



# ControlLogix Digital I/O Modules

Catalog Numbers 1756-IA8D, 1756-IA16, 1756-IA16K, 1756-IA16I, 1756-IA16IK, 1756-IA32, 1756-IA32K, 1756-IB16, 1756-IB16K, 1756-IB16D, 1756-IB16DK, 1756-IB16I, 1756-IB16IK, 1756-IB16IF, 1756-IB16IFK, 1756-IB16XT, 1756-IB32, 1756-IB32K, 1756-IB32XT, 1756-IC16, 1756-IG16, 1756-IG16K, 1756-IH16I, 1756-IH16IK, 1756-IM16I, 1756-IM16IK, 1756-IN16, 1756-IN16K, 1756-IV16, 1756-IV16K, 1756-IV32, 1756-IV32K, 1756-OA8, 1756-OA8D, 1756-OA8E, 1756-OA16, 1756-OA16K, 1756-OA16I, 1756-OA16IK, 1756-OB8, 1756-OB8K, 1756-OB8EI, 1756-OB8EIK, 1756-OB16D, 1756-OB16DK, 1756-OB16E, 1756-OB16EK, 1756-OB16I, 1756-OB16IEF, 1756-OB16IEFK, 1756-OB16IEFS, 1756-OB16IEFSK, 1756-OB16IS, 1756-OB32, 1756-OB32K, 1756-OB32XT, 1756-OC8, 1756-OC8K, 1756-OG16, 1756-OG16K, 1756-OH8I, 1756-ON8, 1756-ON8K, 1756-OV16E, 1756-OV32E, 1756-OV32EK, 1756-OW16I, 1756-OW16IK, 1756-OX8I, 1756-OX8IK



**Allen-Bradley**

by **ROCKWELL AUTOMATION**

User Manual

Original Instructions

# Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.



**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.



**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

## IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

These labels may also be on or inside the equipment to provide specific precautions.



**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.



**ARC FLASH HAZARD:** Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

The following icon may appear in the text of this document.



Identifies information that is useful and can help to make a process easier to do or easier to understand.

	<b>Preface</b>	
	About This Publication .....	9
	Summary of Changes .....	9
	Download Firmware, AOP, EDS, and Other Files .....	9
	Additional Resources .....	9
	<b>Chapter 1</b>	
<b>ControlLogix Digital I/O Modules</b>	About Digital I/O Modules .....	11
	Available Features .....	14
	Module Identification and Status Information .....	14
	Module Definitions, Communication, or Connection Formats .....	15
	Ownership .....	16
	Multiple Owner-Controllers of Input Modules .....	16
	Configuration Changes in an Input Module with Multiple Owners .....	17
	1756-OB16IEF Redundant Owner .....	17
	Listen-only Mode .....	17
	Connections .....	18
	Direct Connections .....	18
	Rack-optimized Connections .....	19
	Recommendations for Rack-optimized Connections .....	20
	Internal Module Operation .....	20
	Input Module Operation .....	21
	Input Modules .....	21
	Input Modules in a Remote Chassis .....	22
	RPI .....	22
	COS .....	22
	Trigger Event Tasks .....	23
	Input Modules in a Remote Chassis .....	23
	Remote Input Modules Connected Via the EtherNet/IP Network .....	23
	Output Module Operation .....	24
	Output Modules .....	24
	Output Modules in a Local Chassis .....	25
	Output Modules in a Remote Chassis .....	25
	Remote Output Modules Connected Via the EtherNet/IP Network .....	25
	Common Digital I/O Module Features .....	26
	Removal and Insertion Under Power .....	26
	Module Fault Reporting .....	26
	Software Configurable .....	27
	Electronic Keying .....	27
	Module Inhibiting .....	28
	Producer/Consumer Communication .....	28
	Status Indicator Information .....	29
	Use the System Clock to Timestamp Inputs and Schedule .....	29
	Use CIP Sync Time with Fast I/O Modules .....	30
	Mixing CST and CIP Sync Modules in a ControlLogix System .....	30
	Input Module Features .....	31

- Data Transfer on Either Cyclic Time or Change of State (COS)..... 31
- Isolated and Nonisolated Varieties of Input Modules ..... 31
- Multiple Input Point Densities ..... 31
- RPI..... 32
- Change of State..... 32
- Software Configurable Filter Times ..... 32
- Output Module Features..... 33
  - Output Data Echo ..... 33
  - Isolated and Nonisolated Varieties of Output Modules..... 34
  - Time-scheduled Output Control ..... 34
  - Configurable Point-level Output States ..... 35
  - Electronic Fusing ..... 35
  - Field Power Loss Detection ..... 37
  - Diagnostic Latch of Information ..... 37

**Chapter 2**

**Digital Diagnostic Module Features**

- Common Diagnostic Features..... 39
  - Diagnostic Timestamp ..... 39
  - 8-Point AC/16-Point DC..... 39
  - Point-level Fault Reporting ..... 39
  - Diagnostic Latch of Information ..... 40
- Diagnostic Input Module Features ..... 41
  - Fault and Status Reporting between Input Modules and Controllers ..... 41
  - Diagnostic Change of State for Input Modules ..... 42
  - Open Wire Detection..... 42
  - Field Power Loss Detection ..... 43
- Diagnostic Output Module Features ..... 43
  - Field Wiring Options ..... 43
  - Pulse Test..... 43
  - Diagnostic Change of State for Output Modules ..... 44
  - Fault and Status Reporting Between Output Modules and Controllers..... 44
  - No Load Detection..... 46
  - Field-side Output Verification..... 46

**Chapter 3**

**Fast Digital Module Features**

- Fast Module Features..... 47
  - Response Time ..... 47
- Fast Input Module Features..... 48
  - Pulse Capture ..... 48
  - Dedicated Connection for Event Tasks ..... 48
  - Fault and Status Reporting between Input Modules and Controllers ..... 50
  - Per Point Timestamping and Change of State ..... 51
  - Software Configurable Filter Times ..... 52
- Fast Output Module Features ..... 54
  - Fault and Status Reporting between Output Modules and Controllers..... 54
  - Programmable Fault State Delay..... 55
  - Pulse-width Modulation ..... 55
  - Cycle Limit and Execute All Cycles ..... 57
  - Minimum On Time, Extend Cycle, and Stagger Output ..... 59

<b>Configure Digital I/O Modules</b>	<b>Chapter 4</b>	
	Configure ControlLogix Digital I/O Modules . . . . .	61
	Create a New Module . . . . .	62
	Communication Formats, Connection Formats, and Module Definitions . . . . .	64
	View and Change Module Tags . . . . .	65
	Edit the Configuration . . . . .	66
	Connection Properties . . . . .	67
	Configure Input Modules . . . . .	68
	Enable COS . . . . .	68
	Configure Input Filter Times . . . . .	69
	Configure Input Module with Multiple Owners . . . . .	69
	Configure Output Modules . . . . .	70
	Enable or Disable Field Power Loss (1756-0A8E only) . . . . .	70
	Enable or Disable Diagnostic Latch of Information (1756-0A8E only) . . . . .	71
	Reset Latched Fault (0A8E only) . . . . .	71
	Configure Diagnostic Modules . . . . .	72
	Reset Latched Fault . . . . .	72
	Configure Diagnostic Input Modules . . . . .	73
	Configure Open Wire Detection . . . . .	73
	Configure Diagnostic Output Modules . . . . .	74
	Enable Field-side Output Verification . . . . .	74
	About Redundant Owners . . . . .	75
	Requirements . . . . .	75
	Restrictions . . . . .	76
	Single Controller Behavior . . . . .	76
	Multiple Controller Behavior . . . . .	77
	Response Times for Output Modules . . . . .	79
	Input Modules . . . . .	79
	Configure Fast Input Modules . . . . .	80
	Configure Input Filter Times . . . . .	82
	Configure Fast Output Modules . . . . .	83
	Configure Pulse-width Modulation . . . . .	83
	Configure the Output Module as Redundant Owner . . . . .	86
	Implement Redundant Ethernet Modules . . . . .	87
	Response Times for Output Modules . . . . .	88
<b>Troubleshoot Your Module</b>	<b>Chapter 5</b>	
	Status Indicators . . . . .	89
	Status Indicators for Output Modules . . . . .	90
	Use the Logix Designer Application for Troubleshooting . . . . .	92
	Fault Type Determination . . . . .	93
	Troubleshooting Fast Modules . . . . .	93
<b>Wiring Diagrams</b>	<b>Chapter 6</b>	
	1756-IA8D . . . . .	96
	1756-IA16, 1756-IA16K . . . . .	97
	1756-IA16I, 1756-IA16IK . . . . .	97
	1756-IA32, 1756-IA32K . . . . .	98

1756-IB16, 1756-IB16K, 1756-IB16XT .....	98
1756-IB16D, 1756-IB16DK .....	99
1756-IB16I, 1756-IB16IK .....	100
1756-IB16IF, 1756-IB16IFK .....	101
1756-IB32, 1756-IB32K, 1756-IB32XT .....	102
1756-IC16 .....	102
1756-IG16, 1756-IG16K .....	103
1756-IH16I, 1756-IH16IK .....	104
1756-IM16I, 1756-IM16IK .....	104
1756-IN16, 1756-IN16K .....	105
1756-IV16, 1756-IV16K .....	105
1756-IV32, 1756-IV32K .....	106
1756-OA8 .....	106
1756-OA8D .....	107
1756-OA8E .....	107
1756-OA16, 1756-OA16K .....	108
1756-OA16I, 1756-OA16IK .....	109
1756-OB8, 1756-OB8K .....	110
1756-OB8EI, 1756-OB8EIK .....	111
1756-OB16D, 1756-OB16DK .....	111
1756-OB16E, 1756-OB16EK .....	112
1756-OB16I .....	112
1756-OB16IEF, 1756-OB16IEFK .....	113
1756-OB16IEFS, 1756-OB16IEFSK .....	113
1756-OB16IS .....	114
1756-OB32, 1756-OB32K, 1756-OB32XT .....	114
1756-OC8, 1756-OC8K .....	115
1756-OG16, 1756-OG16K .....	116
1756-OH8I .....	117
1756-ON8, 1756-ON8K .....	117
1756-OV16E .....	118
1756-OV32E, 1756-OV32EK .....	118
1756-OW16I, 1756-OW16IK .....	119
1756-OX8I, 1756-OX8IK .....	119

**Appendix A**

**Tag Definitions**

Standard and Diagnostic Input Module Tags .....	121
Standard and Diagnostic Output Module Tags .....	123
Fast Input Module Tags .....	125
Fast Output Module Tags .....	129
1756-OB16IEF Module .....	129
1756-OB16IEFS Module .....	136
Redundant Owner Configuration Tags .....	143
Redundant Owner Input Tag Layout .....	143
Redundant Owner Output Tag Layout .....	143
Array Data Structures .....	144

<b>Use Ladder Logic To Perform Run Time Services and Reconfiguration</b>	<b>Appendix B</b>	
	Using Message Instructions.....	145
	Processing Real-time Control and Module Services.....	145
	One Service Performed Per Instruction.....	145
	Create a New Tag.....	146
	Enter Message Configuration.....	148
	Configuration Tab.....	148
	Communication Tab.....	149
	Timestamped Inputs and Scheduled Outputs - Standard or Diagnostic Modules	149
	Timestamped Inputs and Scheduled Outputs - Fast Modules.....	151
	Reset a Fuse, Perform Pulse Test, and Reset Latched Diagnostics.....	154
	Perform a WHO to Retrieve Module Identification and Status.....	155
	Review of Tags in Ladder Logic.....	157
<b>Choose a Correct Power Supply Motor Starters for Digital I/O Modules</b>	<b>Appendix C</b>	
	<b>Appendix D</b>	
	Determine the Maximum Number of Motor Starters.....	162
<b>Major Revision Upgrades</b>	<b>Appendix E</b>	
	If Using a Compatible or Disabled Keying I/O Configuration.....	163
	If Using an Exact Match Keying Configuration.....	163
<b>1492 IFMs for Digital I/O Modules</b>	<b>Appendix F</b>	
	Cable Overview.....	165
<b>ControlNet Communication</b>	<b>Appendix G</b>	
	Rack-optimized Connections.....	173
	Recommendations for Rack-optimized Connections.....	174
	Remote Input Modules Connected Via the ControlNet Network.....	175
	Best Case RPI Multicast Scenario.....	175
	Worst Case RPI Multicast Scenario.....	175
	Remote Output Modules Connected Via the ControlNet Network.....	176
	Best Case RPI Multicast Scenario.....	176
	Worst Case RPI Multicast Scenario.....	176
	Use the System Clock to Timestamp Inputs and Schedule Outputs.....	177
	Create a New Module.....	177
<b>History of Changes</b>	<b>Appendix H</b>	
	Change Log.....	179
	<b>Index</b> .....	181



## About This Publication

This manual describes how to install, configure, and troubleshoot your ControlLogix® digital I/O modules. There is also a complete listing of digital input and output modules, including specifications and wiring diagrams. You must be able to program and operate a ControlLogix controller to use your digital I/O module efficiently.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

## Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Topic	Page
Added missing K variants to the IFMs and Prewired Cables table	166
Added catalog numbers 1756-IB16XT, 1756-IB32XT, and 1756-OB32XT	Throughout
This publication was reformatted and updated to the most current template	Throughout

## Download Firmware, AOP, EDS, and Other Files

Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at [rok.auto/pcdc](http://rok.auto/pcdc).

## Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
1756 ControlLogix I/O Modules Specifications Technical Data, publication <a href="#">1756-TD002</a>	Provides specifications for ControlLogix I/O modules.
ControlLogix Peer I/O Control Application Technique, publication <a href="#">1756-AT016</a>	Describes typical peer control applications and provides details about how to configure I/O modules for peer control operation.
ControlLogix DC Digital I/O Modules Installation Instructions, publication <a href="#">1756-IN062</a>	Describes how to install and wire the ControlLogix DC digital I/O modules.
ControlLogix AC Digital I/O Modules Installation Instructions, publication <a href="#">1756-IN064</a>	Describes how to install and wire the ControlLogix AC digital I/O modules.
ControlLogix System User Manual, publication <a href="#">1756-UM543</a>	Describes how to install, configure, program, and operate a ControlLogix system.
Industrial Automation Wiring and Grounding Guidelines, publication <a href="#">1770-4.1</a>	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, <a href="http://rok.auto/certifications">rok.auto/certifications</a> .	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at [rok.auto/literature](http://rok.auto/literature).

**Notes:**

## ControlLogix Digital I/O Modules

ControlLogix® digital I/O modules are input and output modules that provide On/Off detection and actuation. By using the Producer/Consumer network model, digital I/O modules can produce information when needed while providing additional system functions.

### About Digital I/O Modules

ControlLogix modules mount in a ControlLogix chassis and require either a removable terminal block (RTB) or a Bulletin 1492 wiring interface module (IFM) to connect all field-side wiring.

---

**IMPORTANT** The ControlLogix system has been agency certified using only the ControlLogix RTB catalog numbers 1756-TBCH, 1756-TBCHXT, 1756-TBNH, 1756-TBNHXT, 1756-TBSH, 1756-TBSHXT, 1756-TBS6H, and 1756-TBS6HXT. Any application that requires agency certification of the ControlLogix system using other wiring termination methods may require application-specific approval by the certifying agency.

---

Before you install and use your module, you must do the following:

- Install and ground a 1756 chassis and power supply.
- Order and receive an RTB or IFM and its components for your application.

---

**IMPORTANT** RTBs and IFMs are not included with your module purchase.

---

**Table 1 - Types of ControlLogix Digital I/O Modules**

Digital I/O Type	Description
Diagnostic	These modules provide diagnostic features to the point level. These modules have a <b>D</b> at the end of the catalog number.
Electronic fusing	These modules have internal electronic fusing to help prevent too much current from flowing through the module. These modules have an <b>E</b> at the end of the catalog number.
Individually isolated	These modules have individually isolated inputs or outputs. These modules have an <b>I</b> at the end of the catalog number.
Fast	These modules provide fast response times. These modules have an <b>F</b> at the end of the catalog number.
Conformal coated	These modules provide extended protection in harsh, corrosive environments. These modules have a <b>K</b> or an <b>XT</b> at the end of the catalog number.

---

**IMPORTANT** When a ControlLogix product that is rated for harsh environments (corrosive atmosphere, extended temperature, etc.) is used in a system with other ControlLogix products that have lower specification values, the system is derated to the lowest common value.

**EXAMPLE:** If the maximum operating temperature specification found in the Technical Data for your ControlLogix-XT™ module is 70 °C (158 °F) and you pair it with a ControlLogix chassis that is temperature rated to 60 °C (140 °F), your system is derated to 60 °C (140 °F).

To ensure that your system is equipped for harsh environments, compare the corrosive atmosphere, temperature, and other specifications found in the Technical Data publication for each product.

