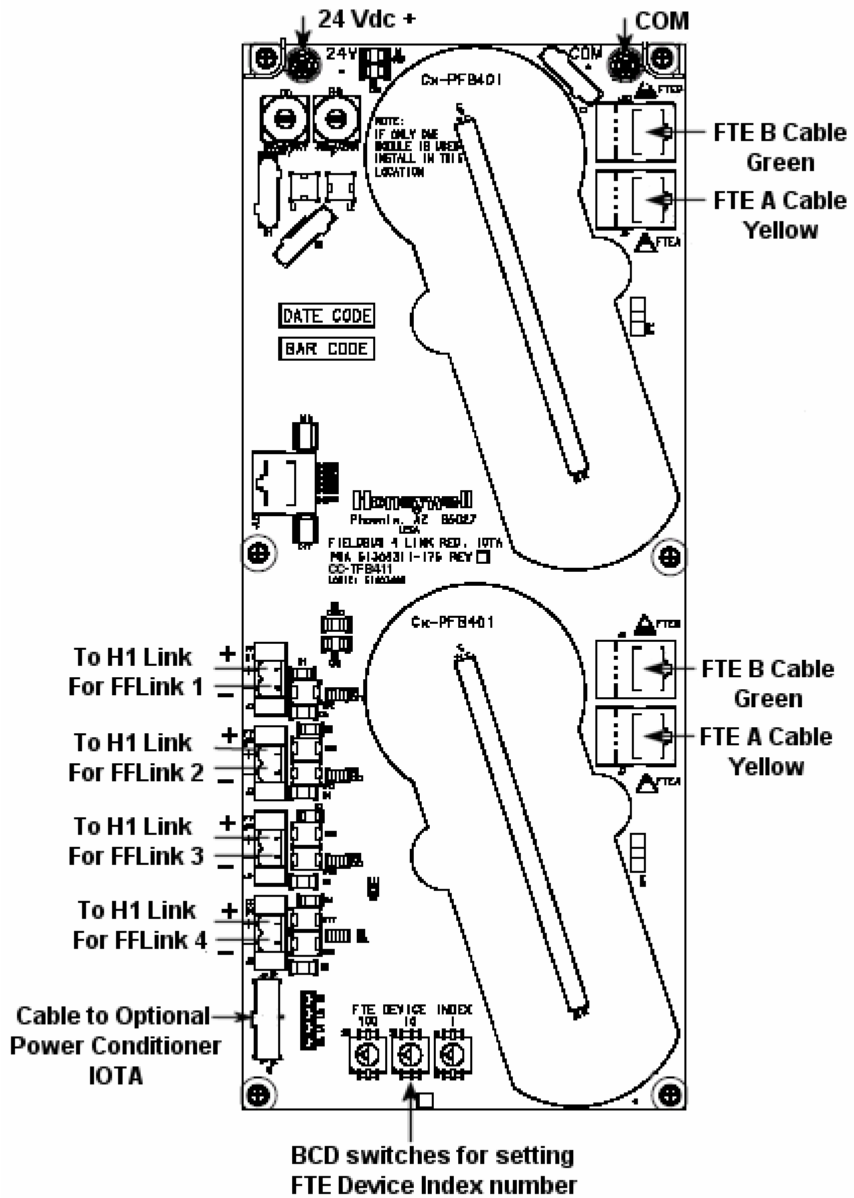
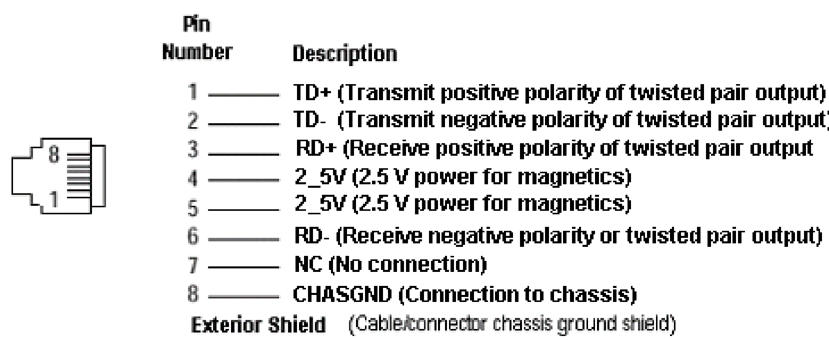


Figure 6 Typical Connector Locations on Redundant Model CC/CU-TFB411 IOTA

Wiring CC/CU-TFB402 or CC/CU-TFB412 IOTA

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>

Connect yellow FTE cable from Control Firewall (CF9) to J8 FTE A connector on non-redundant IOTA. For Series C FIM mounted in the upper module slot on a redundant IOTA, connect yellow FTE cable from CF9 to J10 FTE A on the IOTA.
Connect green FTE cable from Control Firewall (CF9) to J7 FTE B connector on non-redundant IOTA. For Series C FIM mounted in the upper module slot on a redundant IOTA, connect green FTE cable from CF9 to J9 FTE B on the IOTA.



See the following figures for typical connector locations on non-redundant model CC/CU-TFB402 and redundant model CC/CU-TFB412 IOTAs.

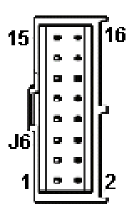
</td></tr>
<tr><td>2</td><td>Repeat Step 1 to connect FTE yellow and green cables for partner Series C FIM in the lower module slot on a redundant IOTA to J8 FTE A and J7 FTE B connectors, respectively. Otherwise, go to Step 3.</td></tr>
<tr><td>3</td><td>Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for FFLink 1 on a non-redundant or redundant IOTA:

Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal J2-1 and the negative (-) lead to J2-2. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.

</td></tr>
<tr><td>4</td><td>Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for FFLink 2 on a non-redundant or redundant IOTA:

Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal J3-

</td></tr>
</table>

Step	Action
	1 and the negative (-) lead to **J3-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
5	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 3** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J4-1** and the negative (-) lead to **J4-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
6	Observing polarity and using the molded plug provided in the connector, make applicable connections as follows for **FFLink 4** on a non-redundant or redundant IOTA: • Connect the positive (+) lead for the fieldbus H1 link 1 cable to terminal **J5-1** and the negative (-) lead to **J5-2**. Connect the cable shield to the field wiring shield landing bus bar in the enclosure.
7	Tighten the screws in terminals **24 Vdc** + and **COM** (logic ground) to the vertical bus bar to connect to the cabinet resident 24 Vdc power supply.
8	If you have the optional redundant power conditioner IOTA, use the cable provided to connect to **J6** on non-redundant or redundant IOTA.  Pin Number — Description 1 — Alarm switch (high side) from power conditioner on H1 Link 1 2 — H1 Link 1- Positive 3 — To Common 4 — H1 Link 1 - Negative 5 — Alarm switch (high side) from power conditioner on H1 Link 1 6 — H1 Link 2 - Positive 7 — To Common 8 — H1 Link 2 - Negative 9 — Alarm switch (high side) from power conditioner on H1 Link 1 10 — H1 Link 3 - Positive 11 — To Common 12 — H1 Link 3 - Negative 13 — Alarm Switch (high side) from power conditioner on H1 Link 1 14 — H1 Link 4 - Positive 15 — To Common 16 — H1 Link 4 - Negative
9	This completes the procedure. Go to next section *Setting Unique Device Index for Series C FIM*.

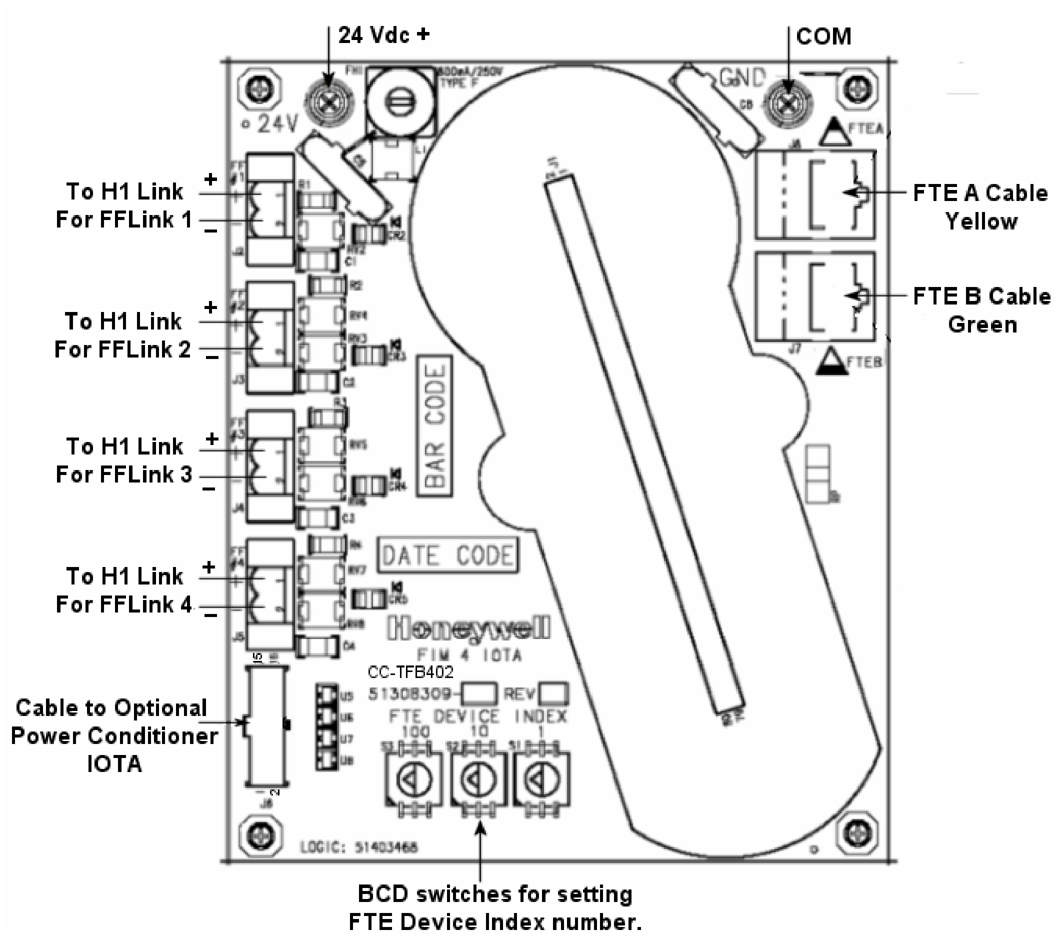


Figure 7 Typical Connector Locations on Non-Redundant Model CC/CU-TFB402 IOTA

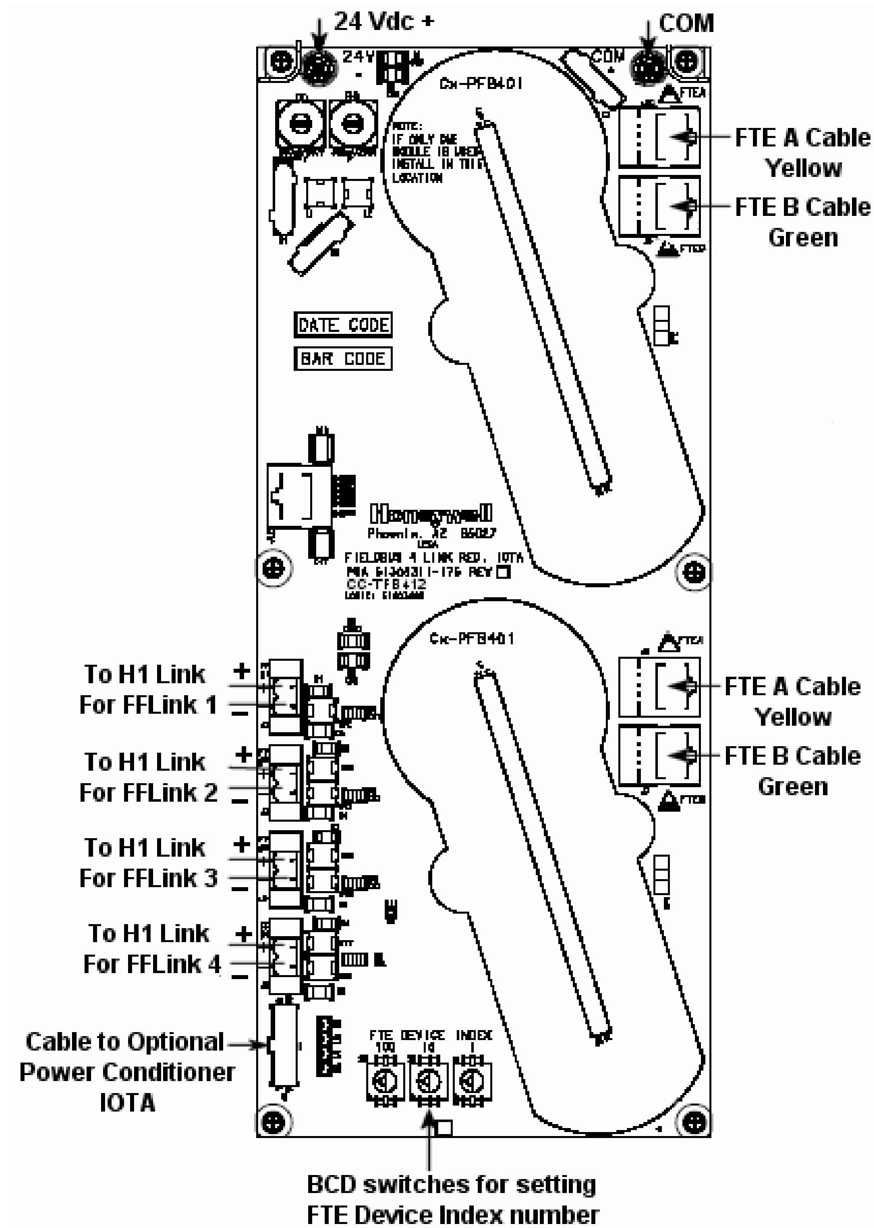


Figure 8 Typical Connector Locations on Redundant Model CC/CU-TFB412 IOTA

Setting Unique Device Index for Series C FIM

Prerequisites

You have mounted and wired the IOTA for a given non-redundant Series C FIM or redundant Series C FIM pair.

Considerations

- The Device Index setting on a redundant Series C FIM IOTA is assigned to the upper module slot.
- For a redundant Series C FIM application, be sure to set the Device Index to be an odd number. The odd number will be used for the primary Series C FIM and a one (1) will be added to the number to be used as a device index for the secondary Series C FIM. For example, if you set the Device Index to 17 for the primary Series C FIM, the Device Index for the secondary Series C FIM would be 18.
- We recommend setting the Device Index for a non-redundant Series C FIM application to an odd number also, so you can easily upgrade to a redundant Series C FIM application at a later date, if desired.

Setting switches

Step	Action
1	Locate the three binary-coded decimal rotary switches on the left side of the IOTA.
2	Use a small flat-bladed screwdriver or your thumbnail to set the left-hand switch (100) to the desired most significant decimal number, middle switch (10) to next desired most significant decimal number and the right-hand (1) switch to the desired least significant decimal number. The applicable setting range is 01 to 511.
3	The completes the procedure. Go to the next section *Installing Series C FIM CC- or CU-PFB401*.

Installing Series C FIM CC- or CU-PFB401

Prerequisites

You have installed non-redundant or redundant IOTA, as applicable.

Considerations

- You can install a non-redundant Series C FIM on a redundant IOTA. Just be sure to install the non-redundant Series C FIM in the upper module slot on the IOTA.
- The IOTA and Series C FIM are keyed, so you cannot install any other Series C module on a Series C FIM non-redundant or redundant IOTA.

Mounting Series C FIM

Step	Action
1	Align Series C FIM connector pins over connector/slot on IOTA. On redundant IOTA, be sure it is the upper slot location.
2	Carefully press down on the Series C FIM until it is fully seated in the connector.
3	Use the three screws provided to secure the Series C FIM to the IOTA.
4	For redundant applications, repeat Steps 1 to 3 to mount a second Series C FIM on the bottom or secondary connector location on the redundant IOTA.
5	This completes the procedure. Go to the next Section.

Installing Redundant Power Conditioner IOTA F660A or F860

Prerequisites

You have installed non-redundant or redundant IOTA, as applicable.

Considerations

- The optional power conditioner is supplied by the vendor MTL-Relcom. Refer to the vendor supplied documentation for more information about the product.
- The F660A redundant power conditioner includes the following components.
  - One unpopulated IOTA part number F660A-C

  - Eight power modules part number FPS-IPM
  - One Alarm module part number F660A-ALM
- The F660A-NR non-redundant power conditioner includes the following components.
  - One unpopulated IOTA part number F660A-C
  - Four power modules part number FPS-IPM
  - One alarm module part number F660A-ALM
  - Four blanking modules part number FPS-BLK
- The F860 8-segment redundant power conditioner includes the following components.
  - One unpopulated IOTA part number F860-CA
  - Two 8-segment power modules F801
- You should mount the Power Conditioner IOTA above or below the associated Series C FIM IOTA on the carrier.
- The optional power conditioner supplies redundant power to all four H1 links on the associated Series C FIM IOTA as well as alarm contacts to monitor circuit status.
- The following procedure is based on installing a model F660A redundant power conditioner. The procedure is very similar for a model F860 redundant power conditioner except that it has two 16-pin connectors for powering 4-segments on two separate Series C FIMs.

CAUTION	If you use a **non-isolated** power conditioner, you must provide a separate **isolated** input power source for the power conditioner IOTA, since the Series C power subsystem is not isolated.

Mounting power conditioner

Step	Action
1	Select desired mounting location on carrier and align mounting holes in IOTA with screw hole locations on the carrier. See the following dimension drawing for details.
2	Be sure component side of IOTA is facing up. Secure IOTA to carrier using screws, washers and spacers provided. Insert spacers and washers between bottom of IOTA and top of carrier.
3	This completes the procedure. Go to *Wiring power conditioner IOTA* for connection details.

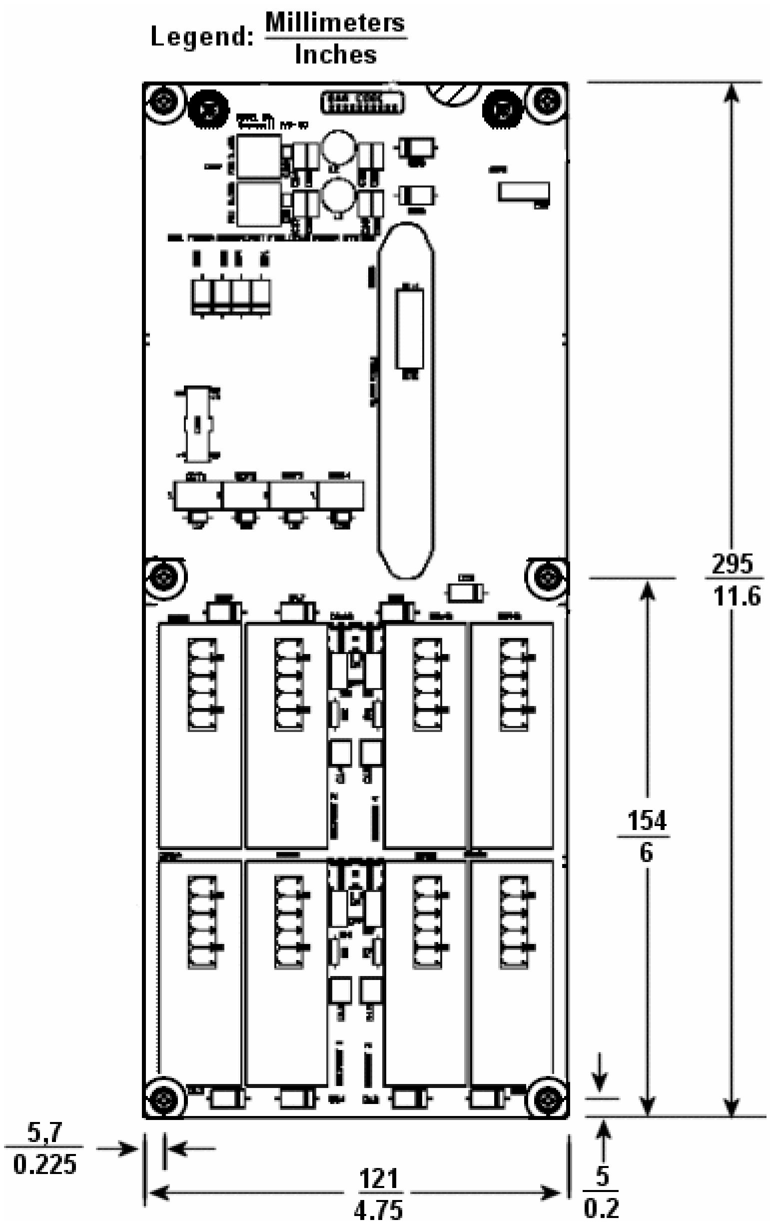


Figure 9 Mounting Dimensions for Redundant Power Conditioner IOTA F660A

Wiring power conditioner IOTA

Step	Action
1	Tighten the screws in both sets of terminals for **24 Vdc** + and **COM** (logic ground) to the vertical bus bar to connect to the cabinet resident 24 Vdc power supply.
2	Connect the cable plug from the 16-pin connector on the IOTA to the mating 16-pin connector on the associated Series C FIM IOTA.
3	This completes the procedure. Go to the next section.

Upgrading Series C FIM Firmware

REFERENCE - INTERNAL

Please refer to the Upgrading Firmware in Multiple Series C Devices section in the *Control Hardware and I/O Module Firmware Upgrade Guide* for information about upgrading firmware in a Series C FIM.

Converting Non-Redundant Series C FIM to Redundant Series C FIM

The following procedure outlines how to convert an existing non-redundant Series C FIM to a redundant one given that specified prerequisites are met.

Prerequisites

- The non-redundant Series C FIM to be converted is currently mounted in the upper module slot on a redundant IOTA.
- The non-redundant Series C FIM to be converted has an Odd number device index.
- You have logged on with sufficient security level to make changes in a control strategy in Control Builder.

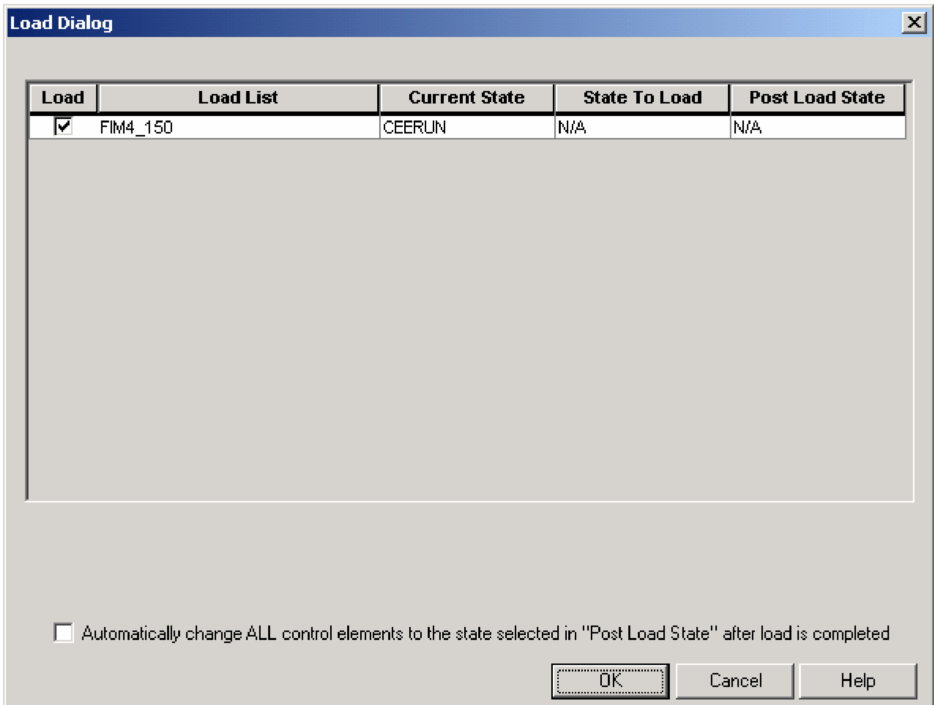
Considerations

- The Series C FIM redundant pair must have odd/odd+1 device index number combination, which means the primary or non-redundant Series C FIM must have an odd address to support an odd+1 address for the secondary Series C FIM. For example, if you set the Device Index to 17 for the primary Series C FIM, the Device Index for the secondary Series C FIM would be 18.

- You can make the conversion to Series C FIM redundancy while the process is online.
- The partner Series C FIM to be added is at the same firmware revision level as the existing primary Series C FIM.

Converting to redundant Series C FIM

Step	Action
1	Install Secondary Series C FIM in the lower module slot on redundant IOTA. See previous Wiring CC/CU-TFB401 or CC/CU-TFB411 IOTA and Mounting procedures for more information.
2	In the **Project** tab, double-click the FIM4 icon for the current non-redundant Series C FIM to open the **FIM4 Block Parameters** form.
3	Click the **Module is redundant** check box to select it.
4	Check that **Secondary Tag Name** is specified. The default name is **Tag Name** plus SEC suffix.
5	Click the **OK** button to save the changes and close the form.
6	Check that FIM4 icon is now redundant and the icon for the secondary FIM4 appears in the **Project** tree.
7	Double-click the Secondary FIM4 icon to open **FIM4SEC Block Parameters** form. Configure block as required. See *Creating FIM4 block* for reference.
8	Click the **OK** button to save changes and close the form.
9	• Right-click the Primary FIM4 icon and select **Load** from the list. • Click the **Continue** button to acknowledge the **Warning** prompt, if applicable. • Confirm that only the FIM4 block is selected for download, as shown in the sample illustration below. • Click the **OK** button to load the block.

Step	Action
	
10	Right-click the Secondary FIM4 icon and select **Load** from the list. Click the **Continue** button to acknowledge Warning, if applicable; and click the **OK** button to load the block.
11	If both Series C FIMs are installed and auto synchronization is enabled, the modules will synchronize when power is applied.
12	This completes the procedure. Go to the next section.

Converting Redundant Series C FIM to Non-Redundant Series C FIM

The following procedure outlines how to convert an existing redundant Series C FIM to a non-redundant one given that specified prerequisites are met.

Prerequisites

- The Primary Series C FIM has an odd number for its device index.
- The Secondary Series C FIM has an odd+1 number for its device index and it is installed in the lower module slot on the redundant IOTA.
- You have logged on with sufficient security level to make changes in a control strategy in Control Builder.

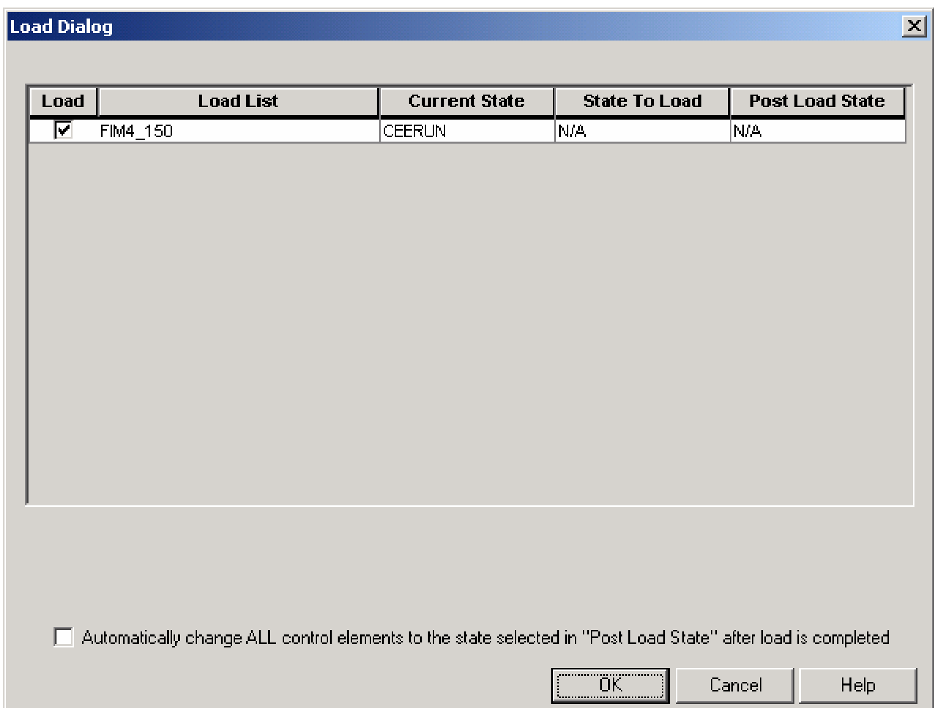
Considerations

- Only the Series C FIM with odd+1 device index number, which is installed in the lower module slot on a redundant IOTA, can be removed from the system.
- If you try to delete a Secondary Series C FIM from the **Monitoring** tab while the redundant Series C FIM pair is synchronized, the command will be rejected.
- If the Secondary Series C FIM is still installed when the Primary/non-redundant Series C FIM is loaded, an alarm will signal the presence of an unconfigured partner.
- You can use a redundant IOTA to support a non-redundant Series C FIM application. Just be sure the non-redundant Series C FIM is installed in the upper module slot on the IOTA.

Converting to non-redundant Series C FIM

Step	Action
1	In the **Monitoring** tab, double-click the Primary FIM4 block icon to open the FIM4 Block Parameters form.
2	On the **Main** tab, confirm that block **Redundancy Role** is Primary and **Device Index** is an odd number.
3	If the Series C FIM with the odd device index number is not the Primary, click the **Redundancy** tab and click the **Initiate Switchover** button to issue a switchover command.
4	On the **Redundancy** tab, click the **Disable Synchronization** button to disable auto synchronization. Click the **Yes** button to confirm the action.
5	Click the **OK** button to close the form.
5	Loosen the three screws in the Secondary Series C FIM and remove it from the lower module slot on the redundnat IOTA. Remove FTE cable connections, as applicable.
6	In the **Monitoring** tab, right-click the Secondary FIM4 icon and select **Delete**

Step	Action
	from the list. Click the **Delete selected object(s)** button to confirm the action.
7	In the **Project** tab, right-click the Secondary FIM4 icon and select **Delete** from the list. Click the **Delete selected object(s)** button to confirm the action.
8	• Right-click the Primary/non-redundant FIM4 icon and select **Load** from the list. • Click the **Continue** button to acknowledge the **Warning** prompt, if applicable. • Confirm that only the FIM4 block is selected for download, as shown in the sample illustration below. • Click the **OK** button to load the block.



Step	Action
9	This completes the procedure. Go to the next section.

Series C FIM Configuration

This section includes the following topics on information and tasks associated with Configuring FIM4, FFLinks, and Fieldbus Devices for use in control strategies created in Control Builder.

Topic
Adding FIM4 Block to Project
Checking FFLink configuration
FFLink Configuration Parameters Reference
Making Fieldbus Device Type from Vendor DD
Editing Device Block Parameters
Adding a Fieldbus Device to Project
Assigning a Device to a FFLink in Project
Checking device configuration
Making fieldbus block template and assigning function block to device

REFERENCE - INTERNAL

Please refer to the *Control Building Guide* for basic functionality details about calling up, navigating and interacting with the application. Also, be aware that menus, selections, tree views, and configuration forms may vary depending upon the licensed options installed on your system.

Third-Party Interface Precautions

You can use available **third-party handheld** and **standalone** Fieldbus **interfaces** to change parameters in Fieldbus devices. These interfaces are particularly useful for bench configuration and calibration functions. You can connect third-party interfaces to the Fieldbus network through one of the four data link visitor addresses (252 to 255) reserved for them. While some interface products appear to successfully co-exist with the Experion Fieldbus Interface Module, others may impede or disrupt the ability to perform some system operations including tag and address reassignment, device commissioning (loading), and Control Module loading. Third-party interfaces also allow access to parameters without regard to any Experion security. They do not honor Experion access rights, so access to make parameter changes is not restricted and changes are not recorded. Be aware that the potential for control disruption exists, if you access and change any parameter in the Experion system. Parameter changes are not logged in Experion and, in some cases, may not be updated in Experion displays.

CAUTION

Be sure you observe the following precautions when using a third-party interface with Fieldbus devices in an Experion system.

- **Only highly trained individuals** who completely understand Foundation Fieldbus and the Experion configuration and support role for Foundation Fieldbus should use third-party interfaces with devices in an Experion system.
- **Never connect** third-party handheld and standalone Fieldbus interfaces while commissioning devices and/or loading (or reloading) Control Modules containing Fieldbus function blocks.
- **Never make** any parameter changes without the **knowledge** and **cooperation** of **plant/mill operations personnel**.
- **Never use** third-party interfaces to **load** or **commission** devices.
- **Always connect** third-party Interfaces to the network as **visitor only** at **one of** these four **addresses: 252, 253, 254, or 255**.
- **Always connect** third-party Interface to the network as **Basic Device** Type. **Never connect** as **Link Master** Type.
- You can use any spare connection terminal on the bus to connect third-party interfaces; but you **must** make sure that the connection is secure or it could disrupt link communications.
- **Carefully consider** the need for using a third-party interface in the field, since most of them are very **SLOW** in comparison to Experion operations.
- **Be aware** that third-party interfaces can use significant bandwidth

that could affect Experion performance.

- **Always remove** third-party interfaces from the network when their intended task is completed.

Adding FIM4 Block to Project

Prerequisites

- You have started Configuration Studio and launched the Control Builder application.
- You have logged on with sufficient privileges to create control strategies using Control Builder.
- You have configured the applicable IP addresses when you set up your FTE network.
- You have configured applicable Base IP address and IP addresses for network time protocol (NTP) servers through the **System Preferences** dialog in Control Builder. See the Control Building User's Guide for more information about setting IP addresses.

Considerations

- You cannot add a FIM4 block to the **Project** tab without first configuring the applicable IP addresses for the FTE network and NTP servers.
- You can only view FIM4 blocks in **Project** tab set for the **Assignment view**. The **Assignment view** shows the relationship among all blocks while the **Containment view** only shows templates that contain other templates or Control Modules (CM), Sequential Control Modules, (SCM) and basic blocks. To toggle the view, right-click in an open area of the tab window and select **Assignment View** or **Containment View** from the list, as applicable.
- You can configure a FIM4 block in the Control Builder **Project** tab without the FIM4 hardware being installed. However, it is good idea to have the FTE communications driver and hardware that is going to be used for the system installed, configured, and running. The FIM4 needs its device index number specified on its configuration form to complete its IP address data. Like the C300 Controller, the Series C FIM represents a hardware module and the block configuration specifies the communication path to the hardware.
- Each FIM4 block is automatically assigned a unique default tag name when it is created. If your system will include multiple FIM4s, you may want to adopt a more structured syntax for naming them.

- The block tag name can be up to 16 characters long and must contain at least one letter (A-Z). It must not contain an embedded space or leading space, and dots are allowed in parameter naming only.
- The item name can be up to 40 characters long and must contain at least one letter (A-Z). It is a name by which an entity is known within the context of the enterprise model. The Item name must be unique among children of the same containment parent in the Enterprise Model Builder hierarchy and should conform to the standard convention for names within the system.
- The Series C FIM module includes a temperature sensor that monitors its ambient temperature. If the module's ambient temperature exceeds the configurable Temperature High Alarm value, it will trigger an alarm. The module's temperature sensing function automatically compensates for wild swings in the ambient to minimize nuisance alarms.
- If you are using a power conditioning solution that is not integrated with the Series C FIM and does not connect to the power conditioner connector on the IOTA, check the Suppress H1 Power Alarm check box on FIM4's Main tab to suppress power alarms.

Creating FIM4 block

Step	Action
1	On the **File** menu, Click **New**>**Interface Modules**>**FIM4 - Fieldbus Interface Module**. Calls up the FIM4 block parameters configuration form **Main** tab with **Tag Name** box highlighted.
2	In the **Tag Name** box, key in desired block tag name or accept the default one.
3	Click the **Item Name** box and key in the name of the item that this object will be associated with in the Enterprise Model Builder hierarchy.
4	Click the **Device Index** box and key in the device index number set on the Series C FIM's IOTA.
5	Click **Alarming Enabled** to enable the alarming option for the block.
6	Click **Temperature High Alarm (deg. C)** box and key in desired threshold at which an alarm is generated for Series C FIM hardware temperature, or accept the default setting of 80 deg. C.
7	Click **CPU Free Low Alarm Limit (%)** box and key in desired threshold at which an alarm is generated for Series C FIM CPU availability in percent, or accept the default value of 15 percent.

Step	Action
8	If Series C FIM is to be part of a redundant pair, click **Module is redundant** check box to select it. Otherwise, go to Step 10.
9	For redundant FIM4, the **Secondary Tag Name** is a combination of configured Tag Name plus SEC suffix.
10	If you want to suppress H1 power alarms, click the **Suppress H1 Power Alarm** check box to select it. Otherwise, go to the next step.
11	If you want to enable the safe handling of new devices option, click the **Safe Handling of New Devices Enable** check box to select it. Otherwise, go to the next step. See *Using Optional Safe Handling of New Devices* for more information.
12	If you want to set a delay time for the device drop off link alarm, click the **Device Drop Off Alarm Delay**. box and key in the desired delay time in seconds. Otherwise, go to the next step.
12	Refer to the *Control Building Guide* and/or the *online help* to complete configuration data on the **Redundancy**, **Server History**, **Server Display**, **Control Confirmation**, and **Identification** tabs. The **Statistics** and **Version** tabs only contain read-only parameters.
13	When you completed entering configuration data, click the **OK** button to close the configuration form and add icon for non-redundant or redundant FIM4 to the **Project** tab. The FIM4 also includes icons for the four H1 fieldbus links that it can support.
14	If you have redundant FIM4 configuration, right-click the Secondary FIM4 icon and select **Module Properties** from the list to call up the **Parameters** configuration form. Configure as required and click the **OK** button to close the form.
15	This completes the procedure. Go to the next section.

Checking FFLink configuration

Prerequisites

You have completed the previous procedure to add a FIM4 block to the **Project** tab.

Considerations

- Like the Series C FIM, you can configure a FFLink through the **Project** tab of Control Builder without having the H1 link installed. However, some parameters on the FFLink configuration form can only be viewed through the **Monitoring** tab with the Series C FIM and FFLink installed and communicating with the system.
- Be sure to click the plus sign in front of the FIM4 icon to open its directory tree and expose the FFLink icons. With the Hierarchal Building license option, must have the Assignment View active to view FIM4 and FFLink blocks.
- In a redundant Series C FIM application, the FFLink blocks are shown with the primary FIM4 block only.
- We recommend that you accept the default values listed for parameters on the **System Management** and **Network Management (LM)** tabs, unless you have prior experience with the devices you are adding to the FFLink. The default settings were carefully chosen to provide the most logical initial settings for most applications. It is more efficient to make any needed adjustments to these parameters in the **Monitoring** mode.
- A wizard will guide you though the FFLink configuration process.
- The block name can be up to 16 characters long and must contain at least one letter (A-Z). It must not contain an embedded space or leading space, and dots are allowed in parameter naming only.
- The item name can be up to 40 characters long and must contain at least one letter (A-Z).
- The following illustration shows the entry fields that are common for time related values. The values shown are the maximums for the given field. Use the Left and Right Arrow keys on the keyboard to move the cursor left or right in the field and use the increase and decrease buttons provided to adjust values in the selected field.

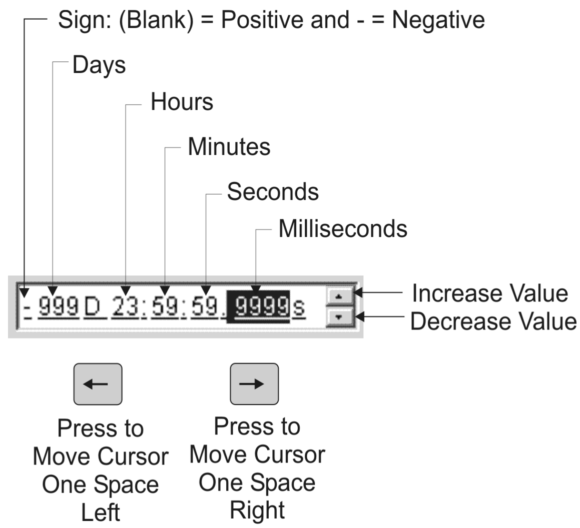


Figure 10 Typical Entry Fields for Time Related Values.

Checking FFLink values

Step	Action
1	Right-click the FFLink icon  and select Module Properties from the list. Calls up the FFLink Block Parameters configuration form with the Name box highlighted.
2	In the **Name** box, key in desired block tag name or accept the default one.
3	Click the **Item Name** box and key in the name of the item that this object will be associated with in the Enterprise Model Builder hierarchy.
4	Click the **Description** box, key in desired descriptive text of up to 132 characters. This text will appear in applicable detail and group displays associated with this block.
5	Click the **Link Schedule Optimization** box, select desired setting from the list: **OPT**, **OPT_NOFORCE**, or **NO_OPT**. Default selection is **OPT**.
6	Click the **Sample Skew** box; select the desired setting from the list: **SKEW** or **NOSKEW**. The default selection is **SKEW**.
7	Refer to the *Control Building Guide* and/or the *online help* to complete configuration data on the **Server History**, **Server Display**, **Control**

Step	Action
	Confirmation, and **Identification** tabs. The **Network Management (Basic)**, **Statistics**, and **Version** tabs only contain read-only parameters.
8	When you completed entering configuration data, click the **OK** button to close the configuration form.
9	Repeat this procedure for Links 2, 3, and 4, as required.
10	This completes the procedure. Go to the next section.

FFLink Configuration Parameters Reference

The following table summarizes the fieldbus related parameters that are user configurable through the **FFLINK Block** configuration form. These parameters let you uniquely identify the FFLink component in the control strategy and tune the performance of FFLink communications. Only the FFLink parameters that are user configurable are listed. Some parameters are only available in the **Monitoring** mode and they are identified as Monitor Only in the table.

Tab	Parameter	Description
Main	Acting Primary Link Master (ACTINGLM)	Monitor Only This read-only parameter indicates whether or not this node is the acting primary Link Master. A check mark in the check box means that the node is the acting primary Link Master. A blank check box means that the node is **not** the acting primary Link Master.
	Link Schedule Optimization (SCHDOPTIMIZE)	Specifies if the link schedule is to be automatically optimized or not. Use the **OPT** selection for Link Scheduling Optimization that changes ORDERINCM values. The optimized schedule is used to update the natural schedule details. The ORDERINLINK/ORDERINCM parameter values are updated to match the sequence of the optimized schedule. The next time that the schedule is viewed/loaded no prompt asks for confirmation of the action. Use the **OPT_NOFORCE** selection for Link Scheduling Optimization that ignores ORDERINCM

Tab	Parameter	Description
		values. The optimized schedule is used to update the natural schedule details. The ORDERINLINK/ORDERINCM parameter values are left unchanged. The next time that the schedule is viewed/loaded a prompt will ask for confirmation of this action. Use the **NO_OPT** selection to disable Link Schedule Optimization for this Link and use ORDERINCM values. The natural schedule is used. The DISABLELINKOPT is set true so that an optimized schedule is no longer created for the link. If an optimized schedule is required later, change this parameter selection to **OPT** or **OPT_NOFORCE**.
	Sample Skew (SAMPLESKEW)	Determines when input blocks will execute. Use the **SKEW** selection to schedule each input block to execute at the time closest to where they converge on a common block like an input selector (minimum loop latency). Use the **NO_SKEW** selection to schedule input blocks to execute at the same time where they converge on a common block like an input selector (minimum scan skew).
System Management	Step Timer Preset (T1)	Specifies the preset time for the System Management step timer in seconds. The default value is specified by the communications profile for the application area. This is an inter-sequence timer. It ensures that devices responding to system management requests have sufficient time to carry out necessary actions and transmit a response.
	Preset Set Addr Seq Timer (T2)	Specifies the preset value for the System Management set address sequence timer in seconds. The default value is specified by the communications profile for the application area. This is a sequence duration timer. It ensures that incorrectly executed or incomplete sequences are aborted in the SMK.