---

Table 2 - ControlLogix Digital I/O Modules

Cat. No.	Description	Page
1756-IA8D	79...132V AC 8-point diagnostic input module	96
1756-IA16, 1756-IA16K	74...132V AC 16-point input module	97
1756-IA16I, 1756-IA16IK	79...132V AC 16-point isolated input module	97
1756-IA32, 1756-IA32K	74...132V AC 32-point input module	98
1756-IB16, 1756-IB16K, 1756-IB16XT	10...31.2V DC 16-point input module	98
1756-IB16D, 1756-IB16DK	10...30V DC diagnostic input module	99
1756-IB16I, 1756-IB16IK	10...30V DC 16-point, isolated input module	100
1756-IB16IF, 1756-IB16IFK	10...30V DC, 16-point, isolated, fast peer control input module	101
1756-IB32, 1756-IB32K, 1756-IB32XT	10...31.2V DC 32-point input module	102
1756-IC16	30...60V DC 16-point input module	102
1756-IG16, 1756-IG16K	Transistor-transistor logic (TTL) input module	103
1756-IH16I, 1756-IH16IK	90...146V DC 16-point isolated input module	104
1756-IM16I, 1756-IM16IK	159...265V AC 16-point isolated input module	104
1756-IN16, 1756-IN16K	10...30V AC 16-point input module	105
1756-IV16, 1756-IV16K	10...30V DC 16-point sourcing current input module	105
1756-IV32, 1756-IV32K	10...30V DC 32-point sourcing current input module	106
1756-OA8	74...265V AC 8-point output module	106
1756-OA8D	74...132V AC 8-point diagnostic output module	107
1756-OA8E	74...132V AC 8-point electronically fused output module	107
1756-OA16, 1756-OA16K	74... 265V AC 16-point output module	108
1756-OA16I, 1756-OA16IK	74...265V AC 16-point isolated output module	109
1756-OB8, 1756-OB8K	10...30V DC 8-point output module	110
1756-OB8EI, 1756-OB8EIK	10...30V DC 8-point electronically fused, isolated output module	111
1756-OB16D, 1756-OB16DK	19.2...30V DC 16-point diagnostic output module	111
1756-OB16E, 1756-OB16EK	10...31.2V DC 16-point electronically fused output module	112
1756-OB16I	10...30V DC 16-point isolated output module	112
1756-OB16IEF, 1756-OB16IEFK	10...30V DC, 16-point, isolated, fast peer control output module	113
1756-OB16IEFS, 1756-OB16IEFSK	10...30V DC, 16-point, isolated, fast, scheduled per point output module	113
1756-OB16IS	10...30V DC scheduled, isolated output module	114
1756-OB32, 1756-OB32K, 1756-OB32XT	10...31.2V DC 32-point output module	114
1756-OC8, 1756-OC8K	30...60V DC 8-point output module	115
1756-OG16, 1756-OG16K	Transistor-transistor logic (TTL) output module	116
1756-OH8I	90...146V DC 8-point isolated output module	117
1756-ON8, 1756-ON8K	10...30V AC 8-point output module	117
1756-OV16E	10...30V DC 16-point electronically fused, sinking current output module	118
1756-OV32E, 1756-OV32EK	10...30V DC 32-point electronically fused, sinking current output module	118
1756-OW16I, 1756-OW16IK	10...265V, 5-150V DC 16-point isolated contact module	119
1756-OX8I, 1756-OX8IK	10...265V, 5-150V DC 8-point isolated contact module	119

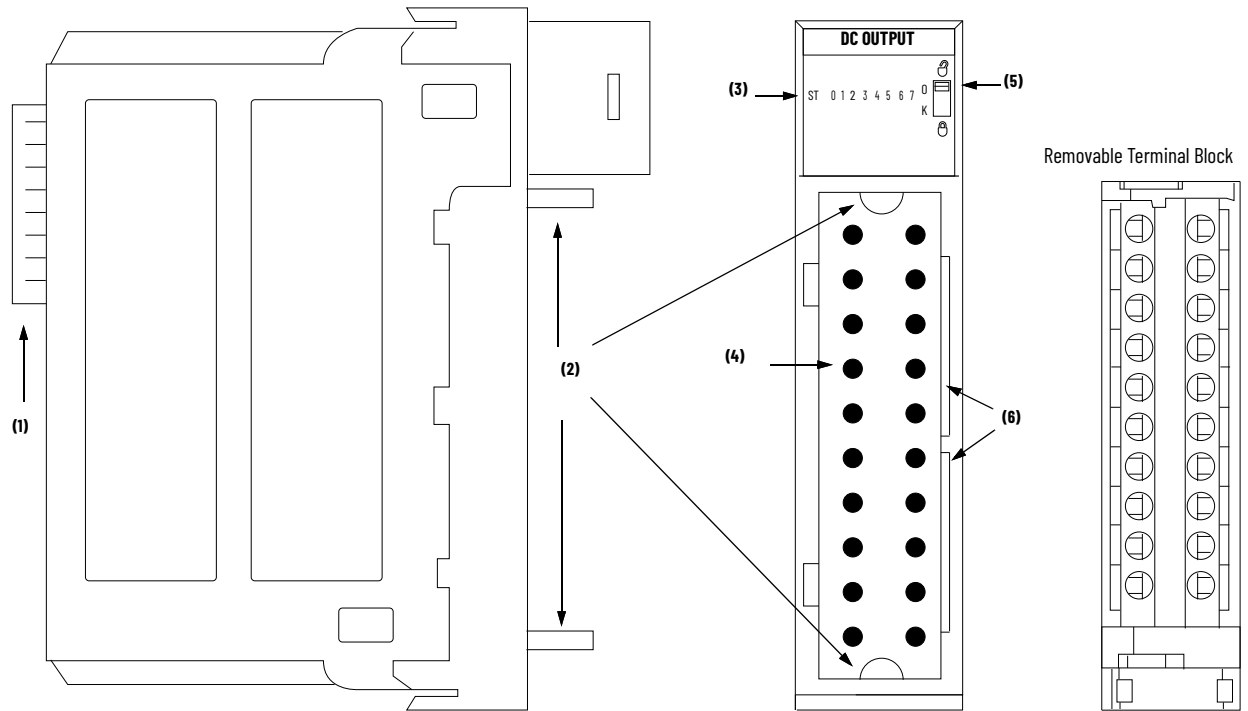
**IMPORTANT**

ControlLogix Digital I/O Modules that have an **XT** at the end of the catalog number are shipped in packaging that provides a degree of protection from corrosive atmospheres.

Once the factory packaging seal is broken, the module and the corresponding ControlLogix-XT Removable Terminal Block (RTB) must remain installed at all times, and the RTB door must remain closed, for the product to maintain its corrosive atmosphere rating.

If maintenance is required, you can open the RTB door or remove the module or RTB. Close the RTB door or reinstall the module or RTB after temporary access is complete.

Figure 1 - Parts Illustration



Note No.	Description
1	<b>Backplane Connector</b> —Interface for the ControlLogix system that connects the module to the backplane.
2	<b>Top and bottom guides</b> —Guides assist in seating the RTB or IFM onto the module.
3	<b>Status indicators</b> —Indicators display the status of communication, module health, and input/output devices. Indicators help in troubleshooting anomalies.
4	<b>Connector pins</b> —Input/output, power, and grounding connections are made to the module through these pins with the use of an RTB or IFM.
5	<b>Locking tab</b> —The locking tab anchors the RTB or IFM on the module, maintaining wiring connections.
6	<b>Slots for keying</b> —Mechanically keys the RTB to help prevent making the wrong wire connections to your module.

## Available Features

The table below lists several features available on ControlLogix digital I/O modules.

**Table 3 - Digital I/O Module Features**

Feature	Description
Removal and Insertion Under Power (RIUP)	You can remove and insert modules and removable terminal blocks (RTB) while power is applied.
Producer/Consumer communication	This communication method is an intelligent data exchange between modules and other system devices in which each module produces data without first being polled.
System time stamp of data	A 64-bit system clock places a time stamp on the transfer of data between the module and its owner-controller.
Module-level fault reporting and field-side diagnostic detection	Fault and diagnostic detection capabilities to help you effectively and efficiently use your module and troubleshoot your application.
Agency Certification	Class 1, Division 2 agency certification for any application that requires approval.

## Module Identification and Status Information

Each ControlLogix I/O module maintains specific identification information that separates it from all other modules. This information assists you in tracking all the components of your system.

For example, you can track module identification information to know which modules are in any ControlLogix chassis at any time. While retrieving module identity, you can also retrieve module status.

Item	Description
Product type	Module product type, such as digital I/O or analog I/O
Product code	Module catalog number
Major revision	Module major revision number
Minor revision	Module minor revision number
Status	Module status, including these items: <ul style="list-style-type: none"> <li>• Controller ownership</li> <li>• Whether the module has been configured</li> <li>• Device-specific status, such as the following:               <ul style="list-style-type: none"> <li>- Self-test</li> <li>- Update in progress</li> <li>- Communications fault</li> <li>- Not owned (outputs in Program mode)</li> <li>- Internal fault (needs update)</li> <li>- Run mode</li> <li>- Program mode (outputs only)</li> </ul> </li> <li>• Minor recoverable fault</li> <li>• Minor unrecoverable fault</li> <li>• Major recoverable fault</li> <li>• Major nonrecoverable fault</li> </ul>
Vendor	Module manufacturer vendor, such as Allen-Bradley
Serial number	Module serial number
Length of ASCII text string	Number of characters in module text string
ASCII text string	Module ASCII text string description

**IMPORTANT** You must perform a WHO service to retrieve this information. For more information, see [See Perform a WHO to Retrieve Module Identification and Status on page 155](#).

## Module Definitions, Communication, or Connection Formats

The initial configuration of a module requires you to choose a module definition or a communication or connection format. The term that is used depends on the Add-on Profile (AOP) for your module. Earlier AOPs use communication formats, later AOPs use connection formats, and the newest AOPs use module definitions.

A module definition or communication or connection format determines the following:

- Available configuration options
- Type of data that is transferred between the module and its owner-controller
- Which tags are generated when the configuration is complete

---

**IMPORTANT**

Communication formats cannot be changed whether online or offline after a program is downloaded to the controller.

However, connection formats can be changed when offline after a program is downloaded to the controller.

---

The communication or connection format also defines the connection between the controller writing the configuration and the module. The number and type of choices varies depending on what module you are using and whether it is in a local or remote chassis.



When you choose a Listen-only format, only the General and Connection tabs appear when you view module properties in Logix Designer application. Controllers that want to listen to a module, but not own it, use a Listen-only format.

For more information on configuration options for module definitions, communication, or connection formats, see [Module Definitions, Communication, or Connection Formats on page 15](#).

## Ownership

I/O modules in a ControlLogix system can be owned by a Logix 5000® controller. An owner-controller fulfills these functions:

- Stores configuration data for every module that it owns
- Sends I/O modules configuration data to define module behavior and begin module operation with the control system
- Resides in a local or remote chassis in regard to the I/O module position

Each ControlLogix I/O module must continuously maintain communication with its owner-controller to operate normally.

Typically, each module in the system has only one owner-controller. Input modules can have multiple owner-controllers. Output modules, however, are limited to a single owner-controller.

For more information about using multiple owner-controllers, see [Configuration Changes in an Input Module with Multiple Owners on page 17](#).

### Multiple Owner-Controllers of Input Modules

If a connection is lost between an owner-controller and a module, the connection is also lost between any controllers listening to that module. As a result, the ControlLogix system lets you define more than one owner-controller for most input modules. Input modules that have Output tags do not allow more than one owner-controller.

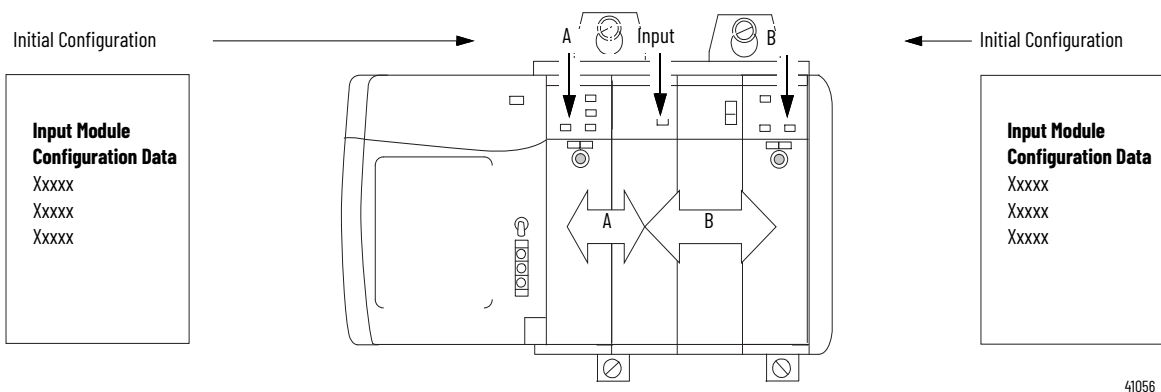
---

**IMPORTANT** Only input modules can have multiple owner-controllers. If multiple owner-controllers are connected to the same input module, they must maintain identical configurations for that module.

---

In the figure below, controller A and controller B both have been configured to be owner-controllers of the same input module.

**Figure 2 - Identical Owner-Controller Configurations for Input Module**



As soon as a controller receives its user program, it tries to establish a connection with the input module. A connection is established with the controller whose configuration data arrives first. When the second controller's configuration data arrives, the module compares it to its current configuration data, which was received and accepted from the first controller.

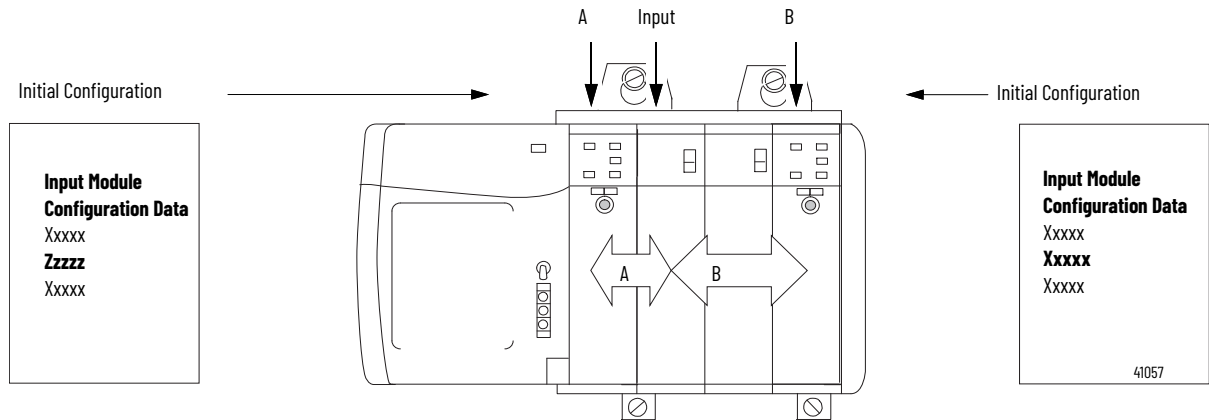
If the configuration data sent by the second controller matches the data sent by the first controller, that connection is also accepted. If any parameter of the second configuration data differs from the first, the module rejects the connection and the user is informed by an error in the software or via program logic.

The advantage of multiple owners over a Listen-only connection is that either of the controllers can break the connection to the module, and the module continues to operate and multicast data to the system through the connection maintained by the other controller.

## Configuration Changes in an Input Module with Multiple Owners

You must be careful when changing the configuration data of an input module in a multiple owner scenario. If the configuration data is changed in owner A and sent to the module, that configuration data is accepted as the new configuration for the module. Owner B continues to listen unaware that any changes have been made in the behavior of the input module, as illustrated.

**Figure 3 - Module Configuration Changes with Multiple Owners**




---

**IMPORTANT** A message in Logix Designer application alerts you to the possibility of a multiple owner-controller situation and lets you inhibit the connection before changing the module configuration. When changing the configuration for a module with multiple owners, we recommend that you inhibit the connection.

---

### 1756-OB16IEF Redundant Owner

Redundant ownership lets a single controller use redundant network paths to communicate with a single output or lets two separate controllers coordinate to control the output.

See [Table 34](#) for a list of digital input modules that do not support multiple module owners.

## Listen-only Mode

Any controller in the system can listen to the data from any I/O module, such as input data, echoed output data, or echoed diagnostic information. Even if a controller does not own a module, or hold the module configuration data, the controller can still listen to the module.

During the module configuration process, you can specify one of several Listen modes. For more information, see [Module Definitions, Communication, or Connection Formats on page 15](#).

Choosing a Listen mode lets the controller and module establish communication without the controller sending any configuration data. In this instance, another controller owns the module being listened to.

---

**IMPORTANT** In Listen-only mode, controllers continue to receive data multicast from the I/O module as long as the connection between the owner-controller and I/O module is maintained.

If the connection between the owner-controller and module is broken, the module stops multicasting data and connections to all listening controllers are also broken.

---

## Connections

With ControlLogix I/O modules, a connection is the data transfer link between a controller and an I/O module. A connection can be one of these types:

- Direct
- Rack-optimized

This table lists the advantages and disadvantages of each connection type.

Connection Type	Advantages	Disadvantages
Direct	All input and data echo information is transferred, including diagnostic information and fusing data.	With more data transferring over the network, your system does not operate as efficiently as with rack connections.
Rack-optimized	Connection usage is economized. The owner-controller has a single RPI value for each connection.	Input and data echo information is limited to general faults and data.

### Direct Connections

A direct connection is a real-time data transfer link between the controller and the device that occupies the slot that the configuration data references. When module configuration data is downloaded to an owner-controller, the controller attempts to establish a direct connection to each of the modules that is referenced by the data.

If a controller has configuration data referencing a slot in the control system, the controller periodically checks for the presence of a device there. When the presence of a device is detected there, the controller automatically sends the configuration data.

If the data is appropriate to the module found in the slot, a connection is made and operation begins. If the configuration data is not appropriate, the data is rejected and an error message appears in the software. In this case, the configuration data can be inappropriate for any of a number of reasons. For example, module configuration data may be appropriate except for a mismatch in electronic keying that helps prevent normal operation.

The controller maintains and monitors its connection with a module. Any break in the connection causes the controller to set fault status bits in the data area associated with the module. Breaks in the connection can be caused by a module fault or the removal of the module from the chassis while under power. Logix Designer application monitors fault status bits to annunciate module failures.

## Rack-optimized Connections

When a digital I/O module is in a remote chassis with respect to its owner-controller, you may be able to choose Rack Optimization or Listen-only Rack Optimization during module configuration. The option that you choose depends on the communication module configuration. If the communication module uses Listen-only Rack Optimization, then the I/O module must also use Listen-only Rack Optimization.

A rack-optimized connection economizes bandwidth between owner-controllers and digital I/O modules in the remote chassis. Rather than having several direct connections with individual RPI values, an owner-controller has a single rack connection with a single RPI value. That RPI value accommodates all digital I/O modules in the remote chassis.

### IMPORTANT

Because rack-optimized connections are applicable only in applications that use a remote chassis, you must configure the communication format, as described in [Chapter 4](#), for both the remote I/O module and the remote 1756-CNB module or EtherNet/IP™ module.

Make sure you configure both modules for rack optimization. If you choose another communication format for each module, the controller makes two connections to the same chassis (one for each format) and the same data travels across the Ethernet network.

If you use rack optimization for both modules, you preserve bandwidth and configure your system to operate more efficiently.

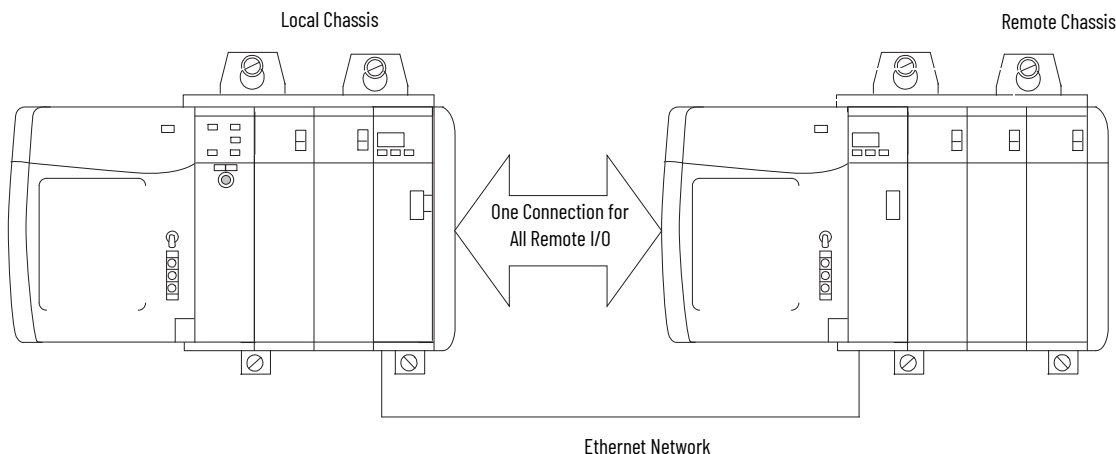
The input, or data echo, information is limited to general faults and data. No additional status, such as diagnostic information, is available.

### IMPORTANT

Each controller can establish connections in any combination of direct or rack-optimized. In other words, you can use a rack-optimized connection between an owner-controller and multiple remote I/O modules while simultaneously using a direct connection between that same controller and any other I/O modules in the same remote chassis.

This figure shows how a rack-optimized connection eliminates the need for three separate connections. The owner-controller in the local chassis communicates with all the I/O modules in the remote chassis but uses only one connection. The EtherNet communication module sends data from the modules simultaneously at the RPI.

**Figure 4 - Rack-optimized Connection**



## Recommendations for Rack-optimized Connections

We recommend that you use a rack-optimized connection for these applications:

- Standard digital I/O modules
- Non-fused digital output modules
- Owner-controllers running low on connections

---

**IMPORTANT** Rack-optimized connections are available only to digital I/O modules. However, do not use a rack-optimized connection for diagnostic I/O modules or fused output modules. Diagnostic and fused output data is not transferred over a rack-optimized connection. This defeats the purpose of using those modules.

---

## Internal Module Operation

ControlLogix I/O modules experience signal propagation delays that must be accounted for during operation. Some of these delays are user-configurable, and some are inherent to the module hardware.

For example, there is a small delay, typically less than 1 ms, between when a signal is applied at the RTB of a ControlLogix input module and when a signal is sent to the system over the backplane. This time reflects a filter time of 0 ms for a DC input.

This section offers an explanation of the time limitations with ControlLogix I/O modules.

## Input Module Operation

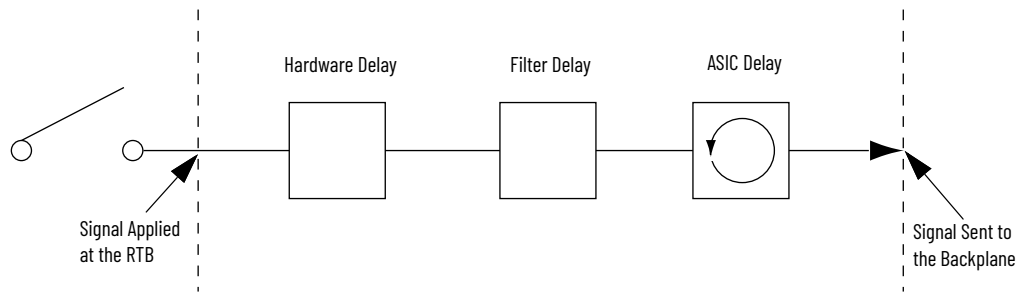
In traditional I/O systems, controllers poll input modules to obtain their input status. In the ControlLogix system, a controller does not poll digital input modules. Instead, the modules multicast their data either upon change of state (COS) or requested packet interval (RPI). The frequency depends on the options that are chosen during configuration and whether the input module is local or remote. This method of communication uses the Producer/Consumer model. The input module is the producer of input data and the controller is the consumer of the data.

All ControlLogix inputs are updated asynchronously in relation to the controller's task execution. In other words, an input may be updated in the controller at any time during the controller's execution of the tasks it is configured to run. The input device determines when the input is sent based on its configuration.

The behavior of an input module also varies depending upon whether it operates in the local chassis or in a remote chassis. The following sections detail the differences in data transfers between local and remote installations.

### Input Modules

As shown in the illustration, ControlLogix input modules receive a signal at the RTB and process it internally through hardware, filters, and an ASIC scan before sending a signal to the backplane via the requested packet interval (RPI) or at a Change of State (COS) occurrence. The RPI is a configured interval of time that determines when module data is sent to the controller.



**Table 4 - Delay Factors that Affect Signal Propagation**

Delay	Description
Hardware	How the module is configured and the variance between the type of modules affects how the signal is processed.
Filter	User configuration varies between modules, thus affecting the signal propagation.
ASIC	ASIC scan = 200 $\mu$ s.

#### EXAMPLE

A typical delay time can be estimated despite the number of factors that can contribute. For example, if you are turning on a 1756-IB16 or 1756-IB16K module at 24V DC in 25 °C (77 °F) conditions, the signal propagation delay is affected by these factors:

- Hardware delay to energize the input (typically 290  $\mu$ s on the 1756-IB16 module)
- User-configurable filter time of 0 ms, 1 ms, or 2 ms
- ASIC scan of 200  $\mu$ s

In the worst case scenario with a filter time of 0 ms, the 1756-IB16 or 1756-IB16K module has a 490  $\mu$ s signal propagation delay.

These times are not guaranteed. For nominal and maximum delay times for each module, see the 1756 ControlLogix I/O Modules Specifications Technical Data, publication [1756-TD002](#).

## Input Modules in a Remote Chassis

If an input module physically resides in a chassis other than where the owner-controller resides, the role of the RPI and the module COS behavior changes slightly with respect to getting data to the owner.

The RPI and COS behavior still define when the module multicasts data within its own chassis, as described in the previous section. But, only the value of the RPI determines when the owner-controller receives it over the network.

### RPI

The RPI defines the slowest rate at which a module multicasts its data to the owner-controller. The time ranges from 200  $\mu$ s...750 ms and is sent to the module with all other configuration parameters. When the specified time frame elapses, the module multicasts data. This is also called a cyclic update.

### COS

COS instructs the module to transfer data whenever a specified input point transitions from On to Off or Off to On. The transition is referred to as a change of state.

---

**IMPORTANT** The module COS feature defaults to Enabled for both On-to-Off and Off-to-On.

---

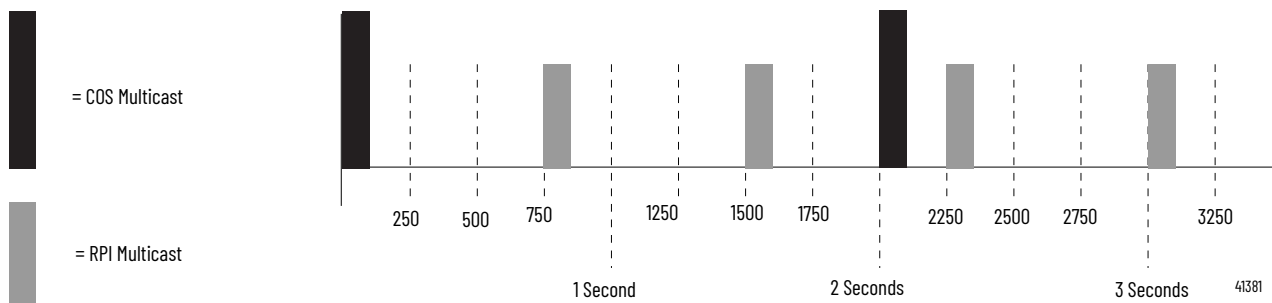
COS configuration occurs on a per-point basis, but all module data is multicast when any point enabled for COS changes state. COS is more efficient than RPI because it multicasts data only when a change occurs.

---

**IMPORTANT** You must specify an RPI regardless of whether you enable COS. If a change does not occur within the RPI time frame, the module still multicasts data at the rate specified by the RPI.

---

For example, if an input is changing state consistently every 2 seconds and the RPI is set at 750 ms, the data transfer looks like the illustration.



Because the RPI and COS functions are asynchronous to the program scan, it is possible for an input to change state during program scan execution. The point must be buffered to help prevent this from occurring. To buffer the point, you can copy the input data from your input tags to another structure and use the data from there.



To minimize traffic and conserve bandwidth, use a larger RPI value if COS is enabled and the module is in the same chassis as its owner-controller.

## Trigger Event Tasks

When configured, ControlLogix digital input modules can trigger an event task. The event task lets you execute a section of logic immediately when an event, or receipt of new data, occurs.

Your ControlLogix digital I/O module can trigger event tasks whenever module input data changes state. Refer to these considerations when using a digital input module to trigger an event task:

- Only one input module can trigger a specific event task.
- Input modules trigger the event task based on the module COS configuration. The COS configuration defines which points prompt the module to produce data if they turn On or Off. This production of data triggers the event task.
- Typically, enable COS for only one point on the module. If you enable COS for multiple points, a task overlap of the event task may occur.

For more information on event tasks, refer to the Logix 5000 Controllers Tasks, Programs, and Routines Programming Manual, publication [1756-PM005](#).

## Input Modules in a Remote Chassis

If an input module physically resides in a chassis other than where the owner-controller resides, the role of the RPI and the module COS behavior changes slightly with respect to getting data to the owner.

The RPI and COS behavior still define when the module multicasts data within its own chassis, as described in the previous section. But, only the value of the RPI determines when the owner-controller receives it over the network.

## Remote Input Modules Connected Via the EtherNet/IP Network

When remote digital input modules are connected to the owner-controller via an EtherNet/IP network, data is transferred to the owner-controller at these times:

- At the RPI, the module produces data within its own chassis.
- At the COS (if enabled), the 1756 EtherNet/IP communication module in the remote chassis immediately sends the module data over the network to the owner-controller as long as it has not sent data within a time frame that is one-quarter the value of the digital input module RPI. This helps prevent flooding the network with data.

For example, if a digital input module uses an RPI = 100 ms, the EtherNet/IP module sends module data immediately on receiving it if another data packet was not sent within the last 25 ms.

For more information about specifying an RPI rate, see the Logix5000 Controllers Design Considerations Reference Manual, publication [1756-RM094](#).

## Output Module Operation

An owner-controller sends output data to an output module when either one of two things occurs:

- At the end of every one of its tasks (local chassis only)
- At the rate specified in the module RPI

When an output module resides in a remote chassis with respect to the owner-controller, the owner-controller sends data to the output module **only** at the RPI rate that is specified for the module. Updates are not performed at the end of the owner-controller’s tasks.

Whenever the module receives data from the controller, it immediately multicasts the output commands it received to the rest of the system. The actual output data is echoed by the output module as input data and multicast back out onto the network. This is called output data echo.

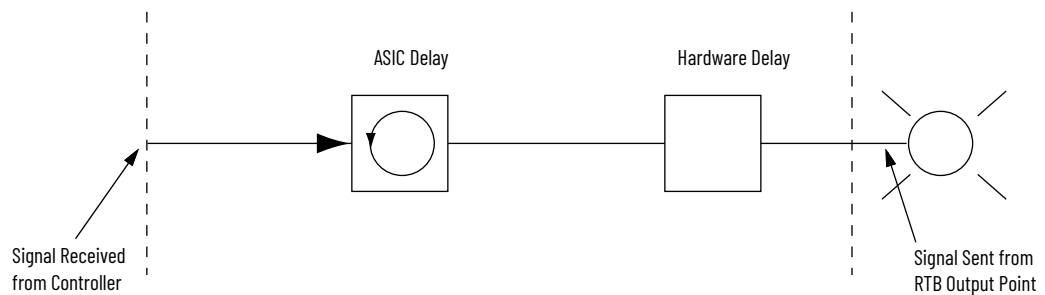
---

**IMPORTANT** In this Producer/Consumer model, the output module is the consumer of the controller’s output data and the producer of the data echo.

---

## Output Modules

ControlLogix output modules receive a signal from the controller and process it internally via hardware and an ASIC scan before sending a signal to the output device via the RTB.



42702

**Table 5 - Delay Factors that Affect Signal Propagation**

Delay	Description
Hardware	How the module is configured and the variance between the type of modules affects how the signal is processed.
ASIC	ASIC scan = 200 μs.

**EXAMPLE**

A typical delay time can be estimated despite the number of factors that can contribute. For example, if you are turning on a 1756-OB16E module at 24V DC in 25 °C (77 °F) conditions, the signal propagation delay is affected by these factors:

- Hardware delay to energize the output (typically 70 μs on the 1756-OB16E module)
- ASIC scan of 200 μs

In the worst case scenario with a filter time of 0 ms, the 1756-OB16E module has a 270 μs signal propagation delay.