Tab	Parameter	Description
	Preset Addr Wait Timer (T3)	Specifies the preset value for the set address wait timer in seconds. The default value is specified by the communications profile for the application area. This timer allows time for a device at a new address to be accessed and added to the delegated token list of the LAS.
Network Management (LM)	Def Min Token Delegation Time (LMINFREC.DMDT)	Defines the default minimum amount of local link capacity that the LAS allocates to a single Pass Token Data Link Protocol Data Unit (PT DLPDU) sent to the Data Link Entity (DLE). Its range is 32 to 32767, its default value is 84 + Data Link Protocol Data Units (DLPDU) Physical Layer Overhead (PhLO) value, which permits the sending of one URGENT DLPDU, and its unit is the transmission duration of one octet.
	Def Token Hold Time (LMINFREC.DTHT)	Defines the default initial amount of local link capacity that the LAS DLE allocates to each DLE, in one cycle of “circulating the token”, when the LAS sends one or more PT DLPDUs to the DLE. Its range is 276 to 65000 and its default value is 276 + Data Link Protocol Data Units (DLPDU) Physical Layer Overhead (PhLO) value, which permits the sending of one URGENT DLPDU, and its unit is the transmission duration of one octet.
	Target Token Rotation Time (LMINFREC.TTRT)	Specifies the desired upper bound on the time required for one cycle of the "circulating the token" to all the DLEs on the local link that the LAS DLE uses. One cycle of "circulating the token" is measured as the interval between successive occurrences of the LAS DLE sending a PT DLPDU, with a token-use sub field specifying Restart to the lowest numbered node address represented in the local link live list. Its range is 1 to 60000. The unit is one millisecond.

Tab	Parameter	Description
	Link Maint Token Hold Time (LMINFREC.LTHT)	Specifies the initial amount of local link capacity that the LAS DLE should allocate to LAS related link maintenance activities in one cycle of "circulating the token". Its range is 292 to 65000 and its default value is 292 + Data Link Protocol Data Units (DLPDU) Physical Layer Overhead (PhLO) value + Immediate Response Recovery Delay (IRRD) value + Slot Time (ST) value, which permits the probing of one node address and the sending of one time available DT DLPDU. The unit is the transmission duration of one octet.
	Time Distribution Period (LMINFREC.TDP)	Determines the minimum frequency of time distribution on the local link. Its initial value is the minimum value required for the link's time synchronization class and its range is 5 milliseconds to 55 seconds. The unit is one millisecond.
	LAS DB Status Distribution Period (LMINFREC.LDP)	This parameter is used by the LAS to determine the time between two successive distributions of the LAS' database by means of the LAS database status SPDUs sent on the local link. Its range is 100 milliseconds to 55 seconds. The unit is one millisecond.
	Slot Time (LMLSTREC.ST)	Indicates the configured slot time value. It is a fundamental link parameter with multiple uses. Each Link Master DLE connected to the link uses Slot-time to determine how long that DLE monitors the link for inactivity before sending a CLAIM LAS (CL) DLPDU. Slot-time is defined such that the nominal link-inactivity monitoring periods of two DLEs that have consecutive node addresses and do not hold any token differ by exactly one slot time. Its range is 1 to 4095, and its unit is the transmission duration of one octet. The default value is 8, which should be valid for most applications. Slot-time defines the minimum upper bound on the maximum two-way asynchronism in immediate communications among interacting DLEs on the local link when trying to initialize the link, maximized across all pairs of DLEs on that local link. It is an aggregate measure of the worst-case implementation delays within the intervening media,

Tab	Parameter	Description
		the PhL, and the PhL/DLL interfaces, all of which limit the speed of two-way DLE interaction on the local link.
	DLPdu Physical Layer Overhead (LMLSTREC.PHLO)	Indicates the configured Data Link Protocol Data Units (DLPDU) Physical Layer Overhead (PhLO) value. The DLE uses it to account for the Physical Layer induced delay between the end of the last octet of one DLPDU, as it appears on the link, and the beginning of the first octet of any other DLPDU, as it appears on the link. It is measured in units of one octet-duration. Its range is 2 to 63.
	Max Response Delay (LMLSTREC.MRD)	Indicates the configured maximum response delay value. It is a measure, in units of one slot time, greater than the worst-case period of **local** Physical Entity inactivity that a DLE can observe. The DLE uses it to record the Link's maximum response delay. Its range is 1 to 11.
	First Unpolled Node ID (LMLSTREC.FUN)	Indicates the first unused node address value. It is the first node address of a series of consecutive node addresses, which are to be omitted from the orderly probe of node addresses for DLE's not specified by the local link live list. Its range is 21 to 247.
	This Link (LMLSTREC.TL)	This parameter is a two-octet primary identifier for the local link, within the extended link, whose values are constrained. The DLE uses it to record the local link ID. Its default un-initialized value is zero, which does not permit a bridge to forward to another link. Its range is 0, and 4096 to 65279.
	Minimum Inter Pdu Delay (LMLSTREC.MID)	Specifies the minimum duration for an interval of non-transmission that a DLE provides after either receiving or transmitting a DLPDU. It is measured in units of the transmission duration of one octet. The default value is 16 and its range is 0 to the smaller of 120 or (Maximum Response Delay -1) x Slot Time.

Tab	Parameter	Description
	Number of Consecutive Unpolled (LMLSTREC.NUN)	Specifies the number of consecutive unused node addresses, which are to be omitted from the orderly probe of node addresses for DLE's not specified by the local link live list. Its range is 0 to 228 and its default value is 196
	Preamble Extension (LMLSTREC.PMBEXTN)	Specifies an extension period for the preamble. A preamble is transmitted at the beginning of each Physical Layer Protocol Data Unit (PhPDU) to synchronize bit times. The period may be extended, but not reduced. It is an Unsigned8 variable with allowable values between 0 and 7.
	Post Trans Gap Extension (LMLSTREC.PTGEXTN)	Specifies an extension to the minimum period during which a subsequent transmission shall not commence after transmission of a PhPDU. For the same minimum period after reception of a PhPDU, the receiving Physical Layer entity shall ignore all received signaling. A Medium Dependent Sublayer (MDS) entity shall set a minimum post transmission period of four nominal bit times. The period may be extended, but not reduced. It is an Unsigned8 variable with allowable values between 0 and 7.
	Max Inter Channel Signal Skew (LMLSTREC.MXICSSW)	Defines the maximum accepted differential delay between any two active channels, as measured from the first PhPDU of a start delimiter, if the device is to receive concurrently on more than one channel. The delay shall not exceed five nominal bit times. This period may be extended, but not reduced. The value of the post transmission gap shall be greater than the value of inter channel skew. It is an Unsigned8 variable with allowable values between 0 and 7.
	Time Sync Class (LMLSTREC.TIMSCLS)	Specifies one of the time synchronization classes. The DLE uses it to record the Link's time synchronization class and to determine the DLE's own requirement for the minimum time distribution period it needs to maintain the specified level of time synchronization. Its default value is 10 ms. The eight classes are None, 1microsecond, 10 microseconds, 100 microseconds, 1 millisecond, 10 milliseconds, 100 milliseconds, and 1 second.

Tab	Parameter	Description
	Primary Link Master (PRIM_LM_FLG)	Monitor Only Indicates whether or not this node is the primary Link Master. A check mark in the check box means that the node is the primary Link Master. A blank check box means that the node is **not** the primary Link Master. A write to this parameter orders the receiving Link Master to try to become the Link Active Scheduler (LAS). If the Link Master becomes the LAS, it returns an FMS Write.rsp(+), which turns the flag On (check mark). If this attempt fails, it returns an FMS Write.rsp(-), which keeps the flag Off (blank). Notes: • It is a configuration error to have more than one device on a link configured as the primary Link Master. • The Series C FIM is the designated primary Link Master for both of its Links. • The value of this parameter may only be set to On (check mark) through a FMS Write or a local interface. • When this parameter is On and the device is attached to the network at one of the default addresses, the NMA regards the value as a configuration value and does not attempt to acquire the LAS. • When this parameter is On or it receives a FMS Write.ind to turn it On from the NMA of a Link Master device that is not the LAS and is at an assigned address instead of a default one on the network, the NMA instructs the Data Link Management Entity (DLME) to acquire the LAS. The two possible outcomes are as follows. a) If the DLME indicates that the attempt failed, the NMA sets the value of this parameter to Off. If a FMS write response is pending, the NMA returns a negative response. b) If the request succeeds on either the first or

Tab	Parameter	Description
		the second attempt, the NMA sets the value of the parameter to On (if not On already). If an FMS write response is pending, the NMA returns a positive response.
	Boot Operational Function Class (BOOT_OPER)	Specifies the Operational Function Class that the device assumes when it is powered up. The class selections are Basic, Link Master, and Bridge. The **Series C FIM** is the Primary Link Master for all Links.

Making Fieldbus Device Type from Vendor DD

Prerequisites

You must have the Device Description (DD) files for the fieldbus device either on a disk supplied by the manufacturer or downloaded from the vendor's web site. This means you can make a device block type without being connected to a device.

Considerations

- The DD files can be stored in this directory location *C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\ER\Release*. This directory must include these three files:
  - The *.FFO file is the DD binary
  - The *.SYM file is the device symbol file
  - The *.CFF file is the device capability file.
- You can skip this procedure, if a block for the given device type already exists in the Control Builder **Library** database.
- If you use a version 4.02.02 or older Honeywell Fieldbus device or a version 2.02.02 or older FlowServe Fieldbus device as a backup Link Master node with redundant Series C FIMs, we recommend that you assign the device to address **40** (decimal).
- Do not assign addresses below **30** to any backup Link Master devices.
- You need Knowledge Builder version 4.3 or greater installed for device help information to be added to the Dynamic Help book in Knowledge Builder.
- Be sure the Knowledge Builder client's server location is the same as the Server location where the Control Builder's system Knowledge Builder client is pointing.

Knowledge Builder cannot receive updates from beyond more than one server location. If there is a redundant pair of Servers, it is best to define Server B as the target installation location of Knowledge Builder.

Creating fieldbus device type

Step	Action
1	On the **File** menu, click **New**->**Type**->**Fieldbus Device**.
2	Click the **OK** button on any message dialogs that may appear to acknowledge the error messages.
3	In the **Select Device Type** dialog, key in, click **arrow** button, or click the **continue** button to use the **Browse for Folder** dialog to enter the applicable DD file location in the **Device DD Files** box. The default location is **C:\Program Files\Honeywell\Experion PKS\Engineering Tools\System\Er\Release**
4	In **Device List:** box, click the device you want to add to the Control Builder Library.
5	Click the **OK** button to initiate the block type creation. Monitor progress through the **Opening Device Vendor Files**, **Creating Definition Files**, and **Importing the Block Type to ERDB** progress dialogs that open in succession.
6	Click the **OK** button to acknowledge the **Control Builder** dialog message about device help information in Knowledge Builder.
7	Click the **OK** button to acknowledge the completed message.
8	In the **Library** tab, look for the new device block type under the Vendor's name.
9	This completes the procedure. Go to the next section *Editing device block parameters*.

Editing Device Block Parameters

Prerequisites

You have completed the previous procedure to add fieldbus device type to the **Library** tab.

Considerations

- Be sure you review the **Load Destination** column for manufacturer specific parameters listed on the **Vendor** tab in the Parameter Definition Editor.
  - The NOLOAD setting essentially means that the parameter will not be loaded to the device, when the device is downloaded from the **Project** tab in Control Builder, so the corresponding parameter in the device itself will not be overwritten.
  - The LOAD setting means that the parameter will be loaded to the device and will overwrite the corresponding parameter in the device itself. Re-set the Load Destination to LOAD, if you want to change the settings for any given parameter during block configuration.
- Be very judicious about selecting what parameters to edit. By keeping the default values, you may use the same block type for like vendor devices used in multiple locations in your application. You can make adjustments to selected device parameters through Control Builder configuration access. Vendors usually supply a text file that includes definitions for each function block used in their device. This text file includes the help that is added to the Dynamic Help in Knowledge Builder when the device type is created.

Making edits

Step	Action
1	On the **Library** tab, click the plus sign (+) for the fieldbus device type that includes the blocks you want to edit. Do the same for the device icon to expose its blocks.
2	Double-click the block you want to edit to open the **Parameter Definition Editor** in the Control Drawing view pane of Control Builder.
3	View or edit parameters for the selected block in the **Parameter Definition Editor**. Refer to the *Parameter Definition Editor Reference* for more information about using the Parameter Definition Editor.
4	Click the close button on upper right-hand corner of the **PDE** to close it.
5	If prompted to save changes, click the **Yes** button to save edits or the **No** button to cancel them.
6	Repeat the procedure for other device blocks, as required.
7	This completes the procedure. Go to the next section.

Adding a Fieldbus Device to Project

Prerequisites

You have completed the previous procedures to add FIM4, Links, and device Type through Control Builder.

Considerations

- Devices created in **Project** tab automatically include only their fixed function blocks, transducer blocks, and resource blocks. If the device block type includes instantiable blocks, you must drag and drop the instantiable block from the block type in the **Library** tab to the device icon in the **Project** tab to instantiate (create) it in the device in **Project**. See *Instantiable block implementation considerations* for reference.
- You can use drag and drop or menu selections method to assign a device to a FFLink in the **Project** tab.
- We suggest that you wait to check all block configuration parameters until after the device is assigned to a FFLink.

Using drag and drop method

Step	Action
1	Click + sign in front of the applicable vendor type directory in the **Library** tab to show stored device block types.
2	Drag the device icon to an open area in the **Project** tab. (Note that cursor appears with + sign, when positioned in a valid location.) Calls up the **Name New Function Block(s)** dialog box.
3	In the **Destination** box, key in desired name consisting of up to 16 characters or accept the default name for the device. Click the **Finish** button.
4	Check that device instance with given name is created in **Project** tab under the Unassigned category.
5	Repeat this procedure to add other devices or go to the next section *Assigning a device to a FFLink in Project.*

Using file menu method

Step	Action
1	On the **File** menu, click >**New**->**Devices**->**Some Vendor Name Fieldbus Device**. (Note that the **Devices** menu is dynamically updated to include the vendor and device data for block types created through the Type function.) Calls up the **Block Parameters** configuration form for given device with **Name** box highlighted
2	In the **Name** box, key in desired name consisting of up to 16 characters or accept the default name for the device. Click the **OK** button
3	Check that device instance with given name is created in **Project** tab under the Unassigned category.
4	Repeat this procedure to add other devices or go to the next section *Assigning a device to a FFLink in Project.*

Assigning a Device to a FFLink in Project

Prerequisites

You have created a Series C FIM block in the **Project** tab.

Considerations

- An alternate method to the following procedure is to just drag and drop the device to the applicable FFLink in Project.
- If duplicate address error message appears, go to the next section *Checking device configuration* for details on changing the device's network node address. It is possible for device addresses to be set to the same default value of 20.

Device assignment

Step	Action
1	Click **Edit**->**Execution Environment Assignment**. Or, click the assign button  in the toolbar. Calls up the **Execution Environment Assignments** dialog box.
2	Click **Devices** tab. Shows all devices currently in **Project** tab.
3	• Click the device you want to assign to a FFLink to highlight it.

Step	Action
	• Confirm that the desired FFLink is selected in the **Assign To** list box. • Click the **Assign →** button to assign the selected device to the selected FFLink.
4	• Check that device is added to the **Assigned Modules** list box with the assign state icon .
5	Repeat Steps 3 and 4 of other devices as required.
6	Click the **Close** button.
7	This completes the procedure. Go to the next section.

Checking device configuration

Prerequisites

You have assigned the device to a FFLink in the **Project** tab.

Considerations

- While the device has been assigned to a FFLink, this is still *offline* configuration of a matching physical device that is to be connected to this FFLink.
- 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.
- The preferred address range for fieldbus devices is 20 to 40 decimal (0x14 to 0x28 hexadecimal). This provides the most optimized default network range where the first unused node address (FUN) is 41decimal (0x29 hexadecimal) and the number of unused addresses is (NUN) is 206. We recommend that the fieldbus device designated as the backup Link Master be given 21decimal (0x15 hexadecimal) as its assigned address. All address number entries and displays are based on the decimal system in Control Builder, but the Fieldbus Foundation uses the hexadecimal system.
- The Device Identification field is blank until the Project device is matched to an uncommissioned device.

- The Physical Device Tag (PD_TAG) and Device State (DEVSTATE) parameters on the **Main** tab are unavailable, since they are only active in the **Monitoring** tab after the FIM4/FFLINK/DEVICE is loaded and communicating with the system.
- In the Experion system, 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.
- Note that 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 device configuration form matches the device name configured in the device.
- We recommend that you do **not** change the default value settings for the active parameters on the **System Management** tab unless you are familiar with tuning the performance of fieldbus links. In most cases, the parameter values will mirror those configured for the FFLink. The unavailable parameters are only accessible in the **Monitoring** tab with the FIM4/FFLINK communicating with the system.
- If the form includes a **Block Instantiation** tab, this device supports instantiable blocks. Please see *Instantiable block implementation considerations* in this book for more information.

Configuring Fieldbus device

Step	Action
1	Double-click the device icon in the **Project** tab. Calls up the parameter configuration dialog box with **Tag Name** (**Tag NAME**) box highlighted.
2	Accept the assigned name or key in a new one of up to 16 characters. Must be unique within the system.
3	Click the **Item Name** (**Item Name**) box. Key in the name of the Entity that this object will be associated with in the Enterprise Model Builder hierarchy.
4	Click the **Description** box. Key in desired description for the device consisting of up to 59 characters or leave it blank.
5	Click he **Device Network Node Address** (**ADDR**) box. Accept the default or key in the device's current address assignment.

Step	Action
6	Click the **System Management** tab to display it.
7	Click the **Step Timer Preset** (**T1)** box. Accept the default value or key in a new value. This is the preset value for the System Management step timer. The default value is 15. 0000s.
8	Click the **Preset Set Addr Seq Timer** (**T2**) box. Accept the default value or key in a new value. This is the preset value for the System Management set address sequence timer. The default value is 01:30. 0000s.
9	Click the **Preset Set Addr Wait Timer** (**T3**) box. Accept the default value or key in a new value. This is the preset value for the SM set address wait timer. The default value is 45. 0000s.
10	Click the **Network Management (LM)** tab to display it. We recommend that you accept the default values for the parameters on this tab, and revisit this tab in the **Monitoring** mode, when the FIM4/FFLINK is communicating with the system.
11	Refer to the *Control Building Guide* and/or the *online help* to complete configuration data on the **Server History**, **Server Display**, **Control Confirmation**, and **Identification** tabs. The **Network Management (Basic)** and **Version** tabs only contain read-only parameters.
12	Click the **OK** button.
13	Repeat this procedure for another device, as required.
14	This completes the procedure. Go to the next section.

Making fieldbus block template and assigning function block to device

Prerequisites

- You have created a Control Module in the **Project** tab.
- You have created a fieldbus device and assigned it to a FIM4 link in the **Project** tab.

Considerations

- The following procedure makes and assigns a fieldbus analog input block template for example purposes only. You can easily adapt this procedure to apply to another Fieldbus Library block.

- When assigning a user template to a non-Honeywell fieldbus device, warning messages may be generated because of ***manufacturer specific parameters*** associated with the device. Be sure that default values for manufacturer specific parameters are applicable for your given application.

Configuring fieldbus block template

Step	Action
1	In **Project** mode, click **Library** tab, click + sign for **FIELDBUS** icon, and click **AI** block icon.
2	Click **Templates**>**Make Template** to call up **AI Block** configuration form..
3	On the **Process** tab, fill in configuration data in the appropriate boxes to configure the block for operation in your application using the *Control Building Guide* and *online Help* for reference.
4	Repeat Step 3 for the other tabs: **Alarm**, **Maintenance**, **Tune**, **Ranges**, **Identification**, **Block Pins**, **Configuration Parameters**, **Monitoring Parameters**, **Block Preferences**, and **Template Defining**, as applicable.
5	When configuration entries are completed, click the **OK** button to close the **AI Block** configuration form and add the configured AI block as a template under its assigned name in the USER Library on the **Library** tab.
6	In the **Project** tab, select and double-click an instantiated Control Module (CM) that is to contain the **AI** block template to open the CM in the control drawing of Control Builder. .
7	Drag and drop the **AI** block icon from the USER Library to the open Control Module in the control drawing to add the **AI** block to the CM.
8	Right-click the **AI** block in the open CM and select **Function Block Assign** from the list to open **Function Block Assignment Dialog** that lists all compatible device blocks that have been assigned to a FIM4 Link. .
9	Click the check box to select the desired device. Click the **Assign** button. • If **Block Assignment Validation** dialog appears with warning message(s), Go to the next Step. • Otherwise, Go to Step 11.
10	Note the warning message and click the Continue button.
11	The **AI** block is assigned to the selected device as noted on the face of the block.

Step	Action
12	Click the close button to close the CM. Click the **Yes** button to save the changes to the CM.
13	This completes the procedure. Go to the next section.

Series C FIM Configuration Form Reference

FIM4 Main Tab Parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Main** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Tag Name	NAME	**Project** Only	System assigned or user configured unique name. Consisting of up to 16 characters and at least one character must be a letter (A-Z).
Item Name	Item Name	**Project** Only	A non-unique name by which an entity is known within the context of the enterprise model.
Application Image Version	IMAGEVER	No	Identifies current version of firmware installed in module.
Controller Command	FIMCOMMAND	**Monitoring** Only	Allows user to initiate commands to a Series C FIM.
Device Index	DEVICEIDX	**Project** Only – Matches Hardware Setting	Unique FTE device index of the Series C FIM. Set on Series C FIM IOTA switches.
Ethernet IP Address	IPADDRESS	No	IP address of the Series C FIM, derived as the Embedded FTE Base Ethernet IP Address plus configured Device Index.
Temperature High Alarm (deg C)	OVERTEMPTHL D	Yes	Set threshold value for modules Temperature High alarm.
CPU Free Low Alarm (%)	CPULOLM	Yes	Set CPU Free Low Capacity alarm limit, in

Plain Text	Parameter Name	User Configurable	Notes
			percent.
CPU Free Low Low Alarm (%)	CPULOLOLM	No	Set CPU Free Low Low Capacity alarm limit, in percent.
Controller State	FIMSTATE	No	Shows Series C FIM's current state.
Redundancy Role	RDNROLESTATE	No	Shows Series C FIM's current redundancy role.
Synchronization State	RDNSYNCSTATE	No	Shows Series C FIM's current synchronization state.
Module is redundant	MODISREDUN	**Project** Only	Module is part of redundant pair.
Secondary Tag Name	SECMODNAME	**Project** Only	System assigned name based on tag name with sec suffix for secondary block.
Suppress H1 Power Alarm	NOPWRDIAG	Yes	Suppresses H1 Link Power Diagnostic alarms. This option should be selected if integrated power alarming functionality is not to be used.
Enable Safe Handling of New Devices	SAFEHANDLING ENABLED	Yes	Enables safe handling of new uncommissioned field devices. Devices previously loaded are not safe handled when rejoining the link. See *Using Optional Safe Handling of New Devices* for more information
Device Drop Off Alarm Delay	DEVDROPOFFD ELAY	Yes	Sets the amount of time, in seconds, a device must be offnet before a device drop off alarm occurs.

FIM4 Redundancy Tab Parameters

The following table summarizes the parameter data you can monitor on the **Redundancy** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Redundancy Status			
Auto Synchronization State	RDNAUTOSYNC	No	Shows current auto synchronization state
Redundancy Compatibility	RDNCMPT	No	Shows redundant partner compatibility
Inhibit Sync Reason	RDNINHIBITSYNC	No	Shows reason that redundancy synchronization is inhibited
Initial Sync. Progress (%)	RDNSYNCPROG	No	Shows current synchronization progress in percent
Last Synchronization Time	SYNCTIMEBEG	No	Shows system time when initial synchronization was completed.
Last Loss of Sync. Time	SYNCTIMEEND	No	Shows sytem time when last synchronization was lost.
Redundancy Controlability	RDNCTLABILITY	No	
Redundancy Statistics			
Redundancy Traffic (bytes/sec)	RDNXFERAVG	No	
Max Redun. Traffic (bytes/sec)	RDNXFERMAX	No	
Redundancy Delay (%)	RDNDELAYAVG	No	
Max Redundancy Delay (%)	RDNDELAYMAX	No	

Plain Text	Parameter Name	User Configurable	Notes
Max Initial Sync Time (sec)	RDNINSTIMEMAX	No	
Max Switchover Time (msec)	RDNSOTIMEMAX	No	
Disable Synchronization	DSBLSYNCCMD	No	Button to initiate command in **Monitoring** mode
Alternate Sync. Over FTE	ALTSYNCCMD	No	Button to initiate command in **Monitoring** mode.
Become Primary	BECMPRICMD	No	Button to initiate command in **Monitoring** mode.
Enable Synchronization	ENBLSYNCCMD	No	Button to initiate command in **Monitoring** mode.
Initiate Switchover	SWITCHCMD	No	Button to initiate command in **Monitoring** mode.
Redundancy History			
Time	RDNHISTTIME	No	Show time of related state.
State	RDNHISTSTATE	No	List the last 16 redundancy related activities.
Reason	RDNHISTREASON	No	Identify reason for redundant history state
Redundancy Link Failed	RDNLINKFAILED	No	Status indicator to signal link failure

FIM4 System Time Tab Parameters

The following table summarizes the parameter data you can monitor on the **System Time** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
System Time			
Current System Time	CURTIME	No	Shows current system time.
Current System Time Source	TIMESOURCE	No	Shows source of system time. Default is Simple Network Time Protocol.
SNTP Status			
SNTP Status	SNTPSTAT	No	Shows status for time source.
SNTP Skew Limit (msec)	SNTPSKEWTHLD	No	
SNTP Skew Limit Exceeded	NUMSNTPSKEWEX	No	Shows number times SNTP knew threshold was exceeded.
Max. SNTP Skew (msec)	MAXSNTPSKEW	No	Shows maximum skew recorded.
System Time Synchronization Status			
Time Synchronization Status	TIMESYNCSTAT	No	
Time of Last Time Sync	TIMELASTSYNC	No	Shows time Controller Time Synced with time source
Skew at Last Time Sync (msec)	TIMELASTSKEW	No	Shows time skew from last synced time.

FIM4 Statistics Tab Parameters

The following table summarizes the parameter data you can monitor on the **Statistics** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Reset All Statistics	STATSRESET	No	Button to initiate reset in **Monitoring** mode.
CPU Utilization			
CPU Free (%)	CPUFREEAVG	No	
Minimum CPU Free (%)	CPUFREEMIN	No	
Notification Statistics			
Notification Rate	TNUMNTFRQUAVG	No	
Maximum Notifications Rate	TNUMNTFRQUMAX	No	
Hardware Temperature			
Current Temperature (deg C)	CTEMP	No	
Maximum Temperature (deg C)	CMAXTEMP	No	
Maximum Temperature (deg C)	CMINTEMP	No	

FIM4 Peer Connections Tab Parameters

The following table summarizes the parameter data you can monitor on the **Peer Connections** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Peer Initiator Connections			
Initiating to ACEs	TNUMACEINCON	No	
Initiating to C300s	TNUMC3INCON	No	
Initiating to C200s	TNUMCPMINCON	No	
Initiating to FIM4s	TNUMSCFIMINCON	No	
Initiating to SIM-C200s	TNUMSCEINCON	No	
Initiating to LIOMs	TNUMLIOMINCON	No	
Peer Responding Connections			
Responding to ACEs	TNUMACEOUTCON	No	
Responding to C300s	TNUMC3OUTCON	No	
Responding to C200s	TNUMCPMOUTCON	No	
Responding to FIM4s	TNUMSCFMOUTCON	No	
Responding to SIM-C200s	TNUMSCEOUTCON	No	
Responding to LIOMs	TNUMLIOMOUTCON	No	

FIM4 Hardware Information Tab Parameters

The following table summarizes the parameter data you can monitor on the **Hardware Information** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Firmware Version			
Boot Image Version	BOOTIMAGEVER	No	
Application Image Version	IMAGEVER_D	No	
Hardware Factory Information			
Module Type	MODTYPE	No	
Serial Number	SERIALNUM	No	
Programmable Logic Set Version	FWREVISION	No	
Hardware Major Revision	HWREVMAJ	No	
Hardware Minor Revision	HWREVMIN	No	
Network Interface Address Information			
FTE Interface A MAC Address	MACADDRA	No	
FTE Interface B MAC Address	MACADDRB	No	
Redun. Interface MAC Address	MACADDRR	No	

FIM4 FTE Tab Parameters

The following table summarizes the parameter data you can monitor on the **FTE** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
FTE Mart Statistics (Provides statistics related to the MAC Address Resolution Table (MART). This deals with on-line media access control (MAC) address mapping. Two separate tables are maintained – One for FTE nodes and one for non-FTE nodes. The media access control address entry information is obtained by extracting the source media access control address from received diagnostic messages.)			
Address Count	FTEMARTADDRCOUNT	No	Number of IP addresses contained in FTE MART.
Max Depth	FTEMARTMAXDEPTH	No	Maximum depth that the FTE MART has reached (largest number of entries in table).
Average Depth	FTEMARTAVGDEPTH	No	Average depth of FTE MART (average number of entries in table).
Address Collisions	FTEMARTCOLLCOUNT	No	Number of collisions that have occurred when hashing the FTE MART.
Current FTE Traffic			
LAN_A Tx Rate (kBit/sec)	LANATXRATE	No	Indicates communication transmission rate in kilobits per second (kbps) for port A (Yellow Tree Port) on the FTE Bridge.
LAN_B Tx Rate (kBit/sec)	LANBTXRATE	No	Indicates communication transmission rate in kilobits per second (kbps) for port B (Green Tree Port) on the FTE Bridge.
LAN_A Rx Rate (kBit/sec)	LANARXRATE	No	Indicates communication receive rate in kilobits per second (kbps) for port A (Yellow Tree Port) on the

Plain Text	Parameter Name	User Configurable	Notes
			FTE Bridge.
LAN_B Rx Rate (kBit/sec)	LANBRXRATE	No	Indicates communication receive rate in kilobits per second (kbps) for port B (Green Tree Port) on the FTE Bridge.
FTM Statistics			
Number of FTE nodes	NUMFTENODES	No	Current number of FTE nodes within FTE community.
Max number of FTE nodes	MAXFTENODES	No	Maximum number of FTE nodes that have been detected within FTE community.
Max node ID	MAXNODEID	No	Highest Device Index supported within FTE community.
IP checksum errors	BADIPCSUM	No	Number of FTE IP messages which were received but determined as having bad IP checksums.
UDP checksum errors	BADUDPCSUM	No	Number of FTE UDP messages which were received but determined as having bad UDP checksums.
Non-FTE Mart Statistics			
Address Count	NONFTEMARTADDRCOUNT	No	Number of IP addresses contained in non-FTE MART.
Max Depth	NONFTEMARTMAXDEPTH	No	Maximum depth that non-FTE MART has reached (largest number of entries in table.
Average Depth	NONFTEMARTAVGDEP	No	Average depth of non-FTE

Plain Text	Parameter Name	User Configurable	Notes
	TH		MART (average number of entries in table).
Address Collisions	NONFTEMARTCOLLCOUNT	No	Number of collisions that have occurred when hashing non-FTE MART.
Peak FTE Traffic			
LAN_A Tx Rate Max (kBit/sec)	LANATXRATEMAX	No	Indicates maximum communication transmission rate in kilobits per second (kbps) for port A (Yellow Tree Port) on the FTE Bridge.
LAN_B Tx Rate Max (kBit/sec)	LANBTXRATEMAX	No	Indicates maximum communication transmission rate in kilobits per second (kbps) for port B (Green Tree Port) on the FTE Bridge.
LAN_A Rx Rate Max (kBit/sec)	LANARXRATEMAX	No	Indicates maximum communication receive rate in kilobits per second (kbps) for port A (Yellow Tree Port) on the FTE Bridge.
LAN_B Rx Rate Max (kBit/sec)	LANBRXRATEMAX	No	Indicates maximum communication receive rate in kilobits per second (kbps) for port B (Green Tree Port) on the FTE Bridge.
Number of FTE Nodes	NUMFTENODES	No	
Max Number of FTE Nodes	MAXFTENODES	No	
Max Device Index	MAXNODEID	No	
IP Checksum Errors	BADIPCSUM	No	
UDP Checksum	BADUDPCSUM	No	

Plain Text	Parameter Name	User Configurable	Notes
Errors			
LAN_A (Yellow) failed	LANAFAILED	No	Status indicator for port A (Yellow Tree Port) on the FTE Bridge. If this LED is lit, it is an indication that communications have failed on Port A.
LAN_B (Green) failed	LANBFAILED	No	Status indicator for port B (Green Tree Port) on the FTE Bridge. If this LED is illuminated it is an indication that communications have failed on Port B.
InterLAN comm. failed	INTERLANFAILED	No	Status indicator for Inter-LAN communications – indicates that inter-LAN communications have failed.
Crossover cable failed	XOVERFAILED	No	Status indicator for Crossover cable – indicates that the Crossover cable has failed.

FIM4 UDP/TCP Tab Parameters

The following table summarizes the parameter data you can monitor on the **FTE** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
UDP Statistics (Provides unreliable connectionless packet delivery service between clients.)			
Datagrams Delivered	UDPINDGRAMS	No	Total number of User Datagram Protocol (UDP) datagrams delivered to destination protocol ports.
Datagrams for	UDPNOPORTS	No	Total number of received

Plain Text	Parameter Name	User Configurable	Notes
Unknown Ports			UDP datagrams for which there was no application at the destination port.
Datagrams Dropped for errors	UDPINERRORS	No	Number of received UDP datagrams that could not be delivered.
Datagrams Sent to Applications	UDPOUTDGRAMS	No	Total number of UDP datagrams sent from this entity.
TCP Statistics (Provides reliable stream delivery service between clients.)			
Active Opens	TCPACTIVEOPEN	No	Number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.
Passive Opens	TCPPASSIVEOPENS	No	Number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
Failed Connection Attempts	TCPATTEMPTFAILS	No	Number of times TCP connections have made a direct transition to CLOSED state from either SYN-SENT state or SYN-RCVD state, plus number of times TCP connections have made a direct transition to LISTEN state from SYN-RCVD state.
Connections Resets	TCPESTABRESETS	No	Number of times TCP connections have made a direct transition to CLOSED state from either ESTABLISHED state or CLOSE-WAIT state.
Current Connections	TCPCURRESTAB	No	Number of TCP connections for which

Plain Text	Parameter Name	User Configurable	Notes
			current state is either ESTABLISHED or CLOSE-WAIT.
Segments Received	TCPINSEGS	No	Total number of segments received, including those received in error.
Segments Sent	TCPOUTSEGS	No	Total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
Segments Retransmitted	TCPRETRANSSEGS	No	Total number of segments retransmitted – that is, number of TCP segments transmitted containing one or more previously transmitted octets.
Segments Discarded For Errors	TCPINERRS	No	Total number of segments received in error (for example., bad TCP checksums).
Reset Segments Sent	TCPOUTRESETS	No	Number of TCP segments sent containing the RST flag.
Local UDP Listeners	UDPLISTENERS	No	Shows path information for all open ports on the FTE Bridge module.
Current TCP Connections	TCPCONNTABLE	No	Shows path information for all currently connected clients of the FTE Bridge module.

FIM4 IP/ICMP Tab Parameters

The following table summarizes the parameter data you can monitor on the **IP/ICMP** tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
IP Statistics (Provides packet delivery services between nodes.)			
Datagrams Received from Below.	IPINRECEIVES	No	Total number of input datagrams received from connected nodes, including those received in error.
Datagrams Format Errors Drops	IPINHDRERRORS	No	Number of input datagrams discarded due to errors in their Internet Protocol (IP) headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.
Datagrams Misdelivery Drops	IPINADDRERRORS	No	Number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity.
Unknown Protocol Datagrams	IPINUNKNOWNPORTS	No	Number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
Datagrams Discarded for Resrcs	IPINDISCARDS	No	Number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded; for example, for lack of buffer space.

Plain Text	Parameter Name	User Configurable	Notes
Datagrams Delivered Above	IPINDELIVERS	No	Total number of input datagrams successfully delivered to IP user-protocols, including Internet Control Message Protocol (ICMP).
Datagrams Sent Out	IPOUTREQUESTS	No	Total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.
Out Datagrams Discarded	IPOUTDISCARDS	No	Number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded; for example, for lack of buffer space.
Datagrams Drops for No Routes	IPOUTNOROUTES	No	Number of IP datagrams discarded because no route could be found to transmit them to their destination.
Fragments Needing Reassembly	IPREASSEMREQS	No	Number of IP fragments received which needed to be reassembled at this entity.
Fragments Reassembled	IPREASSEMOKS	No	Number of IP datagrams successfully reassembled.
Fragments Reassembly Fails	IPREASSMFAILS	No	Number of failures detected by the IP reassembly algorithm, for whatever reason: timed out, errors, etc.
Datagrams Fragmented	IPFRAGOKS	No	Number of IP datagrams that have been successfully fragmented at this entity.

Plain Text	Parameter Name	User Configurable	Notes
Datagrams Fragmentation Fails	IPFRAGFAILS	No	Number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be. For example, because their do not Fragment flag was set.
Fragments Created	IPFRAGCREATES	No	Number of IP datagram fragments that have been generated as a result of fragmentation at this entity.
Routing Entries Discarded	IPROUTINGDISCARDS	No	Number of routing entries which were chosen to be discarded even though they are valid.
ICMP Statistics (Controls transmission of error and control messages between hosts and gateways.)			
Messages Received	ICMPINMSGS	No	Total number of ICMP messages which the entity received.
Messages with Format Errors	ICMPINERRORS	No	Number of ICMP messages which entity received but determined as having ICMP-specific errors such as bad ICMP checksums and bad length.
Dest. Unreachable Msgs Recvd	ICMPINDESTUNREAC HS	No	Number of ICMP Destination Unreachable messages received.
Echo Messages Recvd	ICMPINECHOS	No	Number of ICMP Echo (request) messages received.
Echo Reply Messages Recvd	ICMPINECHOREPS	No	Number of ICMP Echo Reply messages received.
Messages Sent	ICMPOUTMSGS	No	Total number of ICMP messages which this entity

Plain Text	Parameter Name	User Configurable	Notes
			attempted to send.
Out Error Messages	ICMPOUTERRORS	No	Number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers.
Dest. Unreachable Msgs Sent	ICMPOUTDESTUNREACHS	No	Number of ICMP Destination Unreachable messages sent.
Echo Messages Sent	ICMPOUTECHOS	No	Number of ICMP Echo (request) messages sent.
Echo Reply Messages Sent	ICMPOUTECHOREPS	No	Number of ICMP Echo Reply messages sent.

FIM4 QVCS Tab Parameters

The **QVCS** tab is common to all configuration forms for tagged blocks in Control Builder. If you have a Qualification and Version Control System (QVCS) license, this tab shows current QVCS information for the selected FIM4 block. Please refer to the online help and the Qulification and Version Control Sytem User's Guide for more information about the data on this tab.

FIM4 Server History Tab Parameters

The **Server History** tab is common to all configuration forms for tagged blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected FIM4 block.

ATTENTION

The configuration settings you make for Server Load Options on the **System Preferences** dialog determines whether or not the data entered on the **Server History** tab is loaded to the Experion Server. See the *Control Building Guide*for information about setting system preferences.

Plain Text	Parameter Name	User Configurable	Notes
Access Levels			
Control Level	SCANCTRLLVL	Yes	Indicates Server control level to be associated with this function.
History Configuration			
Number of History Parameters	HIST.NUMPARAMS	Yes	Defines number of history parameters to be included in History Configuration table.
Parameter		Yes	Valid parameter name for a parameter associated with the given point that is to be collected and stored as historical data at predetermined intervals.
Description		No	Provides a brief description of the entered parameter.
FAST		Yes	Select the Fast type of history collection.
STD		Yes	Select the Standard type of history collection
EXTD		Yes	Select the Extended type of history collection
Gating Parameter		Yes	Optional gating parameter to define conditions under which data for this parameter should be collected.
Gate State		Yes	Defines gate state for configured gating parameter.
Create New or Edit Existing Server Scripts (Button)		N/A	Launch the Server scripting configuration utility.

FIM4 Server Displays Tab Parameters

The **Server Displays** tab is common to all configuration forms for tagged blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected FIM4 block.

ATTENTION

The configuration settings you make for Server Load Options on the **System Preferences** dialog determines whether or not the data entered on the **Server Displays** tab is loaded to the Experion Server. See the *Control Building Guide*for information about setting system preferences.

Plain Text	Parameter Name	User Configurable	Notes
Point Detail Display	SCANPNTDTL	Yes	By default, a Display template is already entered into Point Detail Display box (for example, sysDtlFTEB.dsp). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into detail display that is supplied as a template.
Group Detail Display	SCANGRPDTL	Yes	By default, a Display template is already entered into the Group Detail Display box (for example, sysGrpFTEB.dsp). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into detail display that is supplied as a template
Associated Display	SCANASSOCDSP	Yes	Name of the Server display to be associated with this

Plain Text	Parameter Name	User Configurable	Notes
			function block.
Trends			
Number of Trends	TREND.NUMPARAMS	Yes	Defines the number of trend parameters to be included in the Trends Configuration table.
Trend #		Yes	Defines Trend number to be associated with this trend parameter
Pen		Yes	Defines color of pen that will be used to trace assigned parameter on Station Trend display.
Trend Parameter		Yes	Valid parameter name for a parameter associated with given point that is configured for history collection.
Description		No	Provides a brief description of the entered parameter.
Groups			
Number of Groups	GROUP.NUMPARAMS	Yes	Defines the number of group parameters to be included in Groups Configuration table.
Group #		Yes	Defines Group number to be associated with this group parameter.
Pos #		Yes	Defines number of position configured parameter will occupy in the Station Group display.
Group Parameter		Yes	Valid parameter name for a parameter associated with the given point that is

Plain Text	Parameter Name	User Configurable	Notes
			configured in the system.
Description		No	Provides a brief description of the entered parameter.

FIM4 Control Confirmation Tab Parameters

The **Control Confirmation** tab is common to all configuration forms for tagged blocks in Control Builder. If you have an optional Electronic Signature license, you can configure electronic signature information for the tagged block through this tab on the block's configuration form in Control Builder. Please refer to the online help and the *Server and Client Configuration Guide* for information about the data on this tab.

The Electronic Signature function aligns with the identical Electronic Signatures function that is initiated through Quick Builder and Station for Server points. When this block is loaded to a controller, its control cofirmation configuration (electronic signatures) is also loaded to the Server. This means you can view the control confirmation configuation for this tagged object in Station and also make changes to it. If you make changes through Station, you must initiate an **Upload** or **Upload with Contents** function through the **Controller** menu in Control Builder for the object in the **Monitoring** tab to synchronize changes in the Engineering Repository Database (ERDB).

FIM4 Identification Tab Parameters

The **Identification** tab is common to all configuration forms for tagged blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected FIM4 block.