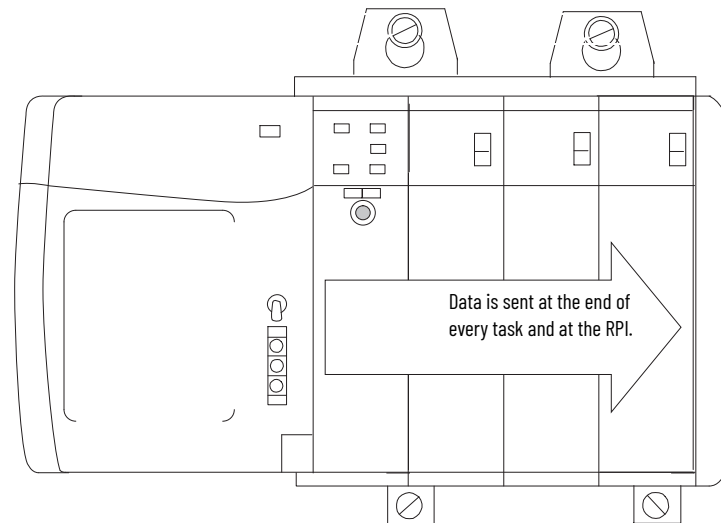
These times are not guaranteed. See [Chapter 6](#) for nominal and maximum delay times for each module.

## Output Modules in a Local Chassis

The owner-controller updates ControlLogix digital output modules in the local chassis at the end of every task and at the RPI.

When you specify an RPI value for a digital output module, you instruct the owner-controller when to broadcast the output data to the module. If the module resides in the same chassis as the owner-controller, the module receives the data almost immediately after the owner-controller sends it. Backplane transfer times are small.

**Figure 5 - Local Output Modules**



Depending on the value of the RPI with respect to the length of the program scan, the output module can receive and echo data multiple times during one program scan.

## Output Modules in a Remote Chassis

If an output module physically resides in a chassis other than that of the owner-controller, the owner-controller normally sends data to the output module at the RPI rate specified. Updates are not performed at the end of the controller's tasks.

In addition, the role of the RPI for a remote output module changes slightly with respect to getting data from the owner-controller.

### Remote Output Modules Connected Via the EtherNet/IP Network

When remote digital output modules are connected to the owner-controller via an EtherNet/IP network, the controller sends output data at these times:

- When the RPI timer expires
- When an Immediate Output (IOT) instruction, if programmed, is executed  
An IOT sends data immediately and resets the RPI timer.
- When a new schedule is created for a 1756-OB16IEFS module from the motion planner for a cam that has been armed by an MAOC instruction

Because the 1756-OB16IEFS module is the only 1756 module that can be used in a remote chassis with the MAOC instruction, it is the only module that receives output data in this scenario.

## Common Digital I/O Module Features

The 1756 digital I/O modules support these features.

### 1756 I/O Module Features

Module Type	Features
1756 digital AC input modules	<ul style="list-style-type: none"> <li>• Change of state: Software configurable</li> <li>• Timestamp of inputs: <math>\pm 200 \mu\text{s}</math></li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>
1756 digital AC output modules	<ul style="list-style-type: none"> <li>• Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>• Fault states per point: Hold last state, on or off (off is default)</li> <li>• States in Program mode per point: Hold last state, on or off (off is default)</li> <li>• Fusing: <ul style="list-style-type: none"> <li>- 1756-0A8D, 1756-0A8E: Electronically fused per point</li> <li>- 1756-0A16: Mechanically fused/group, 3.15 A @ 250V AC slow blow, 1500 A interruption current, Littelfuse p/n H2153.15</li> <li>- All other modules: Not protected. A fused IFM is recommended to protect outputs (see publication <a href="#">1492-TD008</a>)</li> </ul> </li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>
1756 digital DC input modules	<ul style="list-style-type: none"> <li>• Reverse polarity protection: All modules except 1756-IG16 and 1756-IG16K modules</li> <li>• Change of state: Software configurable</li> <li>• Timestamp of inputs: <ul style="list-style-type: none"> <li>- <math>\pm 100 \mu\text{s}</math> for sequence of events modules<sup>(1)</sup></li> <li>- <math>\pm 200 \mu\text{s}</math> for all other modules</li> </ul> </li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB Keying: User-defined mechanical</li> </ul>
1756 digital DC output modules	<ul style="list-style-type: none"> <li>• Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>• Fault states per point: Hold last state, on or off (off is default)</li> <li>• States in Program mode per point: Hold last state, on or off (off is default)</li> <li>• Fusing: <ul style="list-style-type: none"> <li>- 1756-0B8E1, 1756-0B16D, 1756-0B16E, 1756-0B16IEF, 1756-0B16IEFS, 1756-0V16E, 1756-0V32E: Electronically fused per point</li> <li>- All other modules not protected. A fused IFM is recommended to protect outputs. See publication <a href="#">1492-TD008</a>.</li> </ul> </li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>
1756 digital contact modules	<ul style="list-style-type: none"> <li>• Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>• Configurable fault states per point: Hold last state, on or off (off is default)</li> <li>• Configurable states in Program mode per point: Hold last state, on or off (off is default)</li> <li>• Fusing: Not protected. A fused IFM is recommended to protect outputs (See publication <a href="#">1492-TD008</a>)</li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>

(1) For details, see the ControlLogix Sequence of Events Module Installation Instructions, publication [1756-IN592](#), and the ControlLogix Sequence of Events Module User Manual, publication [1756-UM528](#).

---

**IMPORTANT** For the latest I/O module specifications, see the 1756 ControlLogix I/O Modules Technical Specifications, publication [1756-TD002](#).

---

## Removal and Insertion Under Power

All ControlLogix I/O modules may be inserted and removed from the chassis while power is applied. This feature enables greater availability of the overall control system. While the module is being removed or inserted, there is no additional disruption to the rest of the control process. This helps prevent an entire production line from having to be shut down.

## Module Fault Reporting

ControlLogix digital I/O modules provide both hardware and software indication when a module fault has occurred. Each fault status indicator and Logix Designer application graphically display this fault and include a fault message describing the nature of the fault. This feature lets you determine how your module has been affected and what action to take to resume normal operation.

The 1756-0B16IEF module extends this feature by enabling you to define the duration of time before the module transitions to On or Off after a fault occurs. For more information, see [Programmable Fault State Delay on page 55](#).

## Software Configurable

Logix Designer application provides an interface to configure each module. All module features are enabled or disabled through the I/O configuration within the software.

You can also use the software to retrieve this information from any module in the system:

- Serial number
- Firmware revision information
- Product code
- Vendor
- Error and fault information
- Diagnostic counters

## Electronic Keying

Electronic Keying reduces the possibility that you use the wrong device in a control system. It compares the device that is defined in your project to the installed device. If keying fails, a fault occurs. These attributes are compared.

**Table 6 - Electronic Keying**

Attribute	Description
Vendor	The device manufacturer.
Device Type	The general type of the product, for example, digital I/O module.
Product Code	The specific type of the product. The Product Code maps to a catalog number.
Major Revision	A number that represents the functional capabilities of a device.
Minor Revision	A number that represents behavior changes in the device.

The Electronic Keying options below are available.

**Table 7 - Electronic Keying Options**

Keying Option	Description
Compatible Module	Lets the installed device accept the key of the device that is defined in the project when the installed device can emulate the defined device. With Compatible Module, you can typically replace a device with another device that has these characteristics: <ul style="list-style-type: none"> <li>• Same catalog number</li> <li>• Same or higher Major Revision</li> <li>• Minor Revision as follows:               <ul style="list-style-type: none"> <li>• If the Major Revision is the same, the Minor Revision must be the same or higher.</li> <li>• If the Major Revision is higher, the Minor Revision can be any number.</li> </ul> </li> </ul>
Disable Keying	Indicates that the keying attributes are not considered when attempting to communicate with a device. With Disable Keying, communication can occur with a device other than the type specified in the project. <b>ATTENTION:</b> Be extremely cautious when using Disable Keying; if used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We <b>strongly recommend</b> that you <b>do not use</b> Disable Keying. If you use Disable Keying, you must take full responsibility for understanding whether the device being used can fulfill the functional requirements of the application.
Exact Match	Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.

Carefully consider the implications of each keying option when selecting one.

### IMPORTANT

Changing Electronic Keying parameters online interrupts connections to the device and any devices that are connected through the device. Connections from other controllers can also be broken. If an I/O connection to a device is interrupted, the result can be a loss of data.

For more detailed information about Electronic Keying, see Electronic Keying in Logix5000™ Control Systems Application Technique, publication [LOGIX-AT001](#).

## Module Inhibiting

Module inhibiting lets you indefinitely suspend a connection between an owner-controller and a digital I/O module without having to remove the module from the configuration. This process lets you temporarily disable communication to a module, such as to perform maintenance. You can use module inhibiting in these ways:

- You write a configuration for an I/O module but inhibit the module to help prevent it from communicating with the owner-controller. In this case, the owner does not establish a connection and the configuration is not sent to the module until the connection is uninhibited.
- In your application, a controller already owns a module, has downloaded the configuration to the module, and is exchanging data over the connection between the devices. In this case, you can inhibit the module and the owner-controller behaves as if the connection to the module does not exist.

---

**IMPORTANT**

Whenever you inhibit an output module, it enters Program mode, and all outputs change to the state configured for Program mode. For example, if an output module is configured so that the state of the outputs transition to zero during Program mode, whenever that module is inhibited, outputs transition to zero.

---

You may need to use module inhibiting in these instances:

- Multiple controllers own the same digital input module. A change is required in the module configuration. However, the change must be made to the program in all controllers. In this case, you follow these steps.
  - a. Inhibit the module.
  - b. Change configuration in all controllers.
  - c. Uninhibit the module.
- You want to upgrade a digital I/O module. We recommend you use this procedure.
  - a. Inhibit the module.
  - b. Perform the upgrade.
  - c. Uninhibit the module.
- You are using a program that includes a module that you do not physically possess yet, and you do not want the controller to continually look for a module that does not yet exist. In this case, you can inhibit the module in your program until it physically resides in the proper slot.

## Producer/Consumer Communication

By using Producer/Consumer communication, ControlLogix I/O modules can produce data without first being polled by a controller. The modules produce the data and any other owner-controller device can decide to consume it.

For example, an input module produces data and any number of processors can consume the data at the same time. This eliminates the need for one processor to send the data to another processor. For more information about this process, see [Input Module Operation on page 21](#).

## Status Indicator Information

Each ControlLogix digital I/O module has a status indicator on the front of the module that lets you check the health and operational status of a module. The status indicator displays vary for each module.

**Table 8 - Status Indicator Displays**

Status	Description
I/O status ST	This yellow display indicates the On/Off state of the field device. <b>IMPORTANT:</b> For the 1756-0A8D and 1756-0A8E modules, the I/O status indicator does not illuminate without field power applied.
Module status OK	This green display indicates the module communication status.
Fault status FLT	This display is only found on some modules and indicates the presence or absence of various faults.
Fuse status Fuse	This display is only found on electronically fused modules and indicates the state of the module fuse.

See [Status Indicators on page 89](#) for examples of status indicators on ControlLogix digital I/O modules.

## Use the System Clock to Timestamp Inputs and Schedule

This section describes how to use CST timestamps in standard and diagnostic I/O modules and the CIP Sync™ timestamps in fast I/O modules.



### *Use Coordinated System Time with Standard and Diagnostic I/O Modules*

Time masters generate a 64-bit coordinated system time (CST) for their respective chassis.

You can configure your digital input modules to access the CST and timestamp input data with a relative time reference of when that input data changes state.

<b>IMPORTANT</b>	Because only one CST value is returned to the controller when any input point changes state, we recommend that you use timestamping on only one input point per module.
------------------	---

**Table 9 - How to use CST timestamps**

Topic	Description
Timestamping for a sequence of events	<p>The CST can be used to establish a sequence of events occurring at a particular input module point by timestamping the input data. To determine a sequence of events, you must do the following:</p> <ul style="list-style-type: none"> <li>Set the communication format of the input module to CST Timestamped Input Data.</li> <li>Enable COS for the input point where a sequence occurs, and disable COS for all other points on the module.</li> </ul> <p> If you decide to configure multiple input points for COS, your module generates a unique CST each time any of those input points change state, as long as the changes do not occur within 500 μs of each other. If multiple input points configured for COS change state within 500 μs of each other, a single CST value is generated for all making it appear that they changed at exactly the same time.</p>
Timestamping in conjunction with scheduled outputs	<p>Timestamping can be used in conjunction with the scheduled outputs feature, so that after input data changes state and a timestamp occurs, an output point actuates at some configured time in the future. You can schedule outputs up to 16 seconds into the future. When you use timestamping of inputs and scheduled outputs, you must do the following:</p> <ul style="list-style-type: none"> <li>Choose a communication or connection format for each input and output module that enables timestamping. For more information, see <a href="#">Module Definitions, Communication, or Connection Formats on page 15</a>.</li> <li>Have a time master in the same chassis as both I/O modules.</li> <li>Disable COS for all input points on the input module except the point being timestamped.</li> </ul> <p> For scheduled outputs to work most effectively, remember these items:</p> <ul style="list-style-type: none"> <li>The time to schedule outputs to transition in the future must account for any controller, backplane, and network delays.</li> <li>The I/O modules must reside in the same rack as the time master.</li> </ul>

## Use CIP Sync Time with Fast I/O Modules

The 1756-IB16IF, 1756-OB16IEF, and 1756-OB16IEFS modules use CIP Sync for both timestamps and scheduling.

CIP Sync is a CIP™ implementation of the IEEE 1588 PTP (Precision Time Protocol). CIP Sync provides accurate real-time (Real-World Time) or Universal Coordinated Time (UTC) synchronization of controllers and devices that are connected over CIP networks. This technology supports highly distributed applications that require timestamping, sequence of events recording, distributed motion control, and increased control coordination.

The 1756-IB16IF, 1756-OB16IEF, and 1756-OB16IEFS modules are CIP Sync slave-only devices. There must be another module on the network that functions as a master clock. For more information on how to use CIP Sync technology, see the Integrated Architecture and CIP Sync Configuration Application Technique, publication [IA-ATO03](#).

Fast I/O modules can be used to capture timestamps and schedule outputs like CST-based modules while providing these advantages:

- Fast I/O modules have much higher precision than CST-based modules.
- Inputs are timestamped by point, so multiple inputs can be configured for COS without losing timestamp data.
- CIP Sync is system-wide, so timestamp and schedule values are consistent across all modules in the system. For instance, using 1756-IB16IF input timestamps to schedule outputs on a 1756-OB16IEF module means that the controller, input module, and output module are not restricted to the same chassis as is the case with CST-based I/O.
- Output modules use all 64 bits of the timestamp to schedule, so there are no limits on schedule ranges.

## Mixing CST and CIP Sync Modules in a ControlLogix System

CST is automatically enabled for each chassis that has been configured to use CIP Sync. Therefore, it is possible to include modules that use CST for their time base into systems that have been configured to use CIP Sync. Also, there is a direct correlation between CIP Sync system time and the local chassis CST time.

The CIP Sync system time and local chassis CST time are related by this equation:

$$\text{CIP Sync system time} = \text{CST time} + \text{offset}$$

The offset in the equation is a value unique to each chassis and can be obtained by using one of these methods:

- `CSTOffset` from the `WallClockTime (WCT)` object of a controller in the chassis
- `SystemOffset` from the `Time Synchronize` object of a controller in the chassis
- `LocalClockOffset` returned in an I/O connection from a CIP Sync capable module in the chassis

This relationship enables CST and CIP Sync-based I/O to interoperate as long as the offset in the chassis containing the CST-based module is accessible.

## Input Module Features

ControlLogix digital input modules interface to sensing devices and detect whether they are On or Off.

ControlLogix input modules convert AC or DC On/Off signals from user devices to appropriate logic level for use within the processor. Typical input devices include the following:

- Proximity switches
- Limit switches
- Selector switches
- Float switches
- Push-button switches

When designing systems with ControlLogix input modules, consider these factors:

- Voltage necessary for your application
- Current leakage
- Whether you need a solid-state device
- Whether your application uses sinking or sourcing wiring

### Data Transfer on Either Cyclic Time or Change of State (COS)

Digital input modules always send data at the RPI, but they send data at a change of state only if the COS feature is enabled. COS is more efficient than RPI because it multicasts data only when a change occurs.

The table describes the two ways that a module sends data to the owner-controller.

**Table 10 -**

Method	Description
RPI	A user-defined rate at which the module updates the information sent to its owner-controller. This is also known as Cyclic Data Transfer.
COS	Configurable feature that, when enabled, instructs the module to update its owner-controller with new data whenever a specified input point transitions from On to Off and Off to On. The data is sent at the RPI rate when there is no change of state. By default, this setting is always enabled for input modules.

### Isolated and Nonisolated Varieties of Input Modules

ControlLogix input modules provide isolated or nonisolated wiring options. Some applications require power for the I/O circuits to originate on separate, isolated power sources. Because these conditions require separate commons for each channel, some input modules use individual isolation, or point-to-point isolation so if one point faults, the others continue to operate.

Other types of isolation available with ControlLogix input modules are channel-to-channel isolation and no isolation. Your application determines what type of isolation is necessary and which input module to use.

### Multiple Input Point Densities

ControlLogix input modules use either 8-, 16-, or 32-point densities for greater flexibility in your application. A point is the termination where a wire attaches to the input module from a field device. The module receives information from the device to this designated point, thus signaling when activity occurs.

## RPI

The Connection tab on the Module Properties dialog box lets you enter an RPI. The RPI guarantees the slowest rate at which data is multicast.

The actual data transfer rate of the module can be faster than the RPI setting. But, the RPI provides a defined, maximum period of time when data is transferred to the owner-controller.

For more information on configuring this feature, see Chapter X - Configure Digital I/O Modules.

## Change of State

The Point column on the left side of the Configuration tab lets you set whether a COS occurs when a field device transitions from Off to On or On to Off.

For more information on configuring this feature, see Chapter X - Configure Digital I/O Modules.

## Software Configurable Filter Times

On-to-Off and Off-to-On filter times can be adjusted through Logix Designer application for all ControlLogix input modules. These filters improve noise immunity within a signal. A larger filter value affects the length of delay times for signals from these modules.

---

**IMPORTANT** Input filters on the 1756-IB16IF module function differently than other digital I/O modules. For information about input filters on the 1756-IB16IF module, see [Fast Input Module Tags on page 125](#).

---

For more information on configuring this feature, see Chapter X - Configure Digital I/O Modules.

## Output Module Features

ControlLogix output modules can be used to drive various output devices. Typical output devices compatible with ControlLogix outputs include these items:

- Motor starters
- Solenoids
- Indicators

Follow these guidelines when designing a system:

- Make sure that the ControlLogix outputs can supply the necessary surge and continuous current for proper operation.
- Make sure that the surge and continuous current are not exceeded. Damage to the module could result.

When sizing output loads, refer to the documentation supplied with the output device for the surge and continuous current needed to operate the device.

The ControlLogix standard digital outputs are capable of directly driving the ControlLogix standard digital inputs. The exceptions are the AC and DC diagnostic input modules. When diagnostics are used, a shunt resistor is required for leakage current.

For information on the compatibility of motor starters with ControlLogix output modules, see [Appendix D](#).

## Output Data Echo

During normal operation, when a controller sends out an output command to the ControlLogix system, the output module that is targeted for that command returns the commanded state of the output to the system. This process verifies that the module has received the command and tried to execute it.

Other devices can use this broadcast signal through a Listen-only connection to determine the desired state of the output without having to interrogate the owner-controller.

### Monitor Fault Bits

The output data echo only matches the commanded state of the outputs if the module is operating under normal conditions. If there is an anomaly with the module, the commanded state and the output data echo may not match.

You can monitor the fault bits for your output points for fault conditions. If a fault occurs, the fault bit is set and your program alerts you to the condition. In this case, the output data echo may not match the commanded state of the outputs.

If there is a mismatch between the commanded state of the outputs and the output data echo, check your output module for these conditions:

- Communication fault.
- Connection is inhibited.
- Blown fuse—The module does not turn on an output if an overload or short circuit is detected.
- (1756-0A8D and 1756-0A8E only) Loss of field power—The module does not turn on an output unless AC power is detected.

Redundant Owner, but this controller is not the Active owner.

## Isolated and Nonisolated Varieties of Output Modules

As with input modules, ControlLogix output modules provide isolated or nonisolated wiring options. I/O modules provide point-to-point, group-to-group, or channel-to-channel wiring isolation. Your specific application determines what type of isolation is necessary and which output module to use.

---

**IMPORTANT** Although some ControlLogix I/O modules provide nonisolated, field-side wiring options, each I/O module maintains internal electrical isolation between the system side and field side.

---

## Time-scheduled Output Control

Time-scheduled output control is available for these modules:

- 1756-OB16IS—Provides time-scheduled output control in CST time for outputs 0...7. Enables schedules with a minimum interval of 100  $\mu$ s.
- 1756-OB16IEFS—Provides time-scheduled output control in CIP Sync time for outputs 0...15. Enables schedules with a minimum interval of 5  $\mu$ s.

By using the time-scheduled output control feature, the module can turn the outputs On or Off at a scheduled time. You can set the time for the output to turn On or Off in program logic. The modules manage the time locally, so that the output is turned On or Off at the specified time.

### MAOC Instructions with Time-scheduled Output Control

The Motion Axis Output Cam (MAOC) instruction provides position-based control of outputs by using position and velocity information of any motion axis. When the 1756-OB16IS or 1756-OB16IEFS module is specified as the output destination for the MAOC instruction, the MAOC instruction automatically handles time-based scheduling for outputs. The benefit of using output scheduling in this manner is that the resolution of the output control is improved from the motion coarse update rate (typically 1...32 ms) to 100  $\mu$ s for outputs 0...7 on the 1756-OB16IS module and 10  $\mu$ s for outputs 0...15 on the 1756-OB16IEFS module.

You can also use outputs 8...15 on the 1756-OB16IS module with the MAOC instruction. However, only outputs 0...7 have 100  $\mu$ s resolution. Outputs 8...15 are updated at the motion coarse update rate.

---

**IMPORTANT** The 1756-OB16IS Scheduled Output Module can be associated with one MAOC axis/execution target only.

---

For more information about using the MAOC instruction with scheduled output modules, refer to the Position-based Output Control with the MAOC Instruction Application Technique, publication [1756-AT017](#).

### Module Major Revision Considerations with Timestamping

When using timestamping for inputs or diagnostic timestamping of I/O modules, these conditions can occur depending on the major revision of the module:

- If the module has a Major Revision = 1 (unless a 1756-IB16IF, 1756-IB16ISOE, or 1756-IH16ISOE module), it always returns a positive timestamping value.
- If the module has a Major Revision > 2 or is a 1756-IB16IF, 1756-IB16ISOE, or 1756-IH16ISOE module, it returns a negative timestamping value until the module is synchronized with the owner-controller and the first change of state condition occurs.

Use the Module Properties dialog box in Logix Designer application to determine if the module has been synchronized with the owner-controller and whether the controller is synchronized with the CST. For more information on synchronizing owner-controllers and modules with the CST, see the ControlLogix System User Manual, publication [1756-UM001](#).

## Configurable Point-level Output States

Individual outputs can be configured to unique output states if the module goes into Program mode or Fault mode.

---

**IMPORTANT** Whenever you inhibit an output module, it enters Program mode, and all outputs change to the state configured for Program mode. For example, if an output module is configured so that the states of outputs turn Off during Program mode, whenever that module is inhibited, outputs turn Off.

---

For more information on configuring this feature, see [page 70](#).

## Electronic Fusing

Some digital outputs have internal electronic or mechanical fusing to help prevent too much current from flowing through the module. This feature protects the module from electrical damage. Other modules require external fusing.

Modules that use electronic fusing are fused on either a per point basis or per group basis to protect output points from the surge of too much current. If too much current begins to flow through a point, the fuse is tripped and a point-level fault is sent to the controller. A corresponding tag can be examined in the event of a fault. For more information about fault tags, see [Appendix A](#).

These modules use electronic fusing:

- 1756-0A8E
- 1756-0B8EI
- 1756-0A8D
- 1756-0B16D
- 1756-0B16E
- 1756-0V16E
- 1756-0V32E
- 1756-0B16IEF
- 1756-0B16IEFS

Refer to this table to determine what fuse to use in your application. If your module does not support fusing, you can use a fused IFM to protect outputs. See publication [1492-TD008](#).

Table 11 - Recommended Fuses

Circuit Type	Cat. No.	Fusing on the module	Recommended Fuse	Fuse Supplier
AC	1756-0A8 <sup>(1)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 6.3 A Medium lag	SAN-O Industry Corp. (SOC) PN MT 4-6.3A
	1756-0A8D <sup>(2) (3)</sup>	Yes—Fused on a per point basis	Electronically fused	
	1756-0A8E <sup>(2) (3)</sup>			
	1756-0A16 <sup>(1) (4) (5)</sup>	Yes—Fused on a per group basis	5x20mm 3.15 A Slo-Blow 1500 A Interruption current	Littelfuse p/n H2153.15
	1756-0A16 <sup>(1)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 6.3 A Medium lag	SOC p/n MT 4-6.3A
	1756-0N8			
DC	1756-0B8 <sup>(6)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 4 A Quick acting	SOC p/n MQ2-4A
	1756-0B8I <sup>(6)</sup>			
	1756-0B8E <sup>(2) (3) (6)</sup>	Yes—Fused on a per point basis	Electronically fused	
	1756-0B16D <sup>(2) (3) (7)</sup>			
	1756-0B16E <sup>(2) (5) (6)</sup>			
	1756-0B16I <sup>(6) (8)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 4 A Quick acting	SOC p/n MQ2-4A
	1756-0B16IEF <sup>(2) (3) (6)</sup>	Yes—Fused on a per point basis	Electronically fused	
	1756-0B16IEFS <sup>(2) (3) (6)</sup>			
	1756-0B16IS <sup>(6) (8)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 4 A Quick acting	SOC p/n MQ2-4A
	1756-0B32 <sup>(6) (8)</sup>		5x20mm 800 mA	Littelfuse p/n SP001.1003 or Schurter p/n 216.800
	1756-0C8 <sup>(6)</sup>		5x20mm 4 A Quick acting	SOC p/n MQ2-4A
	1756-0G16 <sup>(6)</sup>			
	1756-0H8I <sup>(6) (8)</sup>			
	1756-0V16E <sup>(2) (5) (6)</sup>	Yes—Fused on a per group basis	Electronically fused	
1756-0V32E <sup>(2) (5) (6)</sup>				
Relay	1756-0W16I <sup>(8)</sup>	None—Fused IFM can be used to protect outputs	5x20mm 6.3 A Medium lag	SOC p/n MT 4-6.3A
	1756-0X8I <sup>(8)</sup>			

- (1) For voltages above 132V AC, the Interface Modules (IFM) are not an acceptable means to provide external fusing. A rated terminal block for the intended application must be used.
- (2) Electronic protection is not intended to replace fuses, circuit breakers, or other code-required wiring protection devices.
- (3) The electronic protection of this module has been designed to provide protection for the module from short-circuit conditions. The protection is based on a thermal cut-out principle. In the event of a short-circuit condition on an output channel, that channel limits the current within milliseconds after its thermal cut-out temperature has been reached. All other channels with a NUT of that group continue to operate as directed by the module master (CPU, bridge, and so forth).
- (4) A fuse is provided on each common of this module for a total of two fuses. The fuses are designed to protect the module from short circuit conditions. The fuse does not provide overload protection. In the event of an overload on an output channel, it is likely that the fuse does not blow and the output device associated with that channel is damaged. To provide overload protection for your application, install user-supplied fuses externally.
- (5) If a short circuit condition occurs on any channel within this module group, the entire group is turned Off.
- (6) The module does not provide protection against reverse polarity wiring or wiring to AC power sources.
- (7) The electronic protection of this module has been designed to provide protection for the module from short-circuit conditions. The protection is based on a thermal cut-out principle. In the event of a short-circuit condition on an output channel, that channel limits the current within milliseconds after its thermal cut-out temperature has been reached. Other channels could produce a false error on the output verify fault signal due to the supply dropping below the minimum detect level of 19.2V DC. The output channels that are affected by this phenomenon continues to operate as directed by the module master (CPU, bridge, and so forth). What this means is that the output verify fault signals of the other channels must be checked and reset if a short-circuit on one channel occurs.
- (8) The recommended fuse for this module has been sized to provide short circuit protection for wiring only to external loads. In the event of a short circuit on an output channel, it is likely that the transistor or relay associated with that channel is damaged and the module can be replaced or a spare output channel used for the load. The fuse does not provide overload protection. In the event of an overload on an output channel, it is likely that the fuse does not blow and the transistor or relay associated with that channel is damaged. To provide overload protection for your application, user supplied fuse can be installed externally and properly sized to match the individual load characteristics.

You can reset an electronic fuse through the Logix Designer application during online monitoring or through program logic running on a controller. If your module uses point-level fusing, you can reset a fuse with a CIP Generic message instruction, as described on [Reset a Fuse, Perform Pulse Test, and Reset Latched Diagnostics on page 154](#).

## Field Power Loss Detection

For the standard digital output modules, the Field Power Loss detection feature is found on the **1756-0A8E module only**. When field power to the module is lost, or zero cross cannot be detected, a point-level fault is sent to the controller to identify the exact point faulted.

---

**IMPORTANT** Only enable Field Power Loss detection for points that are in use. If this feature is enabled for points that are not in use, you receive faults for those points during operation.

---

This feature has a corresponding tag that can be examined in the user program in the event of a fault. For information on these tags, see [Appendix A](#).

## Diagnostic Latch of Information

The diagnostic latch feature is available for the **1756-0A8E modules only**. Diagnostic latching lets this module latch a fault in the set position once it has been triggered, even if the error condition causing the fault to occur disappears.

**Notes:**

## Digital Diagnostic Module Features

### Common Diagnostic Features

Below are features common to all ControlLogix® digital diagnostic I/O modules. Diagnostic I/O Modules also have the common features described in [Common Digital I/O Module Features](#).

- Diagnostic Timestamp
- 8-Point AC/16-Point DC
- Point-level Fault Reporting
- Diagnostic Latch of Information

### Diagnostic Timestamp

Diagnostic I/O modules can timestamp the time when a fault occurs or when it clears. This feature provides greater accuracy and flexibility in running applications. Modules use the ControlLogix system clock from a local controller to generate timestamps.

To use diagnostic timestamps, you must choose the appropriate communication format during initial configuration. To configure features

### 8-Point AC/16-Point DC

Diagnostic I/O modules provide various grouping of points on different modules. The eight-point AC modules and 16-point DC modules provide additional flexibility when designing module applications. The greater number of points lets more field devices be attached to I/O modules to boost efficiency.

### Point-level Fault Reporting

Diagnostic I/O modules set bits to indicate when a fault has occurred on a point-by-point basis. These fault conditions generate their own unique fault bits.

**Table 12 - Unique Fault Bits for I/O Points**

Input Points	Output Points
These conditions can set a fault bit for an input point: <ul style="list-style-type: none"> <li>• Open wire</li> <li>• Field power loss (1756-IA8D only)</li> </ul>	These conditions can set a fault bit for an output point: <ul style="list-style-type: none"> <li>• Fuse blown</li> <li>• No load</li> <li>• Output verify</li> <li>• Field power loss (1756-IA8D only)</li> </ul>

Using these bits in tandem with data echo and manually performing a pulse test can help to further isolate the fault.

**Table 13 - 1756-0A8D Point-level Fault Scenarios**

Ladder commands output to be On	Ladder commands output to be Off	Possible cause of fault
1. Output Data Echo returns the state of the output as Off. 2. Fuse Blown bit is set.	1. Output Data Echo returns the state of the output as Off. <sup>(1)</sup> 2. Pulse Test fails.	Output is shorted to L2.
1. Output Data Echo returns the state of the output as On. 2. Pulse Test fails. <sup>(2)</sup>	1. Output Data Echo returns the state of the output as Off. 2. No Load bit is off.	No Load or output is shorted to L1.
1. Output Data Echo returns the state of the output as Off. 2. No Load shows a fault. 3. Field Power Loss shows a fault. 4. Pulse Test fails.	1. Output Data Echo returns the state of the output as Off. 2. No Load bit is set. 3. Field Power Loss is set. 4. Pulse Test fails.	L1 or L2 are disconnected or outside the 47...63 Hz frequency range.
1. Output Data Echo returns the state of the output as On. <sup>(3)</sup> 2. Output Verify bit is set. <sup>(4)</sup>	1. Data Echo returns the state of the output as Off. 2. Pulse Test fails.	Hardware point damage. <sup>(5)</sup>

- (1) It is not possible to create a Fuse Blown fault in the Off state. If a short-circuit occurs, the output point is turned Off and the fault appears in the Off state until the point is reset.
- (2) When pulse test is executed, it is normal operation to see a momentary pulsation on the module display.
- (3) The output cannot turn On due to hardware point damage.
- (4) Depending on the characteristics of an applied short-circuit, an output verify fault could be set until the short-circuit is detected by the module and the output is turned Off.
- (5) During normal operating conditions, hardware damage is not possible. An output shorted to L2 may temporarily cause a hardware point fault. See output shorted to L2 as a possible cause.

This table lists possible diagnostic faults on the 1756-0B16D module.