Plain Text	Parameter Name	User Configurable	Notes
Name	NAME	Yes	Unique block name consisting of up to 16 characters to identify the block. At least one character in the name must be a letter (A-Z).
Description	DESC	Yes	Descriptive text appears on detail and group displays to uniquely describe this particular function block

Plain Text	Parameter Name	User Configurable	Notes
Block Comment 1	BLCKCOMMENT1	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 2	BLCKCOMMENT2	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 3	BLCKCOMMENT3	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 4	BLCKCOMMENT4	Yes	Comment to be associated with this block consisting of up to 40 characters.
Library		No	Identifies Control Builder Library that is source of template.
System Template		No	Identifies System Template that is source for this block.
Base Template		No	Identifies Base Template that is used for this block.
Created By	CREATEDBY	No	Identifies user who created block, if operator security is implemented. Otherwise, may just show Default login.
Date Created	DATECREATED	No	Shows date and time template was created. If this block is in Version Control System, shows date and time initial version of template was created.
Last Modified By	MODIFIEDBY	No	Identifies user who made last modifications to block, if operator security is implemented. Otherwise, may just show default login. If this block is in Version Control System,

Plain Text	Parameter Name	User Configurable	Notes
			modifications apply to last version of block.
Date Last Modified	VERSIONDATE	No	Shows date and time last modification was made to block's configuration. If this block is in Version Control System, modification date and time applies to last version of block.

Block Instantiation Support

In addition to the permanent or fixed function blocks that manufacturer's define for their FOUNDATION Fieldbus (FF) devices, they can define optional instantiable function blocks. This gives users the ability to instantiate (create) additional function blocks within the Function Block Application Process for the given fieldbus device. The Fieldbus Foundation refers to this function as Block Instantiation and they include it in their Host Interoperability Support Test (HIST).

REFERENCE - INTERNAL

Please refer to the *Fieldbus Technology Overview* section for more information about Standard Function Blocks in general.

This section includes the following topics on information related to block instantiation support in Control Builder.

Control Builder supports block instantiation

ATTENTION

The Type function in Control Builder only supports Block Instantiation in fieldbus devices with single Capability Levels. It does not support fieldbus devices with multiple Capability Levels.

The **Type** function in Control Builder supports the functions associated with instantiable blocks in fieldbus devices with single capability levels. It lets you build block types for devices that support block instantiation that include both the fixed function blocks and the instantiable function blocks. This means you can view the instantiable blocks as part of the device's block hierarchy in the **Library** tab of Control Builder. The icon for an instantiable function block has a slash across it and a letter **I_** tag prefix as illustrated below, so you can easily distinguish it from a fixed function block

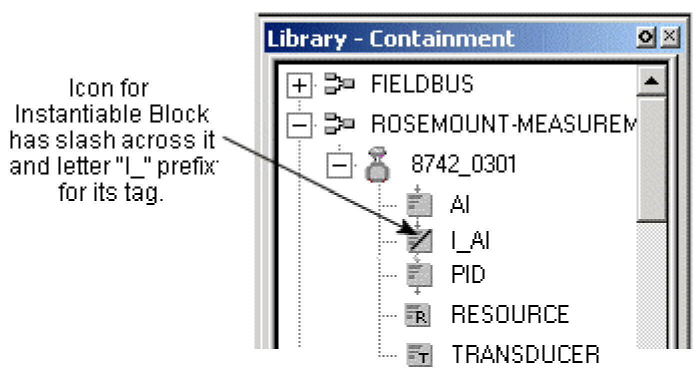


Figure 11 Typical Icon for Instantiable Block in Fieldbus Device.

ATTENTION

A manufacturer may choose to define Transducer blocks as **instantiable** instead of fixed. In this case, be sure you **instantiate** a Transducer block for each physical sensor that is to be used in the device. A Transducer block usually corresponds to a physical sensor in the vendor's device.

Instantiable block implementation considerations

The following topics review considerations that are unique to implementing control strategies that include fieldbus devices with optional instantiated blocks. Please use these considerations to supplement the appropriate procedures in other parts of this document that are tailored to fieldbus devices with only fixed function blocks including resource and transducer blocks.

Adding instantiable block to device in Project tab.

Prerequisites

- You have logged onto Control Builder with sufficient security privileges to make changes in control strategies.
- You have created a device type that includes instantiable blocks.

Considerations

Devices created in **Project** tab automatically include only their **fixed** function blocks, transducer blocks, and resource blocks. If the device block type includes **instantiable** blocks, you **must** add the instantiable block from the block type in the **Library** tab to the device icon in the **Project** tab to instantiate (create) it in the device in **Project**.

Default name of instantiated block may be appended with a letter. For example, a block with default name I_AI becomes I_AIA.

Device does not have to be connected to the link to instantiate a block.

An error message is generated, if device cannot support an additional block.

See *Adding a Fieldbus Device to Project* for reference.

Step	Action
1	Open corresponding manufacturer's device hierarchy in the **Library** tab.
2	Drag the instantiable block onto the applicable device icon in the **Project** tab.
3	Key in desired name for the block or accept the default in the open **Name New Function Block(s)** dialog.
4	Click the **Finish** button to close the dialog and add the block to the device hierarchy in **Project,** as shown in the following illustration.
5	This completes the procedure. Go to the next section.

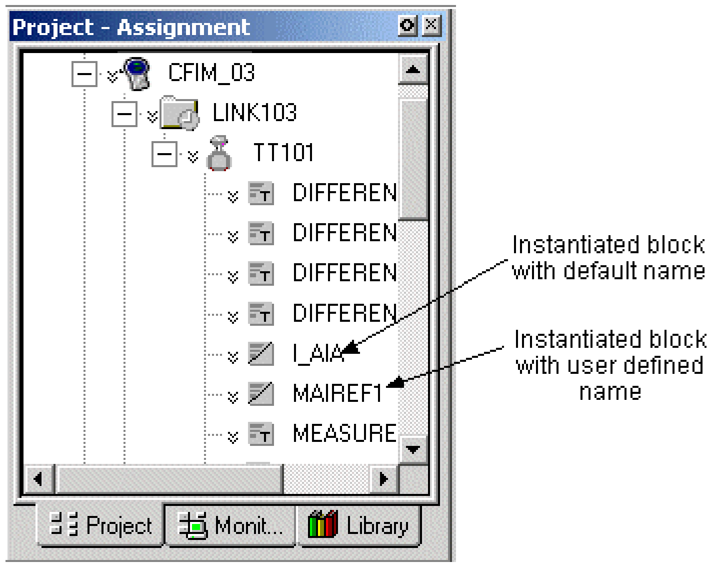


Figure 12 Instantiated block added to device in Project tab.

Checking instantiated block configuration in Project tab.

Prerequisites

You have completed the previous procedure.

Considerations

- The **Instantiation** tab only appears if the device supports Block Instantiation.
- See *Checking device configuration* for reference.

Step	Action
1	In the **Project** tab, right-click applicable device instantiable block and select **Block Properties** from list to open the form.
2	Click **Block Instantiation** tab.
3	Check **Block Template Type** column for list of instantiated blocks in device.
4	Check **Block Tag** column to view assigned tag to see if block is used in a

Step	Action
	Control Module or not as shown in the following illustration.
5	Click the **OK** button to close the form.
6	This completes the procedure. Go to the next section.

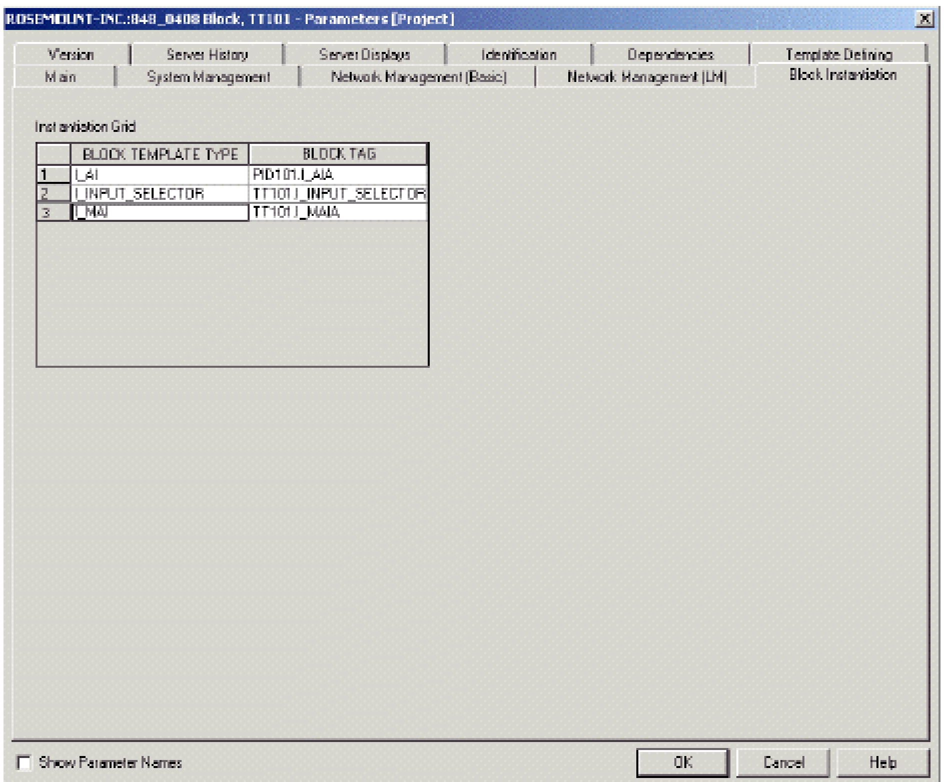


Figure 13 Block Instantiation tab for instantiable block included with device in Project tab.

Making optional instantiated block resource usage check

Prerequisites

You have completed the previous procedures.

Considerations

- Some vendors provide resource information as an option. You can use this information to determine if the configured block contents will fit within the available resources of the device.
- A resource represents a finite entity used by a device such as Central Processing Unit (CPU) or Memory.

Step	Action
1	In the **Project** tab, right-click applicable device instantiable block and select **Block Properties** from list to open the form.
2	Click **Block Instantiation** tab.
3	Check **Resource #** column for the amount of resource used by a particular block. The # (1, 2, …) columns can vary depending on vendor and function block type.
4	In **Resource Statistics** grid, check **Limit** and **Used** columns to view resource allocations against assigned limits, as shown in the following illustration.
5	Click the **OK** button to close the form
6	This completes the procedure. Go to the next section.

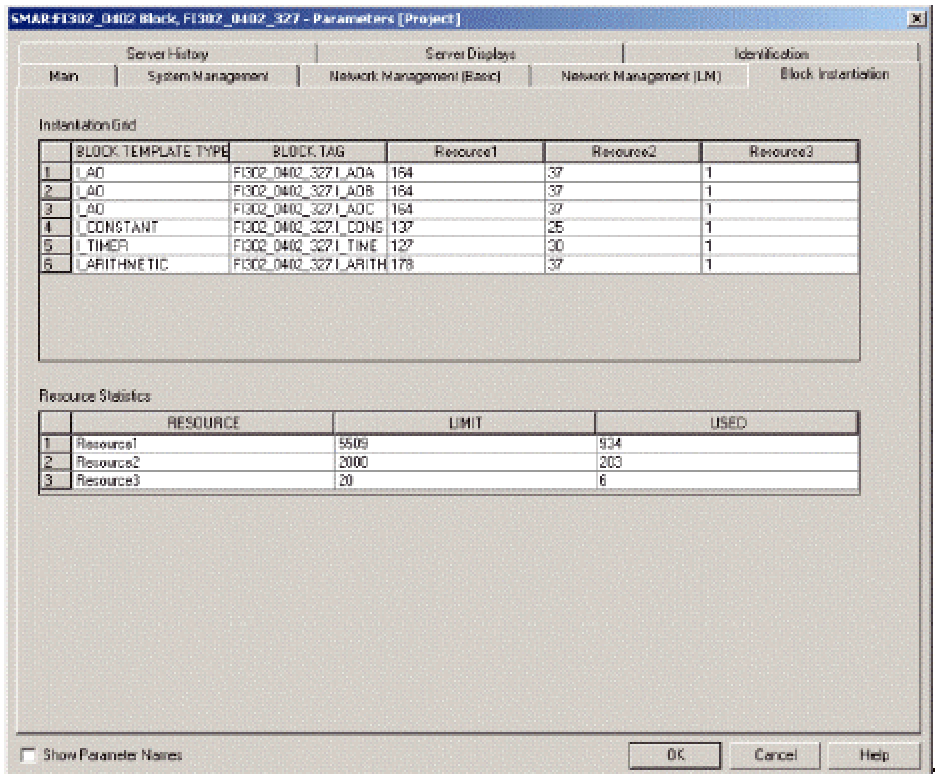


Figure 14 Block Instantiation tab with optional resource information.

Adding instantiated block to CM.

Prerequisites

You have completed the previous procedures.

Considerations

- You add instantiated blocks to a Control Module the same way you do fixed blocks. They also are configured in the same way and have the same graphical appearance as shown in the following figure.

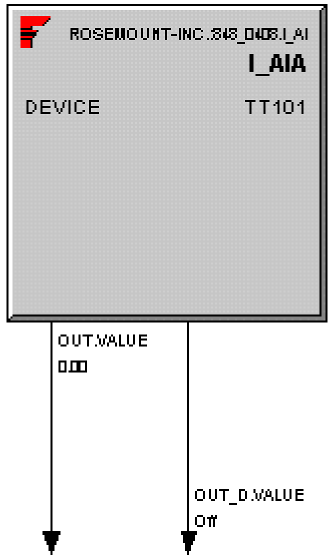


Figure 15 Typical graphical representation of instantiated block in Control Builder.

- See the *Control Building Guide* for reference.

Step	Action
1	In **Project** tab, double-click the icon for the Control Module that is to contain a fieldbus block in its strategy to open the CM in the control chart drawing area..
2	In **Project** tab, click the plus sign for the FIM4, the FFLINK, and the contained device to expand the directories down to the device block level.
3	Drag the applicable instantiated block to the open CM.
4	Double-click the instantiated block to open its configuration form and configure the block, as desired.
5	Click the close button on the CM and click the **Yes** button to save the changes and close the CM .
6	This completes the procedure. Go to the next section.

Loading Project device to physical device

CAUTION

A physical device may go offline while its instantiable blocks are being instantiated through a device load.

Prerequisites

You have completed the previous procedures.

Considerations

- **Block instantiation in a physical device only occurs during a load.**
- Be sure you load a device from **Project** before you load a Control Module that contains instantiable blocks. If you load a Control Module that contains instantiable block(s) before the associated device in **Project** is loaded and its blocks instantiated, the Control Module load will fail.
- The existing load order, FIM4/FFLink load, and device match functions also apply for devices with instantiable blocks.
- Device load from **Project** synchronizes the contents of the **Project** device with the physical device and instantiates applicable instantiable blocks in the physical device.
- You must load the Control Module(s) containing fixed and/or instantiated blocks associated with the device for the fixed and instantiated blocks to appear in the device hierarchy in **Monitoring** tab.
- After loading a device from **Project** or **Monitoring** tab, the physical device function block contents matches that of the corresponding **Project** or **Monitoring** device. This means necessary blocks will be instantiated and unnecessary Blocks will be removed or de-instantiated.

Step	Action
1	See *Load order guidelines* for reference.
2	See Loading a Series C FIM and its Links for reference.
3	See *Matching Uncommissioned Device to Project Device or Vice Versa* for reference.
4	Right-click device icon and select **Load** from list.

Step	Action
5	Review information on **Load** dialog for load details and any warning messages identifying pre-load considerations.
6	Click **OK** button to initiate the load and monitor **Load Progress** dialog for load related activities.
7	Upon successful device load, open device icon in **Monitoring** tab to view block hierarchy. Only transducer and resource blocks appear in the device hierarchy after an initial load.
8	Right-click device icon and select **Upload** from the list. Click **Continue** to load device data into the Monitor database.
9	This completes the procedure. Go to the next section.

Deleting (de-instantiate) instantiated block from Project tab

Prerequisites

- You cannot delete an Instantiable block from **Project**, if it is loaded and viewable in the **Monitoring** tab. See the next section to delete an instantiated block for a physical device.
- You cannot delete an instantiated block that is contained in a Control Module unless you un-assign or delete the block from the CM first.

Step	Action
1	In **Project** tab, click the + plus sign for the FIM4, the FFLINK, and the contained device to expand the directories down to the device block level.
2	Right-click applicable instantiated block icon in the hierarchy and select **Delete** from the list.
3	Click the **Continue** button to initiate action and track progress in **Delete** dialog.
4	Upon successful deletion the instantiated block icon is removed form the device hierarchy.
5	This completes the procedure. Go to the next section.

Deleting (de-instantiate) instantiated block from physical device

Prerequisites

You cannot delete an instantiated block directly from a physical device.

Considerations

- Deleting a block from **Monitoring** tab, does not de-instantiate a block in a device. Only a load of a device will achieve this
- You must re-load **Project** device to re-synchronize contents with physical device after any control strategy changes or physical device maintenance that may alter contents in either device.

Step	Action
1	In the **Monitoring** tab, locate the CM that contains the instantiated block that is to be deleted and delete the CM.
2	In the **Project** tab, unassign or delete the instantiated block from the CM that was deleted in the **Monitoring** tab, and then delete the block from the device in **Project**, as noted in the previous procedure. Make any control strategy changes, as required.
3	Load device from **Project** to re-synchronize contents with physical device.
4	Load device from **Project** to re-synchronize contents with physical device.
5	Load reconfigured CM.
6	Right-click device icon and select **Upload** from the list. Click **Continue** to load device data into the **Monitoring** database.
7	This completes the procedure. Go to the next section.

Replacing failed device that includes instantiable blocks with like device

Prerequisites

- See *Replacing a failed device with a like device having the same model name and device revision* for general replacement procedure.

Considerations

- A device replacement can cause changes in a device's block layout, since the Object Dictionary (OD) index for an instantiated block can change.
- If connections to instantiable blocks of the **replaced** field device are through SCM expressions, such an SCM will not be automatically shown as a part of the Load List under the device in the **Load** Dialog. This means that the SCM will not be automatically loaded. In this case, you must inactivate the SCM and reload it as a separate procedure.
- If a Control Module (CM) or a Sequential Control Module (SCM) has connections to **contained** Fieldbus parameters, they will appear as a part of the Load List under the device in the **Load Dialog** and will be automatically loaded. See the **Load Dialog** example in the following illustration. However, special consideration must be given to inactive Sequential Control Modules.

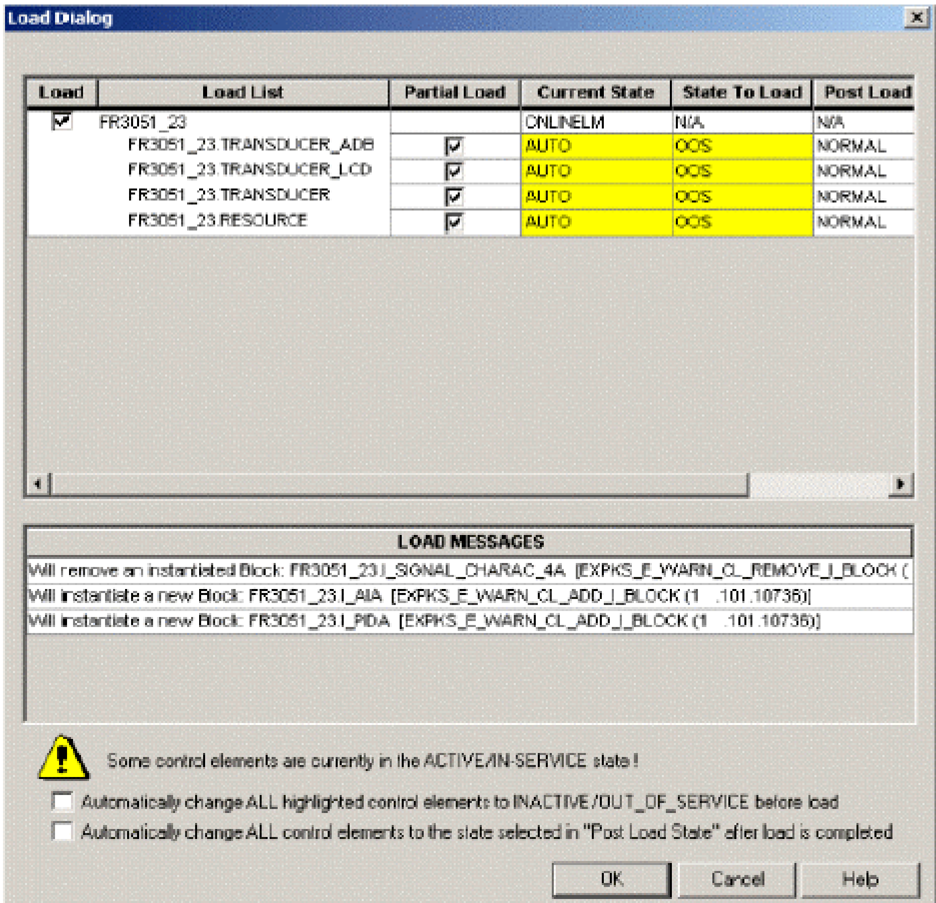


Figure 16 Typical Load Dialog for fieldbus device.

Loading FIM4 Components Online

The Experion system provides the ability to build control strategies offline, without being connected to the actual field components. The process of transferring the Control Strategy to the "live" working components in the field is called the load operation. This section includes the following topics on information and tasks associated with loading FIM4 components through Control Builder.

About load operations

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. The load operation has been expanded to include fieldbus components. It indirectly assures that the planned 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. The following illustration shows a simplified graphical representation of what happens during a load operation.

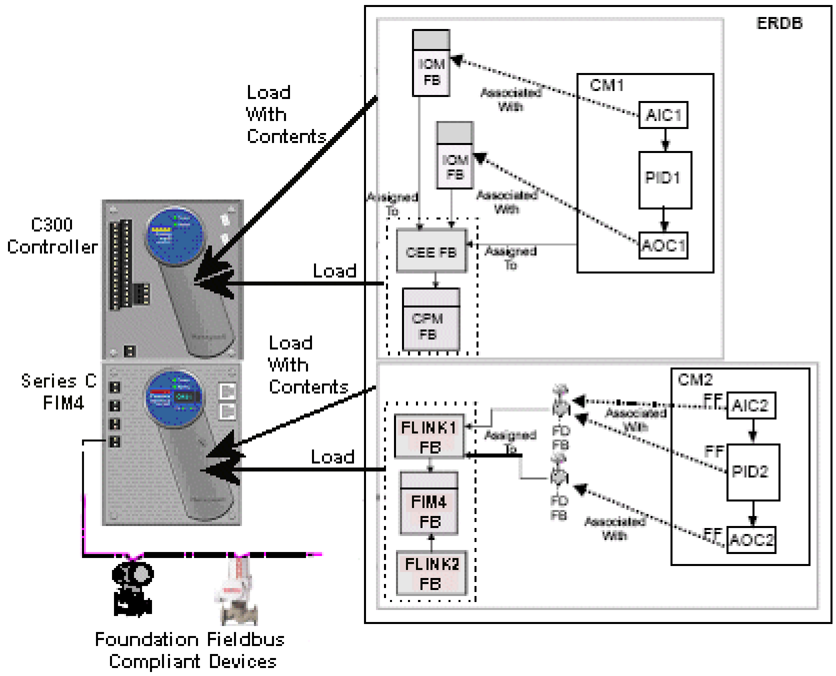


Figure 17 Overview of load operations used to initiate FIM4 components online.

About Load Dialog

The following figure shows a sample Load Dialog invoked for a load with contents operation for a Series C FIM. It provides a brief description of the dialog features for quick reference.

CAUTION

The load operation is still an offline function. The **Load Dialog** provides the ability to automatically inactivate a component during a load and then return the component to its active state. Do **not** use this automatic inactivate/activate function, if your process can**not** tolerate the load disruption and consequent delay in activation. In this case, you must manually toggle the component state through the **Monitoring** tab in Control Builder.

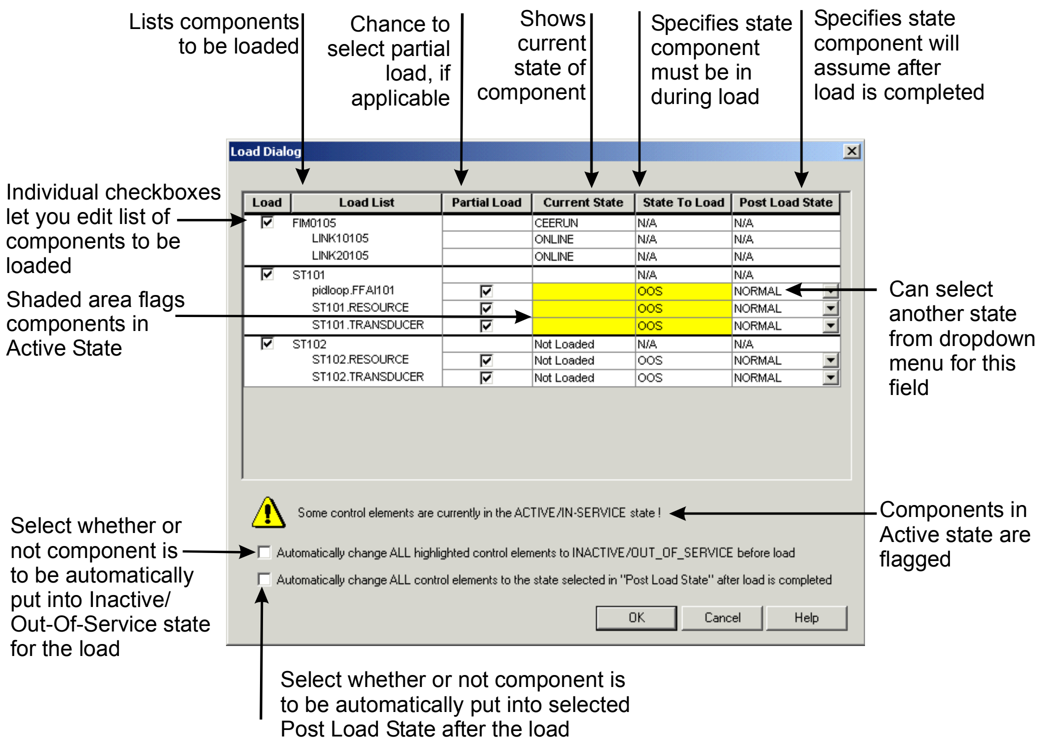
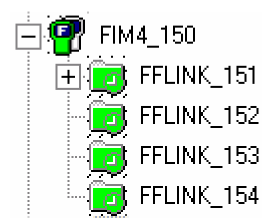
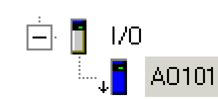
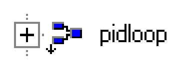


Figure 18 Load Dialog provides more load choices.

Load order guidelines

Load control strategy components from the **Project** tab in the following order to avoid possible load interaction generated error messages.

Order	Component	Typical Loaded Icon in Monitoring Tab
1	C300/CEEC300*	
2	FIM4/FFLINK	
3	Check for Uncommissioned fieldbus devices and initiate match function, as applicable	
4	Fieldbus Device	
5	IOM*	
6	CM or SCM*	

Order	Component	Typical Loaded Icon in Monitoring Tab

*Please refer to the Control Building Guide for more information about loading these components.

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 load problems.

If you are loading . . .	Then, consider this . . .
A FIM4.	Be sure ALIV does not appear in the LED display on the front of the FIM4. If it does, you must first load the FIM4's personality firmware using the Series C hardware firmware tool (Ctool) utility supplied with Experion Engineering Tools.
A CM that contains only fieldbus function blocks.	We recommend that CMs containing only fieldbus function blocks be assigned and loaded to the appropriate FIM4 LINK. They will only take up unnecessary memory and execution time in the CEE. .
A CM that contains both Experion and fieldbus function blocks.	Be sure CMs that contain Experion function blocks are assigned and loaded to a CEE. Control Builder enforces this.
A fieldbus device.	Be sure the device configuration in **Project** tab has the proper PD_TAG specified. While you can load a device to the FFLink 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.

About 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.

SMK State	System Management Services
Unitialized	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

In the **unitialized** 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.

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.

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 Series C 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.

If the Device ID is . . .	And, the Device State Is . . .	Then, the matching rule is . . .
Specified.	Unitialized	The device ID (DEV_ID) must match. If the desired physical device tag (PD_TAG) and node address (ADDR) are available, FIM4 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, FIM4 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, FIM4 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.

Identifying Load Interactions for Fieldbus Related Operations

The following table provides a matrix of load interactions for given fieldbus related operations.

ATTENTION

If you make on-line changes to parameter values in fieldbus devices, the changed values are not written to the **Monitoring** or **Project** copy of the control strategy. You must initiate an **Upload** function to write the new values to **Monitoring** followed by an **Update to Project** function to write the **Monitoring** values to the **Project** copy.

Operation	Build Link Schedules	Load Schedule into FIM4's Link	Load LM Schedules into LM-Capable Devices	Load Used Function Blocks	State Required for Operation
Load Device *or* Replace Device	No	No	This device only, if LM-capable	Yes	Blocks in device = OOS, Device = OOS, FFLink = Active
Delete Device from Loaded	No	No	No	No	Blocks in device = OOS, Device = OOS, FFLink=Active
Load FFLink with Contents	Yes	Yes	All LM-capable	Yes	Blocks in related CMs = OOS, Device = OOS, FFLink = N/A
Load FFLink (without contents)	No	Yes	No	No	FFLink = N/A

Operation	Build Link Schedules	Load Schedule into FIM4's Link	Load LM Schedules into LM-Capable Devices	Load Used Function Blocks	State Required for Operation
Load CM (from Project)	Yes	Yes	All LM-capable	Yes	Blocks in CM = OOS CM = Inactive, FFLink = Active, Device = N/A
Load CM (from Monitor)	No	No	No	Yes	Blocks in CM = OOS CM=Inactive, FFLink=Active, Device=N/A
Delete CM from Loaded	Yes	Yes	All LM-capable	No	Blocks in CM = OOS CM=Inactive, FFLink=Active, Device=N/A

Loading a Series C FIM and its Links

Prerequisites

All Series C FIM components are installed and capable of communicating with the system.

Considerations

- If **ALIV** appears in the Series C FIM's 4-character, alphanumeric display, you must load the Series C FIM's personality firmware before proceeding.
- The associated Series C FIM Links are included with a FIM4 load even if the selected action is **Load** instead of **Load with Contents**. We suggest just loading the FIM4 without all of its contents first to be sure communications paths are working.

Initiating FIM4 load

Step	Action
1	Click the FIM4 block icon in the **Project** tab. On the **Controller** menu, click **Load** to call up the Load dialog.
2	Be sure a check mark appears in the check box for the FIM4 listed in the **Load** Column? Click the **OK** button to initiate the load.
3	Track load status in the **Load Progress** dialog box that appears.
4	When load is completed, click the **Monitoring** tab to open it.
5	Confirm that the FIM4 icon now appears in the **Monitoring** tab. Click + plus sign for FIM4 icon to open the directory and expose the FFLINK icons.
6	This completes the procedure. Go to the next section *Matching uncommissioned device to project device or vice versa*.

Matching Uncommissioned Device to Project Device or Vice Versa

Prerequisites

- You have installed H1 link segments to provide proper power and conditioning for fieldbus devices.
- You have completed the previous procedure to load the FIM4 and its links.

Considerations

- It is best to match devices before you download the fieldbus device from **Project** to the FFLink.
- If you install a device with an address below 20 (hexadecimal 14) on the H1 network, the FIM4 will clear the device's address. The device returns to the link at a temporary address and may be reassigned as needed.
- If you install a **Link Master** class device on the H1 network, the FIM4 performs the following actions to protect the network from an unconfigured or incorrectly configured device disturbing or becoming the **Link Master** of the segment.:
  - **Clears** the link schedule,
  - **Clears** the primary link master flag variable,

- **Sets** the configured link settings,
- **Sets** the device class to basic, and
- **Restarts** the device.

- Refer to the following *Using Optional Safe Handling of New Devices* section for more information about adding an uncommisioned to a loaded Fieldbus link with the optional safe handling function enabled.
- If a connected fieldbus device is **not** being detected by the link, try increasing the Slot Time value through the **Network Management (LM)** tab on the **FFLINK Block** Parameters configuration form.

Initiating device matching

Step	Action
1	On the **Monitoring** tab,confirm that applicable FFLink icon is active (green). • If the FFLink is active (green icon), go to Step 4. • If the FFLink is inactive (blue icon), go to Step 2.
2	Click the applicable FFLink icon and click the  toggle state button to call up the **Change State** dialog.
3	Click the **Yes** button to initiate FFLink startup and wait for the FFLink to startup and become active (green icon).
4	On the **Project** tab, click the FIM4 block and the FFLink that has uncommissioned devices  connected to it.
5	On the **Field Devices** menu, click **Device Match** to call up **FFLink Parameters** form with **Device Matching Dialog** tab selected.
6	Check for block type matches between listed **Uncommissioned Devices** and **Project Devices** in the **Template** column.
7	Do you need to change the Tag and/or Address assignments for the listed devices in either the **Uncommissioned Device** or the **Project Device?** • If the answer is **Yes,** double-click the entry field you want to change to highlight it, key in the desired tag name or address assignment in decimal, and press the **Enter** key. If you are changing data in an Uncommissioned Device, it may take several minutes for the operation to complete.

Step	Action
	• If the answer is **NO**, go to the next Step.
8	For matching devices, click the check box for the Uncommissioned Device and then, click the check box for the matching Project Device to activate the Match from Uncommissioned Device to Project Device and Match from Project Device to Uncommissioned Device buttons.
9	• If the Uncommissioned Devices were previously configured through a Hand-Held Communicator or third-party configuration to contain correct tag and address data, click the **Match from Uncommissioned Device to Project Device** button to preserve the **Uncommissioned Device** data and change the **Project Device** data to match it. • If Uncommissioned device contains factory default data, click the **Match from Project Device to Uncommissioned Device** button to preserve the **Project Device** data and change the **Uncommissioned Device** data to match it.
10	If Warning prompt asks if you want to continue this operation, click the Continue button to Initiate Tag and Address change in the Uncommissioned Device to match the Project Device. Dialog appears to show progress of the change. This may take several minutes to complete. Otherwise, go to the next Step.
11	When operation is completed, confirm that Tag and Address assignments for the selected **Uncommissioned Device** and **Project Device** now match. The match operation was a success.
12	Click the  button to close the configuration form.
13	In the **Monitoring** tab, right-click the device icon and select **Upload** from the list, and click **Continue** to load device data into the Monitoring database. Then, right-click the device icon and select **Update to Project** from the list, and click **Continue** to synchronize the data in the Project database with the Monitoring database
14	Repeat Steps 5 to 13 for another device or repeat this procedure for FFLINK 2, FFLINK 3, and FFLINK 4, as required.
15	This completes the procedure. Go to the next section *Loading FFLink contents or fieldbus device* to download the **Project Device** to the **Uncommissioned Device**.

Using Optional Safe Handling of New Devices

About safe handling option

If you enable the safe handling option for the Series C FIM through its configuration form in Control Builder, the following actions are automatically initiated when you add a new device to a loaded Fieldbus link.

- Clears any existing VCRs configured in the device.
- Clears any existing link schedule.
- Clears the primary link master flag on the device.
- Sets the link settings in the device to the current link settings of the segment.
- If required, changes the device to a basic one and restarts the device.

Enabling and disabling safe handling

You enable and disable the safe handling option through the **Safe Handling of New Devices Enabled** check box on the **Main** tab of the FIM4 Block's configuration form in Control Builder. The default is selected or checked check box, which means safe handling is enabled.

Safe handling and previously commissioned devices

The safe handling option does not apply when removing and adding a previously commissioned device to the Fieldbus link. However, if you remove a commissioned device from one link and add it to another link, the device is treated as an uncommissioned device and the safe handling option applies. If this same device is later moved back to its original link, it will appear as an uncommissioned device and will require a reload.

Removing power during a safe handling operation

The safe handling operation can take up to 30 seconds to complete after adding a new device to a link. If you remove a device while the safe handling operation is in progress, it may have its configuration partially cleared.

Moving a commissioned device to another link and then back to its original link after only 1 to 20 seconds has expired, may leave the device in a half safe handled state. If the FBAP VCR was cleared, the FIM4 and Control Builder may not be able to talk to the device. If this occurs, Fieldbus blocks will appear gray on the **Monitoring** tab. In this case, you will need to delete and reload the device or remove the device from the link and put it back on another link for at least 30 seconds to allow the safe handling operation to complete.

To avoid partial safe device handling, do **not** move previously commissioned devices to other links in the system too quickly.

Loading FFLink Contents or Fieldbus Device

Prerequisites

You have completed the previous procedures to load the FIM4 and match Project and Uncommissioned devices.

Considerations

- It is a good idea to load the fieldbus device before you load the Control Modules containing its associated function blocks to the CEE or FIM4. If you load the Control Module first, you may get an error message indicating that the *parent* block is not available.
- If you just want to load a given fieldbus device, click the device icon and click **Controller**>**Load**. We suggest using FFLINK/**Load With Contents** so you can load more than one device at a time.
- A **Partial Load** selection means that the *contained* or internal block parameters will not be loaded to the device, which reduces the overall load time. It is generally not necessary to load these types of parameters, since no connections are made to them.
- If errors are detected during the load, they will be listed and you will be 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.
- The device's resource and transducer blocks are dedicated to the device and loaded with the device, and can be viewed in the device directory in the **Monitoring** tab. The device's other function blocks will not appear in its directory until the Control Module containing them is loaded to the CEE or FIM4, as applicable.

Initiating device load

CAUTION

If the load generates errors identifying **sub schedule** problems, you must reconfigure the Link Master capable device to become a **Basic** one through Control Builder, and restart the device to initiate the change. Otherwise, if the Series C FIM fails, the potential for a "no schedule" condition exists. Since the schedule is cleared from the Link Master capable device upon the detection of sub schedule load errors, it can**not** function as the backup LAS without a schedule.

Please see the *Checking/Changing Fieldbus Device Functional Class* section for details on reconfiguring a device as a **Basic** type.

Step	Action
1	In the **Project** tab, click the desired FFLINK icon. In the **Controller** menu, click **Load With Contents** to open the **Load Dialog**.
2	Confirm that checks appear in the **FFLINK** and **Device** check boxes listed in the **Load?** column. If you do not want to load a given device, just click it to remove the check from its check box.
3	Accept the default **Partial Load** check box selections for the Resource and Transducer blocks for a device.
4	To change the **Post Load State** for selected components, click the appropriate **Post Load State** row and select another state from the list.
5	Click the check boxes to inactivate the fieldbus devices automatically for the load and to return fieldbus devices automatically to post load state after the load.
6	Click the **OK** button to Initiate the *Load* and open **Load Dialog** to show load progress.
7	Once the load is complete, click the **Monitoring** tab to open it.
8	Confirm that blocks now appear in given device hierarchy.
9	Repeat Steps 1 to 8 to load another FFLink or device, as required.
10	This completes the procedure. Go to the next section.

Fieldbus Device Commissioning

This section outlines a typical process to follow to commission a fieldbus loop in an Experion system. This section does not provide detailed procedures for each stage in the process. Please reference appropriate sections in this document or applicable vendor documentation for detailed information, if required.

The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Getting Started

Read this first

The Series C Fieldbus Interface Module (FIM4) sets many parameters within a fieldbus device during its initial setup and commissioning in an Experion system. While you can change some parameters, such as calibration ones, when the device is being operated on a workbench or at a location apart from the system, you must **not** change certain other parameters. The following table outlines the functions you must consider before commissioning a device.

For This Function. . .	Consider This . . .
Link Settings	These parameters can affect whether or not a device can communicate with other devices on the link, or whether it communicates efficiently and can recover properly from errors. They should never be changed, especially by another host device. If the Experion system notices that these parameters have been changed after a device has been off the network, the Series C FIM takes the following action: a) Changes the parameters back. b) De-commission the device (assuming other parameters have been changed also). c) Change a Link-Master-capable device to a Basic device. To recover, the (now uncommissioned) device must be changed back to a Link-Master device, if desired, and re-loaded from the **Project** or **Monitoring** tab in Control Builder.
Link Schedule	The Link Schedule controls the publication schedule, if no Series C FIM is available on the link. If the device was moved from another link or to another host and was loaded with a different schedule, the link schedule is not the current schedule version number for this link. In this case, the Series C FIM will take the following action. d) De-commission the device. e) Change a Link Master capable device to a Basic device. To recover, the (now uncommissioned) device must be changed back to a Link-Master device, if desired, and re-loaded from the **Project** or **Monitoring** tab in Control Builder.
Virtual Communication Relationships (VCRs)	The Virtual Communication Relationships (VCRs) are communication connections that are established for a given link. They should **not** be re-configured on another link or by another Host. Otherwise, they will very likely not be configured correctly and could interfere with communications when the device is restored to the desired link. When the Series C FIM first encounters a new uncommissioned device, it will clear its VCRs.

For This Function. . .	Consider This . . .
Link-Master Capable	A Link-Master-Capable device can take over control of a link. When Series C FIM first encounters a new uncommissioned device, it checks to see if it is configured as a Link-Master capable device and re-configures it as a Basic device. (When the device is commissioned, it can be changed back to Link-Master capable, since it will be loaded with the correct backup link schedule.)
Binding of Fieldbus Devices to the Experion database	The three most important **identification parameters** for a fieldbus device are its: • Device_ID, • PD_Tag, and • Link Address. The Device_ID is required by Fieldbus Foundation specifications to be device-unique and unchangeable. But, if either the Physical Device Tag (PD_Tag) or the Link Address is changed while the device is off the network, the device is de-commissioned and will have to be re-commissioned when it is returned to the link. If a device is removed from the link and another device is placed on the link at the same address (second device will have different Device_ID and possibly different PD_Tag), the original device will appear de-commissioned upon return to the link. To recover, the (now uncommissioned) device must be re-commissioned or matched again to its device object and then re-loaded.
Number of Fieldbus Device parameters to receive periodic (sustained) stores	Sustained stores are intended for parameters such as FF_VAL, RCAS_IN, TRK_IN_D, and TRK_VAL. Sustained stores should not be used on static parameters or parameters stored in non-volatile memory on a device, such as SP. Performing sustained stores to static or non-volatile parameters may lead to a shortened lifetime for the EPROM or flash memory in a device. Each Fieldbus device will behave differently, so it is best to consult the device manual for information on sustained store limitations.

Initial checks and operations

Stage	Function	Completed
1	Check and test all physical wire connections and verify correct voltage on H1 segment. *References:* Selecting and calculating fieldbus wiring requirements	
2	Load FIM4 block only *References:* Adding FIM4 Block to Project Loading a Series C FIM and its Links	
3	Was FIM4 download successful? **NO** > Go to Stage **4** **YES** > Go to Stage **7**	
4	Is IOTA and Series C FIM powered? **NO** > Go to Stage **5** **YES** > Go to Stage **6**	
5	Be sure required voltage is present at the power supply and it is turned On. Replace power supply, if required. **Return to Stage 2.**	
6	Verify network connections. Replace Series C FIM, if required. **Return to Stage 2.**	
7	Measure and establish H1 segment baseline with Fluke Scopemeter and Relcom FTB-3 or equivalent equipment.	
	Go to Stage 8 in the next section *Connecting Devices*.	