**Table 14 - 1756-0B16D Point-level Fault Scenarios**

Ladder commands output to be On	Ladder commands output to be Off	Possible cause of fault
1. Output Data Echo returns the state of the output as Off. 2. Fuse Blown bit is set. <sup>(1)</sup>	1. Output Data Echo returns the state of the output as Off. <sup>(2)</sup> 2. Pulse Test fails. <sup>(3)</sup>	Output is shorted to GND.
1. Output Data Echo returns the state of the output as On. 2. Pulse Test fails	1. Output Data Echo returns the state of the output as Off. 2. No Load bit is set. 3. Pulse Test passes.	One of the following could be the cause. 1. No Load. 2. Output shorted to DC+. 3. No power at module.
1. Output Data Echo returns the state of the output as On. <sup>(4)</sup> 2. Output Verify sets a bit. <sup>(5)</sup>	1. Output Data Echo returns the state of the output as Off. 2. Pulse Test fails.	Hardware point damage. <sup>(6)</sup>

- (1) The electronic protection of this module has been designed to provide protection for the module from short-circuit conditions. The protection is based on a thermal cutout principal. In the event of a short-circuit condition on an output channel, that channel limits the current within milliseconds after its thermal cutout temperature has been reached. Other channels could produce a false error on the output verify fault signal due to the supply dropping below the minimum detect level of 19.2V DC. The output channels that are affected by this phenomenon continues to operate as directed by the module master (CPU, bridge, and so forth). What this means is that the output verify fault signals of the other channels must be checked and reset if a short-circuit on one channel occurs.
- (2) It is not possible to create a Fuse Blown fault in the Off state. If a short-circuit occurs, the point is turned Off and the fault appears in the Off state until that point is reset.
- (3) When the pulse test is executed, it is normal operation to see a momentary pulsation on the module display.
- (4) The output cannot turn On due to hardware point damage.
- (5) Depending on the characteristics of an applied short-circuit, an output verify fault could be set until the short-circuit is detected by the module and the output is turned Off.
- (6) During normal operating conditions, hardware damage is not possible. An output shorted to GND may temporarily cause a hardware point fault. See output shorted to GND as a possible cause.

## Diagnostic Latch of Information

Diagnostic latching lets diagnostic I/O modules latch a fault in the set position once it has been triggered, even if the error condition causing the fault to occur disappears.

## Diagnostic Input Module Features

When designing systems with ControlLogix diagnostic input modules, consider these factors:

- Voltage necessary for your application
- Current leakage
- Whether you need a solid-state device
- Whether your application must use sinking or sourcing wiring

### Fault and Status Reporting between Input Modules and Controllers

ControlLogix diagnostic digital input modules multicast fault and status data to any owner-controller or listening controller. All diagnostic input modules maintain a module-fault word, the highest level of fault reporting. Some modules use additional words to indicate fault conditions.

[Table 15](#) lists the fault words and the associated tags that can be examined in program logic to indicate when a fault has occurred for a diagnostic input module.

**Table 15 - Fault Words on Diagnostic Input Modules**

Word	Tag Name	Description
Module-fault	Fault	Provides fault summary reporting. Available on all digital input modules.
Field Power Loss	FieldPwrLoss	Indicates loss of field power to a group on the module. Available on the <b>1756-IA8D only</b> . For more information, see <a href="#">Field Power Loss Detection on page 43</a> .
Open Wire	OpenWire	Indicates the loss of a wire from a point on the module. For more information, see <a href="#">Open Wire Detection on page 42</a> .

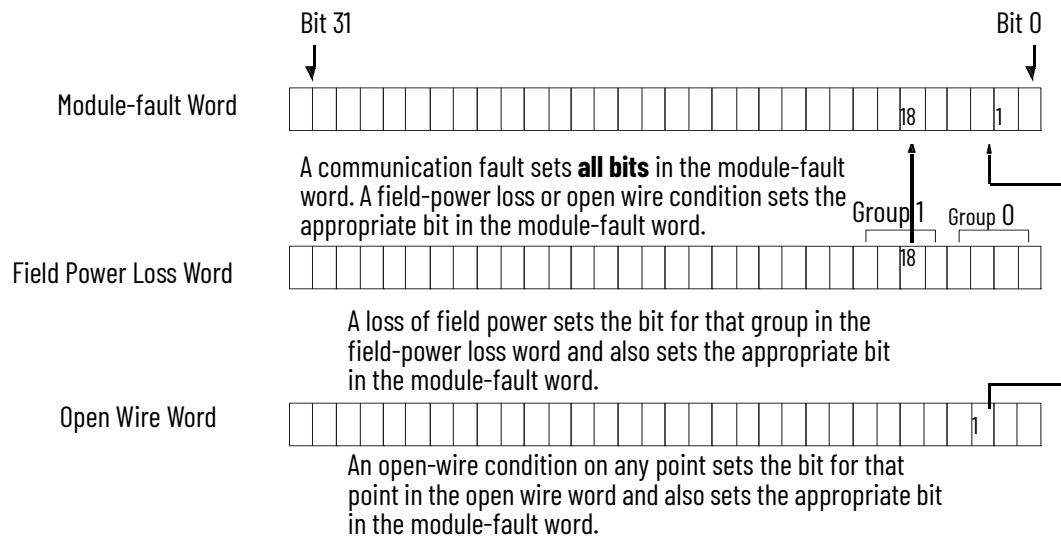
All words are 32 bit, although only the number of bits appropriate for each module's density are used. For example, the 1756-IA16I module has a module-fault word of 32 bits. But, because this is a 16-point module, only the first 16 bits (0...15) are used in the module-fault word.

Fault bits in the field-power loss word and open wire word are logically entered into the module-fault word. Depending on the module type, a bit set in the module-fault word can mean multiple things, as indicated in the table.

**Table 16 - Bits Set in Module-fault Word**

Condition	Bits Set
Communications fault	All 32 bits are set to 1, regardless of the module's density.
Field-power loss	Only the bit affected is set to 1.
Open wire	

This illustration provides an overview of the fault reporting process for digital input modules.



## Diagnostic Change of State for Input Modules

If the diagnostic change of state feature is enabled, a diagnostic input module sends new data to the owner-controller when one of the events that are described in the table occurs.

Event	Description
RPI	A user-defined rate at which the module updates the information sent to its owner-controller. This is also known as Cyclic Data Transfer.
Change of State	Configurable feature that, when enabled, instructs the module to update its owner-controller with new data whenever a specified input point transitions from On to Off and Off to On. The data is sent at the RPI rate where there is no change of state. By default, this setting is always enabled for input modules.
Diagnostic Change of State	Information updates when any change in the diagnostics for an input module occurs.

Although the RPI occurs continuously, the COS feature lets you decide whether changes in a module’s diagnostic detection cause the module to send real-time data to the owner-controller.

1. On the Module Properties dialog box, click the Points tab.

Point	Enable Change of State		Enable Diagnostics for	Enable Diag. Latching
	<input checked="" type="checkbox"/> Off->On	<input checked="" type="checkbox"/> On->Off	<input checked="" type="checkbox"/> Open Wire	
00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
02	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
03	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
04	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
05	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
06	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
07	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
08	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
09	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2. Do the following in the Enable Change of State column:
  - To enable the input module to send new data to the owner-controller at the RPI, on input COS if it is enabled, and if a diagnostic fault occurs, check the corresponding Off ? On or On ?Off checkbox for a point.
  - To disable the feature, clear the corresponding checkbox for a point. Real-time data is not sent when a diagnostic fault occurs but is still sent at the specified RPI or on input COS if it is enabled.
3. Click OK.

## Open Wire Detection

Open Wire is used to verify that the field wiring is connected to the module. The field device must provide a minimum leakage current to function properly.

A leakage resistor must be placed across the contacts of an input device. The resulting current is then expected to exist when the input is open. For more information, see each module’s specifications in [Chapter 6](#).

When an Open Wire condition is detected, a point-level fault is sent to the controller to identify the exact point fault. This feature has a corresponding tag that can be examined in the user program in the event of a fault.

## Field Power Loss Detection

For the standard digital output modules, the Field Power Loss detection feature is found on the **1756-IA8D module only**. When field power to the module is lost, or zero cross cannot be detected, a point-level fault is sent to the controller to identify the exact point faulted.

---

**IMPORTANT** Only enable Field Power Loss detection for points that are in use. If this feature is enabled for points that are not in use, you receive faults for those points during operation.

---

This feature has a corresponding tag that can be examined in the user program in the event of a fault. For information on these tags, see [Chapter 5](#).

## Diagnostic Output Module Features

ControlLogix diagnostic output modules are capable of directly driving the ControlLogix diagnostic digital inputs. When diagnostics are used, a shunt resistor is required for leakage current.

For more information on the compatibility of motor starters with ControlLogix output modules, see [Appendix D](#).

## Field Wiring Options

As with diagnostic input modules, ControlLogix diagnostic output modules provide isolated or nonisolated wiring options. I/O modules provide point-to-point, group-to-group, or channel-to-channel wiring isolation.

Your specific application determines what type of isolation is necessary and which output module to use.

---

**IMPORTANT** Although some ControlLogix diagnostic I/O modules provide nonisolated field-side wiring options, each I/O module maintains internal electrical isolation between the system side and field side.

---

## Pulse Test

Pulse test is a feature found on diagnostic output modules that can verify output-circuit functionality without actually changing the state of the output load device. A short pulse is sent to the targeted output circuit. The circuit responds as if a real change of state command was issued, but the load device does not transition.

See [Reset a Fuse, Perform Pulse Test, and Reset Latched Diagnostics on page 154](#) for instructions on performing a pulse test with a CIP Generic Message instruction.



Consider the following when using the pulse test:

- Only use the test when the output state does not transition for long periods of time. Normal diagnostics catch faults if the outputs are transitioning regularly.
- When first performing the pulse test, verify that the load does not transition. Be at the actual load while the test is performed.

This table explains how a pulse test can be used to perform a preemptive diagnosis of possible future module conditions.

**Table 17 - Pulse Test Descriptions**

Objective	Pulse Test Description
Detect a blown fuse before it happens	The Blown Fuse diagnostic can be used only when an output module is in the On state. However, you can use a pulse test when an output module is in an Off state to determine whether operating conditions may cause a blown fuse. When you perform a pulse test on a module in the Off state, the output point is commanded to be On briefly. Although no diagnostic bits are set in the output data echo, the pulse test reports a failure if the conditions when the point is On indicate a blown fuse may occur. See <a href="#">Point-level Fault Reporting on page 39</a> .
	<b>IMPORTANT</b> The pulse test does not guarantee failure of a fuse when the output point turns On. It merely indicates that a blown fuse is possible.
Detect a No Load condition with an output On	No Load detection can only detect a fault when an output point is in the Off state. However, you can use a pulse test when an output module is in an On state to determine whether operating conditions for a point may cause a No Load condition. If you perform a pulse test on an output point while it is in the On state, the output point is commanded to be Off briefly. The pulse test reports a failure because conditions when the point is Off indicate the possible absence of a field device; in this case, though, the No Load bit is not set. See <a href="#">Point-level Fault Reporting on page 39</a> .
	<b>IMPORTANT</b> The Pulse Test does not guarantee the absence of a load. It merely indicates that a No Load condition is possible.

### Diagnostic Change of State for Output Modules

If the Diagnostic Change of State feature is enabled, a diagnostic output module sends new data to the owner-controller when one of the events that are described in [Table 18](#) occurs.

**Table 18 - Diagnostic Change of State Events**

Event	Description
Receipt of output data	Output module sends data when it echoes back to the owner-controller.
Diagnostic change of state	Output module sends data when any change in the diagnostics output point occurs.

Unlike diagnostic input modules, this feature cannot be disabled for diagnostic output modules. There is no Enable Change of State for Diagnostic Transitions checkbox on the Configuration tab to check or clear for diagnostic output modules.

### Fault and Status Reporting Between Output Modules and Controllers

ControlLogix diagnostic digital output modules multicast fault and status data to any owner-controller or listening controller. Like input modules, output modules maintain a module-fault word, the highest level of fault reporting. However, some output modules use additional words to indicate fault conditions.

The table lists the fault words and the associated tags that can be examined in program logic to indicate when a fault has occurred for a diagnostic output module.

**Table 19 - Fault Words on Diagnostic Output Modules**

Word	Tag Name	Description
Module-fault	Fault	Provides fault summary reporting. Available on all digital output modules.
Fuse Blown	FuseBlown	Indicates a blown fuse for a point on the module. For more information, see <a href="#">Electronic Fusing on page 35</a> .
No Load	NoLoad	Indicates a loss of a load from a point on the module. For more information, see <a href="#">No Load Detection on page 46</a> .
Output Verify	OutputVerify	Indicates when an output is not performing as commanded by the owner-controller. For more information, see <a href="#">Field-side Output Verification on page 46</a> .

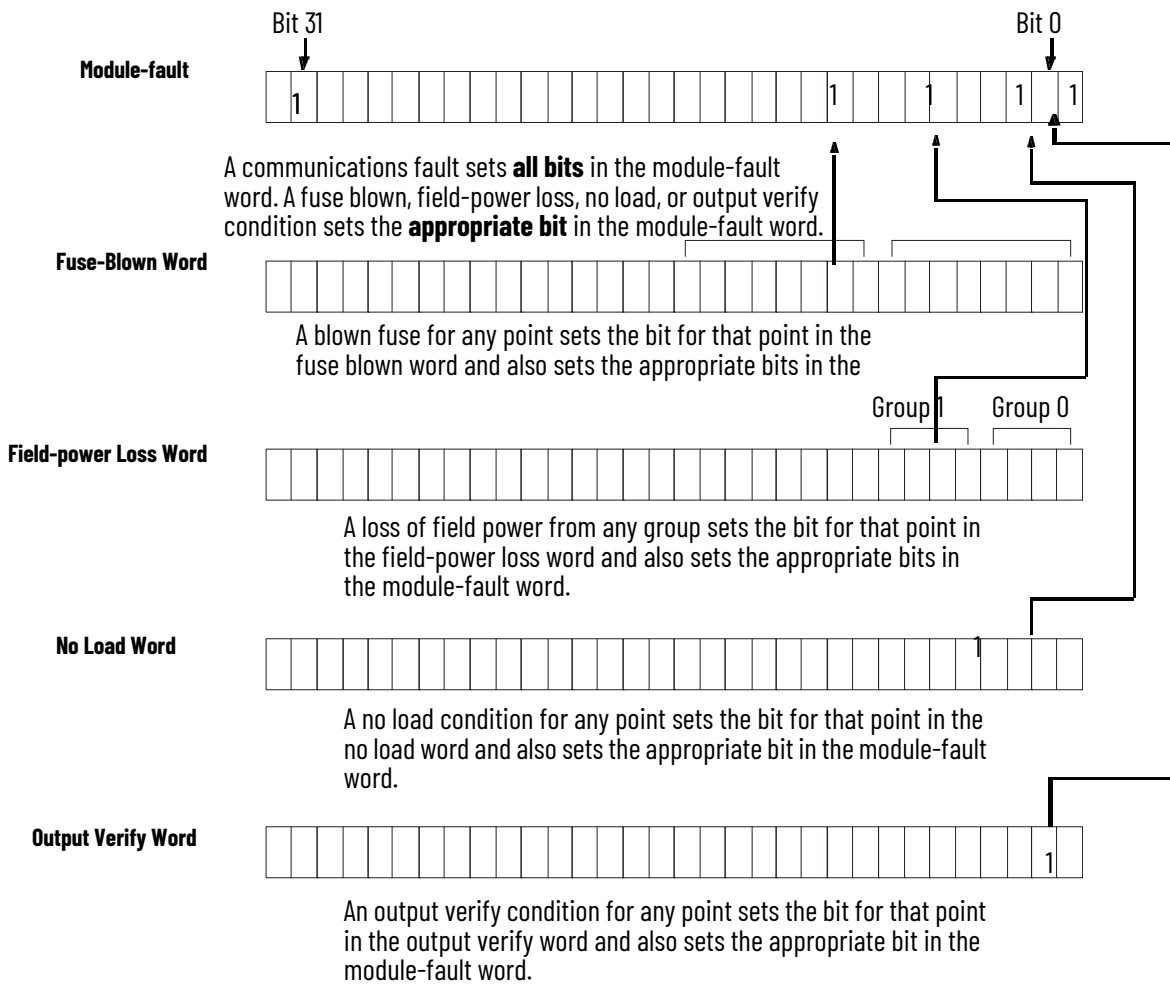
All words are 32 bit, although only the number of bits appropriate for each module’s density are used. For example, the 1756-0B8 module has a module-fault word of 32 bits. But, because the module is an 8-point module, only the first 8 bits (0...7) are used in the module-fault word.

Fault bits in the fuse blown word, field-power loss word, no load word, and output verify word are logically entered into the module-fault word. Depending on the module type, a bit set in the module-fault word can mean multiple things, as indicated in the table.

**Table 20 - Bits Set in Module-fault Word**

Condition	Bits Set
Communication fault	All 32 bits are set to 1, regardless of the module’s density.
Fuse blown	Only the bit affected is set to 1.
Field-power loss	
No load	
Output verify	

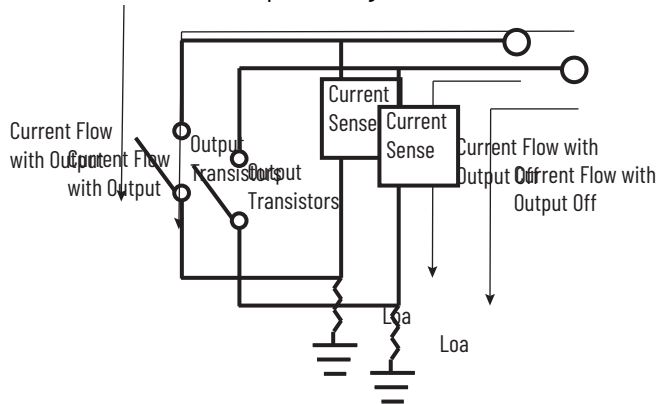
This illustration provides an overview of the fault reporting process for digital output modules.



## No Load Detection

For each output point, no load detection senses the absence of field wiring or a missing load from each output point in the Off state only.

The output circuit on a diagnostic output module has a current sensing opto-isolator used in parallel with the output transistor. Current flows through this sensing circuit only when the output is Off, as shown in the simplified diagram.



Diagnostic output modules list a minimum load current specification (1756-OA8D = 10 mA & 1756-OB16D = 3 mA). In the On state, the module must be connected to a load that draws a minimum current equal to these values.

If a connected load is sized in accordance with the minimum load current specification, diagnostic output modules are capable of sensing current through the opto-isolator and the load when the output point is Off.

For more information on configuring this feature, see [Chapter 4](#).

## Field-side Output Verification

Field-side output verification informs you that logic-side instructions consumed by the module are accurately represented on the power side of a switching device. For each output point, this feature confirms that the output is On when it is commanded to be On.

The diagnostic output module can tell a controller that it received a command and whether the field-side device that is connected to the module has executed the command. For example, in applications that must verify that the module has accurately followed the processor's instructions, the module samples the field-side state and compares it to the system-side state.

This feature has a corresponding tag that can be examined in the user program in the event of a fault. For more information on these tags, see [Appendix A](#).

If an output cannot be verified, a point-level fault is sent to the controller.

For more information on configuring this feature, see [Chapter 4](#).

## Fast Digital Module Features

### Fast Module Features

Module features include all the common features that are described in [Chapter 1](#), as well the extended capabilities described within this chapter.

For higher-speed control, the 1756-OB16IEF output module can be configured to receive input status over the backplane directly from the 1756-IB16IF input module or 1756-LSC8XIB8I counter module without controller processing. This feature, known as peer ownership, is described in the ControlLogix® Peer Ownership Application Technique, publication [1756-AT016](#).

#### IMPORTANT

To configure the modules, you must have the following:

- The 1756-OB16IEF and 1756-OB16IEFS modules requires Studio 5000® environment, version 21.00.00 or later.
- The Add-on Profile (AOP) for each module available for download at [rok.auto/pcdc](http://rok.auto/pcdc)

### Response Time

This table indicates the screw-to-backplane response time of fast input and fast output modules.

**Table 21 - Input and Output Response Time**

Delay	Response Time
<b>Fast Input Modules</b>	
Total On/Off delay (screw to backplane)	14 $\mu$ s nom/23 $\mu$ s max + user-configurable filter time
Hardware delay	< 1 $\mu$ s nom, 2 $\mu$ s max
Firmware delay	13 $\mu$ s nom, 21 $\mu$ s max
User-configurable filter time	0...30,000 $\mu$ s
<b>Fast Output Modules</b>	
Total On/Off delay (screw to backplane)	14 $\mu$ s nom/23 $\mu$ s max
Hardware delay	< 1 $\mu$ s nom, 2 $\mu$ s max
Firmware delay	13 $\mu$ s nom, 21 $\mu$ s max

## Fast Input Module Features

Follow these guidelines when designing a system:

- Make sure that the ControlLogix outputs can supply the necessary surge and continuous current for proper operation.
- Make sure that the surge and continuous current are not exceeded. Damage to the module could result.

### Pulse Capture

The 1756-IB16IF fast input module can be used to detect or latch short-duration pulses. The module can detect incoming pulses with a duration as short as 10  $\mu$ s if the frequency is under 4 kHz (period of 250  $\mu$ s).

When the module detects a short-duration pulse at an input point, it sets the corresponding bit for the Pt[x].NewDataOffOn or Pt[x].NewDataOnOff input tag. This bit remains latched until acknowledged. As a result, you can use this bit to detect a transition that is too fast to be detected by the program scan. You can also determine how rapid the transition was by configuring the module to latch timestamps for the point, as described in [Per Point Timestamping and Change of State on page 51](#).

To acknowledge the last captured pulse and reset the pulse latch, you set the rising edge of the corresponding bit in these output tags:

- Pt[x].NewDataOffOnAck—Acknowledges that the input point has transitioned to an On state and resets the pulse latch.
- Pt[x].NewDataOnOffAck—Acknowledges that the input point has transitioned to an Off state and resets the pulse latch.

You can change output tag values in program logic while normal module operation continues or through the Logix Designer application tag editor. For more information about module tags, refer to [Appendix A](#).

Once a pulse latch is reset for an input point, the next pulse at that point sets the corresponding bit in the Pt[x].NewDataOffOn or Pt[x].NewDataOnOff input tags.

### Dedicated Connection for Event Tasks

The 1756-IB16IF input module can initiate an event task over a dedicated second connection in response to four user-defined input patterns. You can define these patterns in real time during a control process by using these output tags:

- Event[x].Mask—Defines which input points trigger the event task.
- Event[x].Value—Defines whether the masked input points must be in the On or Off state before the event task is triggered.

Each pattern can use any of the module's 16 input points, as shown in the tables below.

In example pattern 1, the input module triggers the event task when input points 0...7 are in the On state.

**Table 22 - Example Pattern 1**

Output Tag	Bit Position															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event[x].Mask	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Event[x].Value	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x

In example pattern 2, the input module triggers the event task when input points 0...7 are in the Off state.

**Table 23 - Example Pattern 2**

Output Tag	Bit Position															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event[x].Mask	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Event[x].Value	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

In example pattern 3, the input module triggers the event task when input points 4, 6, 8, and 10 are in the On state.

**Table 24 - Example Pattern 3**

Output Tag	Bit Position															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event[x].Mask	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0
Event[x].Value	x	x	x	x	1	x	1	x	1	x	1	x	x	x	x	x

In example pattern 4, the input module triggers the event task when input points 0...3 are in the On state, and input points 12...15 are in the Off state.

**Table 25 - Example Pattern 4**

Output Tag	Bit Position															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event[x].Mask	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
Event[x].Value	1	1	1	1	x	x	x	x	x	x	x	x	0	0	0	0

Once you define a pattern, you can disable an event from being triggered without clearing its output data by using the Event[x].Disarm output tag.

---

**IMPORTANT** All event masks and event values must be defined in the output tags of the module.

---

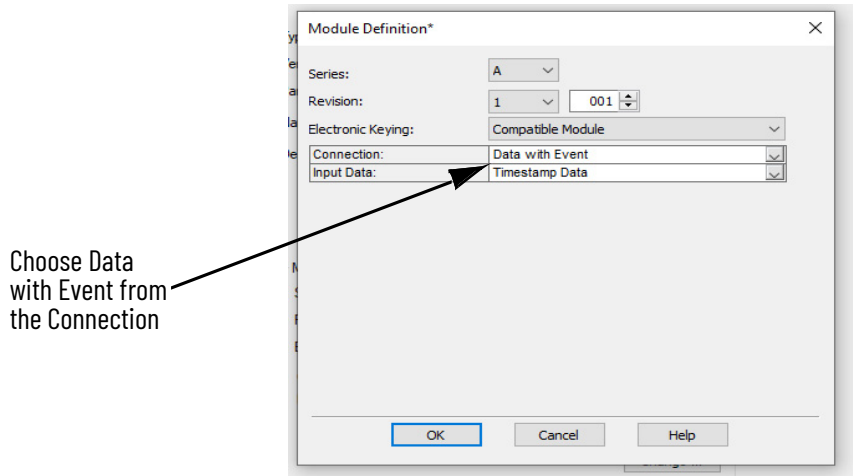
You can change output tag values in program logic while normal module operation continues or through the Logix Designer application tag editor. For more information about module tags, refer to [Appendix A](#).

To use a dedicated connection to trigger event tasks, you must set the module's connection format to Data with Event, as shown in [Figure 6](#). For more information about connection formats, see [Module Definitions, Communication, or Connection Formats on page 15](#).



You can change the connection format at any time after creating a module except when you are online. The AOP applies all the configuration data that is required for the new connection format.

Figure 6 - Event Connection Format



When you choose the Data with Event connection format, the following occurs:

- A second connection dedicated to event data only is established with the module. This dedicated event connection reduces controller overhead when using inputs or input patterns to trigger event tasks in the controller.

A new set of event tags is created, as described in [Table 49 on page 123](#).

### Fault and Status Reporting between Input Modules and Controllers

ControlLogix fast input modules multicast fault and status data to any owner-controller or listening controller. All input modules maintain a Module Fault word, the highest level of fault reporting. Modules configured to use the Data with Event connection format also maintain an Event Fault word to report on the status of an event connection.

This lists the fault words and associated tags that you can examine in program logic to indicate when a fault or event has occurred for a fast input module.

Table 26 - Fault Words on Fast Input Modules

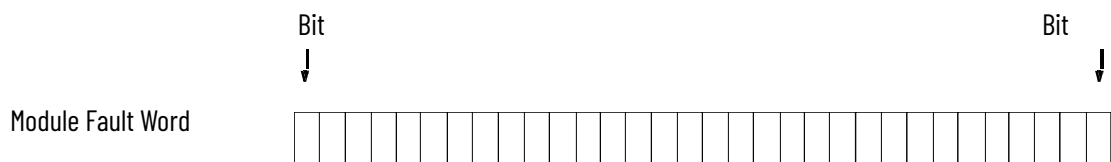
Word	Input Tag Name	Description
Module Fault	I:Fault	Provides fault summary reporting. Available on all digital input modules.
Event Fault	E:Fault	Provides fault summary reporting. Available on all digital input modules that use the Data with Event or Listen Only with Event connection format.

All words are 32 bit, although only the number of bits appropriate for each module's density are used. For example, the 1756-IB16IF module has a Module Fault word of 32 bits.

Table 27 - Bits Set in Module Fault Word

Condition	Bits Set
Communication fault	All 32 bits are set to 1, regardless of the module's density.

This illustration offers an overview of the fault reporting process on ControlLogix fast digital input modules.



A communication fault sets all 32 bits in the Module Fault word.

## Per Point Timestamping and Change of State

With per point timestamping, each input point on the module records timestamps in CIP Sync™ format at these speeds:

- $\pm 4 \mu\text{s}$  for inputs  $< 4 \text{ kHz}$
- $\pm 13 \mu\text{s}$  for inputs  $> 4 \text{ kHz}$

---

**IMPORTANT** Timestamping functions only in a CIP Sync system. If you are using change of state (COS) in a system using Coordinated System Time (CST), all timestamp values and the GrandMasterClockID input tag are set to zero.

To set up CIP Sync time synchronization on the local controller, use the Date/Time tab in the controller properties. For more information about CIP Sync configuration, refer to the Integrated Architecture® and CIP Sync

Configuration Application Technique, publication [IA-AT003](#).

---

You can configure an input point to record a timestamp when the point transitions from On to Off, Off to On, or in both directions. By default, all points are configured to record a timestamp in both directions.

You can also configure the module to latch timestamps for an input point's last transition. When latching is enabled for a specific point, the point records a timestamp in the Pt[x].Timestamp.OffOn or Pt[x].Timestamp.OnOff input tags. The timestamp remains latched, and no new timestamps are recorded for the input point until the timestamp is acknowledged and reset. As a result, you can use the timestamp to determine the speed of a transition that is too fast to be detected by the program scan.

To acknowledge a transition and reset a timestamp latch, you set the corresponding bit in these output tags:

- Pt[x].NewDataOffOnAck—Acknowledges that the input point has transitioned to an On state and resets the timestamp latch.
- Pt[x].NewDataOnOffAck—Acknowledges that the input point has transitioned to an Off state and resets the timestamp latch.

The Pt[x].TimestampDropped input tag indicates whether a new timestamp has not been recorded because a previous timestamp was either latched or unacknowledged.

Once a timestamp latch is reset for an input point, a new timestamp may be recorded in the Pt[x].Timestamp.OffOn or Pt[x].Timestamp.OnOff input tags upon the next transition.

You can configure per point timestamping in three ways:

- Timestamping enabled without latching (default configuration)
- Timestamping enabled with latching
- Timestamping disabled

For more information on configuring this feature, see [Per Point Timestamping and Change of State on page 51](#)

## Software Configurable Filter Times

To account for hard contact bounce, you can configure Off-to-On and On-to-Off input filter times of 0...30,000  $\mu$ s in the Logix Designer application. These filters define how long an input transition must remain in the new state before the module considers the transition valid.

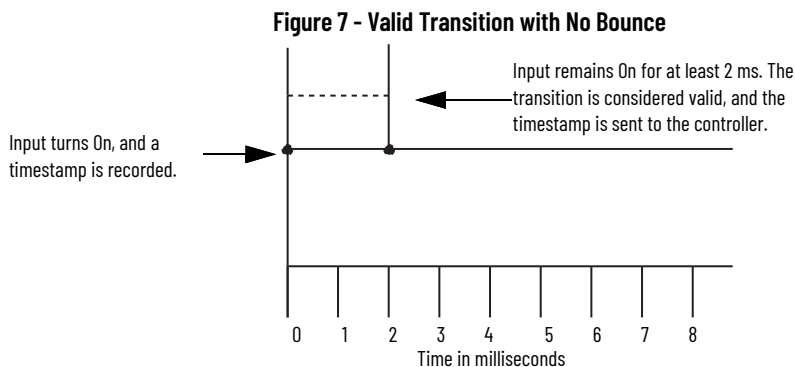
When an input transition occurs, the module timestamps the transition on the edge of the transition and stores timestamp data for the transition. The module then monitors the input for the duration of the filter time to verify that the input remains in the new state:

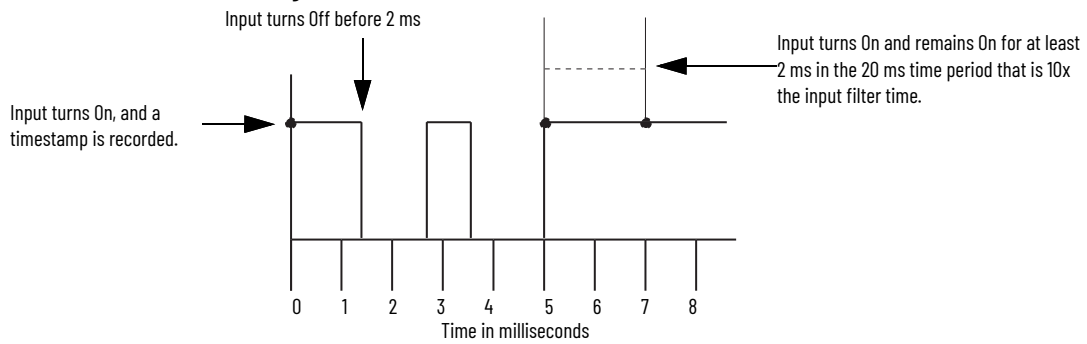
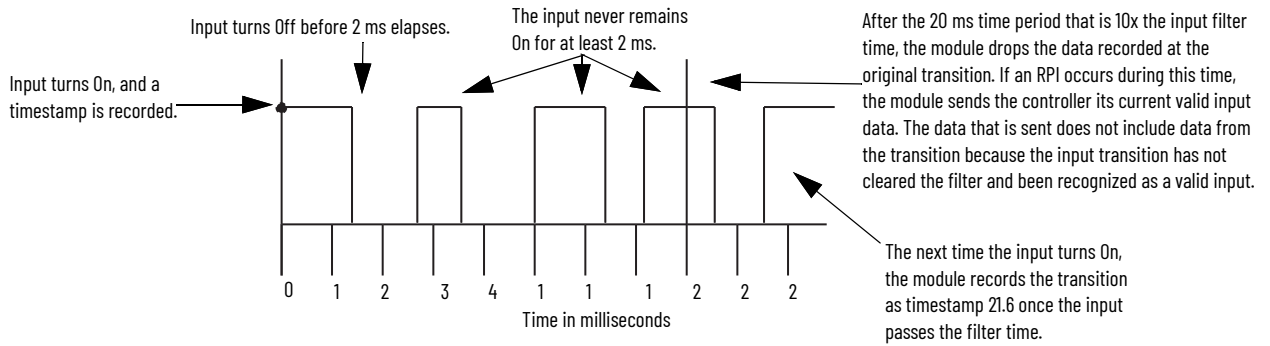
- If the input remains in the new state for a time period equal to the filter time, the input is recognized and recorded. The module sends timestamp data for the transition and the input's On/Off state to the controller.
- If the input changes state again before the duration of the filter time has elapsed, the module continues to scan that input for up to 10x the filter time. During this continued scan period, one of these events occurs:
  - In the time period that is 10x the duration of the filter time, the input returns to the transitioned state for the duration of the filter time. In this case, the module sends timestamp data from the initial transition to the controller.
  - In the time period that is 10x the duration of the filter time, the input never remains in the transitioned state for the duration of the filter time. In this case, the input is recognized, but the module does not consider the original transition valid and drops the timestamp.