Connecting Devices

Stage	Function	Completed
8	Connect the next device. *References:* *Fieldbus Network Wiring and Installation references*	
9	Does device appear on link? **NO** > Go to Stage **10** **YES** > Go to Stage **23** in the next section *Checking Devices*.	
10	Be sure cable is connected to proper segment terminals on IOTA. *References:* Wiring CC/CU-TFB401 or CC/CU-TFB411 IOTA	
11	Connect Relcom Tester and/or Fluke Scopemeter to Link's cable.	
12	Is voltage over 9 volts? **NO** > Go to Stage **13** **YES** > Go to Stage **14**	
13	Test the power supply, check that terminators are Installed properly and check for short circuits. **Return to Stage 9**	
14	Does proper number of devices appear on tester? **NO** > Go to Stage **16** **YES** > Go to Stage **15**	
15	Check that proper FFLink is downloaded and that device is connected to proper link. *References:* Loading a Series C FIM and its Links **Return to Stage 9**	

Stage	Function	Completed
16	Are noise average and peak under 75 millivolts? **NO** > Go to Stage **17** **YES** > Go to Stage **18**	
17	Check the integrity of all cable connections, cables and verify the design of the link. **Return to Stage 9**	
18	Is lowest signal level over 150 millivolts? **NO** > Go to Stage **19** **YES** > Go to Stage **20**	
19	Insufficient power for device. Check power supply and link design. **Return to Stage 9**	
20	Is there an addressing issue? **NO** > Go to Stage **21** **YES** > Go to Stage **22**	
21	Device should be present. Replace the device. **Return to Stage 9**	
22	Check if address is in unpolled range, check for too many devices with temporary address. *References:* Tags, Addresses, and Live List **Return to Stage 9**	

Checking Device

Stage	Function	Completed
23	Does firmware - DD Revision match device revision? **NO** > Go to Stage 24 **Yes** > Go to Stage 25	
24	Upgrade firmware or import new DD file *References:* Upgrading Series C FIM Firmware Making Fieldbus Device Type from Vendor DD	
25	Match device. *References:* Matching Uncommissioned Device to Project Device or Vice Versa	
26	Load (Partial Load) the new device from **Project** tab. References: Using Optional Safe Handling of New Devices	
27	Measure and compare H1 segment baseline with Fluke Scopemeter and Relcom FTB-3 or equivalent equipment.	
28	Are measurements within specifications? **NO** > Go to Stage **29** **YES** > Go to Stage **30**	
29	Identify source of noise or communication problems and repair or replace equipment as required. **Return to Stage 27**	

Stage	Function	Completed
30	Do you need to add more devices on this segment? **NO** > Go to Stage **31** in the next section *Checking Control Strategy*. **YES** > Return to Stage **8** in previous section *Connecting Devices.*	

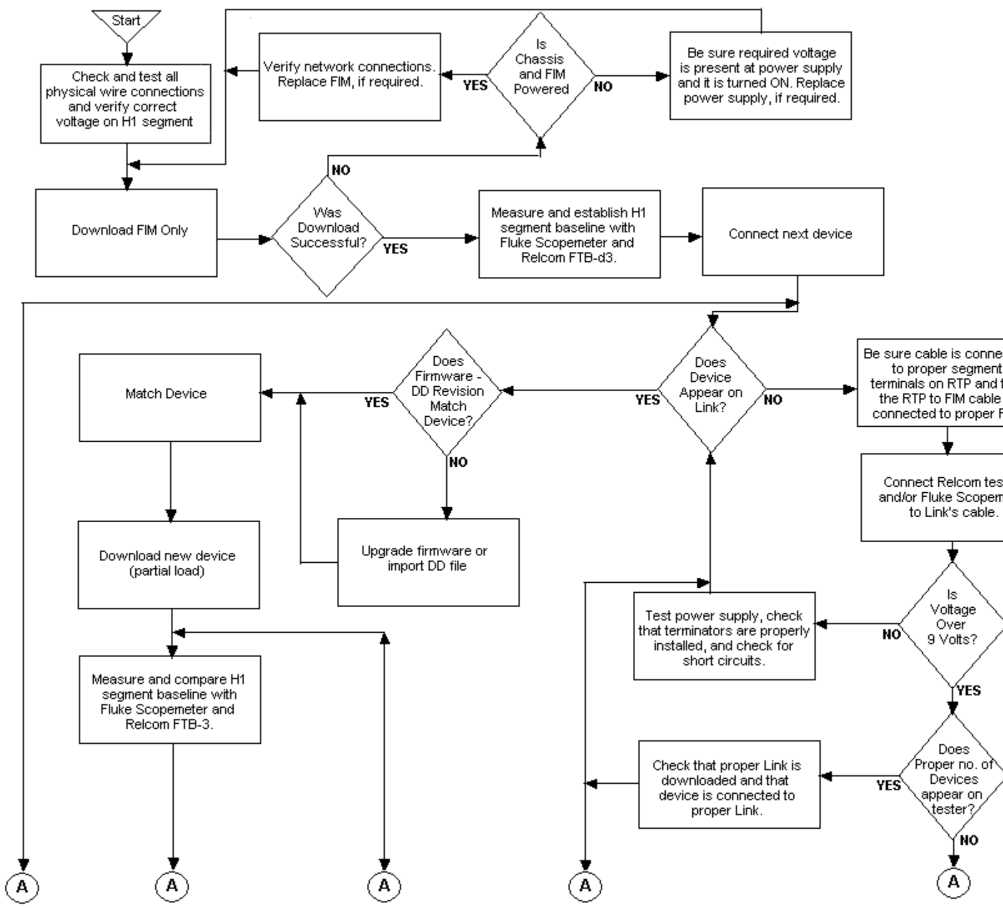
Checking Control Strategy

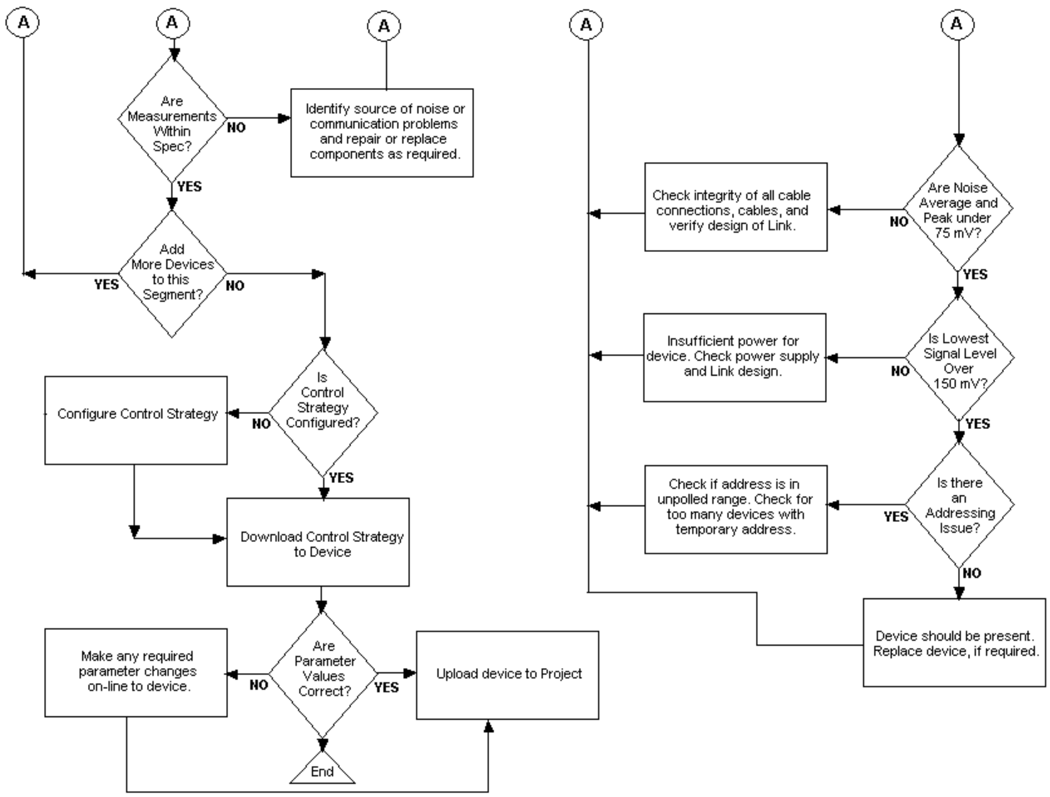
Stage	Function	Completed
31	Is control strategy configured? **NO** > Go to Stage **32** **Yes** > Go to Stage **33**	
32	Configure control strategy. *References:* Series C FIM Configuration	
33	Load control strategy to device. *References*: About load operations	
34	Are parameters correct? **NO** > Go to Stage **35** **YES** > Go to Stage **36**	
35	Make required parameter changes on-line to device.	
36	Upload device to project. *References:* Identifying Load Interactions for Fieldbus Related Operations	

Stage	Function	Completed
	This completes the commissioning process. The commissioning process is summarized in the following flowchart.	

Flowchart Summary

The commissioning stages outlined above are summarized in the following flowchart.




A
Are Measurements Within Spec?
NO
Identify source of noise or communication problems and repair or replace components as required.
YES
Add More Devices to this Segment?
YES
NO
Is Control Strategy Configured?
NO
Configure Control Strategy
YES
Download Control Strategy to Device
Are Parameter Values Correct?
NO
Make any required parameter changes on-line to device.
YES
Upload device to Project
End
Are Noise Average and Peak under 75 mV?
NO
Check integrity of all cable connections, cables, and verify design of Link.
YES
Is Lowest Signal Level Over 150 mV?
NO
Insufficient power for device. Check power supply and Link design.
YES
Is there an Addressing Issue?
YES
Check if address is in unpolled range. Check for too many devices with temporary address.
NO
Device should be present. Replace device, if required.

Series C FIM Redundancy Functionality

This section provides information about using a redundant Series C FIM pair. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Redundant Series C FIM overview

If you have a Series C FIM (FIM4) redundant Input/Output Termination Assembly (IOTA), you can implement redundant Series C FIM operation through a redundant Series C FIM pair. The following figure shows a typical hardware configuration for a redundant Series C FIM pair that includes a C300 Controller. You do not need the C300 Controller to support Series C FIM redundancy.

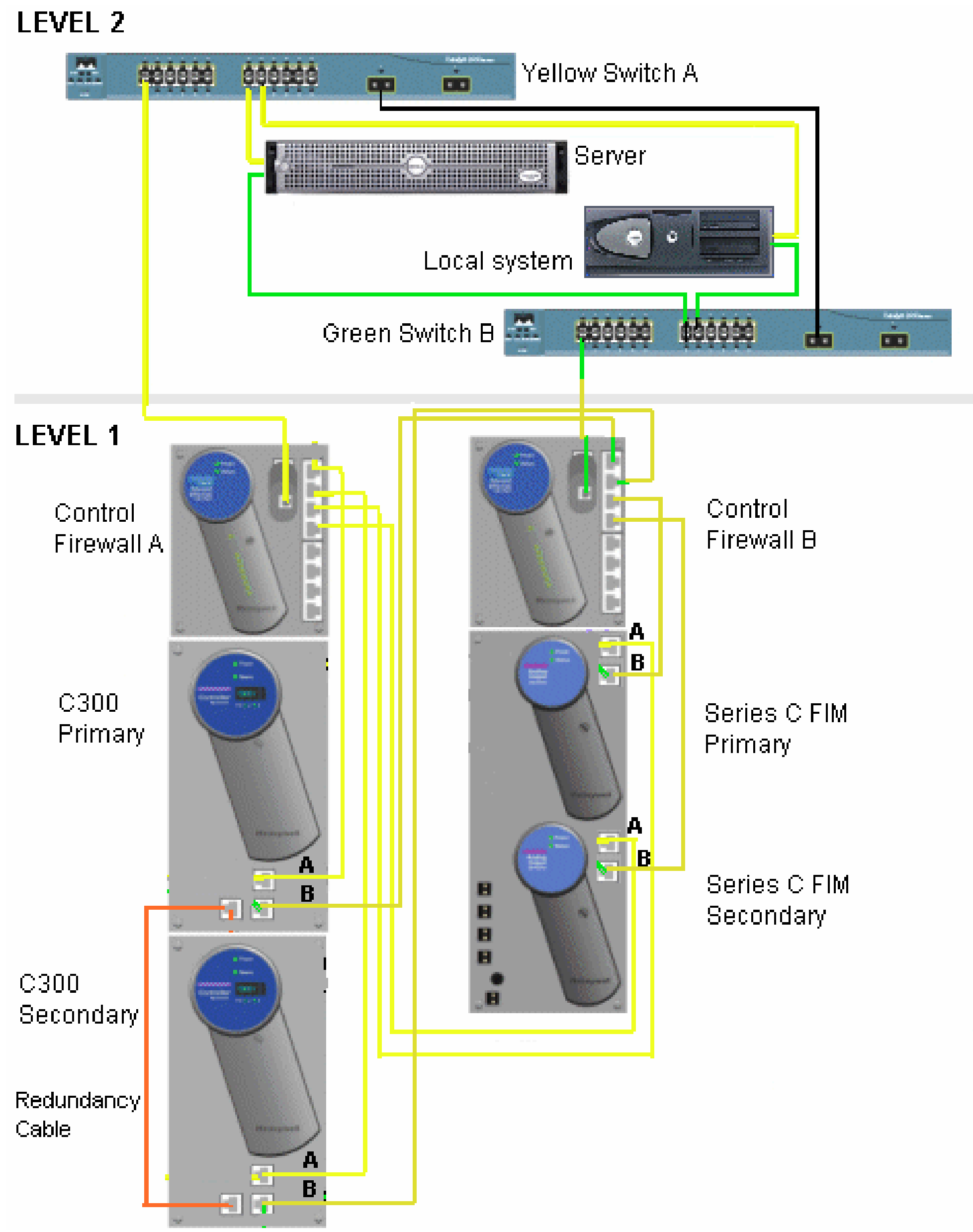


Figure 19 Redundant Series C FIM Pair With Redundant C300 Controller.

This is considered a dual redundant system, which is characterized by the following two main redundancy states.

- Primary — Refers to the Series C FIM executing the assigned control functions.
- Secondary (or Backup) — Refers to the Series C FIM in some state of readiness to assume the responsibilities of the Primary.

Switchover and Secondary readiness

A switchover describes the process where a Secondary Series C FIM assumes the Primary state, and the Primary Series C FIM assumes the appropriate Secondary state of readiness, depending upon what triggered the switchover. A switchover can be triggered immediately upon the detection of a fault in the Primary or upon the receipt of an operator command.

The ability of a Secondary Series C FIM to take over the assigned control functions of the Primary depends upon which one of the following readiness states reflects its current state.

If Secondary Series C FIM State is . . .	Then, the Secondary Series C FIM . . .
Not synchronized	Cannot assume the Primary state. This is a state of non-readiness.
Synchronizing	Cannot assume the Primary State. In this state, the Secondary Series C FIM is copying database information from the Primary.
Synchronized	Can assume the Primary state upon switchover. In this state, the database in the Secondary is aligned with the database in the Primary. The Secondary closely tracks database changes to maintain its synchronization with the database of the Primary. Otherwise, the Secondary will revert to a Not Synchronized state.

Failure conditions and switchover

The following table identifies failure conditions that result in a switchover and those that do not.

ATTENTION

When any failure that results in a switchover occurs in a Secondary, the Secondary Series C FIM loses synchronization.

In addition to the failure conditions, this event is reported as diagnostic notification:

- Loss of view of redundant partner on H1 network (applies to Series C FIM link).
- Loss of private path connection from Primary to Secondary (lonely event).

Failure Conditions That Result in a Switchover	Failure Conditions That Do Not Result in a Switchover
Power to Primary Series C FIM fails.	One or all H1 cables fail.
Both FTE links to Primary Series C FIM are lost.	One or all H1 network conditioners fail.
Primary Series C FIM fails.	The 24-Volt power supply for the H1 link fails.

Fieldbus network switchover considerations

The Series C Fieldbus Interface Module (FIM4) is a Link Master device running the link schedule, performing network maintenance activities, publishing control data and subscribing to control data, listening to alerts, and so on. All of these activities must be transferred from one module to another during switchover. The redundant Series C FIM operation accounts for the following key considerations associated with initiating a H1 network switchover.

For this Switchover Consideration . . .	Series C FIM Redundancy Operation . . .
Bumpless Switchover	Assures that fieldbus devices do not: • Initiate a fault state, • Remote shed, or • Mode shed, during switchover. The new Primary Series C FIM does not publish any data on the network until all switchover actions are completed and it has good data from the Control Data Access (CDA) server.
Device Index number	Does not transition the Device Index for the Primary Series C FIM to the Secondary Series C FIM. The Device Index stays with the physical Series C FIM.
MAC Addresses	Does not transition the MAC addresses for the FTE links on the Primary Series C FIM to the Secondary Series C FIM. The MAC addresses stay with the physical Series C FIM.

For this Switchover Consideration . . .	Series C FIM Redundancy Operation . . .
IP Address	Does swap **IP addresses** for Primary and Secondary Series C FIMs during switchover as follows. • During initial synchronization, Primary and Secondary Series C FIMs exchange their **IP addresses** over the private redundancy link. This is for verification purpose only, since the odd-even convention allows both modules to determine their partner's address based on their own address. • During switchover or swap, both Primary and Secondary Series C FIMs replace their current **IP address** with their partner's **IP address**. This puts redundant pair into transient state in which peer modules are temporarily unable to communicate with either one of them, since their address tables are no longer valid due to the **IP address** swap. After **IP address** swap, both Primary and Secondary Series C FIMs construct Address Resolution Protocol (ARP) request packets with updated IP-to-Ethernet address mapping and broadcast them on the network. Upon reception of ARP packets, peer nodes update their address tables.

For this Switchover Consideration . . .	Series C FIM Redundancy Operation . . .

ATTENTION

The swapping of IP addresses is transparent to the higher-level protocols such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

- TCP packets sent to either Primary or Secondary Series C FIM during switchover will be re-tried. Once retry limit is exceeded, connection breaks and gets re-established by application based on updated address information.
- UDP packets sent to either Primary or Secondary Series C FIM during switchover and before client address table is updated will be lost.

Switchover versus fieldbus network activities

The following table summarizes how given fieldbus network activities behave during a switchover.

ATTENTION

Depending upon the link communication load, number of devices, and type of parameter, fresh viewed data from field devices may be delayed for up to 20 seconds during a Series C FIM switchover. This does not apply to control data.

If Network Activity Is . . .	Then, Behavior During Switchover Is . . .
Link Maintenance	Since Secondary Series C FIM link is a backup Link Active Scheduler (LAS), it resumes network management activities as soon as it detects that the old Primary Series C FIM is no longer there. Both Primary and Secondary C FIM links have the same knowledge of current time. When new Primary takes over as acting Link Master, it also becomes acting time publisher.

If Network Activity Is . . .	Then, Behavior During Switchover Is . . .
System Management Operations	Any System Management operation that is in progress is interrupted. For example, the new Primary does not know anything about pending change tag or address operation. As a result, device whose address or tag is being changed will timeout and abort the operation.
Link Active Scheduler (LAS)	The Primary LAS runs the link schedule by telling devices to publish their data at specific times. The Secondary Series C FIM link is a backup LAS with a valid copy of the link schedule. During switchover, it skips the remainder of the current macrocycle and starts running its schedule from time zero of next macrocycle. Unlike backup LAS devices, the Secondary FIM has full LAS schedule with Series C FIM publications.
Alert Queued, User Triggered, Unidirectional Multicast (QUU) VCR	Both Primary and Secondary Series C FIM links subscribe to alerts from devices and receive them in parallel. Only the Primary Series C FIM link reports received alerts through the CDA server to the operator interface. After switchover, the new Primary Series C FIM starts and the old Primary Series C FIM stops submitting alerts to the CDA server. The new primary Series C FIM regenerates its alarms. This action covers whatever alarms might have been acknowledged just prior to switchover but did not get submitted to Server as a result of event throttling.
Subscriber Buffered, Network scheduled, Unidirectional broadcast (BNU) VCR	Both Primary and Secondary links subscribe to the same data published by a given device. Secondary has its VCR open and receives publications in parallel with the Primary. No specific action needs to be taken during switchover or swap.

If Network Activity Is . . .	Then, Behavior During Switchover Is . . .
Publisher Buffered, Network scheduled, Unidirectional broadcast (BNU) VCR	Only acting Primary Series C FIM link can publish data to fieldbus devices. Both Primary and Secondary can be configured with the same publication endpoint connection, but only the Primary link has this connection open and active at a time. During switchover, the old Primary closes its connection and consequently stops publishing. The old Secondary, that is becoming new Primary, opens its connection and consequently starts publishing.
Client Queued, User triggered, Bi-directional peer-to-peer (QUB) VCR	All client-server connections are broken and re-established. This way new Primary builds fresh connection context. QUB connections are point-to-point and therefore can only be opened from the primary. Series C FIM maintains one Management Information Base (MIB) connection to a device, one Function Block Application Process (FBAP) connection and possibly one control connection.

Switchover events

Switchover results in several Experion system events and alarms. All of them are system information or diagnostics type. No process alarms are generated due to switchover.

Typical events generated during switchover include:

- Connection failure to secondary module alarm
- Switchover event
- Not synchronized alarm

Each of these events appears for Series C FIMs where switchover occurred.

Series C FIM Operation

This section includes the following topics on information and tasks associated with operating Series C FIM, FFLinks, and Fieldbus Devices included in control strategies created in Control Builder.

Series C FIM Startup

The following table summarizes the stages the Series C FIM goes through after power is applied to its IOTA during startup. The Series C FIM repeats these stages every time power is cycled Off/On.

Stage	Description
1	Power Light Emitting Diode (LED) Lights
2	Executes firmware installed in FLASH memory. The initial firmware boot code is installed at the factory.
3	The Status LED briefly displays all of its colors: RED, GREEN, AMBER. Then, remains solid RED until the power-on self test (POST) completes.
4	Initiates a series of horizontal and vertical bars, as well as several brightness levels for evaluating the condition of the 4-character, alphanumeric display. At the end, the Link Status LEDs are blinked in turn.
5	Executes power-on self test to verify that all subsystems are working as intended. Cycles test number codes in the 4-character display to show progress of the test. (Initialization will halt on the first test that finds an invalid or faulty piece of hardware. The Series C FIM will display the test number code associated with the detected device failure until it is reset.)
6	Upon successful completion of the power-on self test, displays information about the application image currently stored through its 4-character display. The status LED should now reflect the true state of its associated hardware.
7	The 4-character display shows -bp- ((BOOTP) until it obtains its IP Address. The display then shows -ts- (Time Server) until time has been obtained from the NTP Server. Once its IP Address is known, it negotiates its redundancy role, either Primary or Secondary, with its partner module, if present. Verifies that the correct Device Index is displayed (`#nnn`) and that no addressing errors are detected. The primary module will have the odd IP address while the secondary module will have the odd + 1 IP address. The odd IP address will follow the primary module during a switchover. The Device Index does not change during a switchover or any other role change. Once a device has obtained its IP Address and NTP Server IP Address(es), it retains them until its Device Index is changed or firmware is reloaded. (The Series C FIM will determine if any other module is using the same IP

Stage	Description
	address. If it does find another module with the same Device Index, it will not join the network but instead transitions to *no address* state waiting for new address assignment. If it finds another node with the same IP Address, it will not join the network but instead transitions to *dup address* state; it must be reset to recover.)
8	The 4-character display cycles this information: Device Index setting, Link status, followed by Redundancy status. See the *Display and LED Descriptions* section for more information about the display and LED indications.

Initiating Series C FIM Shutdown

Use the following procedure to initiate a shutdown command to the Series C FIM, which results in the Series C FIM rebooting to its RDY state or boot firmware.

ATTENTION

Shutting down the Series C FIM, interrupts the transfer of data to the Experion system. Be sure your system can tolerate the lost of live data, while the Series C FIM is in its RDY state.

Step	Action
1	On the **Monitoring** tab, double-click the FIM4 icon to open the **FIM4 Block** configuration form.
2	On the **Main** tab, click **Controller Command** box and select **Shutdown** from the list.
3	Click the **Yes** button to confirm the action.
4	Wait for the FIM4 to reboot to its RDY state.
5	This completes the procedure. Go to the next section.

Series C FIM/IOTA Display and LED Descriptions

The following illustration and table identify and describe the display and LED indications associated with the 4-character alphanumeric display on the Series C FIM and LEDs on the Series C FIM and IOTA.

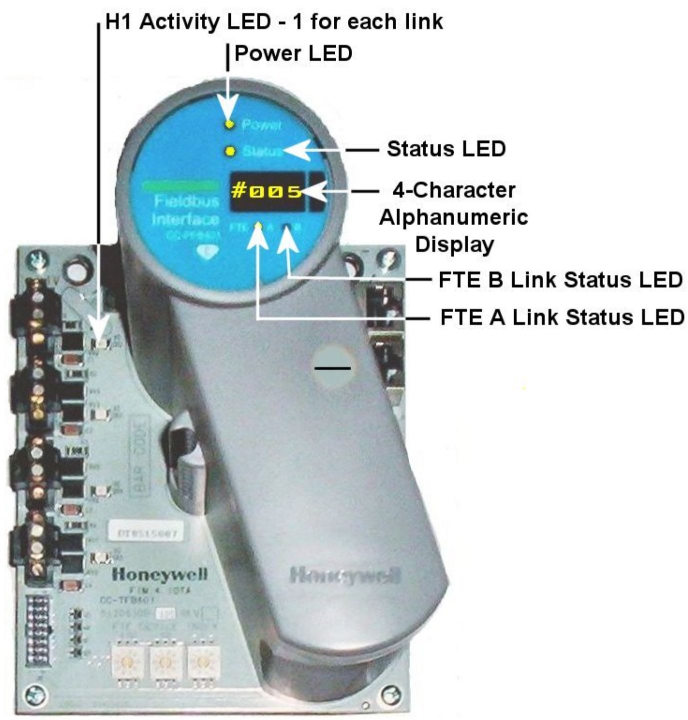


Figure 20 Display and LEDs on Series C FIM and IOTA.

If . . .	Is . . .	Then, it means that
H1 Activity LED	Off	There is no activity on the link.
	Blinking Green	There is transmission activity on the link
Series C FIM Power LED	Off	Series C FIM is not receiving power.
	Green	Series C FIM is powered.

If . . .	Is . . .	Then, it means that
Series C FIM Status LED	Off (Blanked or Frozen Display)	There is a Series C FIM fault, no cause specified; or hardware failure.
	Green	Non-redundant or primary Series C FIM is loaded and operation is okay. For redundant operation, does not imply if the module is synchronized or not.
	Blinking Green	Non-redundant or primary FIM4 is okay, not loaded or there is a soft failure. This may occur if no database is loaded to the module or a soft failure condition exists.
	Amber	Secondary Series C FIM is in the backup or secondary state and the module is synchronized.
	Blinking Amber	Secondary Series C FIM is in the backup or secondary state and is not synchronized or has a soft failure.
	Red	Module is running power on self test, or power on self test has failed.
	Blinking Red	• Series C FIM has no IP Address, no role. This should be a transient condition after power up unless BOOTP server is not running. Display will indicate module is not faulted. The module is not on control. • Series C FIM is in Alive (ALIV) or Ready (RDY) state and can accept a firmware download. Application image is not executing.The module is not on control. • Series C FIM is in LOAD state and accepting a firmware download. Application image is not executing. The module is not on control. • Series C FIM is in the FAIL state. An unrecoverable error has been logged and the module has gone off control.

If . . .	Is . . .	Then, it means that
FTE A and/or FTE B LED	Off	Link integrity signal is present, but no activity. This is unlikely because every node will normally show some activity over several seconds.
	Blinking Green	Link integrity signal is present, along with some activity. Frequency of green LED pulse indicates the amount of transmit traffic on the FTE link.
	Red	Link is not connected. No link integrity signal.
4-Character Alphanumeric Display	**#nnn**	It is the Series C FIM's FTE device index, where nnn can be 001 to 512. Does not change with redundancy role.
	X-CE	Error during link communications Initialization. Where X equals 1, 2, 3, or 4 for the respective link.
	X-DE	Error during link database initialization. Where X equals 1, 2, 3, or 4 for the respective FFLink.
	X-CI	Link is undergoing communications initialization. Where X equals 1, 2, 3, or 4 for the respective FFLink.
	X-DI	Link is undergoing database initialization. Where X equals 1, 2, 3, or 4 for the respective FFLink.
	X-nn	Link is currently online with connected devices. Where X equals 1, 2, 3, or 4 for the respective FFLink. And, nn equals number of devices on the link from 01 to 16.
	BKUP	Series C FIM is currently in the Secondary role.
	-np-	Primary or Secondary Series C FIM is configured for redundant operation, but a partner module is not physically present.
	nrdn	Series C FIM is explicitly configured to be non-redundant.

If . . .	Is . . .	Then, it means that
	!syn	Redundant Series C FIM pair is not synchronized. Primary or secondary Series C FIM is configured for redundant operation and a partner module is physically present. Initial synchronization has not occurred.
	sync	Redundant Series C FIM pair is synchronized.
	XXX%	Initial synchronization is in progress. Where XXX can be 000 to 100 to indicate percentage complete.
	PRI	Series C FIM is operating as primary in redundant Series C FIM pair.
	SEC	Series C FIM is operating as secondary in redundant Series C FIM pair.
	TXXX	Series C FIM is in test mode. Where XXX equals some self test code.
	BOOT	Series C FIM is starting up.
	ALIV	Series C FIM is executing boot firmware. A valid application image has not been loaded.
	LOAD	Series C FIM is executing boot firmware. A new boot or application image is being loaded.
	RDY	Series C FIM is executing boot firmware. A valid application image has been loaded.
	FAIL	Series C FIM has detected a software failure.

Control Builder Block Icon Descriptions

FIM4 block icons

The following table summarizes the various appearances that a FIM4 block icon can assume based on configuration, view, and current Series C FIM operating state and status.

If Icon is . . .	Then, it represents . . .	And, Module State Is. . .
Project Tab		
 (gray)	FIM4 configured for non-redundant operation.	N/A
 (gray/white)	FIM4 primary configured for redundant operation.	N/A
 (white/gray)	FIM4 secondary configured for redundant operation.	N/A
Monitoring Tab		
 (blue)	FIM4 is non-redundant.	Idle
 (blue/white)	Primary FIM4 is synchronized	Idle
 (blue/yellow)	Primary FIM4 is not synchronized and partner FIM4 is visible.	Idle
 (yellow/blue)	Secondary FIM4 is not synchronized and partner FIM4 is visible.	Backup
 (blue/shadow)	Primary FIM4 is not synchronized and partner FIM4 is absent or incompatible.	Idle
 (green)	FIM4 is non-redundant	Run
 (green/white)	Primary FIM4 is synchronized	Run

If Icon is . . .	Then, it represents . . .	And, Module State Is. . .
 (white/green)	Secondary FIM4 is synchronized	Backup
 (green/yellow)	Primary FIM4 is not synchronized and partner FIM4 is visible	Run
 (green/shadow)	Primary FIM4 is not synchronized and partner FIM4 is absent or incompatible	Run
 (yellow)	FIM4 is non-redundant	No Database
 (yellow/white)	Primary FIM4 is synchronized	No Database
 (yellow/yellow)	Primary FIM4 is not synchronized and partner FIM4 is visible	No Database
 (yellow/shadow)	Primary FIM4 is not synchronized and partner FIM4 is absent or incompatible	No Database
 (shadow/yellow)	Secondary FIM4 is not synchronized and partner FIM4 is absent or incompatible	Backup
 (red)	FIM4 is non-redundant and not communicating	Offnet
 (red/white)	Primary FIM4 is not communicating.	Offnet
 (white/red)	Secondary FIM4 is not communicating.	Offnet

FIM4 link icons

The following table summarizes the various appearances that a FIM4 Link block icon can assume based on view and current FFLink state. FFLink blocks apply only to the primary or non-redundant FIM4 block, and do not have matching blocks for secondary FIM4.

If Icon is . . .	Then, it represents . . .
Project Tab	
 (gray)	FFLink associated with configured non-redundant or primary FIM4.
Monitoring Tab	
 (gray/arrow)	Control Builder / Control Data Access (CDA) server is currently establishing communication to the FFLink.
 (blue)	FFLink is inactive.
 (yellow)	FFLink is initializing.
 (green)	FFLink is active.
 (green/asterisks)	FFLink is active and uncommissioned devices exist on the H1 network.
 (red/black exclamation)	Communication to the FFLink is unavailable
 (red/white exclamation)	Communication to the FFLink is available, but the FFLink is in a failed state.

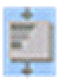
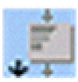
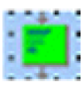
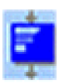
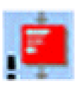
Fieldbus Device icons

The following table summarizes the various appearances that a Fieldbus device block icon can assume based on view and current device state. FFLink blocks apply only to the primary or non-redundant FIM4 block, and do not have matching blocks for secondary FIM4

If Icon is . . .	Then, it represents . . .
Project Tab	
 (gray)	Device added to Project.
Monitoring Tab	
 (gray/arrow)	Control Builder / Control Data Access (CDA) server is currently establishing communication to the device.
 (green/question mark)	Uncommissioned device.
 (green)	Device is online and is configured as a Basic device.
 (green/green clock)	Device is online and is configured as a Link Master device. Green clock means that the backup LAS in the field device is the same version and is in sync with FIM4 Link. It is important to operate with a green clock, so the Link Master device has a valid backup LAS in case of a FIM4 failure.
	Device is online and is configured as a Link Master device. A bad link schedule exists in the device. A Red Clock indicates that either the field device never received a backup LAS or is not at the same version and is out of sync with FIM4 Link. Such a condition can occur if the backup LAS cannot fit in the field device (For example, EXPKS_E_CL_SCHED_TOOBIG error seen during configuration download) or configuration download to the field device was aborted because of other failures.
 (red/black exclamation)	Device is offnet. Communications with the device is unavailable.
 (red/red exclamation)	Communications with the device is available, but the device is in a failed state. The device is a Basic device.
 (red/red exclamation/red clock)	Communications with the device is available, but the device is in a failed state. The device is a Link Master device.

Fieldbus block icons

The following table summarizes the various appearances that a Fieldbus block icon associated with a fieldbus device can assume based on view and current block state. The faceplate of the block will vary to reflect the block type such as Resource, Transducer, Analog Input, and so on.

If Icon is . . .	Then, it represents . . .
Project Tab	
 (gray)	Block added to Project.
Monitoring Tab	
 (gray/arrow)	Control Builder / Control Data Access (CDA) server is currently establishing communication to the block.
 (green)	Block is active
 (blue)	Block is inactive
 (red/exclamation)	Block is offnet. Communications with the block is unavailable.

Activating or Inactivating FIM4 Link

Use the following procedure to change the state of a FIM4 H1 Link through the **Monitoring** tab in Control Builder.

Prerequisites

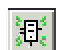
- You can view FIM4 and its Links through the **Monitoring** tab of Control Builder.

Considerations

- Inactivating a FIM4 Link essentially shuts downs the H1 Link and the Series C FIM, if all Links are inactivated, and interrupts the transfer of data to the Experion system. Be sure your system can tolerate the lost of live data, while the FFLink is inactive.

- All fieldbus device blocks associated with the inactivated FFLink are also shutdown, including those contained in Control Modules
- You can also initiate this same function through the corresponding Detail display in Station.

Changing FIM4 Link state

Step	Action
1	• In **Monitoring** tab, click desired **FFLink** icon. • Click the **Toggle State** button  in the toolbar.
2	In the **Change State** dialog, click the **Yes** button to initiate the state change.
3	Check that the **FFLink** icon changes from green to blue, when it is inactivated; or from blue to green when its activated.
4	Repeat this procedure to inactivate or activate a given FFLink as applicable.
5	To inactivate the FIM4, use this procedure to inactivate all 4 FFLinks.
6	This completes the procedure. Go to the next section.

Initiating Series C FIM Link Segment Commands

Use the following procedure to issue a startup, shutdown, or purge NVS (non-volatile storage) command to a Series C FIM H1 Link through the **Monitoring** tab in Control Builder.

Prerequisites

- You can view FIM4 and its Links through the **Monitoring** tab of Control Builder.

Considerations

- Shutting down a Series C FIM Link essentially shuts downs the H1 Link and interrupts the transfer of data to the Experion system. Be sure your system can tolerate the lost of live data, while the FFLink is shutdown.
- All fieldbus device blocks associated with the shutdown FFLink are also shutdown, including those contained in Control Modules
- You can also initiate this same function through the corresponding Detail display in Station.

Using segment commands

Step	Action
1	In **Monitoring** tab, click plus sign + for **FIM4** icon to expose its associated Links in tree hierarchy.
2	Right-click desired **FFLink** icon and select **Module Properties** from the list to call up the **FFLink Block Parameters** form.
3	Click the **SEG Command** box and select **STARTUP**, **SHUTDOWN**, or **PURGENVS** from the list to initiate desired FFLink/segment action. Click the **Yes** button to confirm the action.
4	Check that the **SEG State** read-only parameter changes to reflect the current state of the FFLink.
5	Click the **OK** button to close the **FFLink Block Parameters** form.
6	Repeat this procedure to use segment commands on another FFLink or issue another command on the selected FFLink, as applicable .
7	This completes the procedure. Go to the next section.

Activating or Inactivating Fieldbus Block

Prerequisites

- You can view FIM4, its Links, and assigned fieldbus devices through the **Monitoring** tab of Control Builder.

Considerations

- Inactivating a fieldbus device block interrupts the transfer of data to the Experion system. Be sure your system can tolerate the lost of live data, while the block is inactive.

Changing block state

Step	Action
1	In **Monitoring** tab, click plus signs + for **FIM4**, **FFLink**, and **Device** icons to expose fieldbus blocks in tree hierarchy.
2	Right-click desired **fieldbus block** icon and select **Activate**> or **Inactivate**>**Selected Item(s)** from the list.

Step	Action
3	Check that the **block** icon changes from green to blue, when it is inactivated; or from blue to green when its activated.
4	Repeat this procedure to inactivate or activate a given fieldbus block, as applicable.
5	This completes the procedure. Go to the next section.

Checking/Changing Fieldbus Device Functional Class

Prerequisites

- You can view FIM4, its Links, and assigned fieldbus devices through the **Monitoring** tab of Control Builder.
- :You have logged on with a security level that allows you to make changes in a control strategy.

Considerations

- The Series C FIM is the primary **Link Master** for all Links. If you designate a device as a ***backup*** **Link Master**, be sure it has the capacity to handle the Link Schedule. Otherwise, you may have to reset the device as a **Basic** type and restart it to restore operation, if the Link Schedule is too large for it to handle.
- You can confirm the change by calling up the device's **Parameters** form and checking that **BASIC** appears as the value for the DL Operational Functional Class parameter on the **Network Management (Basic)** tab.
- If you change the functional class for a device in the **Project** tab, be sure you restart the device after it is loaded from **Project** for the change to be made in the device.
- You can manually cycle power to the fieldbus device to restart it.

Changing device functional class

Step	Action
1	In **Monitoring** tab, double-click device icon to open its configuration **Parameters** form.
2	Click **Network Management (LM)** tab to open it.
3	Click the arrow  button in **Boot Operational Function Class** box and

Step	Action
	select **BASIC** or **LINKMASTER** from the list, as applicable.
4	Follow the online Wizard prompts to initiate the change. Click the **Finish** button to complete the change.
5	After fieldbus device restarts, its functional class is changed to **BASIC** or **LINKMASTER** as commanded.
6	Be sure the **Boot Operational Function Class** change is also reflected in configuration form for the **Project** version of the device.
7	Click the **OK** button to close form.
8	This completes the procedure. Go to the next section.

Checking Live List and Commissioning an Uncommissioned Device

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

Prerequisites

- You can view FIM4, its Links, and assigned fieldbus devices through the **Monitoring** tab of Control Builder.

Considerations

- If you add an unconfigured fieldbus device to a live/operating link, it will appear as an **uncommissioned** device in the FFLink block hierarchy in the **Monitoring** tab in Control Builder.
- If you add a device with an address that is below 20 (decimal)/14 (hexadecimal) to the H1 Link, the Series C FIM clears the device's address so it can join the Link at a temporary address that can be reassigned as required.
- If you add an uncommissioned device with a functional class of **LINKMASTER** to the H1 Link, the Series C FIM initiates the following actions to guard the network against incorrect link schedules.
  - **Clears** the Link Schedule,
  - **Clears** the Primary Link Master Flag Variable,
  - **Sets** the configured link settings,

  - **Sets** the device functional class to **BASIC**, and.
  - **Restarts** the device.

- 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 block type 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 the **Uncommissioned Devices** tab on the Device's **Parameters** form in Control Builder. Please see *Upgrading Firmware in Uncommissioned Device* in the next Section *Series C FIM Maintenance* for more information.
- A new Link Schedule must be loaded to the Series C FIM and other Link Master devices anytime there is a change that invalidates the current Link Schedule, such as adding a device, removing a device, or making a configuration change.
- A block type must exist before a device can be integrated/commissioned with the system.

Changing uncommissioned device to commissioned one

Step	Action
1	In **Monitoring** tab, click plus sign + for **FIM4** icon to expose its associated Links in tree hierarchy.
2	Click the plus sign ⊞ for the **FFLink** icon with an asterisks  to expose the uncommissioned device icon .
3	Double-click the uncommissioned device icon to open the FFLink **Parameters** form with the **Uncommissioned Devices** tab open.
4	Click the **Create Project Database Instance for this Device** button to launch a commissioning wizard that automatically steps through the process to commission a device. The process includes creating device Type, changing tag and/or address, creating Project instance, matching Project to device, and loading device, as required.
5	Click the **OK** button to close the **Parameters** form.
6	Repeat this procedure for other uncommissioned devices as needed.
7	This completes the procedure. Go to the next section.

Viewing and Optimizing Link schedule configuration

The following table summarizes how to call up a Link schedule configuration for viewing in the control chart area of the Control Builder application. You can also interact with the natural schedule to optimize it using the considerations listed below. Or, you can configure the FFLink for the system to provide automatic schedule optimization as noted in the following procedure.

Prerequisites

You have completed your control strategy configuration that includes fieldbus blocks integrated with control modules in Control Builder.

Considerations

- The key to optimizing the Link schedule is logically configuring the ORDERINCM parameter for each block to:
  - Sequence blocks in a logical order from inputs to outputs and,
  - Execute non-control blocks after control blocks.
- The following are some general latency related considerations to use as reference for determining the most logical overall sequencing for your Control Strategy.
  - Assume that faster loops are more critical and have a higher priority than slower ones on the same link.
  - Primary or innermost loops are always more critical than secondary or outer loops. The same is true for lower level loops in multi-level cascades, since the timing requirements for each level become less critical.
  - Configurations that do not contain an output type block have a much lower priority than any configuration that does, since these configurations are most likely intended for monitoring only.
  - Back-Calculation publications have the lowest priority, since they only have to be published before they are needed in the next execution sequence.
  - Delay execution of non-control loop auxiliary blocks like integrators through the ORDERINCM parameter sequencing. If they are created last, delay will occur naturally through the default ORDERINCM parameter sequencing.
- See the following procedure to configure the FFLink for automatic schedule optimization.

Calling up Link Schedule

If You Are in the . . .	Then, you can	And, the Link Schedule Configuration shows. . .
Project Tab	Right-click the FFLink icon and select Open Chart from the list. Or, Double-click the FFLink icon.	The block execution and data publications for all Control Modules that are configured in Project and assigned to the FFLINK or CEE. And, they contain fieldbus function blocks that are associated with a device assigned to the given FFLink, and/or Experion blocks that participate in Link publications. See the Sample Link Schedule configuration display in Project Figure that follows this table for reference.
Monitoring Tab	Same as above.	The block execution and data publications for all applicable Control Modules configured and assigned in Project and loaded to the FFLINK or CEE. See the Sample Link Schedule configuration display in Monitoring Figure that follows this table for reference.

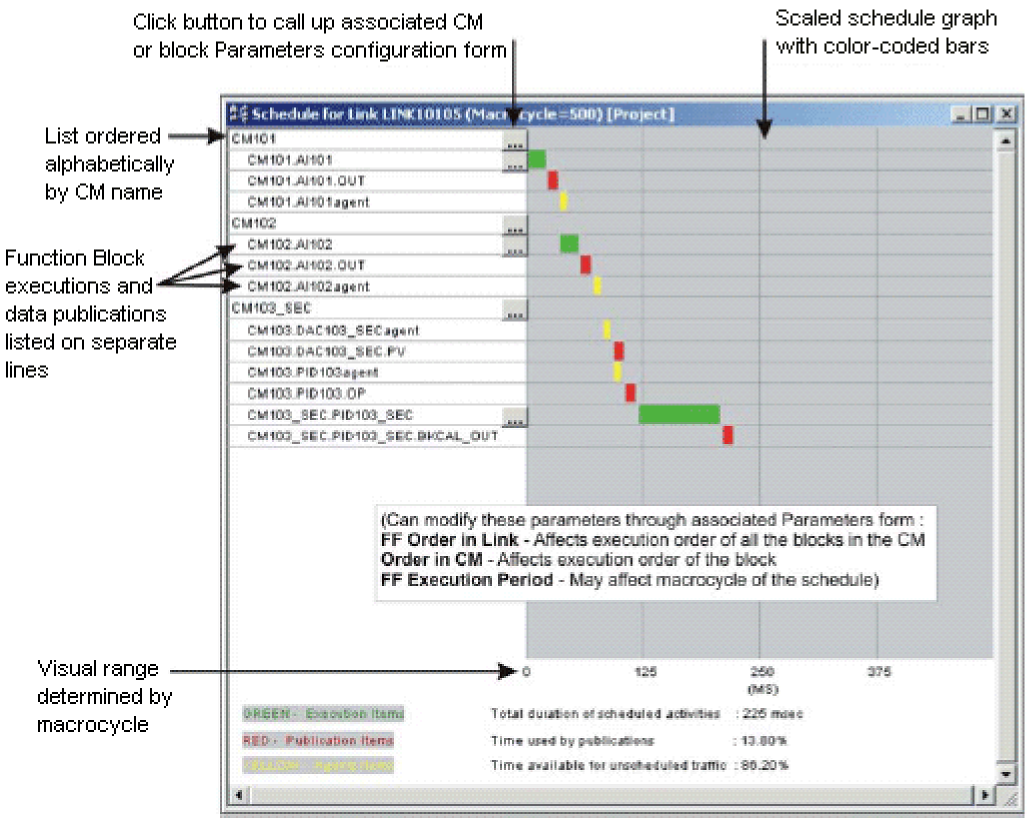


Figure 21 Sample Link Schedule configuration display in Project.

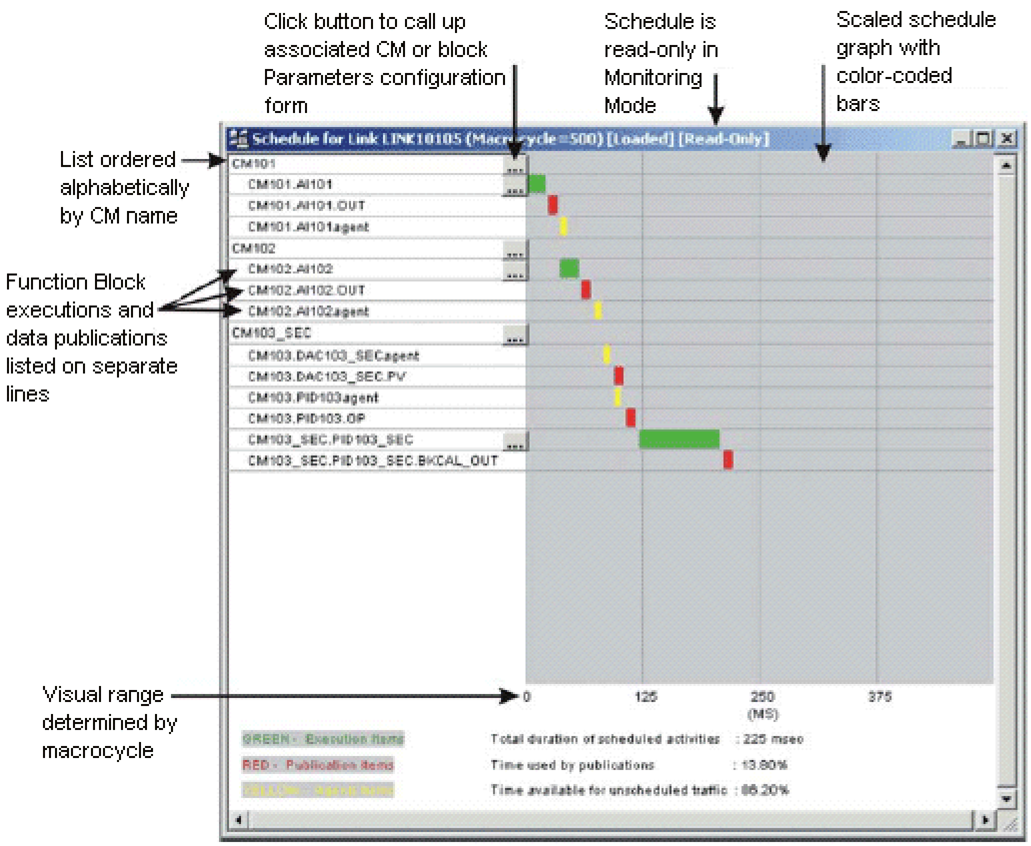


Figure 22 Sample Link Schedule configuration display in Monitoring.

To configure link schedule optimization

Step	Action
1	In the **Project** tab, click the plus sign for the FIM4 icon to expose its FFLinks in the tree.
2	Right-click the first FFLink icon and select **Module Properties** to open the **FFLINK Block Parameters** form.
3	On the **Main** tab, click the down-arrow button in the **Link Schedule Optimization** box and select **OPT** or **OPT_NOFORCE** for automatic link schedule optimization. Otherwise, select **NO_OPT** to disable link schedule

Step	Action
	optimization. Refer to *FFLink Configuration Parameters Reference* for more information about the selections.
4	Click the down arrow button in the **Sample Skew** box and select **SKEW** or **NOSKEW** for desired block execution action. Refer to *FFLink Configuration Parameters Reference* for more information about the selections.
5	Repeat Steps 2 to 4 for other FFLinks, as applicable.
6	Click the **OK** button to close the form and save the settings.
7	This completes the procedure.

Initiating Synchronization Command

Use the following procedure to initiate a synchronization command manually to a redundant Series C FIM pair.

Prerequisites

You can view active redundant FIM4 pair on **Monitoring** tab in Control Builder.

Commanding synchronization

Step	Action
1	In the **Monitoring** tab, double-click the primary **FIM4** icon to call up its **Parameters** configuration form.
2	Click the **Redundancy** tab to display it.
3	Click the **Enable Synchronization** button.
4	Click the **Yes** button to confirm the action and issue the synchronize command.
5	Check the **Redundancy Status** data on the **Parameters** form and confirm that **sync** cycles in the primary Series C FIM's 4-character display.
6	Click the **OK** button to close the **Parameters** form.
7	This completes the procedure. Go to the next section.

Initiating Switchover Command

You can use the following procedure to initiate a switchover command manually to a redundant Series C FIM pair.

Prerequisites

You can view active redundant FIM4 pair on **Monitoring** tab in Control Builder.

The primary and secondary Series C FIMs are synchronized.

Commanding a switchover

Step	Action
1	In the **Monitoring** tab, double-click the primary **FIM4** icon to call up its **Parameters** configuration form.
2	Click the **Redundancy** tab to display it.
3	Click the **Initiate Switchover** button.
4	Click the **Yes** button to confirm the action and issue the switchover command.
5	Confirm that the Secondary Series C FIM assumes the Primary role.
6	Click the **OK** button to close the **Parameters** form.
7	This completes the procedure. Go to the next section.

Using Fieldbus Methods Manager

Prerequisites

Before you can launch Methods Manager you must:

- Create the block type for a device from its manufacturer's DD files that include methods data using the **Type** command in Control Builder.
- Instantiate (create) and load the device block through Control Builder.
- Access the active device block through the **Monitoring** tab in Control Builder.

Launching Methods Manager

You can use one of the following action to launch Methods Manager through Control Builder, assuming the prerequisites have been met.

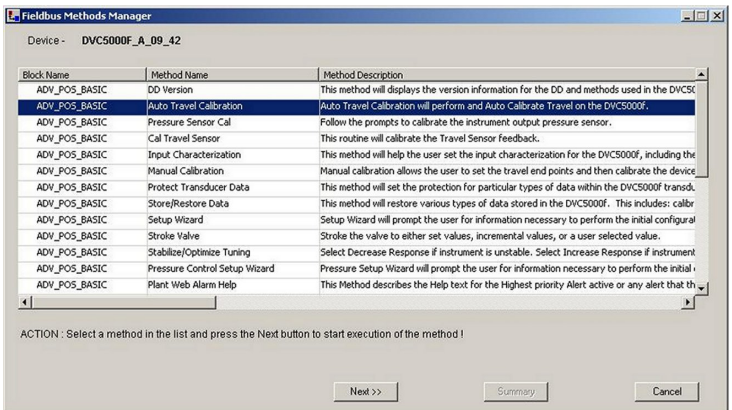
If You Select . . .	Then, . . .	And,
Device in **Monitoring** tab	Click **Field Devices-> Methods Manager…**	Methods Manager dialog provides a selection list that includes all the blocks associated with the device and their available methods with the method description.

Interacting with Methods Manager

The Methods Manager application executes the device's method and prompts users for needed interaction based on the selected method. The following table lists some possible user prompted actions for reference. The actual interaction is determined by the selected method.

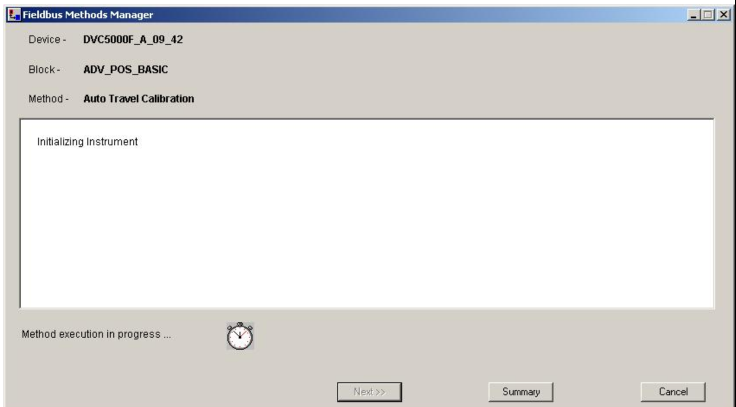
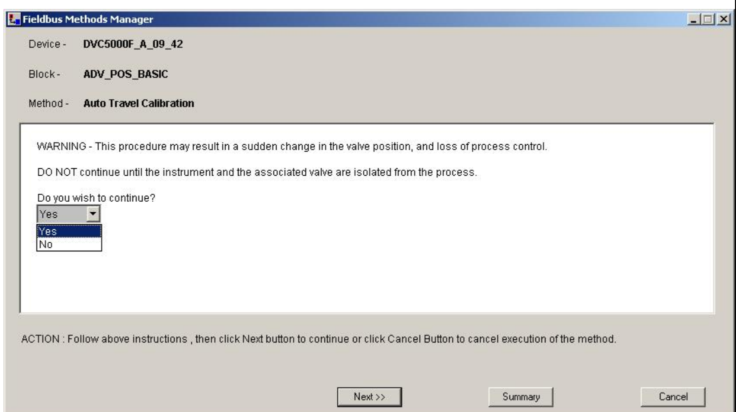
TIP

- When active, press the **Summary** button on the Methods Manager dialog to call up a report that shows all the steps executed during the execution of the selected method up to that point in time.
- When active, press the **Show Methods List** button to see the list of methods and their descriptions for the selected device.

<table>
<tr><th>If Methods Manager Dialog . . .</th><th>Then, You Are Prompted to . . .</th></tr>
<tr><td>Includes list of available Methods.

</td><td>Select the method to be invoked and follow the instructions in the Methods Manager dialog.</td></tr>
</table>

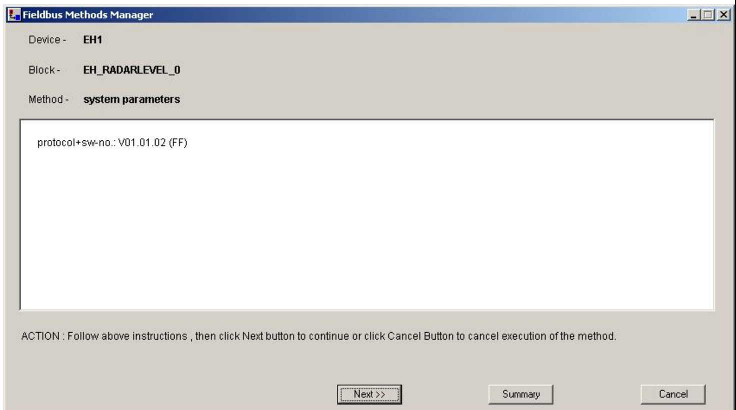
ATTENTION

- While using the Fieldbus Methods Manager to execute a method, do not use Experion applications like Control Builder or Station to change the device/block parameters for the device on which the methods is running.
- While using the Fieldbus Methods Manager to execute a method, some methods may display messages that are very specific to a vendor. In this case, please consult the vendor's user/maintenance manual to verify/validate it.

<table>
<tr><th>If Methods Manager Dialog . . .</th><th>Then, You Are Prompted to . . .</th></tr>
<tr><td>Shows action in progress.
</td><td>Wait for action to complete before continuing.</td></tr>
<tr><td>Shows selection from a list.
</td><td>Select desired action by selecting an option and follow the instruction given in the Methods Manager dialog.</td></tr>
</table>

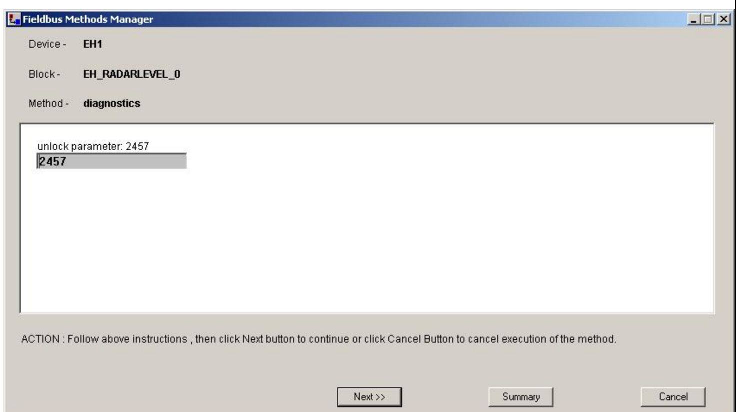
<table>
<tr><th>If Methods Manager Dialog . . .</th><th>Then, You Are Prompted to . . .</th></tr>
<tr><td>

Requests user acknowledgement.

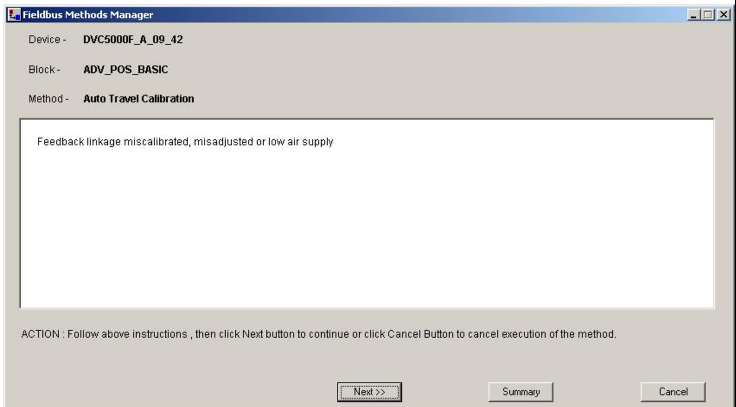
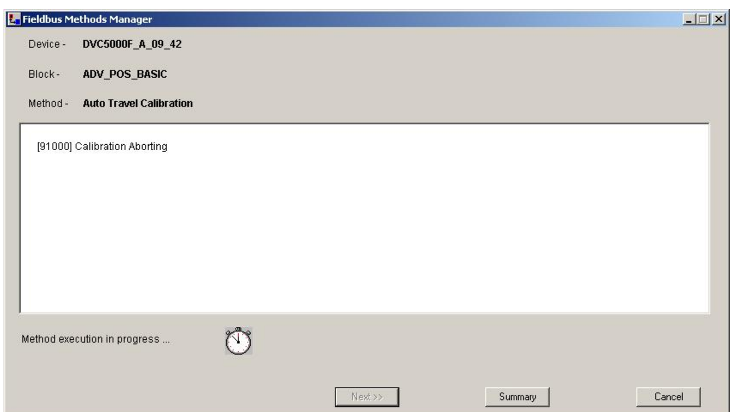


</td><td>Follow the instruction given in the Methods Manager dialog.</td></tr>
<tr><td>

Requests user action.



</td><td>Follow the instruction given in the Methods Manager dialog.</td></tr>
</table>

If Methods Manager Dialog . . .	Then, You Are Prompted to . . .
Displays error message or code 	Note error and follow the instruction given in the Methods Manager dialog.
Shows an abort message 	Wait until the abort sequence is completed.

Using Controller Menu Functions

Many of the Control Builder **Controller** menu functions designed for use with the Controller and CEE components also apply for the FIM4 and Links components. These include **Upload**, **Update**, and **Checkpoint** functions. These functions are useful for correcting mismatches that may occur between the components and the database. Please refer to the *Control Building Guide* for details about a given menu function.

Using Station Detail displays

The Experion server Station application includes pre-configured Detail displays for the Series C 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 Series C FIM with points registered in the Experion server. The Detail displays let you quickly view the component's current state, fault status, and pertinent configuration data.

REFERENCE - INTERNAL

Please refer to the Operator's Guide for detailed information about calling up, navigating, and viewing Station displays.

Using Station Event Summary display

Like the Detail displays, the Station Alarm and Event Summary displays support the integration of fieldbus generated notifications and events. It is integrated with Experion component data and is for the most part self-explanatory. Use the Event Summary display to get a quick review of recent actions that have been initiated within the system.

Series C FIM Maintenance

This section provides information about maintaining the Series C FIM. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Periodic Checks

The following table identifies checks that you should make periodically (every 3 to 6 months) to keep the Series C FIM in good working condition.

Check . . .	Possible Corrective Action . . .
That all segments of the 4-character display and the light emitting diodes (LED) on Series C FIM and IOTA are working.	If segment or LED is not lit or has dimmed, you must replace the Series C FIM or IOTA, since front-panel indicators and LEDs are not field replaceable.
That all connections are secure.	Secure connections, as needed.
That cable insulation is not worn or cracked.	Replace cables, as required.
That IOTA is secure.	Tighten mounting screws.

Recommended Spare Parts

The following table provides a list of parts that you may want to keep on hand for backup.

Part Name	Part Number	Description	Quantity per 10/100
Series C FIM	CC- or CU-PFB401	Series C Fieldbus Interface Module	1/5
Non-Red. IOTA	CC- or CU-TFB401	Non-Redundant Input Output Terminal Assembly	1/2
Non-Red. IOTA	CC- or CU-TFB402	Non-Redundant Input Output Terminal Assembly - Supersedes CC/CU-TFB401 in row above.	1/2
Red. IOTA	CC- or CU-TFB411	Redundant Input Output Terminal Assembly	1/2
Red. IOTA	CC- or CU-TFB412	Redundant Input Output Terminal Assembly- Supersedes CC/CU-TFB411 in row above.	1/2
Fuse	51506348-341	800 mA, quick-acting fuse on Input Output Terminal Assembly	4/25

Part Name	Part Number	Description	Quantity per 10/100
Fuse Holder	51506443-100	Fuse holder on Input Output Terminal Assembly	2/10
Connector	51506471-202	Plug-in connector for H1 link field wiring	4/15
Pwr Conditioner IOTA	F660A, or F860	Redundant Power Conditioner Input Output Terminal Assembly (Optional - order from vendor MTL-Relcom)	1/2

Series C FIM Removal and Installation Under Power

The Series C FIM has been designed to permit removal and installation under power (RIUP) without damaging the module or interrupting IOTA communications.

WARNING

Unless the location is known to be non-hazardous, do **not**:

- connect or disconnect cables,
- install or remove fuses, terminal blocks, and so on,

while the component is powered.

Replacing Failed Series C FIM *CC- or CU-PFB401*

Use the following procedure as a guide for replacing a failed Series C FIM.

CAUTION

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.

Prerequisites

You have logged onto Control Builder with sufficient security level to make control strategy changes.

Considerations

- You can remove and install the Series C FIM under power.
- Be sure you take ESD hazard precautions when handling the Series C FIM and IOTA. See the *Installation Declarations* section for more information.

Step	Action
1	Loosen the screws that hold the Series C FIM to the IOTA and remove the Series C FIM from its connector on the IOTA.
2	Install the new Series C FIM on its connector on the IOTA and tighten the screws.
3	Monitor the front-panel 4-character display to confirm that the Series C FIM receives its IP address and starts running. If this is a redundant configuration, the Series C FIM starts in a secondary role. See *Series C FIM Startup* for more information.
4	Use the **CTool** utility to check that the firmware revision of the new Series C FIM is the correct one for your application. • If firmware revision is correct, go to Step 6, if the Series C FIM is non-redundant; or go to Step 5, if it is redundant. • If firmware revision is not correct, go to the *Upgrading Series C FIM Firmware* section and then return to Step 5, if Series C FIM is redundant; or Step 6, if it is non-redundant.
5	Command the primary Series C FIM to synchronize the secondary Series C FIM though the FIM4 block parameters form in the **Monitoring** tab of Control Builder.
6	Load the block from **Project** tab.
7	This completes the procedure. Go to the next section.

Replacing IOTA *CC-or CU-TFB401, CC-or CU-TFB411, CC- or CU-TFB402, or CC- or CU-TFB412*

Use the following procedure as a guide to replace a non-redundant or redundant IOTA.

Prerequisites

- You have reviewed the previous **CAUTION** note.
- You have logged onto Control Builder with sufficient security level to activate/inactivate components. .

Considerations

- You can replace a non-redundant IOTA with a redundant one, providing that mounting space is available and the Series C FIM has an odd Device Index number assignment.
- Be sure you take ESD hazard precautions when handling the Series C FIM and IOTA. See the *Installation Declarations* section for more information.

Step	Action
1	You may want to inactivate the associated FFLinks through the **Monitoring** tab in Control Builder first, since view of the process will be lost while the IOTA is being replaced
2	Remove the screws from the 24V and COM terminals to disconnect the IOTA from the power bus bars.
3	Record the Device Index number set on the binary coded decimal (BCD) switches on the IOTA.
4	Tag and remove the H1 link connections from connectors on the IOTA.
5	Remove the FTE cables from IOTA connectors.
6	Loosen the screws holding the non-redundant or Primary Series C FIM to the IOTA. Remove the Series C FIM.
7	If you are replacing a redundant IOTA, loosen the screws holding the Secondary Series C FIM to the IOTA. Remove the Series C FIM and be sure you can identify it as the Secondary Series C FIM. Otherwise, go to the next Step.
8	Remove mounting screws, spacers, washers, and remove the IOTA.
9	Install replacement IOTA with screws, spacers, and washers provided.

Step	Action
10	Set the BCD switches to the value noted in Step 3.
11	Connect H1 link connections to connectors on IOTA as noted in Step 4.
12	Connect FTE Cables to appropriate connectors on the IOTA.
13	Tighten screws in 24V and COM terminals to connect the IOTA to the power bus bars.
14	Install non-redundant Series C FIM or Primary Series C FIM on top connector on a redundant IOTA. Tighten screw to hold Series C FIM to IOTA.
15	If you are replacing a redundant IOTA, install Secondary Series C FIM on lower connector on IOTA. Tighten screw to hold Series C FIM to IOTA.
16	Check front-panel display to confirm operational state of the Series C FIM.
17	If necessary, load FIM4 block from **Project** and activate FFLinks in **Monitoring** mode.
18	This completes the procedure. Go to the next section.

Relocating Series C FIM or Reassigning IP Address

The following procedure outlines how to prepare a Series C FIM for relocation to another network or assigning a new device index and IP address.

Prerequisites

- You have logged onto Control Builder with sufficient security level to make control strategy changes.
- Be sure the system is offline or in a safe operating mode.

Considerations

- You can remove and install the Series C FIM under power. See the previous **Warning** before doing so.
- Be sure you take ESD hazard precautions when handling the Series C FIM and IOTA. See the *Installation Declarations* section for more information.
- The Series C FIM retains its assigned IP address in non-volatile storage, which means it must be erased before a new address can be assigned.

- If you are working with a redundant Series C FIM pair, the device index and IP address will be reset in both the Primary and Secondary modules. The node number for a device is dynamically assigned. The primary device in a redundant pair always has an odd node number. The node number is actually the IP address minus the Base IP Address set through **System Preferences** in Control Builder. The Device Index number is set on the Binary Coded Decimal switches on the device's IOTA.
- For a redundant Series C FIM application, be sure to set the Device Index to be an odd number. The odd number will be used for the primary Series C FIM and a one (1) will be added to the number to be used as a device index for the secondary Series C FIM. For example, if you set the Device Index to 17 for the primary Series C FIM, the Device Index for the secondary Series C FIM would be 18.
- We recommend setting the Device Index for a non-redundant Series C FIM application to an odd number also, so you can easily upgrade to a redundant Series C FIM application at a later date, if desired.
- After Device Index and IP address are reset, refer to the *Series C FIM Configuration* and *Loading FIM4 Components Online* sections in this book to configure and load the associated function blocks for the reassigned Series C FIM(s).

Resetting device index and IP address

Step	Action
1	Remove power from the IOTA containing the non-redundant Series C FIM or redundant Series C FIM pair to be reset.
2	Locate the three binary-coded decimal rotary switches on the lower left side of the IOTA.
3	Use a small flat-bladed screwdriver or your thumbnail to set the left-hand switch (100) to zero (0), midddle switch (10) to zero (0) and the right-hand (1) switch to zero (0).
4	Apply power to the IOTA. Wait for Series C FIM(s) to start up and confirm that 4-character display shows device index as **#000**.
5	Remove power from the IOTA.
6	Use a small flat-bladed screwdriver or your thumbnail to set the left-hand switch (100) to the desired most significatnt decimal number, middle switch (10) to the desired next most significant decimal number and the right-hand (1) switch to the desired least significant decimal number. The applicable setting range is 01 to 511.
7	Apply power to the IOTA. Wait for Series C FIM(s) to start up and confirm that

Step	Action
	4-character display shows device index as **#nnn**. Where nnn equals the Device Index number set on the switches in Step 6. The IP address is assigned based on the new device index number.
8	This completes the procedure. Go to the next section.

Upgrading Firmware in Uncommissioned Device

Use the following procedure to upgrade the firmware in an uncommissioned device through Control Builder.

Prerequisites

- You have the vendor supplied upgrade file for the device.
- You are monitoring the FIM4/FFLINK through the **Monitoring** tab in Control Builder.

Considerations

If the **Load Firmware** button is inactive (grayed out) on the **Uncommissioned Devices** tab, try changing the device address to a lower number in the 30 to 40 range.

Initiating Firmware upgrade

Step	Action
1	Double-click the uncommissioned device icon to open the FFLINK Parameters form with the **Uncommissioned Devices** tab open.
2	Click the **Load Firmware** button to open the **Device Firmware Upgrade** dialog.
3	Click the **Upgrade** button to open the **Open** dialog.
4	• Use the **Look In** box to navigate to the directory location that includes the vendor's upgrade file. • Select the file and click the **Open** button to initiates the firmware upgrade. This may take a few minutes to complete.
5	Once the firmware upgrade is completed, wait for the device to rejoin the network. This may take up to three minutes. Click the close button to close the dialog.

Step	Action
6	This completes the Upgrade procedure.
7	See the *Changing uncommissioned device to commissioned one* procedure to commission the device. Otherwise, click the OK button to close the form.
8	Go to the next section.

Replacing a failed device with a like device having the same model name and device revision

Use the following procedure to replace a failed fieldbus device with a like device having the same Model Name and Device Revision (DEV_REV).

TIP

A wizard will guide you through the device replacement process when you initiate the action through the device detail display in Station. In this case, you can ignore the following procedure.

Prerequisites

The system is offline or in a safe operating mode

Considerations

- After the procedure, be sure you complete the device load before you return the system to normal operation.
- If you are replacing a device that includes **instantiated** blocks, you must re-load any Control Module and Sequential Control Module that include client/server connections to the blocks in the replacement device from the **Monitoring** tab before putting the device into operation. See

Replacing failed device that includes instantiable blocks with like device for more information.

Replacing like device

Step	Action
1	Disconnect the failed device from the Link. Fieldbus device icon turns red in **Monitoring** tab.
2	Right-click the failed device icon and select **Device Replacement** from the list to call up the **FFLINK Block Parameters** form with the failed device listed in the bottom half of the **Device Replacement Dialog**.
3	• Connect the replacement fieldbus device to the Link • Wait for the replacement device to appear as an Uncommissioned Device (UCD) listed in the top half of the **Device Replacement dialog**.
4	• Click the check box to the left of the **Tag** column for the Replacement Device. • The **Replace the Failed Device with the Uncommissioned Replacement Device** button becomes active.
5	• Click the **Replace the Failed Device with Uncommissioned Replacement Device** button. • Click the **Continue** button to acknowledge **Warning** prompt. • If another **Warning** prompt appears, the replacement device was previously matched to another device and the previous match will be undone, if you continue. Otherwise, go to the next Step.
6	• **Progress** dialog appears to track the status of the replacement. • Wait for **Replacement Device Load Confirmation** dialog to appear. Click the **Continue** button. The **Load Dialog** for replacement device appears. It is normal for an error code (Err xxxx) to appear in the **Current State** column for the blocks.
7	Click the check boxes for the automatic state changes near the bottom of the dialog. Click the **OK** button. Load dialog appears to track progress of the load.
8	Once the load is completed, check the **Monitoring** tab to confirm that the device icon has turned green and function blocks are inactive (blue) or active (green) per the selected **Post Load State**.
9	This completes the procedure. Go to the next section.

Replacing a failed device with a different device using a different block type

Use the following procedure to replace a failed device with a different device using a different block type.

Prerequisites

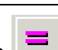
The system is offline or in a safe operating mode

Considerations

- It is a good idea to record the Tag and Address assignments for the failed device before deleting it.
- It is a good idea to print out a parameters and connections report for a CM before you modify it. This will serve as a convenient configuration reference.
- You must unassign the given fieldbus block from its associated device before you can delete the device from **Project**.
- If you unassign a Fieldbus block that has a RCAS_IN or ROUT_IN connection from a Experion block, be aware that the VCR that was formed when the connection was made is **not** removed when the block is unassigned. To remove this VCR, you must delete the block and then recreate it. The same is true, if you want to delete the RCAS_IN or ROUT_IN connection and re-wire it to the block's CAS_IN pin.
- Always initiate match from Project Device to UCD for replacement devices.

Replacing different device

Step	Action
1	Optionally, inactivate any CMs that contain blocks associated with the failed device.
2	Disconnect the failed device from the Link. **Device** icon turns red in **Monitoring** tab.
3	In **Monitoring** tab, locate all Control Modules (CMs) containing fieldbus blocks associated with the failed device. The associated fieldbus block icons will be red.
4	Right-click the CM icon and select **Force Delete** from the list to open **Force Delete** dialog
5	Click the **Delete Selected object(s)** button. Deletion errors are displayed in

Step	Action
	the list box for the dialog. You can ignore these, since you know the failed device is off net.
6	Click the **Close** button to acknowledge any errors or skip this Step, if no errors are detected.
7	Repeat Steps 4 to 6 for other CMs, as required.
8	Right-click the failed device icon and select **Force Delete** from the list.
9	Click the **Delete Selected object(s)** button. Deletes Failed device from the **FFLink** folder in **Monitoring**.
10	In the **Project** tab, double-click the CM that was deleted from the **Monitoring** tab to open CM in control chart.
11	Right-click the fieldbus function block and select **Function Block Unassign** from the list to unassign Fieldbus block from device .
12	Repeat Step 11 for other blocks, if required.
13	Click the close  button. Click the **Yes** button to acknowledge the prompt and close the chart and save changes to CM.
14	Repeat Steps 10 to 14 for other CMs, if required.
15	In **Project** tab, right-click failed device icon and select **Delete** from list to open the **Delete Selected object(s)** dialog.
16	Click the **Delete Selected object(s)** button to delete the device.
17	In the **Library** tab, drag and drop block type for replacement device from applicable vendor directory to open area in **Project** tab to Open **Name New Function Block(s)** dialog. .
18	Key in tag name of failed device in the **Destination** column and click the **Finish** button. Device icon appears in **Project** tab.
19	Click the device icon and click the  **assign** button in the toolbar to Call up the **Execution Environment Assignment** dialog box.
20	With **Devices** tab open and device selected, click the **Assign** button to assign device to selected FFLINK.
21	Click the **Close** button to close the dialog.
22	Click + plus sign to open the FIM4, FFLINK and Device directories in **Project**

Step	Action
	tab to Expand directories to show contents.
23	Double-click the device icon to open **Device Block Parameters** form
24	Check that the **Device Network Node Address** entry matches the address for the failed device. If required, key in matching address. Click the **Ok** button to close form.
25	In **Project** tab, double-click CM that contains unassigned function block for failed device to open CM in control chart.
26	Right-click fieldbus block and select **Function Block Assign** from list to open the **Function Block Assignment Dialog**.
27	Click the check box for **Device Name** and click the **Assign** button. Assigns fieldbus block to new device.
28	Repeat Steps 26 and 27 for other blocks, if required.
29	Click the close button and click the **Yes** button to acknowledge the prompt to save changes and close the form.
30	Double-click the UCD icon to call up the **FFLINK Block Parameters Uncommissioned Devices** form.
31	Click the **Match** button to call up the Device Match dialog.
32	Click check box to left of **Tag** column for matching **Project Device**. Selects project device and activates the **Match** buttons.
33	Click the Match From Project Device to Uncommissioned Device button.
34	Click the Continue button to acknowledge Warning prompt about Tag/Address changes in UCD.
35	Wait for match to be completed and note that the UCD's Tag and Address now match the Project Device. Click the OK button to close the Match dialog.
36	In the Project tab, click FFLINK with new project device and click Controller->Load with Contents to call up the Load Dialog.
37	• Be sure **Load** check boxes for components to be loaded include a check mark. Click check box to add/remove component as applicable. • Click the **OK** button.
38	Wait for load to complete and check that UCD changes to commissioned device, in Monitoring tab.

Step	Action
39	In **Project** tab, click CEE with reconfigured CM and click **Controller**->**Load With Contents** to call up **Load Dialog,**
40	• Be sure **Load** check boxes for components to be loaded include a check mark. Click check box to add/remove component as applicable. • Click the **OK** button.
41	Wait for the load to complete and check that the CM is restored in **Monitoring** tab.
42	This completes the procedure. Go to the next section.

Changing or clearing tag and/or address of uncommissioned device

Use the following general procedure to change the tag and/or address of a uncommissioned device. You can also use this procedure to clear the tag name and/or address from a device to re-initialize it.

Step	Action
1	In the **Monitoring** tab, double-click the uncommissioned device icon  to open the **FFLINK Parameters** form with the **Uncommissioned Devices** tab open.
2	• To change the tag name, key in desired Tag (PD_TAG) name. Usually use the tag name that matches the one used for the **Project** device. Press the **Enter** key or click the **OK** button. • To clear the tag name, just highlight it and press the **Delete** key. Click the **OK** button.
3	Click the **Continue** button to acknowledge **Warning** prompt and initiate change.
4	Wait for change to complete. Press the **Tab** key to move cursor to Address box.
5	• To change the address, key in desired address and press the **Enter** key. • To clear the address, just highlight it and press the **Delete** key. Click the **OK** button.
6	Click the **Continue** button to acknowledge **Warning** prompt and initiate

Step	Action
	change.
7	Wait for change to complete. Click the **Ok** button to close the form.
8	Confirm that the tag for the uncommissioned device icon in the **Monitoring** tab has changed or has been cleared (NoTag_248).
9	This completes the procedure. Go to the next section.

Preparing Fieldbus device for a move or software migration

Use the following procedure to prepare a device before migrating the system to a new software release or moving the device to another Link.

ATTENTION

If you do **not** prepare a device as outlined in the following procedure, you may have problems loading a device after it is moved to another LINK or after the system software has been migrated to a new release.

Step	Action
1	In the **Monitoring** tab, click + plus sign for FIM4, FFLINK, and Device icons to expand directories to block level.
2	Right-click the fieldbus block icon and select **Inactivate**->**Selected Item(s)** to call up the **Change State** dialog.
3	Click the **Yes** button to initiate the action. Check that the function block icon turns Blue.
4	Double-click the fieldbus block icon to call up the **Block Parameters** configuration form.
5	On the **Process** tab, check that the **Actual Mode** has changed to **OOS** to confirm that the block is in Out-Of-Service (OOS) mode.
6	Click the **OK** button to close the form.
7	Repeat Steps 2 to 6 for other fieldbus blocks (if applicable), Transducer, and Resource blocks in the device, in that order.
8	Device is now ready to be decommissioned and moved to another Link or

Step	Action
	migrated to a new system software release.
9	This completes the procedure. Go to the next section.

Using Fieldbus device simulate function

Some of the standard fieldbus function blocks support a signal simulation function for testing purposes only. The following procedure shows the general steps involved in using the simulate function with an Analog Input block in a Honeywell ST3000 transmitter. You can easily adapt this procedure to any applicable fieldbus device.

CAUTION

Only use the simulate function when the system is offline or in a safe operating mode for testing purposes.

Prerequisites

The fieldbus device is capable of supporting a simulate function.

Step	Action
1	Refer to the vendor documentation supplied with the fieldbus device to determine the location of the simulate activating jumper or switch on the device. A switch or jumper is usually included in the device to inhibit accidental activation of the simulate feature.
2	Position the jumper or switch to its simulate enable position.
3	In the **Monitoring** tab, open the directory for the fieldbus device icon and double click the Analog Input function block to call up the **Block Parameters** form.
4	Click the **Maintenance** tab to open it.
5	In the **Simulate En/Disable** box, click the arrow button and select **Simulation Active**.
6	Click the **Yes** button to confirm the action and initiate the simulate function.
7	Click the arrow button in the **Simulate Status** box. For this example,

Step	Action
	select **Good** from the list.
8	Click the **Yes** button to confirm the action and set the **Simulate Status** to Good.
9	Click in the **Simulate Value** box and key in **25** and press the **Enter** key.
10	Click the **Yes** button to confirm the action and set the **Simulate Value** to 25.
11	Click the **OK** button to close the form
12	In the **Monitoring** tab, double-click the Control Module that includes the Analog Input block for the device with the simulated value to open it in the control chart.
13	Check that OUT VALUE for the Analog Input block is 25 to confirm that simulate value is active. Click the close button to close the CM.
14	To disable the simulation, repeat Steps 3 to 6 and select **Simulation Disabled** instead of **Simulation Active** in Step 5.
15	Click the **OK** button to close the form.
16	Reverse the jumper position set in Step 2 to disable the simulate function in the device.
17	This completes the procedure. Go to the next section.

Checking Fieldbus device calibration

Please refer to the manufacturer's documentation for the fieldbus device to determine the recommended calibration schedules and procedures. The **Tune** and **Other** tabs on the **Parameters** form for a device's transducer block provide pertinent calibration information, when accessed through the **Monitoring** tab in Control Builder. The following illustrations show typical **Tune** and **Other** tabs for reference.

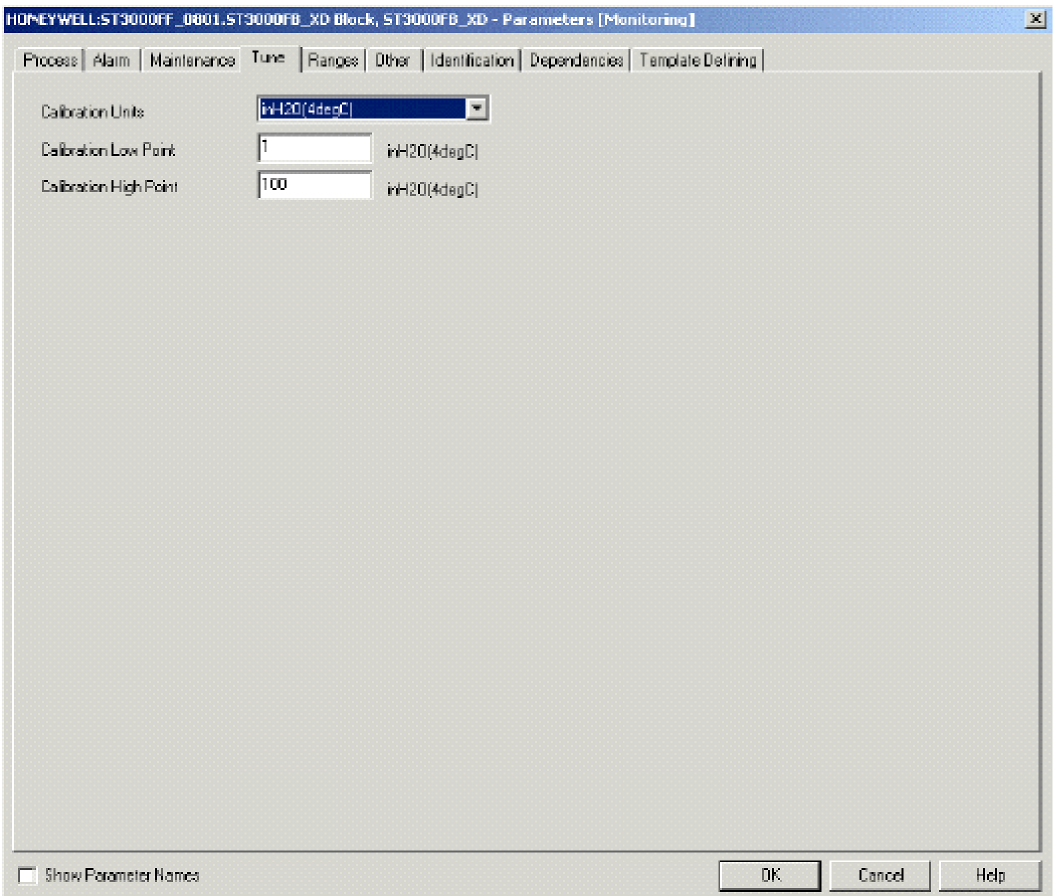


Figure 23 Typical Tune tab for fieldbus device's Transducer block.

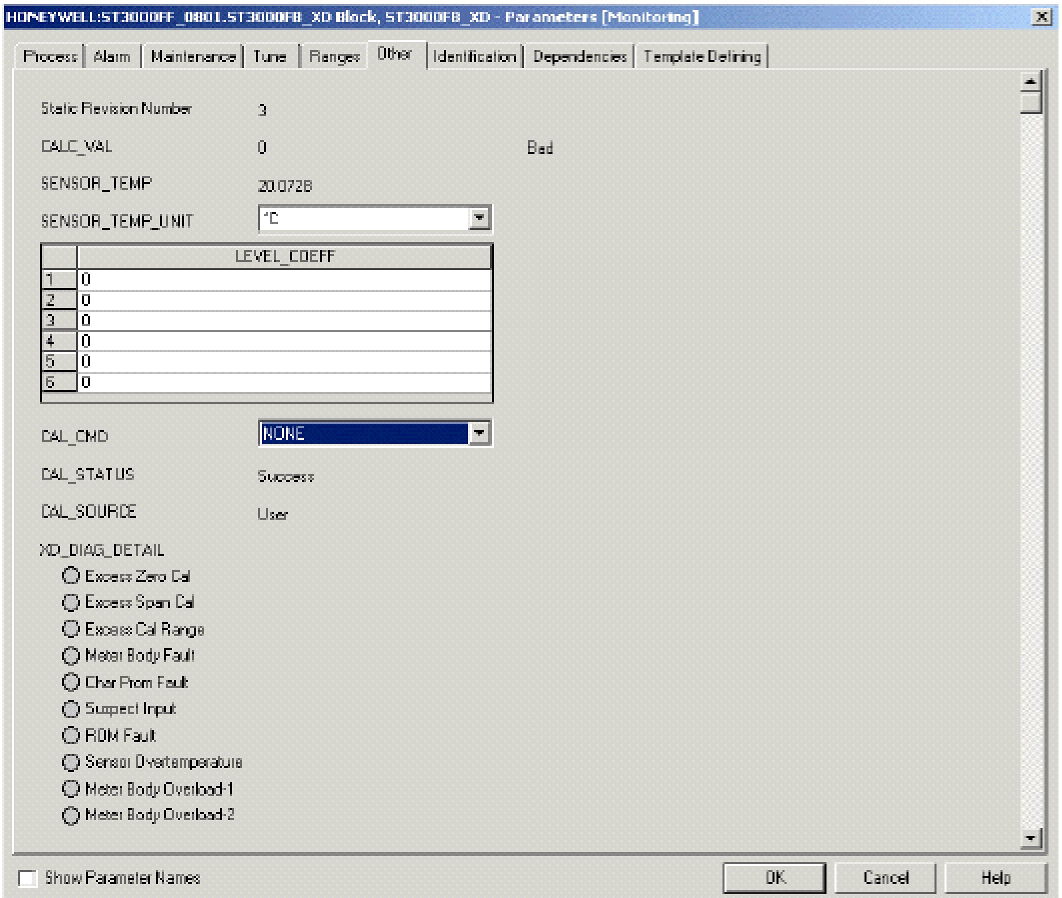


Figure 24 Typical Other tab for fieldbus device's Transducer block.

Series C FIM Troubleshooting

This section provides information about troubleshooting the Series C FIM. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Isolating Problems

This section offers some general data about fault classifications, initial checks, and Series C FIM self-test diagnostic codes that may help you isolate the cause of a problem.

If you have observed a specific symptom and/or identified an error code, go to the Fixing Common Problems section to see if you can find a matching topic.

Fault Classifications

The following table lists some possible ways for classifying faults along with some identifying characteristics.

Fault Classification	Characteristics
Hard Failure	Failure detected by hardware; operation cannot continue. If the fault does not prevent software processing the problem, the affected node will be rebooted under software control into the FAIL State. • If failure occurs on Primary node, it triggers a switchover to synchronized Secondary node. • If failure occurs on Secondary node, it results in loss of synchronization and reduced availability until the problem is resolved.

Fault Classification	Characteristics
	• If failure occurs on non-redundant node, it results in loss of control and loss of view.
Severe Failure	Failure detected by software; operation cannot continue. The affected node will be rebooted under software control into the FAIL State. The failure scenarios are as listed in the row above for the Hard Failure.
Soft Failure	Failure detected by software; operation continues with full control and full view. Soft failures are alarmed to the operator. FTE will be monitored by the FTE System Management Tool. • If failure occurs on Primary node, it does not trigger a switchover to synchronized Secondary node. • If failure occurs on Secondary node, it does not result in loss of synchronization. • If failure occurs on non-redundant node, it does not result in loss of control and loss of view.
Installation/Startup Failure	Detected by software. Node may not become operational. • This failure does not apply to synchronized Primary node, since installation and startup must be successful to reach synchronized Primary state. • If failure occurs on Secondary node, it results in the inability to complete the initial synchronization and to view the node on the network. • If failure occurs on non-redundant node, it results in the inability to commence control and to view the node on the network.
Communication	Communication errors between peer nodes and/or I/O devices, including Fault Tolerant Ethernet Bridge (FTEB), do not cause any node state change.

Initial Checks

This section offers some checks that you can make to help isolate the problem. The checks are arranged in no particular order.

Checking Control Builder error code reference

An indication of a problem may be in the form of an error dialog that includes an error message and possibly an error code in Control Builder.

The syntax for a typical Control Builder error message is as follows:

Connection to device is not open EPKS_E_CL_NOCONN(6L.101.3326)

In this syntax, the error code is the last four digits in the message or **3326**.

Please refer to the *Control Builder Error Codes Reference* book for applicable error code information.

Checking front panel display and LEDs

Check the Series C FIM 4-character display and Series C FIM/IOTA light emitting diodes (LEDs) indications and compare results with data in the *Series C FIM/IOTA Display and LED Descriptions* section of this book.

Viewing flash log

The Flash.txt log provides a list of firmware updates that have been initiated.

To view the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\bin\Flash.txt.

Viewing release information log

The ReleaseInfo.txt log provides a list of Experion software releases that have been installed on the computer.

To view the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\bin\ReleaseInfo.txt.

Viewing trace log

The TraceLogRs.txt log provides a list of definitions for strings associated with *breadcrumbs* data for given hardware components. The *breadcrumbs* provide a way to trace operations leading up to an event.

To view the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\bin\TraceLogRs.txt.

Checking version and revision log

The ver_rev.txt log provides a list of components by model number with software version/revision along with compatible Experion software release(s).

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\bin\ver_rev.txt.

Checking server point build log

The SvrPtBld.txt log provides list of process (CB) points built in the server database.

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\temp\SvrPtBld.txt.

Checking server point build error log

The svrptblderr.txt log provides list of any errors associated with process (CB) points built in the server database

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS\Engineering Tools\system\temp\svrptblderr.txt.

Checking error log

The Errlog_n.txt log provides a running list of Control Builder detected errors in chronological order. The n represents any number that is assigned to the most recent log.

To check the log, navigate to this file location on the server: C:\Documents and Settings\All Users\Application Data\Honeywell\Experion PKS\Errlog_n.txt.

Viewing link schedule log

The FFLinkSchedule20000nnnPROJECT.txt log provides a description of the link schedule associated with the given FFLink in Control Builder. The nnn represents the identifying portion of the FFLink block name configured in Control Builder. The link schedule must be viewed in Control Builder for this log to be generated. The log includes the following information categories.

- **NATURAL SCHEDULE**: Shows the execution and publish entries in the schedule by Control Module (CM) point.
- **SCHEDULE STATS**: Shows the schedule statistics for the natural schedule.
- **FB LIST**: Lists the function blocks added to the list of scheduled objects.
- **PUB LIST**: Lists the connections added to the list of scheduled objects.
- **CHAIN LIST**: Lists the chains and the elements contained in each chain.

- **OPTIMIZED SCHEDULE**: Shows the execution and publish entries in the schedule with the optimized schedule start times by Control Module (CM) point.
- **SCHEDULE STATS**: Shows the schedule statistics for the optimized schedule.
- **SCHEDULE FACTORS**: Shows the schedule factors for the *optimized* schedule compared to the *natural* schedule.

To view the log, navigate to this file location on the server: C:\Documents and Settings\All Users\Application Data\Honeywell\Experion PKS\FFLinkSchedule20000nnnPROJECT.txt.

Checking fieldbus library manager log

The FieldbusLibraryManagerLog.txt log provides list of information captured during generation of given fieldbus device type from vendor DD file in chronological order.

To check the log, navigate to this file location on the server: C:\Documents and Settings\All Users\Application Data\Honeywell\Experion PKS\FieldbusLibraryManagerLog.txt.

Using CTool to capture diagnostic data

REFERENCE - INTERNAL

Please refer to the *Using CTool to capture diagnostic data* section in the *Control Hardware Troubleshooting and Maintenance Guide* for more information.

Series C FIM Self-Test Diagnostic Codes

The following table lists the self-test diagnostic codes that the Series C FIM steps through upon power up.

Test Code	Target Device(s)	Failure Modes	Function
T000			Signals start of selftest.
Display Diagnostics			
T001	Display Test	Bad display, Incorrect assembly	Steps through horizontal/vertical bars, then flashes all segments of the display through stages of brightness. For this test to work, CPU, Flash ROM, address bus, data bus, and some I/O ports must be working. The module status LED is set to GREEN, OFF, then

Test Code	Target Device(s)	Failure Modes	Function
			RED for the duration of self test.
CPU Diagnostics			
T002	CPU Mask Revision Test	Obsolete or unexpected new, CPU	This test is done before the ROM tests, because a wrong CPU could appear to work, but might fail timing tests. The Instruction Cache may be enabled after this test.
T003	CPU Speed Test	Invalid jumper configuration	The software-visible hardware jumpers are verified. This is necessary because the PPC Timebase and Decrementer run at CPU clock /4.
T004	Miscellaneous Speed Test	Register configuration	Checks the value of the system clock mode register to ensure that bus division factor, CPM division factor, the PLL predivision factor, and the PLL multiplication factor are correct.
T010	Boot Image Test	Bad flash memory, Bad instruction cache	The CRC-32 and other parameters of the boot image in ROM are computed and verified. Running this test with the Instruction and Data Cache enabled provides an acceptable SIL-2 cache test because it involves comparing against a known value. Data cache is disabled after this test.
T012	First Stuck Interrupt Test	Bad wiring, Improper PLD configuration, Improper FPGA configuration	Checks for interrupts stuck on at power-on. Clears the interrupt pending register (SIPNR) and then re-checks if any interrupts are still asserted.
T013	Main RAM Initialization	None	Initializes all main RAM locations with 0 while error detection functionality is turned off
Main RAM Diagnostics			

Test Code	Target Device(s)	Failure Modes	Function
T020	Main RAM Data Bus Test	Stuck high and Stuck low data bus lines caused by shorts and unconnected data lines	Tests the data bus lines from the CPU to the SDRAM for stuck high and stuck low bus lines.
T021	Main RAM Address Bus Test	Stuck high and stuck low address bus lines caused by shorts and unconnected data lines	Tests the address bus lines from the CPU to the SDRAM for stuck high and stuck low bus lines.
T022	Main RAM Device Test	Bad memory device	Confirms that each bit in every memory location can be set to either a 1 or 0 and can maintain that value for a period of time. This diagnostics consists of writing 0 and 0xFFFFFFFF, walking 1s, and walking 0s to all memory locations in SDRAM.
T030	Main RAM Byte Access Test	Bad memory controller configuration, Faulty wiring between CPU and SDRAM	Confirms that Main RAM can support byte-sized memory accesses.
T031	MainRAM HalfWord Access Test	Bad memory controller configuration, Faulty wiring between CPU and SDRAM	Confirms that main RAM can support half-word (16-bit) sized memory access.
T032	Main RAM Word Access Test	Bad memory controller configuration, Faulty wiring between CPU and SDRAM	Confirms that main RAM can support word (32-bit) sized memory access.
T033	Main RAM Double	Bad memory	Confirms that main RAM can

Test Code	Target Device(s)	Failure Modes	Function
	Word Access Test	controller configuration, Faulty wiring between CPU and SDRAM	support double word (64-bit) sized memory access.
T038	Main RAM SBE Test	Improper EDAC configuration in BR and/or MCR register, improper wiring from CPU to SDRAM	Confirms that the memory controller's error detection and correction functionality works properly with respect to single bit errors. This includes ensuring that single bit errors are corrected and recorded in the TESCR1 register's ECC1 bit.
T039	Main RAM MBE Test	Improper EDAC configuration in BR and/or MCR register, improper wiring from CPU to SDRAM	Confirms that the memory controller's error detection and correction functionality works properly with respect to double bit errors. This includes ensuring that double bit errors are correctly flagged in the TESCR1 register's ECC2 bit.
Cache Diagnositcs			
T040	Instruction Cache Test	Bad CPU, Instruction cache not activated	Confirms that executing code with instruction cache turned on speeds up execution time. Significantly faster execution time will indicate that the instruction cache works as expected.
T045	Data Cache Test	CPU, Data cache is not activated	Confirms that executing code with the data cache turned on speeds up execution time when there are repeated data accesses to the same memory location. Significantly faster execution time will indicate that the data cache works as expected.
T046	Motorola Data Cache Test	Bad CPU, Data cache not activated	Tests the data cache as rigorously as it can be tested. Vendor wrote the source code for this test.

Test Code	Target Device(s)	Failure Modes	Function
EEPROM Diagnostics			
T050	EEPROM Read Test	Bad wiring between CPU and EPROM, Invalid port configuration	Confirms that we can read from EEPROM. (This does not include validating the contents of EEPROM.)
T051	EEPROM Write Protect Test	Bad wiring between CPU and EPROM,Invalid port configuration	Verifies that the write protect line defaults to the protected state
Thermometer Diagnostics			
T052	Thermometer Test	Bad wiring from CPU to thermometers, Bad thermometers, invalid ambient temperature, invalid IO port configuration	SPI Digital Thermometers- Verify that data and clock lines can be set to both states. Verify that chip select lines can be set to both states. Read temperature from all digital thermometers, verify acceptable values (acceptable operating temperature range -10 to +70 degrees C indicates parts are working).
Miscellaneous Diagnostics			
T065	EEPROM Checksum Test	Bad EEPROM, Unprogrammed EEPROM, Bad wiring to CPU	Calculates the checksum of the factory data stored in EEPROM.
T066	PHY Interface Test	Stuck high or low PHY configuration lines	Wiggles MDIO and MDCLK lines.
T067	Second Stuck Interrupt Test	Bad FPGA image, faulty wiring	Checks for any asserted interrupt lines. Once the FPGA has been configured there should initially be no asserted interrupts from the application board (IRQ1).
Fieldbus Network Number 1 Diagnostics			
T110	Communication Controller Data	Bad CPLD image, Bad interboard	Writes a set of values to a communication controller register

Test Code	Target Device(s)	Failure Modes	Function
	Path	connections, Invalid memory controller configuration	in which all the bits are writeable. The values written will include walking ones and walking zeros on all data bus bits.
T111	Initialize Communication Controller	Bad memory controller configuration, Faulty wiring, Bad communication controller chip	Initializes the Fieldbus communication controller by setting up its internal registers with a base configuration used during power-on self-test. Since this involves writing to the communication controller's registers, this function implicitly tests the data path from the host processor to the communication processor via the local bus. This test is performed on Fieldbus network number 1.
T112	Communication Controller Token Timer	Bad communication controller	Checks the communication controller's internal token timer. This timer is used to indicate the token holding time remaining.
T113	Communication Controller Node Timer	Bad communication controller	Checks the communication controller's internal node timer.
T114	Communication Controller Watch Timer Test	Invalid communication controller configuration	Checks the communication controller's internal watch timer. The watch timer expires and generates an interrupt, if there is no bus activity after a specificed period of time.
T115	Communication Controller Gap Timer	Bad communication controller	Checks the communication controller's internal gap timer. The gap timer governs the minimum amount of time between two transmissions.
T116	FIFO Threshold Interrupt Generation	Bad communication controller	Checks the implementation of the FIFO threshold interrupts (TFI and RFI). These interrupts are generated when the TX and/or RX FIFOs exceed a specified limit.

Test Code	Target Device(s)	Failure Modes	Function
T117	Communication Controller Interrupt Generation	Bad communication controller interrupt aggregation	Checks to see that an interrupt generated by the communication controller can be seen by the host processor.
T118	Communication Controller Loopback	Bad communication controller	Puts the communication controller into internal loopback mode and tests to see that data can be transmitted and received with internal loopback on. This test is performed on Fieldbus network number 1
Fieldbus Network Number 2 Diagnostics			
T120	Communication Controller Data Path	Bad CPLD image, Bad interboard connections, Invalid memory controller configuration	Writes a set of values to a communication controller register in which all the bits are writeable. The values written will include walking ones and walking zeros on all data bus bits.
T121	Initialize Communication Controller	Bad memory controller configuration, Faulty wiring, Bad communication controller chip	Initializes the Fieldbus communication controller by setting up its internal registers with a base configuration used during power-on self-test. Since this involves writing to the communication controller's registers, this function implicitly tests the data path from the host processor to the communication processor via the local bus. This test is performed on Fieldbus network number 2.
T122	Communication Controller Token Timer	Bad communication controller	Checks the communication controller's internal token timer. This timer is used to indicate the token holding time remaining.
T123	Communication Controller Node Timer	Bad communication controller	Checks the communication controller's internal node timer.

Test Code	Target Device(s)	Failure Modes	Function
T124	Communication Controller Watch Timer Test	Invalid communication controller configuration	Checks the communication controller's internal watch timer. The watch timer expires and generates an interrupt, if there is no bus activity after a specificed period of time.
T125	Communication Controller Gap Timer	Bad communication controller	Checks the communication controller's internal gap timer. The gap timer governs the minimum amount of time between two transmissions.
T126	FIFO Threshold Interrupt Generation	Bad communication controller	Checks the implementation of the FIFO threshold interrupts (TFI and RFI). These interrupts are generated when the TX and/or RX FIFOs exceed a specified limit.
T127	Communication Controller Interrupt Generation	Bad communication controller interrupt aggregation	Checks to see that an interrupt generated by the communication controller can be seen by the host processor.
T128	Communication Controller Loopback	Bad communication controller	Puts the communication controller into internal loopback mode and tests to see that data can be transmitted and received with internal loopback on. This test is performed on Fieldbus network number 2
Fieldbus Network Number 3 Diagnostics			
T130	Communication Controller Data Path	Bad CPLD image, Bad interboard connections, Invalid memory controller configuration	Writes a set of values to a communication controller register in which all the bits are writeable. The values written will include walking ones and walking zeros on all data bus bits.
T131	Initialize Communication Controller	Bad memory controller configuration, Faulty wiring, Bad communication	Initializes the Fieldbus communication controller by setting up its internal registers with a base configuration used during power-on self-test. Since this

Test Code	Target Device(s)	Failure Modes	Function
		controller chip	involves writing to the communication controller's registers, this function implicitly tests the data path from the host processor to the communication processor via the local bus. This test is performed on Fieldbus network number 3.
T132	Communication Controller Token Timer	Bad communication controller	Checks the communication controller's internal token timer. This timer is used to indicate the token holding time remaining.
T133	Communication Controller Node Timer	Bad communication controller	Checks the communication controller's internal node timer.
T134	Communication Controller Watch Timer	Invalid communication controller configuration	Checks the communication controller's internal watch timer. The watch timer expires and generates an interrupt, if there is no bus activity after a specificed period of time.
T135	Communication Controller Gap Timer	Bad communication controller	Checks the communication controller's internal gap timer. The gap timer governs the minimum amount of time between two transmissions.
T136	FIFO Threshold Interrupt Generation	Bad communication controller	Checks the implementation of the FIFO threshold interrupts (TFI and RFI). These interrupts are generated when the TX and/or RX FIFOs exceed a specified limit.
T137	Communication Controller Interrupt Generation	Bad communication controller interrupt aggregation	Checks to see that an interrupt generated by the communication controller can be seen by the host processor.
T138	Communication Controller Loopback	Bad communication controller	Puts the communication controller into internal loopback mode and tests to see that data can be transmitted and received with

Test Code	Target Device(s)	Failure Modes	Function
			internal loopback on. This test is performed on Fieldbus network number 3.
Fieldbus Network Number 4 Diagnostics			
T140	Communication Controller Data Path	Bad CPLD image, Bad interboard connections, Invalid memory controller configuration	Writes a set of values to a communication controller register in which all the bits are writeable. The values written will include walking ones and walking zeros on all data bus bits.
T141	Initialize Communication Controller	Bad memory controller configuration, Faulty wiring, Bad communication controller chip	Initializes the Fieldbus communication controller by setting up its internal registers with a base configuration used during power-on self-test. Since this involves writing to the communication controller's registers, this function implicitly tests the data path from the host processor to the communication processor via the local bus. This test is performed on Fieldbus network number 4.
T142	Communication Controller Token Timer	Bad communication controller	Checks the communication controller's internal token timer. This timer is used to indicate the token holding time remaining.
T143	Communication Controller Node Timer	Bad communication controller	Checks the communication controller's internal node timer.
T144	Communication Controller Watch Timer	Invalid communication controller configuration	Checks the communication controller's internal watch timer. The watch timer expires and generates an interrupt, if there is no bus activity after a specificed period of time.
T145	Communication Controller Gap Timer	Bad communication controller	Checks the communication controller's internal gap timer. The gap timer governs the minimum

Test Code	Target Device(s)	Failure Modes	Function
			amount of time between two transmissions.
T146	FIFO Threshold Interrupt Generation	Bad communication controller	Checks the implementation of the FIFO threshold interrupts (TFI and RFI). These interrupts are generated when the TX and/or RX FIFOs exceed a specified limit.
T147	Communication Controller Interrupt Generation	Bad communication controller interrupt aggregation	Checks to see that an interrupt generated by the communication controller can be seen by the host processor.
T148	Communication Controller Loopback	Bad communication controller	Puts the communication controller into internal loopback mode and tests to see that data can be transmitted and received with internal loopback on. This test is performed on Fieldbus network number 4.
Pre-OS Diagnostics			
T400	Application Image Validation	Missing Application Image, Incomplete load of Application Image, Bad flash memory	Checks for valid Application Image stored in flash memory.
T401	Unused Port Pin	Unanticipated usage of spare port pins	Confirms that unused parallel port pins on the CPU can be driven into both high and low states.
T511	End of POST Diagnostics		Confirms that self test is successful and there is a valid Application Image stored in Flash.

Fixing Common Problems

This section identifies some common problems and describes how you might fix them.

Loss of power

The power supply has failed or the main power source has been shut down or is experiencing a *brownout* or *blackout* condition.

Diagnostic Check	• The 4-character display on the Series C FIM and LEDs on the Series C FIM and the IOTA are off. • In the **Monitoring** tab, the FIM4 icon turns red.
Cause 1	Main power source has been disconnected or shut down either manually or temporarily by *brownout* or *blackout* condition.
Solution	Re-connect the main power source or turn it On or wait for temporary *brownout* or *blackout* condition to pass.
Cause 2	The 24 Vdc power supply failed or power cable has been disconnected or failed.
Solution	Replace the 24 Vdc power supply or re-connect/replace the power cable.
Cause 3	Power fuse opens on IOTA.
Solution	Replace power fuse.

Poweron Self-Test Fault

The Series C FIM poweron selftest has detected a fault.

Diagnostic Check	A self-test diagnostic code remains on display and the poweron selftest does not continue.
Cause	The Series C FIM poweron selftest has detected a failure that does not allow operation to continue.
Solution	Try shorting the reset pads on the IOTA to re-start the Series C FIM. If error persists, replace the Series C FIM. See *Replacing Failed Series C FIM CC- or CU-PFB401* for details.

Fatal ECC error

The Series C FIM software has detected a fatal Error Checking and Correction (ECC) condition that can be a multiple-bit error or excessive single bit errors in main Random Access memory (RAM).

Diagnostic Check	• The 4-character display on the Series C FIM displays FAIL. • In the **Monitoring** tab, the FIM4 icon turns red.
Cause	The Series C FIM software has detected a failure that does not allow operation to continue. There can be many causes for a failure including hardware, use the Series C Platform Analyzer to capture a crash block for the device to determine the possible cause before proceeding.
Solution	Try shorting the reset pads on the IOTA to re-start the Series C FIM. If error persists, replace the Series C FIM. See *Replacing Failed Series C FIM CC- or CU-PFB401* for details. Check the Trace log for breadcrumbs that occurred prior to the event. See *Viewing trace log* and *Using CTool to capture diagnostic data* for more information. Provide the results of the trace log to our Solutions Support Center (SSC) for analysis.

Background Diagnostic failure

The Series C FIM detects failure in system integrity checks, such as Watch Dog Timer (WDT), error detection circuits, Field Programmable Gate Array (FPGA) readback, microprocessor static configuration registers, and Read Only Memory (ROM) checksum.

Diagnostic Check	• The 4-character display on the Series C FIM displays FAIL. • In the **Monitoring** tab, the FIM4 icon turns red.
Cause	The Series C FIM software has detected a background diagnostic failure that does not allow operation to continue.
Solution	Try recycling power to the Series C FIM. If error persists, replace the Series C FIM. See *Replacing Failed Series C FIM CC- or CU-PFB401* for details. Check the Trace log for breadcrumbs that occurred prior to the event. See *Viewing trace log* and *Using CTool to capture diagnostic data* for more information. Provide the results of the trace log to our Solutions Support Center (SSC) for analysis.

No IP address

The Series C FIM cannot obtain an IP address.

Diagnostic Check	The 4-character display on the Series C FIM displays -bp-.
Cause 1	A valid IP address has not been configured.
Solution	Configure a valid IP address through the **System Preferences** dialog in Control Builder.
Cause 2	The BOOTP server service is not running.
Solution	Check that the Experion PKS BOOTP Server service is running through the **Services** dialog of the Control Panel application.
Cause 3	The BOOTP server was not re-started after a change in the Device Index or System Preferences.
Solution	Stop and start the Experion PKS BOOTP Server service through the **Services** dialog of the Control Panel application.
Cause 4	Communication problems.
Soulution	Check your FTE network and cables for proper operation.

No time reference

The Series C FIM cannot obtain a system time reference.

Diagnostic Check	The 4-character display on the Series C FIM displays -ts-.
Cause 1	Valid (S)NTP Server addresses have not been configured.
Solution	Configure valid (S)NTP Primary and Secondary Server IP addresses through the **System Preferences** dialog in Control Builder.
Cause 2	A prior release version of the BOOTP Server is running.
Solution	Be sure that only an R300 Experion PKS BOOTP Server service is running on the servers. Only the latest BOOTP server can provide all of the necessary information for the current and prior-release devices.
Cause 3	Not all FTE System Preferences for all Experion clusters have been configured with the same information.
Solution	Check that all FTE System Preferences are configured identically through the **System Preferences** dialog in Control Builder.

Cause 4	The designated (S)NTP Servers are not properly configured and operating.
Solution	See Setting up time synchronization in the *Supplementatry Installation Tasks Guide* for information.
Cause 5	The BOOTP Server service on both servers was not re-started after a configuration change.
Solution	Re-start the BOOTP Server service on both servers through the **Services** dialog of the Control Panel application..
Cause 6	The Windows Time Service on Experion servers was not re-started after NTP server was enabled.
Solution	Be sure the Windows Time Service is running on Experion servers. Normally, NTPSetup takes care of this.

Isolated (lonely) Node

For redundant Series C FIM pair, Fault Tolerant Ethernet (FTE) communications with partner and FTE network are lost.

Diagnostic Check	• The Primary Series C FIM negotiates whether to initiate a switchover or not. if the Secondary was known to be in better condition than the Primary at the time of fault determination, then the Primary should fail so the Secondary will switchover. But, the new Secondary (old Primary) still cannot restore FTE communications. • The Secondary Series C FIM should reboot once, in an attempt to restore communications. The Primary Series C FIM will be able to report the problem in the Secondary. If the Secondary cannot restore FTE communications, it would still be able to resynchronize over the redundancy link and be a partially functional backup.
Cause 1	The Secondary Series C FIM is defective.
Solution	Replace the Secondary Series C FIM that initiated switchover when fault was detected. See Replacing Failed Series C FIM CC- or *CU-PFB401* for details. If Secondary Series C FIM synchronizes after replacement, the removed Series C FIM is defective. Otherwise, go to Cause 2.
Cause 2	The Primary Series C FIM is defective
Solution	Replace the Primary Series C FIM. See Replacing Failed Series C FIM CC- or *CU-PFB401* for details.

	If you can command synchronization after replacement, the removed Series C FIM is defective. Otherwise, go to Cause 3.
Cause 3	The redundant IOTA is defective.
Solution	Replace the redundant IOTA. See *Replacing IOTA CC-or CU-TFB401, CC-or CU-TFB411, CC- or CU-TFB402, or CC- or CU-TFB412* for details. If the Series C FIM pair synchronize after IOTA replacement, The removed IOTA is defective. Other wise, go to Cause 4.
Cause 4	There is a software problem.
Solution	Contact our Solution Support Center (SSC).

Device Index setting does not match displayed value

The Device Index number shown in the Series C FIM's 4-character display does not match the value set on the IOTA's binary-coded decimal rotary switches.

Diagnostic Check	May be detected as a soft failure that triggers an alarm for the operator. The Device Index setting is important only at initial application startup, when the Device Index number is used to obtain the IP Address. The Device Index and IP Address are saved in non-volatile memory and reused on subsequent application startups.
Cause 1	Someone changed the setting on the binary-coded decimal rotary switches on the IOTA.
Solution	Change binary-coded decimal rotary switches to correct setting. See *Resetting device index and IP address* for more information.
Cause 2	The Primary or non-redundant Series C FIM is defective
Solution	Replace the Primary or non-redundant Series C FIM. See *Replacing Failed Series C FIM CC- or CU-PFB401* for details. If Device Index switch setting matches Device Index number in 4-character display upon Series C FIM power up, the removed Series C FIM is defective. Otherwise, go to Cause 3.
Cause 3	The redundant or non-redundant IOTA is defective.
Solution	Replace the IOTA. See *Replacing IOTA CC-or CU-TFB401, CC-or CU-TFB411, CC- or CU-TFB402, or CC- or CU-TFB412* for details. If Device Index switch setting matches Device Index number in 4-character display upon Series C FIM power up, the removed IOTA is

	defective. Otherwise, go to Cause 4.
Cause 4	There is a software problem.
Solution	Contact our Solution Support Center (SSC).

Duplicate Device Index setting

The FTE subsystem detects duplicate Device Index settings in separate nodes.

Diagnostic Check	All nodes will stop tracking cable status for the detected duplicate Device Index value. Communications will continue and will not impact system performance until there is a cable fault. This fault will also be detected by the FTE System Management Tool. A duplicate Device Index could cause a duplicate IP Address. In most cases, the duplicate IP Address would be detected first and prevent the FTE diagnostic messages from being sent.
Cause 1	Have set binary-coded decimal rotary switches on separate IOTA's to same value.
Solution	Change binary-coded decimal rotary switches setting on one of the IOTA's to a unique value. See *Resetting device index and IP address* for more information.

Device Index value is zero upon power up

The Series C FIM's 4-character display shows a Device Index value of zero (#000).

Diagnostic Check	Be sure binary-coded decimal rotary switches on the IOTA were not intentionally set to zero to initiate a Device Index/IP Address reset.
Cause 1	Binary-coded decimal rotary switches set to zero.
Solution	Change binary-coded decimal rotary switches to correct setting. See *Resetting device index and IP address* for more information.
Cause 2	The Primary or non-redundant Series C FIM is defective
Solution	Replace the Primary or non-redundant Series C FIM. See *Replacing Failed Series C FIM CC- or CU-PFB401* for details. If Device Index switch setting matches Device Index number in 4-character display upon Series C FIM power up, the removed Series C FIM is defective. Otherwise, go to Cause 3.

Cause 3	The redundant or non-redundant IOTA is defective.
Solution	Replace the IOTA. See *Replacing IOTA CC-or CU-TFB401, CC-or CU-TFB411, CC- or CU-TFB402, or CC- or CU-TFB412* for details. If Device Index switch setting matches Device Index number in 4-character display upon Series C FIM power up, the removed IOTA is defective.

Getting Further Assistance

Other troubleshooting sources

The following table lists other documents and sections that contain troubleshooting information for other Experion subsystems.

All of these documents are available from Knowledge Builder. Some documents are also supplied as part of Station Help. For documents that can be accessed directly from this page, click on the link, otherwise look for the document within Knowledge Builder. The XXX represents the current Experion release number.

Document/Section	Comments
Experion RXXX > Reference	There is a separate interface reference for each type of controller other than the Process Controller; for example, the *ASEA Interface Reference*. Most of these references contain an interface-specific troubleshooting section.
Experion RXXX > Reference > TPS Integration Guide > Troubleshooting	Troubleshooting an integrated system that uses Experion "TPS Integration" option.
Experion RXXX > Reference > Control Builder Error Codes Reference	Describes error codes generated from within Control Builder.
Experion RXXX > Troubleshooting and Maintenance > Control Hardware Troubleshooting and Maintenance Guide	The main repository for troubleshooting, maintenance and repair of Process Controllers.
Experion RXXX > Configuration > DeviceNet Implementation Guide > Troubleshooting DeviceNet Status Failures	Describes error codes generated from DeviceNet Interface Board.
Experion RXXX > Installation and Upgrades > Fault Tolerant Ethernet Bridge Implementation	Troubleshooting FTE bridges.

Document/Section	Comments
Guide > Service > Troubleshooting	
Experion RXXX > Installation and Upgrades > Fault Tolerant Ethernet Installation and Service Guide > Troubleshooting FTE Nodes	Troubleshooting FTE nodes.
Experion RXXX > Reference > Honeywell TDC 3000 Data Hiway Interface Reference > TDC error codes and Troubleshooting	Troubleshooting TDC 3000 Hiway problems.
Experion RXXX > Configuration > Qualification and Version Control System User Guide > QVCS Troubleshooting	Troubleshooting QVCS.
Experion RXXX > Operations > SafeView User's Guide > Appendix D - SafeView Error Messages	Describes the meaning of SafeView configuration errors.
Experion RXXX > Reference > Server Scripting Reference > Server scripting error messages	Describes the meaning of error messages in the server log specific to server scripting.
Experion RXXX > Reference > System Management Configuration Guide > Troubleshooting System Management	Describes the meaning of System Management Configuration errors.
Experion RXXX> Reference > System Management Configuration Guide > Troubleshooting SES	Describes the meaning of SES Configuration errors.
Experion RXXX > Reference > System Management Configuration Guide > Troubleshooting SPS	Describes the meaning of SPS Configuration errors.
Experion RXXX > Planning and Design > Planning, Installation, and Service for WS360	Troubleshooting workstation nodes used in Experion and TPN.

Guidelines for requesting support

If you cannot resolve a problem using this guide, you can request support from your Honeywell Solution Support Center.

When requesting support, please supply as many relevant details as possible, including:

- **Short summary of the problem**
- **Product Name and release.**

- **Recent changes**, such as upgrades/service packs, to Experion software, Windows or other applications.
- **Subsystem and its version/build**, if the problem relates to a particular subsystem, such as Station or Quick Builder.
  If the problem relates to **Display Builder**, please specify whether it is HMIWeb Display Builder (for HMIWeb displays) or Display Builder (for DSP displays).
- **Operating system, variant and service pack**, for example "Windows 2000 Server, SP5".
- **Instructions on how to reproduce the problem**. If the problem is reproducible, please supply step-by-step instructions — the more detailed the steps the better.
- **Diagnostic package which contains any relevant logs**.

Fieldbus Integration with Experion Control Reference

This section provides conceptual type information about the integration of fieldbus devices with Experion control functions. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Series C FIM handles data integration

The Series C Fieldbus Interface Module (FIM4) functions as a dual network bridge using a dynamic data cache to facilitate the exchange of data between the control communication network and the fieldbus H1 links. It supports both publish/subscribe and client/server communications methods to implement control connections between Control Builder function blocks and fieldbus function blocks. The Series C FIM capability includes converting Control Builder value-status structure to fieldbus value-status by mapping similar fields to one another and defaulting others. This means the Experion system can monitor fieldbus control functions, fully integrate with control functions, or provide a combination that includes using fieldbus based control as backup for selected Experion system control functions.

The Series C 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.

About link object

The link object represents one fieldbus H1 link. It consists of Network Management and System Management information, along with several application parameters related to link functions such as startup, shutdown, and diagnostics.

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 Schedule;
- 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 device objects

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 Series C 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, Series C FIM retrieves MIB information, SM directory, and NM directory. Knowledge of these directories allows Series C 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 VFD objects

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 Series C 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 Series C 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.

Type creation makes integration possible

The fieldbus device Type function included with Control Builder makes integrating fieldbus function blocks with Control Builder ones possible. The **Type** function reads the manufacturer's DD for the fieldbus device and creates a device block type that is stored in the Control Builder **Library** tab. The device block type includes the device's fieldbus function blocks, so it can be configured and integrated with control strategies through Control Builder.

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 Experion Control Builder application. The Proportional, Integral, Derivative (PID) function block is a typical regulatory control type function block.

The following simplified functional diagram shows 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).

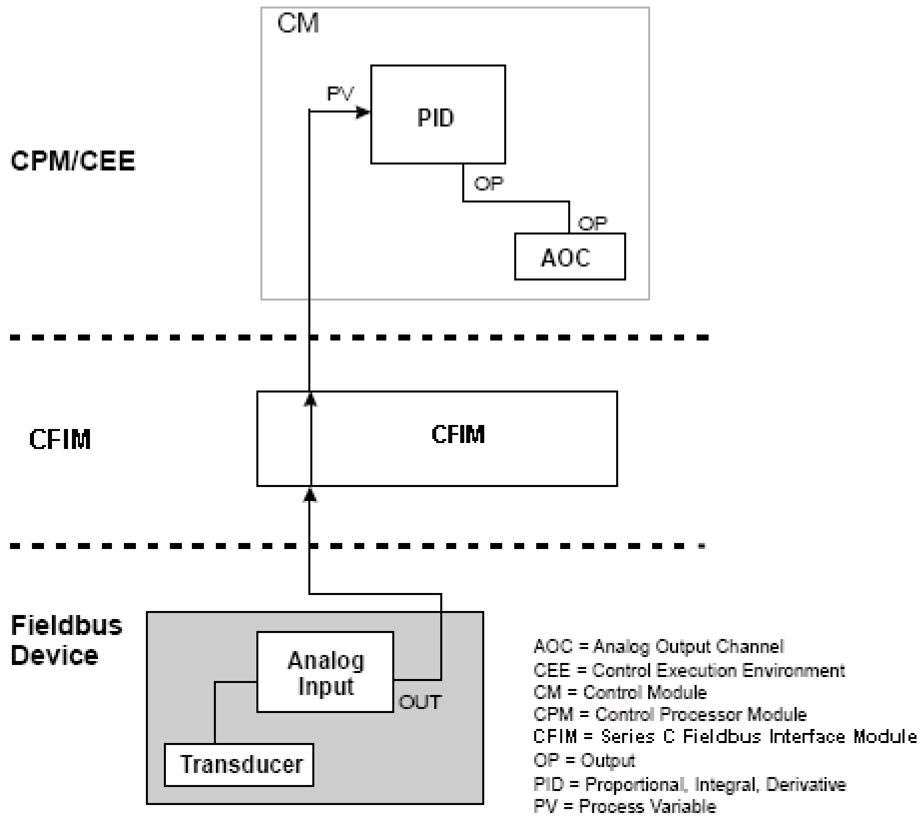


Figure 25 Integration of fieldbus device analog input signal with Control Builder control strategy

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 Control Builder PV with status parameter. It interprets the value portion in fieldbus terms and converts it to Control Builder representation. The floating-point representation is identical, in most cases, but the fieldbus +/-infinity value must be converted to a Control Builder representation.

If the fieldbus status byte indicates "BAD", the value must be converted to Not a Number (NaN) for Control Builder representation. The fieldbus data quality of good, bad, and uncertain is mapped to the appropriate Control Builder parameter of PVSTS, PVSTSFL.NORM, PVSTSFL.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 Control Builder blocks.

The fieldbus data sub status indicator maps only the limited number of sub status conditions that have corresponding Control Builder indications. Note that the handshaking provided by the sub status 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 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 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 Control Builder regulatory control type function block.

The following simplified functional diagram shows 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.

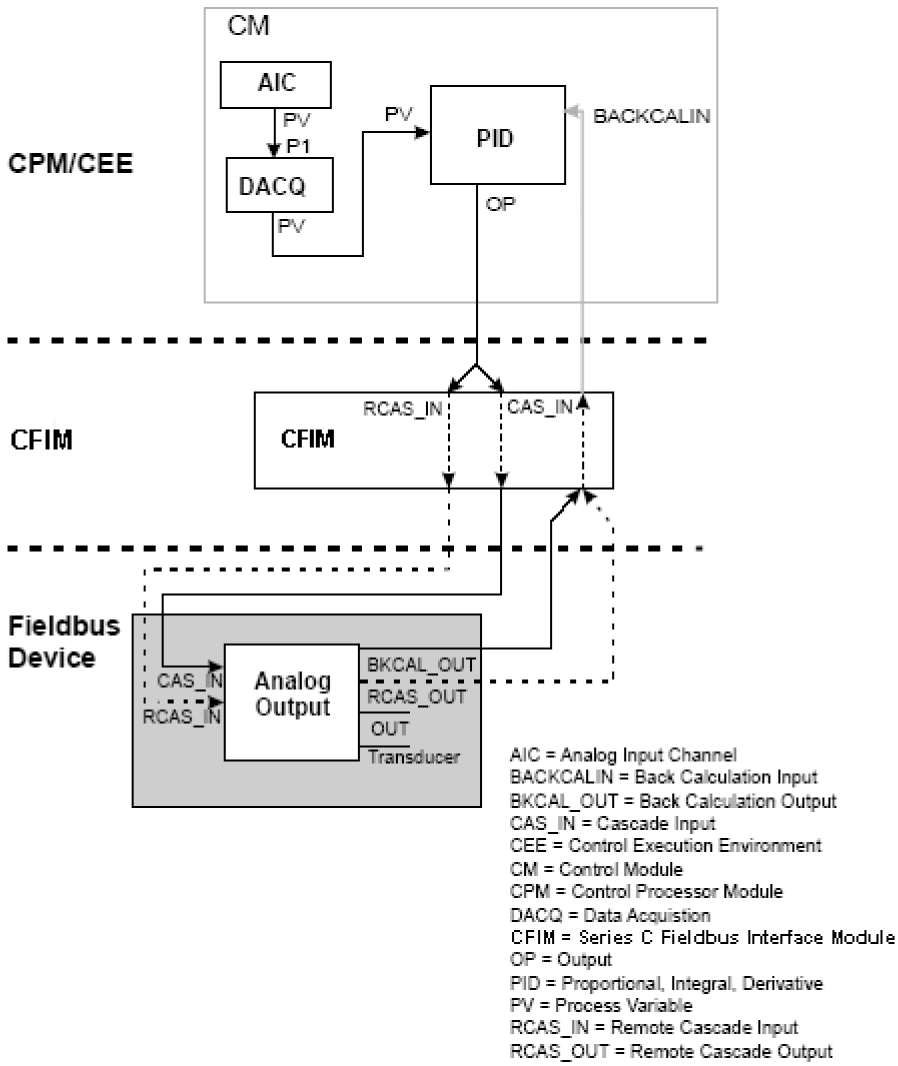
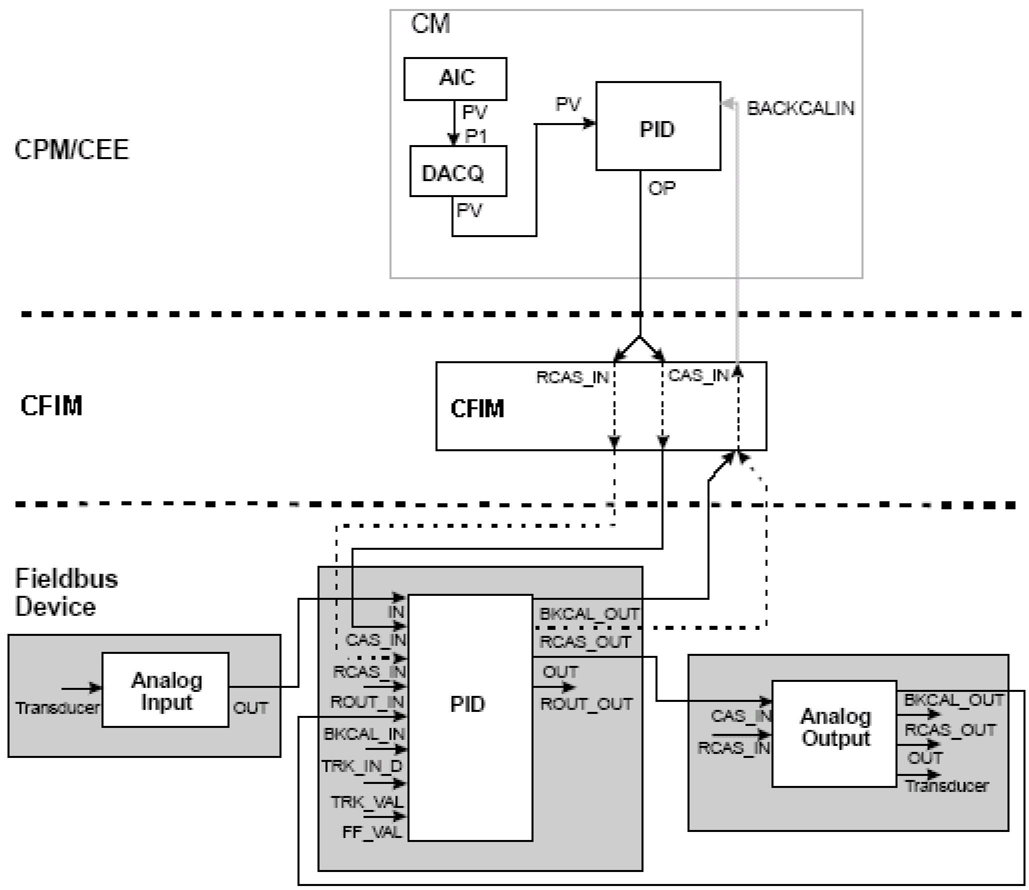


Figure 26 Integration of fieldbus device analog output signal with Control Builder control strategy

The following simplified functional diagram shows 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.



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
CFIM = Series C 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

Figure 27 Integration of fieldbus device PID control with Control Builder control strategy

Fieldbus Analog Output or PID data manipulation

When the Output 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 Cont OP with status parameter to the fieldbus data structure (DS-65) of the CAS_IN parameter. It interprets the value portion in Experion terms and converts it to fieldbus representation. The floating-point representation is identical, in most cases, but the Experion +/-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".

WARNING

The Fieldbus Foundation specifications do not clearly state the expectations of fieldbus devices for handling Not-a-Number (NaN) values. This means that some may behave as the user desires and others may not.

If you compute a data type float + status value that uses data structure (DS-65), which includes all **FOUNDATION** Fieldbus inputs, outputs and some contained parameters, a resulting NaN value passed outbound through the Series C FIM sets the status element to BAD and changes the value element from NaN to 0.0 (zero). If you compute a data type float value, which includes many contained values such as alarm trip points, set points or output limits, gains, and filter time values, a resulting NaN value that is to be passed outbound through the FIM4 will not be written and usually results in retention of the previous value. If the device would not behave in the application as you desire, you must add appropriate function blocks or logic to test for and replace the NaN with an acceptable value.

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 Experion terms.

ATTENTION

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

Like the Series C FIM, the output agent supports both publish/subscribe and client/server communication methods. The publish/subscribe method allows the Series C FIM to appear as a fieldbus device on the H1 link. The Series C 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 Series C FIM.

The client/server method allows the Series C FIM to appear as a computing device on the H1 link. The Series C 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 Series C FIM.

The client/server method also allows the Series C FIM to function in a Direct Digital Control (DDC) mode or the Remote Out mode in fieldbus terms. In this case, the Series C 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 Series C FIM.

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

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

The Control Builder control initialization indicators map only to the limited number of sub status 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 Experion Control Builder application.

The following simplified functional diagram shows 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).

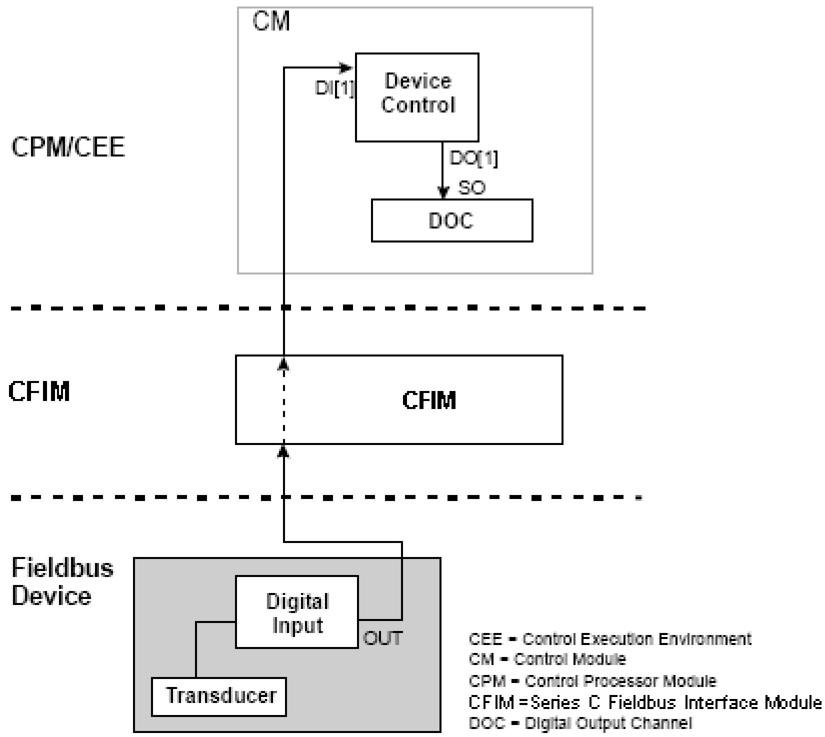


Figure 28 Integration of fieldbus device digital input signal with Control Builder control strategy

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 Experion 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 Control Builder 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 Control Builder 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 Control Builder function block like Device Control to the input of a Discrete Output block in a fieldbus device residing on an H1 link.

The following simplified functional diagram shows 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.

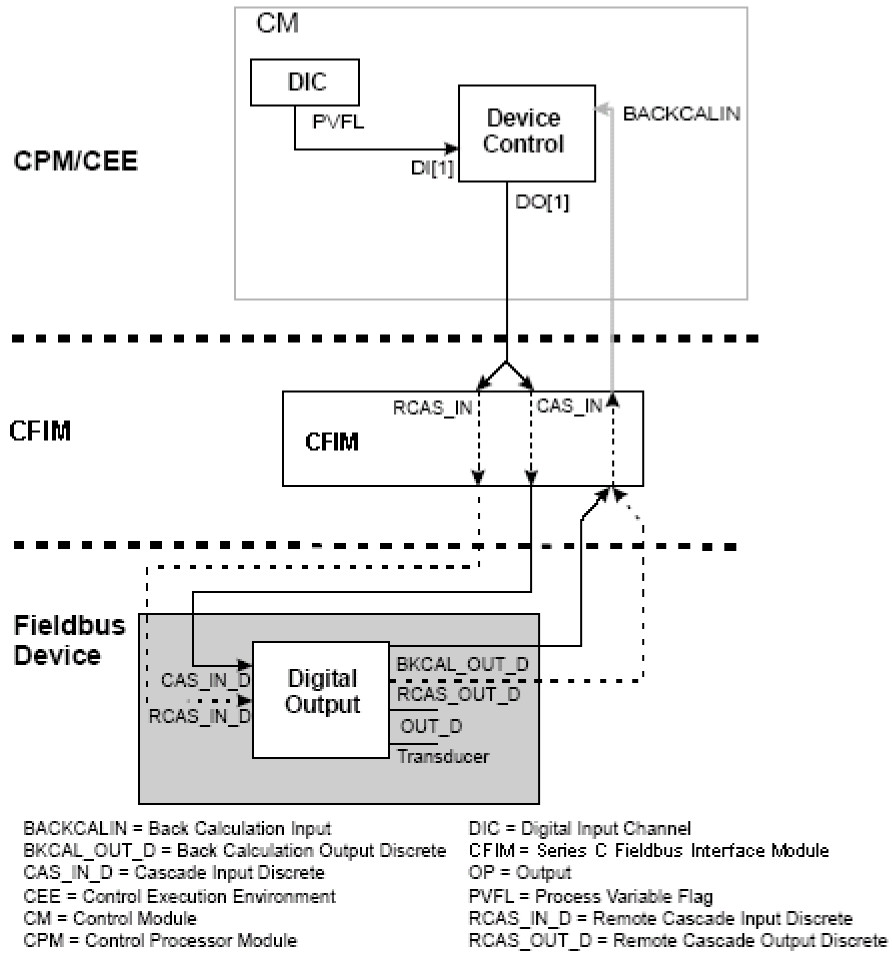


Figure 29 Integration of fieldbus device digital output signal with Control Builder control strategy

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 Control Builder DO[n] with status parameter to the fieldbus data structure (DS-66) of the CAS_IN_D parameter. It interprets the value portion in Control Builder 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 Control Builder 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 Control Builder data quality is converted to fieldbus data quality. The Control Builder Good indication is represented as fieldbus Good (Cascade).

The Control Builder control initialization indicators map only to the limited number of sub status 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 FIM4.

1. Analog process value into the FIM4.
2. Discrete process value into the FIM4.
3. Analog process output from the FIM4.
4. Discrete process output from the FIM4.
5. Analog process output from the FIM4 with backcalculation feedback.
6. Discrete process output from the FIM4 with backcalculation feedback.

Note that types 5 and 6 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 FIM4. The FIM4 can also monitor (subscribe to) the data published between the functions blocks of these fieldbus devices.

A word about SCM parameter interaction

Sequential Control Module (SCM) Steps and Transitions can write to any contained parameter of a fieldbus function block. The value of a contained parameter can be configured, set by an operator, higher level device, or calculated. For example, the GRANT_DENY parameter can be used to check access rights, the MODE_BLK parameter can be used to request mode changes, and the RCAS_IN and RCAS_OUT parameters can be used to control set points in the Remote Cascade mode.

If the contained parameter has a DS-65 or DS-66 data structure, you must use multiple SCM Step outputs to write to the different fields in the parameter's data structure. For example, if you use a Step output to write to the SP parameter of a fieldbus PID block, when it is in Auto mode, you will need Step outputs for the related parameter attributes of value, status, sub status and limits, as defined by its structure. A typical SCM write SP scenario would include the following actions.

- Transition checks if Target MODE of the fieldbus block is AUTO.
- Step output sets Target MODE to AUTO, if required.
- Step output sets SP value and GOOD (Non Cascade) status.
- Optional Step output to set SP Limits.
- Optional Transition to verify that the value got stored in the fieldbus device.

The SCM will issue a fail alarm, if the store fails for any reason.

ATTENTION

Avoid SCM configurations that would push a NaN (Not a Number) value as an output to a fieldbus device. If this is not possible, be sure to test the field device to observe how it reacts to the possible NaN value.

Fieldbus status data details

According to Fieldbus Foundation 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 sub status 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 sub status field indication.

Bit	Quality	Sub status, if Quality field is				Limits
		BAD	***UNCERTAIN***	***GOOD (Non-Cascade)****	***GOOD (Cascade)****	
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 Acknowledge (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	Unacknowledged Block Alarm	Not Selected (NS)	

Bit	Quality	Sub status, if Quality field is				Limits
		BAD	***UNCERTAIN***	***GOOD (Non-Cascade)****	***GOOD (Cascade)****	
5		No Comm-unication, with Last Usable Value	Engineering Unit Range Violation	Unacknow-ledged Advisory Alarm	Do Not Select (DNS)	
6		No Comm-unication, 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)	

* The Good (non-cascade) sub status is used by output connections for fieldbus blocks such as Analog Input and Discrete Input. The Good (cascade) sub status is used by output connections for fieldbus blocks such as PID. Both of these sub statuses are converted to the single system data quality of Good.

Fieldbus status indications

The following tables include a list of possible display indications associated with a given fieldbus status and definitions of related sub statuses.

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
BAD	Non-Specific	No Limits	B	Bad
BAD	Non-Specific	Low Limit	B	Bad,LowLimit

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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	<blank>	<blank>
GOOD (Non-Cascade)	Non-Specific	Low Limit	L	Low Limit
GOOD (Non-Cascade)	Non-Specific	High Limit	H	High Limit
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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	<blank>	<blank>
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
GOOD (Cascade)	Request Initialization (IR)	No Limits	R	ReqInit
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

Fieldbus Status Byte			1-Char Suffix	12-Char Independent Status Field
Quality	***Sub status***	***Limits***		
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	<blank>	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

Sub status Definitions for Quality Status

Quality	Sub status	Description
Bad	Non-Specific	There is no specific reason why the value is bad. Used for propagation.

Quality	Sub status	Description
	Configuration Error	Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect.
	Not Connected	Set if this input is required to be connected and is not.
	Device Failure	Set if the source of the value is affected by a device failure.
	Sensor Failure	Set it the device can determine this condition. The Limits define which direction has been exceeded.
	No Communication, with last usable value	Set if this value had been set by communication, which has now failed.
	No Communication, with no usable value	Set if there has never been any communication with this value, since it was last Out of Service.
Uncertain	Non-Specific	There is no specific reason why this value is uncertain. Used for propagation.
	Last Usable Value	Whatever was writing this value has stopped. (This happens when one disconnects an input.)
	Substitute	Set when the value is written when the block is Out of Service.
	Initial Value	Set when the value of an input parameter is written while the block is Out of Service.
	Sensor Conversion Not Accurate	Set if the value is at one of the sensor limits. The limits define which direction has been exceeded. Also, set if the device can determine that the sensor has reduced accuracy, in which case no limits are set.
	Engineering Unit Range Violation	Set if the value lies outside of the range of values defined for this parameter. The limits define which direction has been exceeded.

Quality	Sub status	Description
	Sub-Normal	Set if value derived from multiple values has less than the required number of Good sources.
Good (Non-Cascade)	Non-Specific	There is no specific reason why the value is good. No error or special condition is associated with this value.
	Active Block Alarm	Set if the value is good and the block has an active Block alarm.
	Active Advisory Alarm	Set if the value is good and the block has an active alarm with a priority less than 8.
	Active Critical Alarm	Set if the value is good and the block has an active alarm with a priority greater than or equal to 8.
	Unacknowledged Block Alarm	Set if the value is good and the block has an unacknowledged Block alarm.
	Unacknowledged Advisory Alarm	Set if the value is good and the block has an unacknowledged alarm with a priority of less than 8.
	Unacknowledged Critical Alarm	Set if the value is good and the block has an unacknowledged alarm with a priority greater than or equal to 8.
Good (Cascade)	Non-Specific	There is no specific reason why the value is good. No error or special condition associated with this value.
	Initialization Acknowledge	The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).
	Initialization Request	The value is an initialization value a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.

Quality	Sub status	Description
	Not Invited	The value is from a block, which does not have a target mode that would use this input. This covers all cases other than Fault State Active, Local Override, and Not Selected. The target mode can be the next permitted mode of higher priority in the case of shedding a supervisory computer.
	Not Selected	The value is from a Control Selector that has not selected the corresponding input. This tells the upper block to limit in one direction, not initialize.
	Do Not Select	The value is from a block that should not be selected by a control selection block, due to conditions in or above the block. As an option, a block may generate this output status if its actual mode is not Auto or not Cascade.
	Local Override	The value is from a block that has been overridden by a local key switch or is a complex AO/DO with interlock logic active. The failure of normal control must be propagated to a PID block for alarm and display purposes. This also implies Not Invited.
	Fault State Active	The value is from a block that has Fault State active. The failure of normal control must be propagated to a PID block for alarm and display purposes. This also implies Not Invited.
	Initiate Fault State	The value is from a block that wants its downstream output block to go to Fault State. This is determined by a block option to initiate Fault State if the status of the primary input and/or cascade input goes Bad.

Control Mode Interaction Reference

This section provides conceptual type information about fieldbus block mode interaction with Experion control modes. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Fieldbus block modes versus control 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. The eight possible modes are described in the About modes of operation paragraph in the

Standard Function Blocks section in this book.

For use within the Experion system, the structure of the MODE_BLK parameter is expanded to add MODE to the existing Actual, Target, Permitted, and Normal modes as outlined in the following table.

Control Mode Structure	Data Type	Description
MODE	Enumeration	Write Only Experion system style mode enumeration MAN, AUTO, CAS, NORMAL, BCAS, NONE
MODE.TARGET	Enumeration	Read/Write Target mode OOS, MAN, AUTO, CAS, RCAS, ROUT
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
MODE.NORMAL	Enumeration	Read/Write Normal mode MAN, AUTO, CAS, RCAS, ROUT OOS is not Normal

The Experion system uses the name MODE instead of MODE_BLK and its enumeration set is identical to that used in fieldbus. Mode is a write only parameter and the FIM4 will reject all reads. The FIM4 captures all writes to MODE and maps valid changes to MODE.TARGET. If the value NORMAL is written to the MODE.TARGET, the FIM4 replaces it with the value from MODE.NORMAL. If a new MODE.NORMAL value is entered, it is validated against the MODE.PERMITTED values. The OOS mode is not an acceptable normal mode.

When a new MODE.PERMITTED value is entered, the FIM4 forces the OOS mode to be permitted. Only a user with an access level of Engineer is allowed to put a block into its OOS mode or to return the block to an operating mode. This means a user must have an access level of Engineer to put a block in Man, Auto, Cas, Rcas, or Normal, if the blocks current target mode is OOS. When a mode is changed, the FIM4 clears the GRANT_DENY.GRANT parameter bits. It is standard Experion system behavior to take a grant back after an operator changes a mode relative to its mode attribute (MODATTR) parameter.

The following table shows how control modes are mapped to fieldbus ones.

Control 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

The following table 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 Fieldbus Foundation 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 from the target mode, and to determine if the particular actual mode allows write access.

Mode	Mode Abbreviation		Priority Interpretation		
	2-Char	***4-Char***	***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

While every 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 below for reference.

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.

Experion 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

Experion 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.

ATTENTION

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.

The Experion system adheres to the following additional rules for setting fieldbus target mode bits for its MODE supported subset of combinations.

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 cannot be set as the target mode. Never set the IMan as the target mode.
LO	This is a Read Only parameter and cannot 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.

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 the figure below.

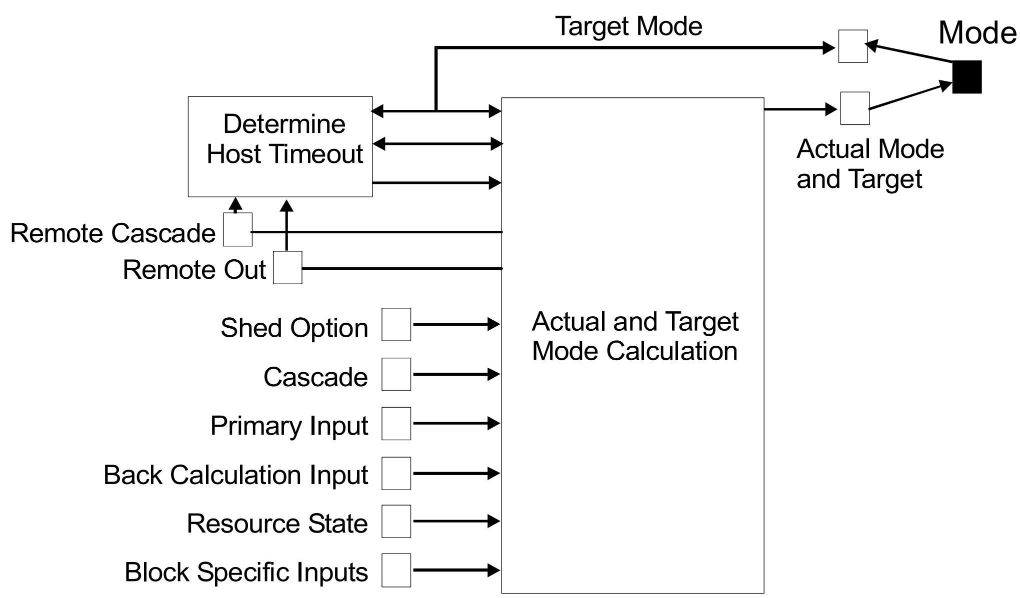


Figure 30 Block mode calculation summary

Mode Change Conditions

The following table provides a list of conditions, which will change the mode in order of priority with Good (Non-Cascade) status on input parameter as the lowest priority.

Input Parameters Status Attributes, Inputs Values and Resource States	Mode Parameter - Target Attribute Value					
	Out of Service (OOS)	Manual (MAN)	Auto-matic (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.	OOS	MAN	AUTO	CAS	Shed based on Shed Option	ROUT
Remote out in has BAD status.	OOS	MAN	AUTO	CAS	RCAS	Shed based on Shed Option
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 executio n or AUTO	N/A

Input Parameters Status Attributes, Inputs Values and Resource States	Mode Parameter - Target Attribute Value					
	Out of Service (OOS)	**Manual (MAN)**	**Auto-matic (Auto)**	**Cascade (CAS)**	**Remote Cascade (RCAS)**	**Remote Output (ROUT)**
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 or mode as determined by Shed Option
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 or MAN	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 – fault state 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

Input Parameters Status Attributes, Inputs Values and Resource States	Mode Parameter - Target Attribute Value					
	Out of Service (OOS)	Manual (MAN)	Auto-matic (Auto)	Cascade (CAS)	Remote Cascade (RCAS)	Remote Output (ROUT)
Tracking is enabled, initiated and track override manual is Disabled.	OOS	MAN	LO	LO	LO	LO
Fault State 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 or evaluate d mode	MAN or evaluated mode	MAN or evaluate d mode	MAN or evaluate d mode	MAN or evaluate d mode
Resource state as reflected in the resource block parameter resource state is Standby.	OOS	OOS	OOS	OOS	OOS	OOS

*Shed to the next lowest priority that is permitted and supported by the available inputs.

Access control through GRANT_DENY parameter

Every fieldbus function block includes a GRANT_DENY parameter. It allows users to Grant and Deny access permission to groups of function block parameters by other devices. The following table summarizes the attributes for the Grant and Deny elements of the parameter.

Element and Bit	Attribute	Description
GRANT_DENY.GRANT:0	Program Operations	Grants permission to higher level device to change the target mode, set point or output of a block, depending on block mode.
GRANT_DENY. GRANT:1	Tune	Grants permission to higher level device to change tuning parameters of the block.

Element and Bit	Attribute	Description
GRANT_DENY.GRANT:2	Alarm	Grants permission to higher level device to change alarm parameters of the block
GRANT_DENY.GRANT:3	Local Operations	Grants permission to a local operator's panel or hand-held device to change target mode, set point, or output of the block, depending on block mode.
GRANT_DENY.DENY:0	Program Operations	Turns off Program Operations permission grant.
GRANT_DENY.DENY:1	Tune	Turns off Tune permission grant.
GRANT_DENY.DENY:2	Alarm	Turns off Alarm permission grant.
GRANT_DENY.DENY:3	Local Operations	Turns off Local Operations grant.

The Grant selections are mutually exclusive. If you select Program Operations, the Local Operations selection is turned off. If Local Operations is selected, neither an operator nor a high level device (program) has the right to modify a parameter value of the block. Note that access by another function block is always granted and is independent of GRANT_DENY selections. Also, an operator can always change the MODE of a block with Program or Local Operations Granted. When an operator changes the MODE, it automatically resets all four Grants to off.

The GRANT_DENY parameter will not track the mode of an associated SCM.

Link and Block Schedules Reference

This section provides conceptual type information about fieldbus Link Active Scheduler and Link Master functions. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Topic
Link Active Scheduler (LAS) and Link Master
Link Schedule
Function block execution schedule

Link Active Scheduler (LAS) and Link Master

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 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 FIM4 is Link Master capable and supports both primary and backup link schedules. It is designated as the primary Link Master. It is always assigned the lowest network address.

Upon startup or failure of the existing LAS, the Link Master capable 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. The Link Master capable device with the lowest address usually wins the bid. 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 the 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.

Most Link Master capable devices can only support one sub schedule and can**not** support blocks that are running two different macrocycles. Each macrocycle includes one sub schedule. Each sub schedule includes sequences that specify the start time of each element in the sequence. The elements are actual publications that initiate schedule operations. Devices have predefined limits on the number of sub schedules that can exist per LAS and the number of sequences that can exist in a sub schedule. If the number of sub schedules and/or sequences in the current schedule (backup LAS) exceed the Link Master capable device's predefined limits, a load of the device through Control Builder will generate load errors identifying sub schedule problems. In this case, the user must reconfigure the device to become a **Basic** one through Control Builder, and restart the device to initiate the change. Otherwise, if the FIM4 fails, the potential for a "no schedule" condition exists. Since the schedule is cleared from the Link Master capable device upon the detection of load errors, it can**not** function as the backup LAS without a schedule.

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. A Link Schedule is provided for the FIM4 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 Experion system or Experion system 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 FIM4.
- 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.

Note that users will be able to add extra time before and after the execution of a fieldbus function block through the Pre-execution delay (PREXEGAP) and the Post execution delay (POSTEXEGAP) parameters that appear on the block configuration forms in Control Builder. This means users can adjust the schedule to achieve such actions as forcing multiple AI blocks to execute at the same time (even though they must publish serially).

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 Fieldbus Foundation mandated feature. They may run asynchronously.

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

1. Preprocessing — Snap of parameter values
2. Execution — Function block outputs are determined
3. Post processing — 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 the following figure.

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 2 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.

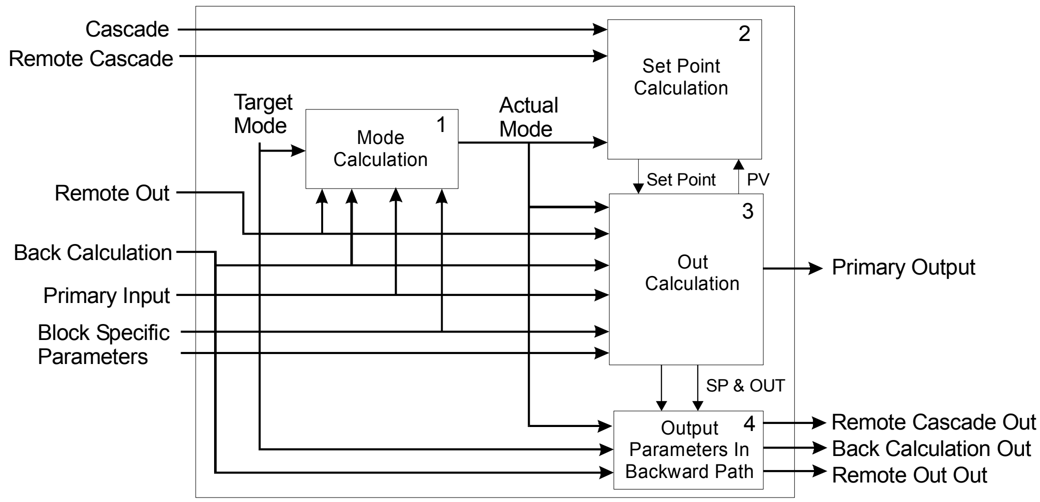


Figure 31 Algorithm execution phase sequence

Tags, Addresses, and Live List Reference

This section provides conceptual type information about fieldbus tags, addresses, and live list functions. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Topic
Tag and address assignments
A word about fieldbus address assignments in Control Builder
Live List and Uncommissioned Devices

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.

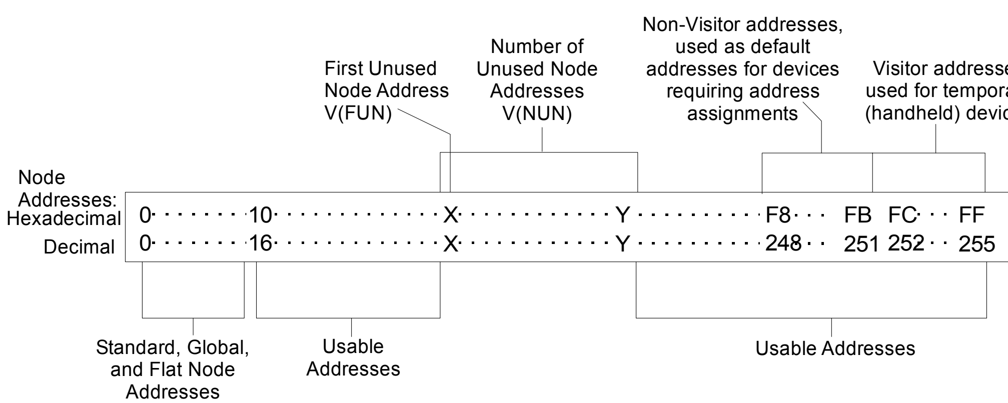


Figure 32 Summary of address allocations for fieldbus devices

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.

Note that the Experion system will show address assignments in Decimal notation rather than Hexadecimal in Control Builder configuration forms and Station displays. The address range in Decimal is 0 to 255 and 0 to FF in Hexadecimal.

A word about fieldbus address assignments in Control Builder

The Fieldbus Interface Module will be given 16 as its assigned address. The preferred address range for fieldbus devices is 20 to 40 decimal (14 to 28 hexadecimal). This provides the most optimized default network range where the first unused node address (FUN) is 41decimal (29 hexadecimal) and the number of unused addresses (NUN) is 206. We recommend that the fieldbus device designated as the backup Link Master be given 21decimal (15 hexadecimal) as its assigned address.

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 FIM4 constantly monitors the live list for each fieldbus link 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 FIM4 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.

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

Name	Description	Data Type	Access
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 FIM4 must be configured and loaded through Control Builder before you can view it and its links through the **Monitoring** tab of Control Builder.

Notification Scheme Reference

This section provides conceptual type information about how fieldbus alarms relate to Experion system alarms. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

Fieldbus versus Experion Alarm Priorities

The fieldbus alarms will be closely integrated with the existing Experion notification system. The Experion Server will handle FIM4 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. Function blocks that support alarming also allow each different alarm condition to be disabled. It is always good to check that the disables are set appropriately.

Fieldbus devices use 0 to 15 as numeric priorities for alarm reporting. Priority 0 clears any alarm status, clearing reports and preventing new ones from being issued. Priority 1 clears any alarm reports and prevents new ones from being issued, but allows the alarm status to continue to be observed in the ALARM_SUM and alarm object status. Priorities 2 through 7 are considered *advisory* and priorities 8 through 15 are considered *critical* in fieldbus. Experion alarms use Journal, Low, High and Urgent as priorities with a sub-priority of 0 to 255. The following table shows how fieldbus priorities are mapped to Experion priorities and severities.

Fieldbus Alarm Priority	Experion Alarm Priority	Experion Alarm Severity
0	(Can never be seen by FIM4 or above)	(Can never be seen by FIM4 or above)
1	(Can never be seen by FIM4 or above)	(Can never be seen by FIM4 or above)
2 BLOCK_ERR bit 14 (power-up) BLOCK_ERR bit 15 (Out-	Journal (Event System Only)	2

Fieldbus Alarm Priority	Experion Alarm Priority	Experion Alarm Severity
of-Service)		
2 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 process alarms are associated with process variable conditions and they are reported as process alarms into the Experion system. 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 Experion notification system. The following table 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.

ATTENTION

Only the first alarm in an overlapping series of alarms associated with BLOCK_ERR and XD_ERROR bitstring alarms is reported due to a flaw in the Fieldbus Foundation Specifications regarding the alert state machine. This has been reported to the Fieldbus Foundation through their Action Request system and the Foundation has agreed to permit future devices to issue alerts on changes to the status of any element of these bitstring alarms.

If no more than one of the bits is set at a time, the alarms are reported and cleared properly

TIP

Fieldbus alarm functions do not support rate of change (ROC) alarms. ROC alarms can only be generated in applications that use Control Builder Data Acquisition blocks for input signal conditioning.

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

Enumeration	Description	Alarm/Event Type
BLOCK	BLOCK_ERR: 0 (Other (LSB)	FFOTHER (Unspecified Error)
BLOCK	BLOCK_ERR: 1 (Block Configuration Error)	FFBLKCFG (Block Config Error)
BLOCK	BLOCK_ERR: 2 (Link Configuration Error)	FFLNKCFG (Link Config Error)
BLOCK	BLOCK_ERR: 3 (Simulate Active)	FFSIMACT (Simulation Activ)
BLOCK	BLOCK_ERR: 4 (Local Override)	FFLO (Local Override)
BLOCK	BLOCK_ERR: 5 (Dev Fault State Set)	FFFLSAFE (Device Fault State)
BLOCK	BLOCK_ERR: 6 (Dev Needs Maintenance Soon)	FFDEVNMS (Maintain Device Soon)
BLOCK	BLOCK_ERR: 7 (I/P Failure or PV BAD Status)	FFINFL (Device Input Failure)
BLOCK	BLOCK_ERR: 8 (O/P Failure)	FFOUTFL (Device Output Fail)
BLOCK	BLOCK_ERR: 9 (Memory Failure)	FFMEMFL (Memory Failure)
BLOCK	BLOCK_ERR: 10 (Lost Static Data)	FFLSTDTA (Static Data Lost)
BLOCK	BLOCK_ERR: 11 (Lost NV Data)	FFLNVDTA (Non-Vol Data Lost)
BLOCK	BLOCK_ERR: 12 (Readback Check Failed)	FFRBCKFL (Readback Check Fail)
BLOCK	BLOCK_ERR: 13 (Dev Needs Maintenance Soon)	FFDEVNMS (Maintain Device Now)
BLOCK	BLOCK_ERR: 14 (Power Up)	FFPWRUP (Powered-Up)

Enumeration	Description	Alarm/Event Type
BLOCK	BLOCK_ERR: 15 (Out-Of-Service)	FFOOS (Out-Of-Service)
BLOCK	XD_ERROR: 16 (Unspecified Error)	TBUNSPEC (Unspecified TB Error)
BLOCK	XD_ERROR: 17 (General Error)	TBGENRAL (General Error)
BLOCK	XD_ERROR: 18 (Calibration Error)	TBCALERR (Calibration Error)
BLOCK	XD_ERROR: 19 (Configuration Error)	TBCFGERR (Configuration Error)
BLOCK	XD_ERROR: 20 (Electronics Failure)	TBELECFL (Electronics Failure)
BLOCK	XD_ERROR: 21 (Mechanical Failure)	TBMECHFL (Mechanical Failure)
BLOCK	XD_ERROR: 22 (I/O Failure)	TBIOFL (I/O Failure)
BLOCK	XD_ERROR: 23 (Data Integrity Error)	TBDTAERR (Data Integrity Error)
BLOCK	XD_ERROR: 24 (Software Error)	TBSWERR (Software Error)
BLOCK	XD_ERROR: 25 (Algorithm Error)	TBALGERR (Algorithm Error)
UPDATE	TB Static Data Update Event	TBSTCHNG
UPDATE	FB Static Data Update Event	FBSTCHNG (Static Revision)
WRITE	Write Protect Change Alarm	RBWPCHNG (Write Lock)
UPDATE	Link Object Update Event	FBLOCHNG (Link Object Changed)
UPDATE	Trend Object Update Event	No Action

Fieldbus Technology Overview

This section provides an overview of **FOUNDATION** Fieldbus technology concepts. The following table lists the topics included in this section. If you are viewing this information online, just click the topic title to jump to the information location.

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" field 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 the following figure, 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.

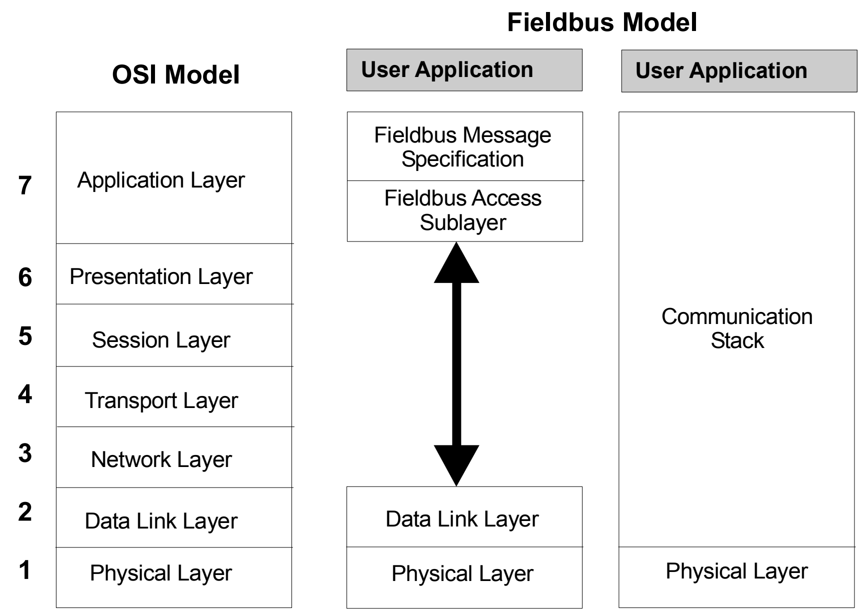
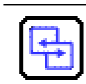


Figure 33 OSI versus Fieldbus communication model.

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.



REFERENCE - INTERNAL

Please refer to *Fieldbus Network Wiring and Installation references* for a list of available documents or visit the Fieldbus Foundation web site http://www.fieldbus.org/ for additional details and ordering data.

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)

Layer	Functional Description	Associated Terms
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)

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 the illustration that follows this table. 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 next 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

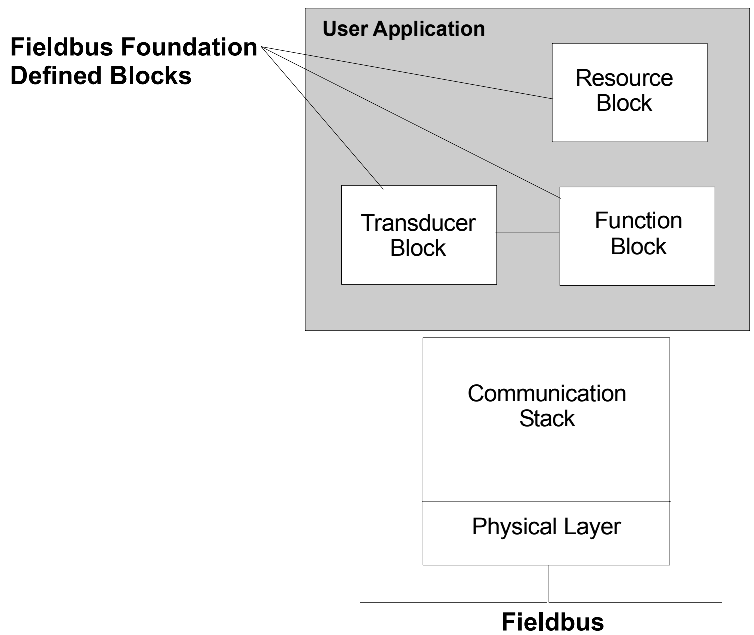


Figure 34 User Application (or Function Block Application Process) based on blocks.

Standard Function Blocks

WARNING

The Fieldbus Foundation specifications do not clearly state the expectations of fieldbus devices for handling Not-a-Number (NaN) values. This means that some may behave as the user desires and others may not.

If you compute a data type float + status value that uses data structure (DS-65), which includes all **FOUNDATION** Fieldbus inputs, outputs and some contained parameters, a resulting NaN value passed outbound through the FIM4 sets the status element to BAD and changes the value element from NaN to 0.0 (zero). If you compute a data type float value, which includes many contained values such as alarm trip points, set points or output limits, gains, and filter time values, a resulting NaN value that is to be passed outbound through the FIM4 will not be written and usually results in retention of the previous value. If the device would not behave in the application as you desire, you must add appropriate function blocks or logic to test for and replace the NaN with an acceptable value.

Blocks for basic control functionality

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 Experion 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.

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 the following figure.

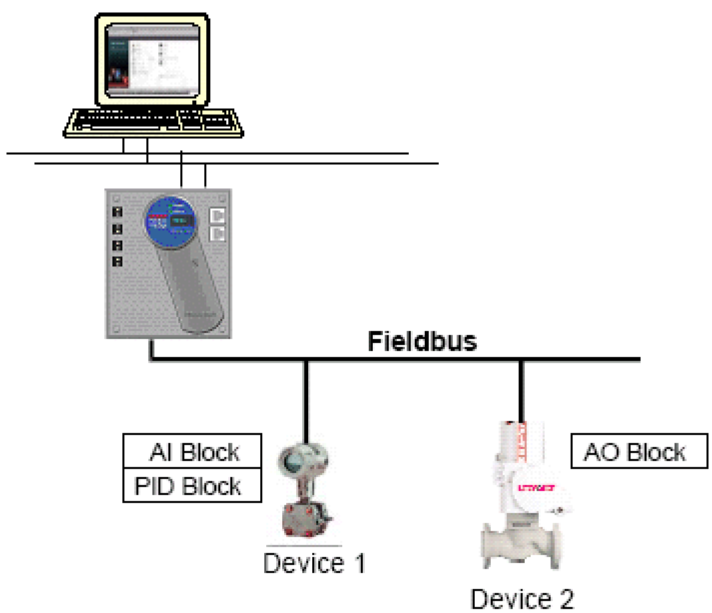


Figure 35 Using Function Blocks in fieldbus devices to form a control loop.

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.

Mode	Abbreviation	Operation Effect
Out of Service	OOS	The block is not being evaluated. The output is maintained at the last value, an assigned fault state value -last value or configured fault state 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.

Mode	Abbreviation	Operation Effect
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

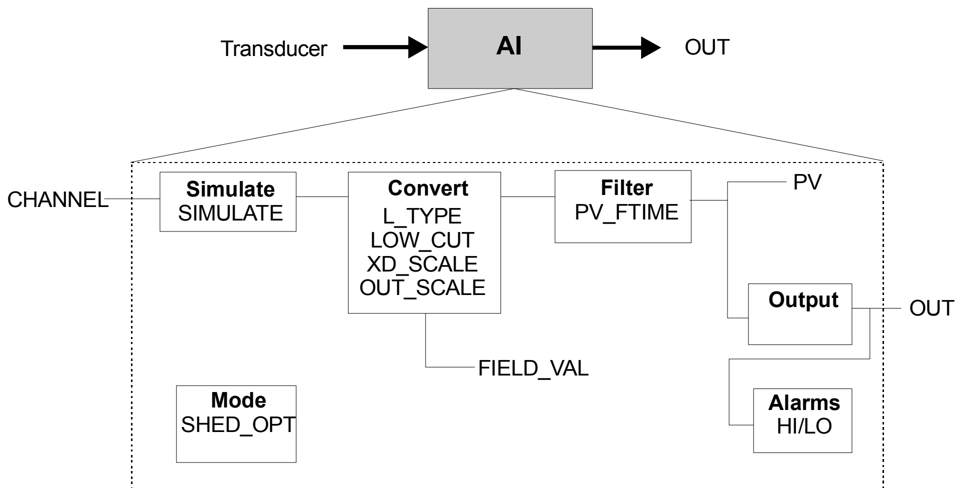


Figure 36 Functional schematic for Analog Input function block.

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 the previous illustration 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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.

Analog Output block

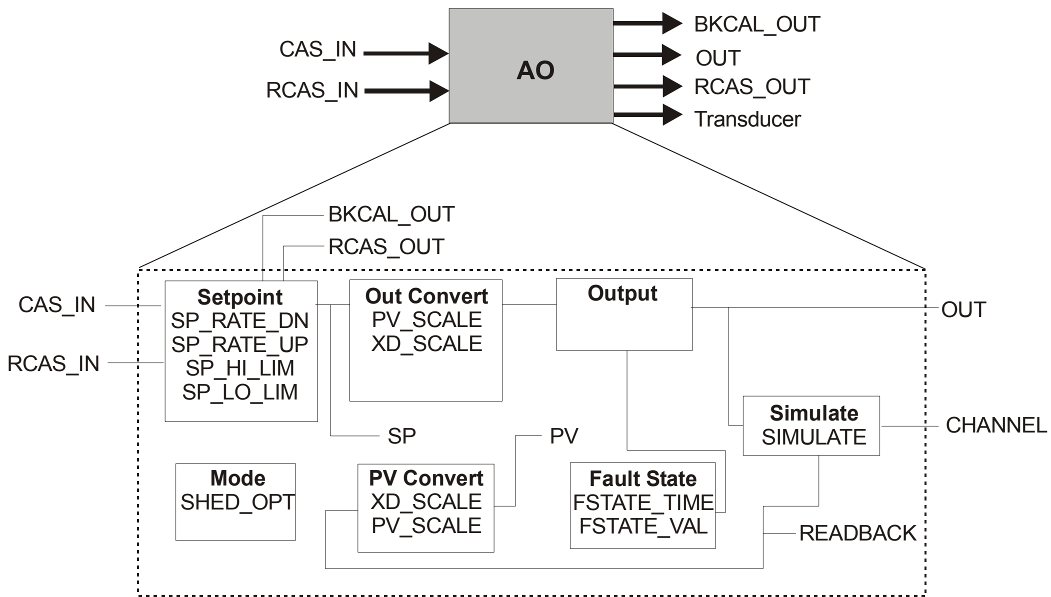


Figure 37 Functional schematic for Analog Output function block.

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 the previous illustration 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	ALERT_KEY BKCAL_OUT BLOCK_ALM BLOCK_ERR CAS_IN CHANNEL FSTATE_TIME FSTATE_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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Bias/Gain block

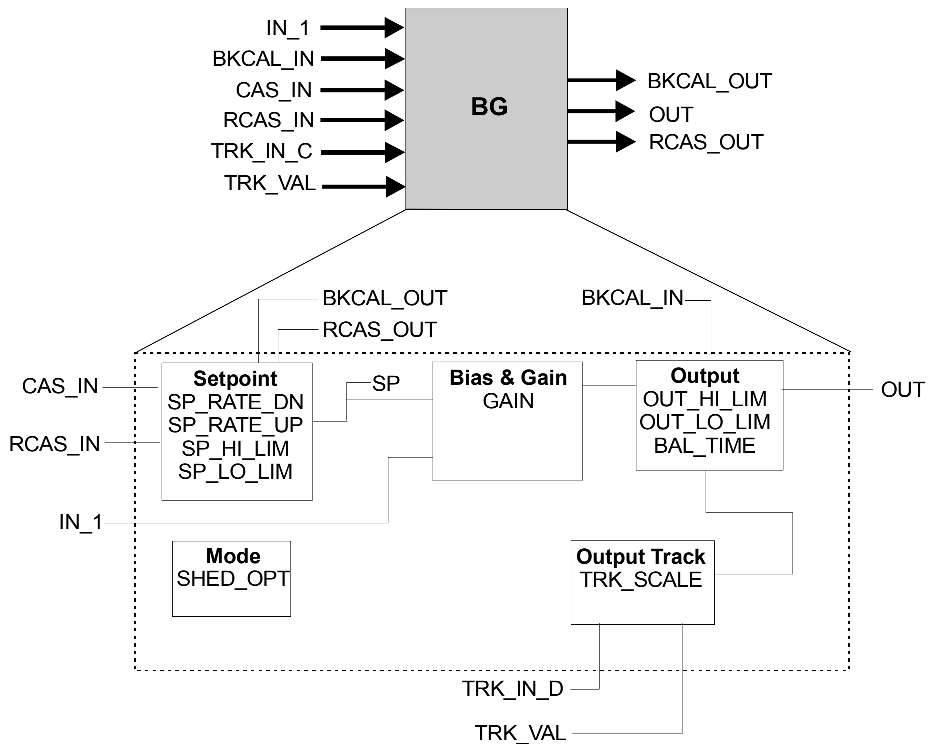


Figure 38 Functional schematic for Bias/Gain function block.

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 the previous illustration 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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Control Selector block

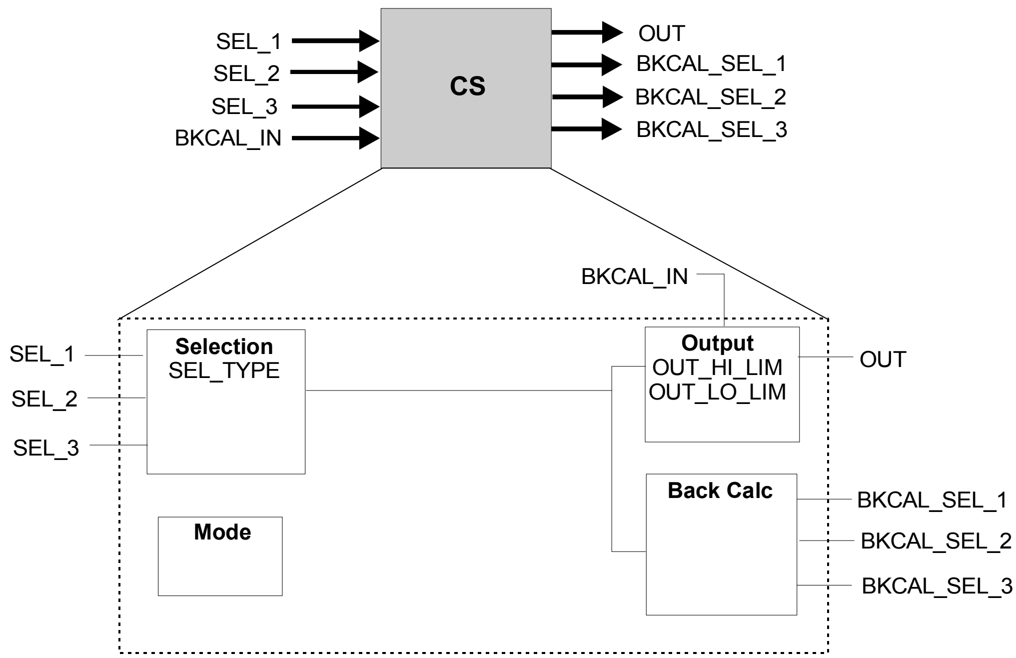


Figure 39 Functional schematic for Control Selector function block.

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 the previous illustration 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 Fault State (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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Discrete Input block

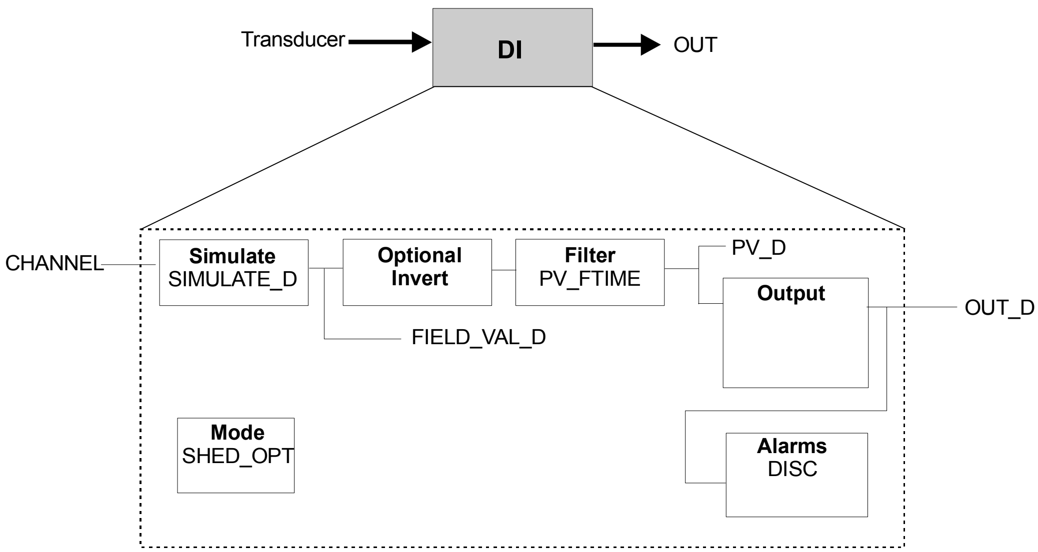


Figure 40 Functional schematic for Discrete Input function block.

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 the previous illustration 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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Discrete Output block

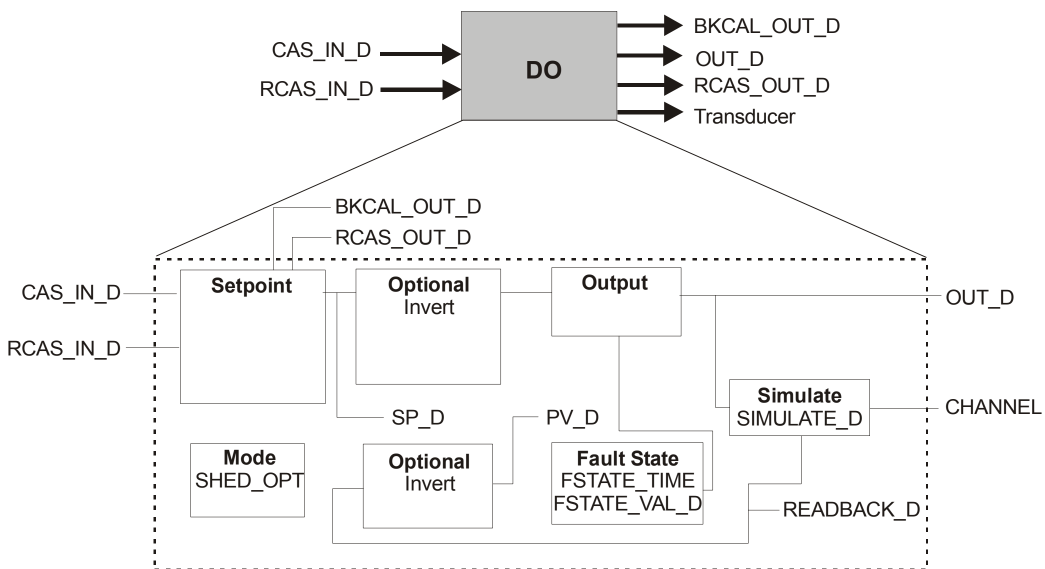


Figure 41 Functional schematic for Discrete Output function block.

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 the previous illustration 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	ALERT_KEY BKCAL_OUT_D BLOCK_ALM BLOCK_ERR CAS_IN_D CHANNEL FSTATE_TIME FSTATE_VAL_D GRANT_DENY IO_OPTS MODE_BLK OUT_D PV_D	PV_STATE RCAS_IN_D RCAS_OUT_D READBACK_D SHED_OPT SIMULATE_D SP_D ST_REV STATUS_OPTS STRATEGY TAG_DESC UPDATE_EVT XD_STATE
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Manual Loader block

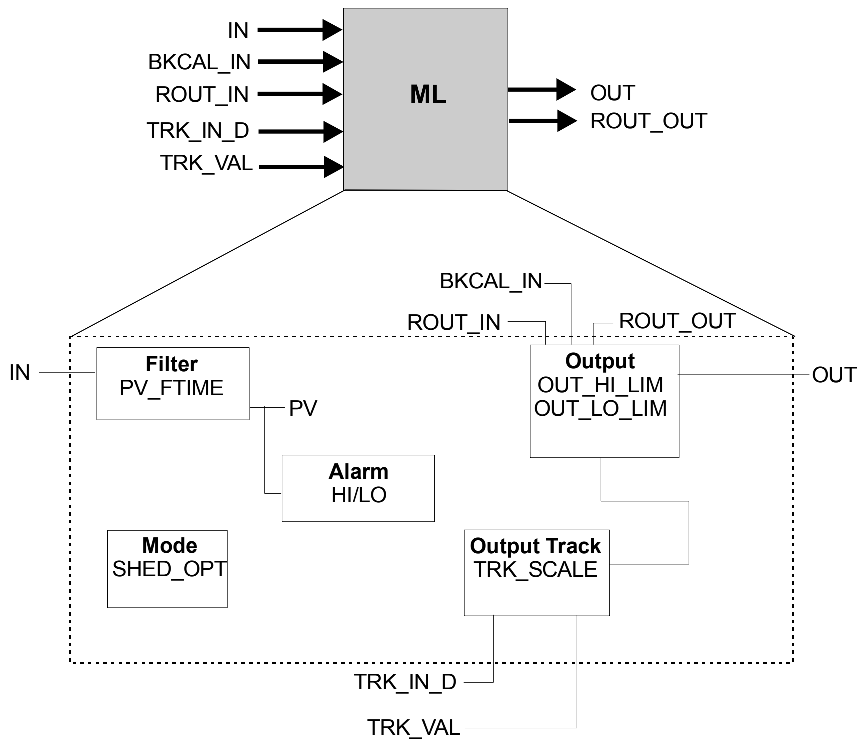


Figure 42 Functional schematic for Manual Loader function block.

Description	The Manual Loader function block output is not set by the block's algorithm. An operator can set its output in the Manual mode or a program in the Remote-Out mode. A functional schematic of the block is shown in the previous illustration 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	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_PRI HI_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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Proportional/Derivative block

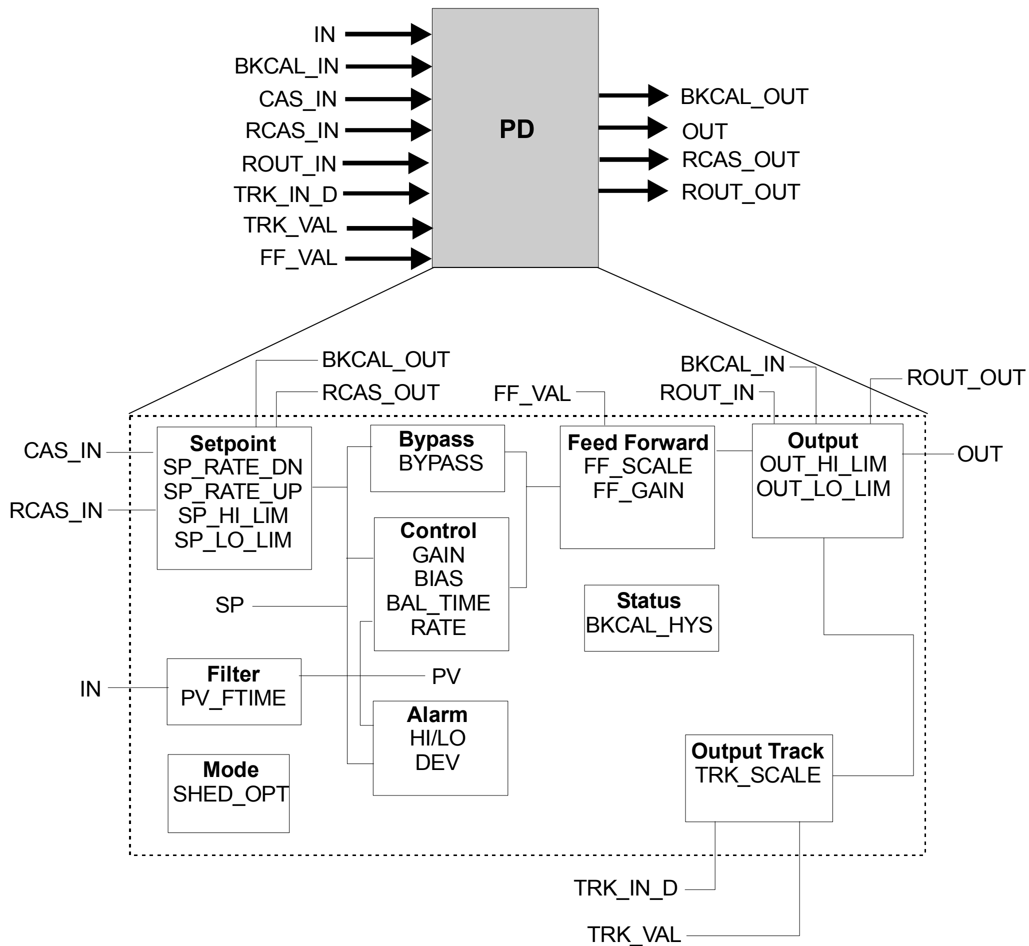


Figure 43 Functional schematic for Proportional/Derivative function block.

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 the previous illustration 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. • 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. • 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. • 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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Proportional/Integral/Derivative block

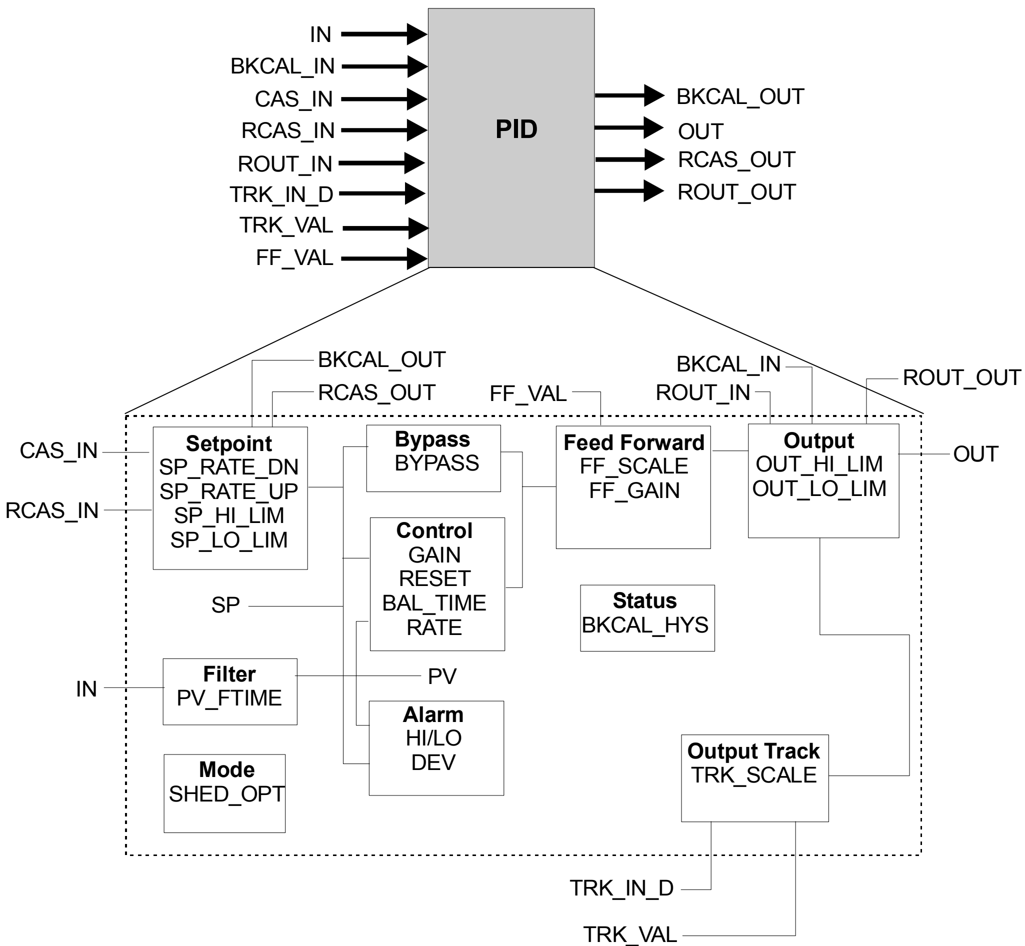


Figure 44 Functional schematic for Proportional/Integral/Derivative function block.

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 the previous illustration 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. • 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. • 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. • 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.	
Parameters	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 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 RESET 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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Ratio block

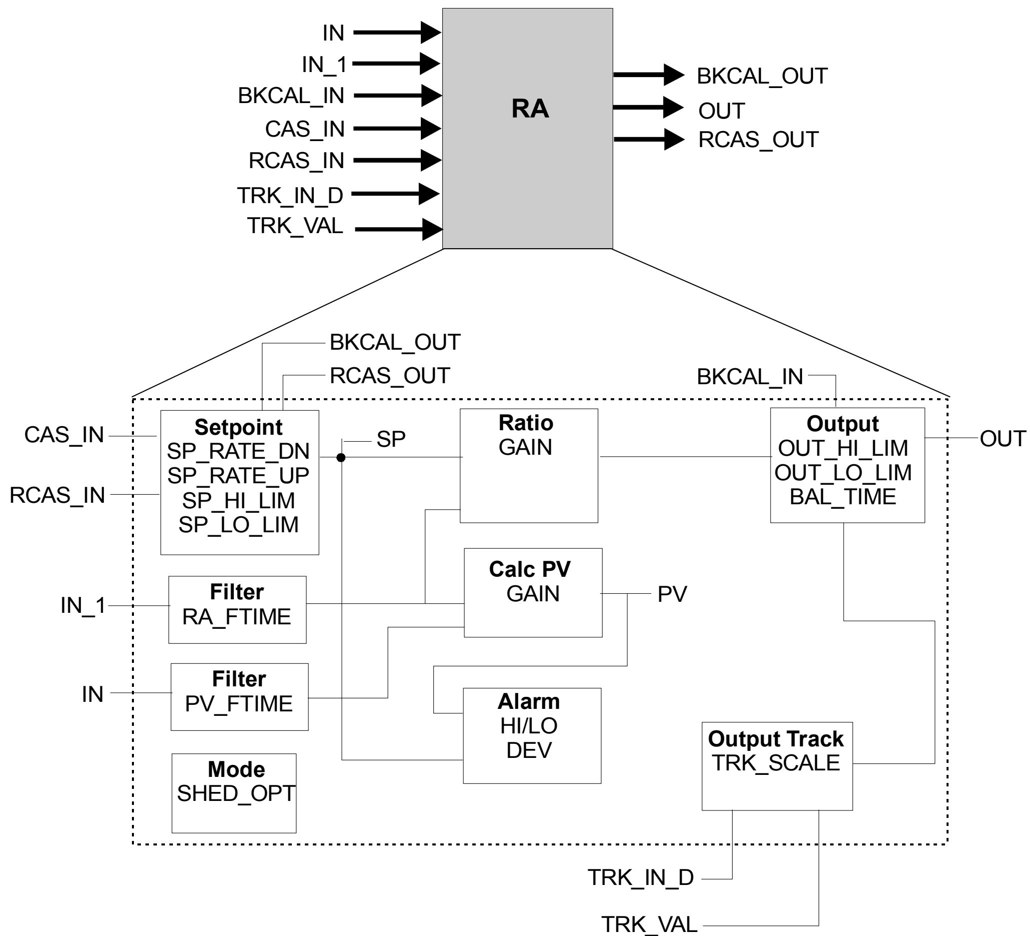


Figure 45 Functional schematic for Ratio function block.

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 the previous illustration 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. • 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	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
Reference	Refer to the *Standard Function Block Parameters Reference* section in this book for definitions of each parameter.	

Device Descriptions and Block Parameters

About Device Descriptions

Device Descriptions (DD) are critical to the interoperability of fieldbus devices. They define the data needed to establish communications between different fieldbus devices from multiple vendors 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 and on its web site. Manufacturer's may 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 the following illustration for a graphical representation of the DD infrastructure.

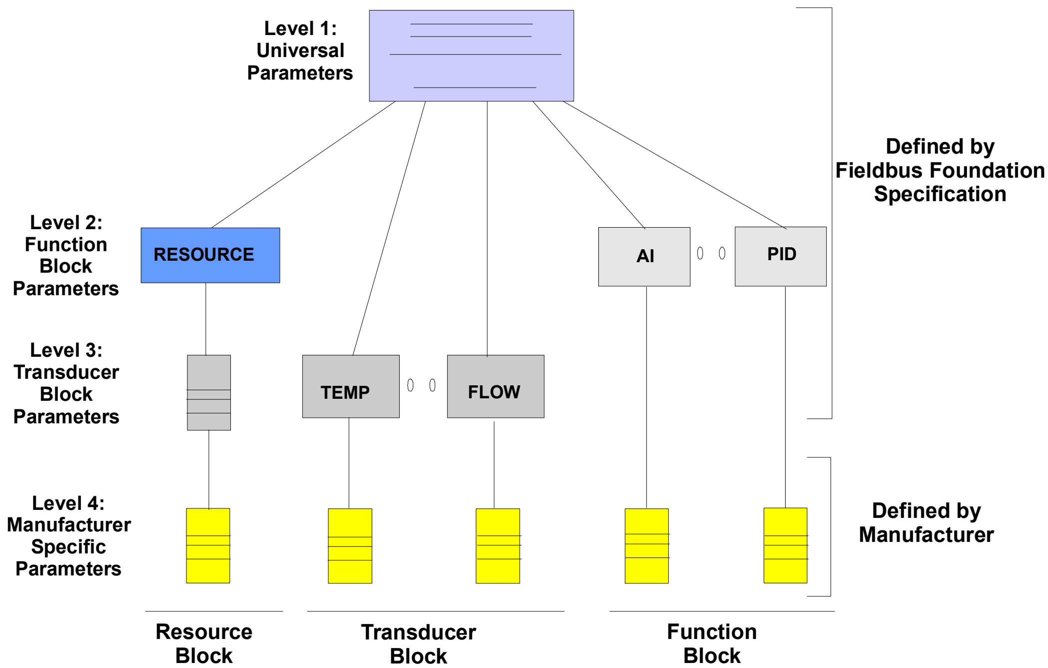


Figure 46 Device Descriptions infrastructure.

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.

The first three levels are the Device Descriptions provided by the Fieldbus Foundation.

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 in the DD supplied by the manufacturer.

Standard Function Block Parameters Reference

This section provides definitions for some fieldbus standard function block parameters for convenient reference.

Parameter Definitions

ACK_OPTION

Classification	Simple Variable
Description	Selects whether alarms associated with the block will be automatically acknowledged.
FF Data Type	Bit String
Range	1: Unacknowledged
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	Initial value is 0 All function blocks that support alarming, such as AI, PID, and DI, provide an option to support operator acknowledgement status stored in the device or stored in the control system's alarm server. Experion systems support operator acknowledgement stored in the server to provide alarm acknowledgement operation consistent with the rest of Experion components, particularly with respect to multiple console independence of alarm acknowledgement. Therefore the fieldbus parameter ACK_OPTION is always automatically set by Experion systems to ones for supported alarms, and the parameter is generally hidden to avoid confusion.

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

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
Range	
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

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

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

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.

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

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
Usage	C/Alarm
Length	13
Valid Views	
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

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
Range	
Usage	C/Block Error
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Dynamic
Remarks	

BYPASS

Classification	Simple Variable
Description	Provides the means to bypass the normal control algorithm. When BYPASS is On, the set point value is directly transferred to the output. To prevent a bump upon BYPASS switching, the set point 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.

CAS_IN

Classification	Record
Description	Represents the remote set point value that must come from another fieldbus block or a distributed control system (DCS) block through a defined link.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

CAS_IN_D

Classification	Record
Description	Represents the remote set point 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
Range	
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. • 1 Status • 2 Value

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.

CLR_FSTATE

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

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.

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
Range	
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	See the following Table 2 for a list of the control options by bit and applicable function block.

Table 1 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						

CYCLE_SEL

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

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

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

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

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
Range	
Usage	C/DD Resource
Length	32
Valid Views	
Storage	Static
Remarks	Read Only

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
Range	
Usage	C/Contained
Length	1
Valid Views	VIEW_4
Storage	Static
Remarks	Read Only

DISC_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the discrete alarm.
FF Data Type	DS-72
Range	
Usage	C/Alarm
Length	13
Valid Views	
Storage	Dynamic
Remarks	Read Only The data type consists of data that describes discrete alarms. It uses the following five elements. • 1 Unacknowledged • 2 Alarm State • 3 Time Stamp • 4 Subcode • 5 Value

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
Remarks	

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
Remarks	

DV_HI_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the high deviation alarm.
FF Data Type	DS-71
Range	
Usage	C/Alarm
Length	16
Valid Views	
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

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

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

DV_LO_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the low deviation alarm.
FF Data Type	DS-71
Range	
Usage	C/Alarm
Length	16
Valid Views	
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

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

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

FAULT_STATE

Classification	Simple Variable
Description	Condition set by loss of communication to an output block, failure promoted to an output block or a physical contact. When the fault state action is active, the output function blocks will perform their FSTATE 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

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

FEATURE_SEL

Classification	Simple Variable
Description	Bit string that identifies the selected resource block options.
FF Data Type	Bit String
Range	
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static
Remarks	This parameter must have the **Reports** bit set (FEATURE_SEL[1]). Without it, the device never

	issues any alarm notifications.

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
Range	
Usage	C/Contained
Length	4
Valid Views	VIEW_4
Storage	Static
Remarks	

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. • 1 Engineering Units at 100 percent • 2 Engineering Units at 0 percent • 3 Units Index • 4 Decimal Point

FF_VAL

Classification	Record
Description	Represents the feed-forward value.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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

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

FSTATE_TIME

Classification	Simple Variable
Description	Represents the reaction time in seconds from the detection of a failure at the output block remote set point 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
Remarks	

FSTATE_VAL

Classification	Simple Variable
Description	Defines the preset analog set point to use when a failure occurs. Value is ignored, if the IO_OPTS Fault state 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
Remarks	

FSTATE_VAL_D

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

GAIN

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

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
Range	
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. • 1 Grant • 2 Deny

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

HI_ALM

Classification	Record
Description	Identifies the status and time stamp associated with the high alarm.
FF Data Type	DS-71
Range	
Usage	C/Alarm
Length	16
Valid Views	
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

HI_HI_ALM

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

Length	16
Valid Views	
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

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

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

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

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

IO_OPTS

Classification	Simple Variable
Description	Identifies user-selectable options for altering the Input and Output block processing.
FF Data Type	Bit String
Range	
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 2 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	Fault state to value			X	X
7	Use Fault state value to restart			X	X
8	Target to Man if fault state 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				

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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. This parameter describes the number of alerts that the device can send out at the same time. A setting of zero results in never transmitting any alarm notifications. It must be set greater than zero for alarming to work.

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
Remarks	

LO_ALM

Classification	Record
Description	Represents the status of the low alarm and its associated time stamp.
FF Data Type	DS-71
Range	
Usage	C/Alarm
Length	16
Valid Views	
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

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

LO_LO_ALM

Classification	Record
Description	Represents the status of the low low alarm and its associated time stamp.
FF Data Type	DS-71
Range	
Usage	C/Alarm
Length	16

Valid Views	
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

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

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
Remarks	

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
Remarks	

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 chosen 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.

MANUFAC_ID

Classification	Simple Variable
Description	Defines the manufacturer's identification number. An interface device uses this number 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

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

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

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

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
Range	
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. • 1 Target • 2 Actual • 3 Permitted • 4 Normal

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

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. • 1 Status • 2 Value

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. • 1 Status • 2 Value

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.

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.

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
Range	
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

OUT_STATE

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

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

PV_FTIME

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

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
Range	

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

PV_STATE

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

RA_FTIME

Classification	Simple Variable
Description	Defines the time constant of a single exponential 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
Remarks	

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
Remarks	

RCAS_IN

Classification	Record
Description	Represents target set point and status provided by a supervisory host to the analog control or output block.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

RCAS_IN_D

Classification	Record
Description	Represents target set point and status provided by a supervisory host to the analog control or output block.
FF Data Type	DS-66
Range	
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. • 1 Status • 2 Value

RCAS_OUT

Classification	Record
Description	Represents block set point 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
Range	
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. • 1 Status • 2 Value

RCAS_OUT_D

Classification	Record
Description	Represents block set point 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
Range	
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. • 1 Status • 2 Value

READBACK

Classification	Record
Description	Represents the "readback" of the actual continuous valve or other actuator position in transducer units.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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
Remarks	

RESTART

Classification	Simple Variable
Description	Allows a manual restart to be initiated. The following degrees of restart are possible. • 1: Run • 2: Restart Resource • 3: Restart with Defaults • 4: 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
Valid Views	
Storage	Dynamic
Remarks	Changing this parameter, may be fatal to communication. Note that the **Restart with Defaults** function restarts the fieldbus device with **factory** default values instead of **Experion** ones. To restore Experion default values, re-load the device block through Control Builder with associated blocks set for **Full** instead of **Partial** Load.

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
Range	
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. • 1 Status • 2 Value

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
Range	
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. • 1 Status • 2 Value

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

SEL_1

Classification	Record
Description	Represents first input value to the selector.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

SEL_2

Classification	Record
Description	Represents second input value to the selector.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

SEL_3

Classification	Record
Description	Represents third input value to the selector.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

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
Remarks	

SET_FSTATE

Classification	Simple Variable
Description	Allows the fault state condition to be manually initiated by selecting Set.
FF Data Type	Unsigned 8
Range	1: Off 2: Set
Usage	C/Contained
Length	1
Valid Views	
Storage	Dynamic
Remarks	The operator can control PROGRAM or LOCAL access to this value.

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 (Not used in Experion) 8: Shed to Retained target, no return ((Not used in Experion)
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.

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
Remarks	

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
Remarks	

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
Range	
Usage	C/Simulate
Length	11
Valid Views	
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

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
Range	
Usage	C/Simulate
Length	5
Valid Views	
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

SP

Classification	Record
Description	Defines the set point of any analog block.
FF Data Type	DS-65
Range	PV_SCALE +/- 10 percent
Usage	C/Set point
Length	5
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	SP (SP_D) is normally written by an operator occasionally while in Auto mode. It is not intended to be written as a cascade setpoint input. Instead, a connection to CAS_IN is intended to provide a value into SP (SP_D) when in Cas mode and a write into RCAS_IN is intended to provide a value into SP (SP_D) when in RCas mode. However, SP (SP_D) may be updated when operating in ROut mode in order to provide a reasonable setpoint, if failing from ROut to Auto mode upon a failure of the ROUT_IN update. The data type consists of the value and status of floating

	point parameters that are Inputs or Outputs. It uses the following two elements. • 1 Status • 2 Value

SP_D

Classification	Record
Description	Defines the set point of any discrete block.
FF Data Type	DS-66
Range	PV_STATE
Usage	C/Set point
Length	2
Valid Views	VIEW_1, VIEW_3
Storage	Non-Volatile
Remarks	SP (SP_D) is normally written by an operator occasionally while in Auto mode. It is not intended to be written as a cascade setpoint input. Instead, a connection to CAS_IN is intended to provide a value into SP (SP_D) when in Cas mode and a write into RCAS_IN is intended to provide a value into SP (SP_D) when in RCas mode. However, SP (SP_D) may be updated when operating in ROut mode in order to provide a reasonable setpoint, if failing from ROut to Auto mode upon a failure of the ROUT_IN update. The data type consists of the value and status of discrete value parameters. It uses the following two elements. • 1 Status • 2 Value

SP_HI_LIM

Classification	Simple Variable
Description	Defines the high limit for set point 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

SP_LO_LIM

Classification	Simple Variable
Description	Defines the low limit for set point 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)

SP_RATE_DN

Classification	Simple Variable
Description	Defines the downward ramp rate in PV units per second for set point 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 set point change is invoked immediately.

SP_RATE-UP

Classification	Simple Variable
Description	Defines the upward ramp rate in PV units per second for set point 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 set point change is invoked immediately.

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
Range	
Usage	C/Static Revision
Length	2
Valid Views	VIEW_1, VIEW_2, VIEW_3, VIEW_4
Storage	Static
Remarks	Read Only

STATUS_OPTS

Classification	Simple Variable
Description	Defines user-selectable options for the block processing of status.
FF Data Type	Bit String
Range	
Usage	C/Contained
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	See the following Table 4 for a list of the control options by bit and applicable function block.

Table 3 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	Propagate Failure Forward	X	X								
4	Propagate 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										

STRATEGY

Classification	Simple Variable
Description	Assists in grouping blocks. This data is not checked or processed by the block.
FF Data Type	Unsigned 16
Range	
Usage	C/Strategy
Length	2
Valid Views	VIEW_4
Storage	Static
Remarks	

TAG_DESC

Classification	Simple Variable
Description	Serves as user defined description of the block.
FF Data Type	Octet String
Range	
Usage	C/Tag Description
Length	32
Valid Views	
Storage	Static
Remarks	Initial value is 32 space characters

TEST_RW

Classification	Record
Description	Defines read/write test parameter.
FF Data Type	DS-85
Range	
Usage	C/Test
Length	112
Valid Views	
Storage	Dynamic
Remarks	The data type consists of function block test read/write data. It uses the following 15 elements. • 1 Value 1 (Boolean) • 2 Value 2 (Integer 8) • 3 Value 3 (Integer 16) • 4 Value 4 (Integer 32) • 5 Value 5 (Unsigned 8) • 6 Value 6 (Unsigned 16) • 7 Value 7 (Unsigned 32) • 8 Value 8 (Floating Point) • 9 Value 9 (Visible String) • 10 Value 10 (Octet String) • 11 Value 11 (Date) • 12 Value 12 (Time of Day) • 13 Value 13 (Time Difference) • 14 Value 14 (Bit String) • 15 Value 15 (Time Value)

TRK_IN_D

Classification	Record
Description	Represents the discrete input for initiation of the external tracking function.
FF Data Type	DS-66
Range	
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. • 1 Status • 2 Value

TRK_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 associated with TRK_VAL.
FF Data Type	DS-68
Range	
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

TRK_VAL

Classification	Record
Description	Represents the input value for external tracking.
FF Data Type	DS-65
Range	
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. • 1 Status • 2 Value

UPDATE_EVT

Classification	Record
Description	Represents an alert generated by any change to the static data.
FF Data Type	DS-73
Range	
Usage	C/Event Update
Length	1, 4
Valid Views	
Storage	Dynamic
Remarks	Read Only The data type consists of data that describes a static revision alarm. It uses the following five elements. • 1 Unacknowledged • 2 Update State • 3 Time Stamp • 4 Static Revision • 5 Relative Index

WRITE_ALM

Classification	Record
Description	Represents alert that is generated if the write lock is cleared.
FF Data Type	DS-72
Range	
Usage	C/Alarm
Length	1, 3
Valid Views	
Storage	Dynamic
Remarks	Read Only The data type describes discrete alarms using the following five elements: • 1 Unacknowledged • 2 Alarm State • 3 Time Stamp • 4 Subcode • 5 Value

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.

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
Remarks	

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
Range	
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

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
Range	
Usage	C/Contained
Length	2
Valid Views	VIEW_2
Storage	Static
Remarks	

Glossary of Fieldbus Terms

Description

The following table lists some fieldbus terms and abbreviations for general reference.

Term	Abbreviation	Description
Application Process	AP	
Capability File		The Capability file contains some or all of the given fieldbus device's information that can be read from a device online. It consists of both resource information (what the device can potentially do) and value information (how the device should actually be set up).
Common File Format	CFF	The format of a Capability file is a readable text document based on a Windows Initialization (INI) file type.
Connection Manager		A Fieldbus Foundation service to manage connection information about device types, devices, and blocks that are actively communicating with a host application.
Data Link Entity	DLE	
Data Link Layer	DLL	Defined in ISO 7498
Data Link Layer Management Entity	DLME	
Data Link Protocol Data Unit	DLPDU	
Device Description	DD	A binary file that provides the definition for parameters in the FBAP of a device. For example, what Function Blocks a device contains, and what parameters are in those blocks.

Term	Abbreviation	Description
Device Description Item	DD Item	Item is a fundamental concept of the Device Description Language (DDL). It makes up the description of the device and can be any of the following constructs: • Array* • Block* • Collection • Domain • Edit Display • Item Array • Menu • Method • Program • Record* • Refresh Relation • Response Code • Variable* • Variable List* • WAO Relation * These items are of the most interest to Block Type templates.
Device Description Language	DDL	The language that vendors use to define their device's Function blocks and parameters.
Device Description Object	DDO	The suffix name for incremental DD binary files supplied by vendors that are to be converted to full and complete DD binary files by the Fieldbus Foundation Synthesizer.
Device Description Service	DDS	A software library developed by the Fieldbus Foundation that provides a generic access to a DD.

Term	Abbreviation	Description
DD Synthesizer		A tool supplied by the Fieldbus Foundation. It combines incremental DDs, with unresolved references, with Fieldbus Foundation standard DDs to produce a complete/full DD that can be used with DDS.
Fieldbus Access Sublayer	FAS	
Fieldbus Foundation Object	FFO	The suffix name for the complete/full DD binary file.
Fieldbus Foundation Tokenizer		A Fieldbus Foundation tool that converts an ASCII text file written in conformance with the Device Description Language specification into a DD binary file.
Fieldbus Message Specification	FMS	
H1 Fieldbus Segment		An independent electrical environment consisting of wire, terminators and a power source per IEC 61158-1/ISA S50.1 for supporting fieldbus devices.
H1 Fieldbus Link		The logical medium by which H1 Fieldbus devices are interconnected. It is composed of one or more physical segments interconnected by bus Repeaters or Couplers. All of the devices on a link share a common schedule, which is administered by that link's current LAS.
Immediate Response Recovery Delay	IRRD	
International Organization For Standardization	ISO	
Link Active Scheduler	LAS	
Medium Attachment Unit	MAU	For wire media MAU = transceiver.
Medium Dependent Sublayer	MDS	

Term	Abbreviation	Description
Network Management Agent	NMA	
Object Dictionary	OD	Contains the Object Descriptions for communication objects such as DataType, DataTypeStructureDescription, and SimpleVariable.
Open Systems Interconnection	OSI	
Pass Token Data Link Protocol Data Unit	PT DLPDU	
Physical Layer	PhL	
Physical Layer Overhead	PhLO	
Physical Layer Protocol Data Unit	PhPDU	
Production Rules		The grammar, in BNF format, used to describe a language. The Common File Format used to construct the Resource file follows specific production rules.
Release Directory		A directory structure used to store the DD and related files. Each manufacturer has a directory name that is the registered manufacturer ID number. There is a directory for each device type that a manufacturer produces under their ID directory.
Resource File		The part of the Capability file that is supplied by the manufacturer. It describes the communication profile, device capacity, internal record address assignments, and certain default values for a field device. A Capability file that has only this content is often called a Resource file.
Standard Dictionary		A Fieldbus Foundation file that contains standard strings and enumerations. A compressed form of the file is called the IMPORT.DCT and it is used by the Type function in Control Builder.
Symbol File	SYM	Contains symbolic names that were used in the DDL file for the FBAP of the device. It is produced by the Fieldbus Foundation Tokenizer.

Term	Abbreviation	Description
System Management Kernel	SMK	
System Management Kernel Protocol	SMKP	
Virtual Communications Relationship	VCR	Sets up communications for host to talk to a device. It acts like a speed dialer to provide for the transfer of data between applications. **FOUNDATION** Fieldbus describes these three types of VCRs: • Publish/Subscribe • Client/Server • Source/Sink
Virtual Field Device	VFD	The management or FBAP section of a device that is addressable by a VFD tag.

Honeywell

Honeywell International
Process Solutions
2500 West Union Hills
Phoenix, AZ 85027