### EXAMPLE

A 1756-IB16IF module is configured for a 2 ms filter time for Off-to-On transitions. In this example, three possible scenarios can result after an input transitions from Off to On:

- **Scenario 1**—The input turns On and remains On for the full 2 ms filter time. The module considers the transition valid and sends the data recorded at the transition to the controller ([Figure 7 on page 52](#)).
- **Scenario 2**—The input turns On but turns Off before the 2 ms filter time elapses. The module continues to monitor the input for 10x the duration of the filter time. Within that time period, the input turns On again and remains On for at least 2 ms. The module considers the transition valid and sends the data timestamped at the original transition to the controller ([Figure 8 on page 53](#)).
- **Scenario 3**—The input turns On but turns Off before the 2 ms filter time elapses. The module continues to monitor the input for 10x the duration of the filter time. Within that time period, the input never remains On for at least 2 ms. The module considers the transition invalid and drops the data timestamped at the original transition ([Figure 9 on page 53](#)).



**Figure 8 - Transition Valid with Bounce****Figure 9 - Invalid Transition**

For more information on configuring this feature, see [Software Configurable Filter Times on page 52](#).

## Fast Output Module Features

ControlLogix fast output modules can be used to drive a variety of output devices. Typical output devices compatible with ControlLogix outputs include these items:

- Solenoids
- Indicators

Follow these guidelines when designing a system:

- Make sure that the ControlLogix outputs can supply the necessary surge and continuous current for proper operation.
- Make sure that the surge and continuous current are not exceeded. Damage to the module could result.
- When sizing output loads, refer to the documentation supplied with the output device for the surge and continuous current needed to operate the device.
- Outputs on fast output modules can be directly wired to inputs on fast input modules.

### Fault and Status Reporting between Output Modules and Controllers

ControlLogix fast digital output modules multicast fault and status data to any owner-controller or listening controller. Like input modules, output modules maintain a Module Fault word, the highest level of fault reporting. However, output modules use an additional word to indicate a fault condition.

[Table 28](#) lists the fault word and the associated tag that you can examine in program logic to indicate when a fault has occurred for a fast output module.

**Table 28 - Fault Words on Fast Output Modules**

Word	Input Tag Name	Description
Module Fault	I:Fault	Provides fault summary reporting. Available on all digital output modules.

All words are 32 bit, although only the number of bits appropriate for each module's density are used. For example, the 1756-OB16IEF module has a Module Fault word of 32 bits. But, because the module is a 16-point module, only the first 16 bits (0...15) are used in the Module Fault word.

Bits set in the FuseBlown tag are logically entered into the Module Fault word. Depending on the module type, a bit set in the Module Fault word can mean multiple things, as indicated in the table.

**Table 29 - Bits Set in Module Fault Word**

Condition	Bit set
Communication fault	All 32 bits are set to 1, regardless of the module's density.
Fuse blown	Only the bit affected is set to 1.

This illustration provides an overview of the fault reporting process for digital output modules.

Bits set for the Fault input tag indicate I/O data may be incorrect due to a fault due to one of these conditions:

- FuseBlown = 1
- PWMCycleTime outside the valid range of 0.001...3600.0 seconds
- PWMOnTime outside the valid range of 0.0002...3600.0 seconds or 0...100 percent

$PWMCycleTime \leq PWMOnTime$

## Programmable Fault State Delay

You can define these states for an output point that is in Fault mode due to a communication failure:

- **Duration**—Defines the length of time that the output remains in the Fault mode state before transitioning to a final state of On or Off. By default, the output remains in the Fault mode state as long as the fault condition persists.
- **Final state**—Defines whether the output transitions to the On or Off state after the duration of the Fault mode state elapses. By default, the output transitions to the Off state.

---

**EXAMPLE** You define a duration of 1 second and a final state of On for an output point. If a fault occurs at that point, the output remains in its Fault mode state (Off, On, or Hold) for 1 second before transitioning to the On state.

---

**IMPORTANT** If a connection is re-established after an output point goes into Fault mode but before the duration time elapses, the settings you specify for the duration and final state no longer apply. For example, if you specify a 10 second duration and a final state of Off, and the fault ends in 3 seconds, the output point never transitions to the final state of Off.

---

For more information about defining a Fault mode state, refer to [Configurable Point-level Output States on page 35](#).

For more information on configuring this feature, see [page 70](#).

## Pulse-width Modulation

Pulse-width modulation (PWM) provides precise, onboard control of an output's pulse train with no program variability. To configure a PWM signal, you define two real-time values for the pulse train in the module's output tags:

- **Cycle time**—The duration of a pulse cycle in seconds from 1 ms...1 hour.
- **On time**—The pulse width, or length of time that a pulse is active within a cycle from 200  $\mu$ s...1 hour. You can define the On time in seconds or as 0...100 percent of the cycle time. You may want to use a steady-state On time, such as for gluing applications, or a dynamic On time that is defined by program logic.

If the cycle time or On time is outside the valid range for an output, the corresponding bit in the Fault input tag is set and the module responds as described.

Condition	Result
PWMCycleTime < minimum of 1 ms	PWMCycleTime = 1 ms
PWMCycleTime > maximum of 1 hour	PWMCycleTime = 1 hour
PWMCycleTime $\leq$ PWMOnTime	Output is always On
PWMOnTime < minimum of 200 $\mu$ s	Output is always Off
PWMOnTime > maximum of 1 hour	PWMOnTime = 1 hour

If the cycle time or On time value changes while the output is generating a PWM signal, the changes are not applied until the next cycle of the PWM output. For instance, if the cycle time is erroneously set to an hour, a new cycle time does not go into effect until the last cycle of the

hour is complete. To trigger the PWM output to restart immediately with a new cycle time or On time, turn the output Off and then back On.

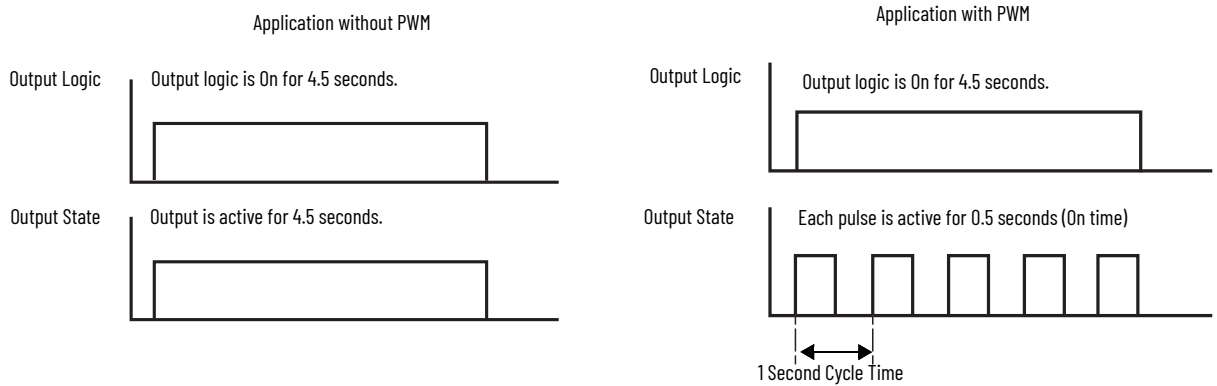
**EXAMPLE** If PWMOnTime is 0.1 second and PWMCycleTime is 1.0 second and the PWMCycleTime is changed to 0.5 second just after the output turns On, the output stays on for 0.1 second and then turns Off for 0.9 seconds to complete the cycle before the new 0.5 second cycle begins.

**IMPORTANT** Before PWM functions, you must enable PWM during configuration and define the PWM cycle time and On time in the PWMCycleTime and PWMOnTime output tags.  
 If PWM is enabled (PWMEnable = 1) and the output is instructed to turn On (Data = 1), the output generates a PWM signal.

The figure below compares two applications in which the output is instructed to turn On for 4.5 seconds:

- In the application without PWM, a single pulse is generated. The pulse remains active for the same length of time the Data output tag is On (4.5 seconds).
- In the application with PWM, a series of pulses is generated. Each pulse is active for a configured On time of 0.5 seconds or 50% of the 1 second cycle time. The Data output tag is On for 4.5 seconds

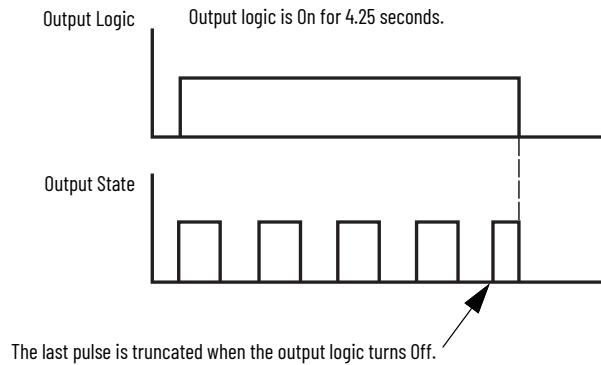
**Figure 10 - PWM Timing**



By default, PWM is configured to continue the output pulse train until the output logic turns Off. When the output logic turns Off, the output pulse train immediately stops.

**EXAMPLE** In this figure, the output logic is On for 4.25 seconds and then turns Off in the middle of the last pulse. Even though the PWM On time is configured for 0.5 seconds, the last pulse is only active for 0.25 seconds because it is truncated when the output logic turns Off.

**Figure 11 - PWM with Truncated Pulse**



**IMPORTANT** The Program and Fault mode states that are configured for the module override the PWM output state unless the point is configured to hold the last state while in Program or Fault mode. If a point is configured to hold the last state and the output is currently On, the output continues to use PWM until the PWM cycle limit is reached, the module transitions out of Program or Fault mode, or a final fault state goes into effect.

For more information, see the following:

- [Configurable Point-level Output States on page 35](#)
- [Programmable Fault State Delay on page 55](#)
- [Cycle Limit and Execute All Cycles on page 57](#)

You can modify the default PWM configuration for each of a module's 16 outputs for further control of an output's pulse train, as described in [Configure Pulse-width Modulation on page 83](#).

Configuration options include the following:

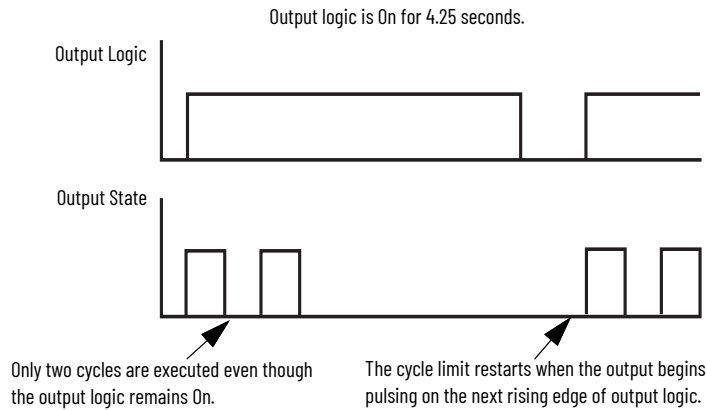
- [Cycle Limit and Execute All Cycles on page 57](#)
- [Minimum On Time, Extend Cycle, and Stagger Output on page 59](#)

## Cycle Limit and Execute All Cycles

You can limit the number of pulse cycles that occur while an output is On. This feature is useful when you want to apply a level of output control when a process is stopped. For example, in a gluing application, you may want to apply four drops of glue to a product when the product is within a fixed window on a conveyor belt. By configuring a cycle limit of four, you can make sure that only 4 drops of glue are applied even if the conveyor belt stops with the product in the window. Controlling the process with the Cycle Limit feature eliminates the need to write complex logic to detect a stopped conveyor belt.

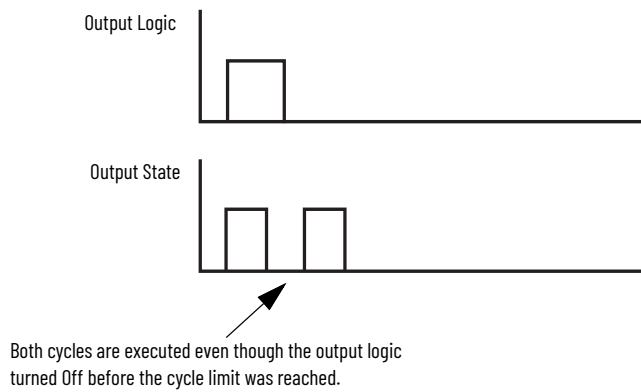
This figure shows a PWM pulse train that is configured with a cycle limit of 2. The PWMCycleLimitDone input tag indicates when the PWM cycle limit has been reached. The corresponding bit is reset upon the next rising edge of the output that restarts PWM.

**Figure 12 - PWM Cycle Limit**



If the output logic turns Off before the cycle limit is reached, you can configure the pulse cycles to continue until the cycle limit is reached by enabling the Execute All Cycles option. [Figure 13](#) shows a cycle limit of 2 with the Execute All Cycles option enabled.

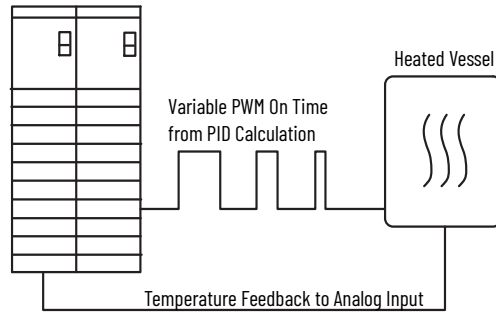
**Figure 13 - PWM Cycle Limit with Execute All Cycles Option**



## Minimum On Time, Extend Cycle, and Stagger Output

The Minimum On Time, Extend Cycle, and Stagger Output configuration options are useful in time-proportional control applications, such as temperature control. In these applications, PID calculations compare the actual temperature to the desired setpoint and vary the PWM On time to a heating element in real time to regulate temperature as it approaches the setpoint, as shown in [Figure 14](#).

**Figure 14 - PWM for Time Proportioned Control**



In this type of application, the Minimum On Time, Extend Cycle, and Stagger Output configuration options provide these benefits:

- **Minimum On Time and Extend Cycle**—Ensures that output devices that require a minimum time to turn On or that cannot react to a short pulse cycle can react with any given PWM On time calculation rather than not turning On.

To ensure that the output device turns On when the calculated On time is less than the minimum On time, you must enable the Extend Cycle option. When Extend Cycle is enabled, the cycle time is extended proportionately up to 10 times the calculated On time while taking into account the minimum On time.

---

### EXAMPLE

A solenoid requires at least 40 ms to turn On. During configuration, you enable the output for PWM, specify a minimum On time of 40 ms, and enable the Extend Cycle option.

If the calculated On time in the PWMOnTime output tag drops below the 40 ms minimum On time, the module automatically extends the On time to 40 ms and proportionally extends the cycle time in the PWMCycleTime output tag.

If the On time drops below 4 ms, the output turns Off because the cycle cannot extend beyond 10 times the 40 ms On time.

---

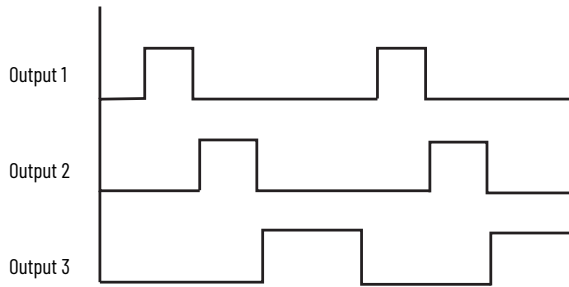
If Extend Cycle is not enabled and the calculated On time is less than the minimum On time, the output of the module does not energize.

- Stagger Output**—Mitigates the power surge from outputs that drive high-power loads by helping prevent the outputs from turning On simultaneously. Enabling the Stagger Output option for multiple output points addresses surges by staggering the leading edge of those outputs (Figure 15). When the Stagger Output feature is not enabled, output points turn On immediately at the start of the cycle (Figure 16).

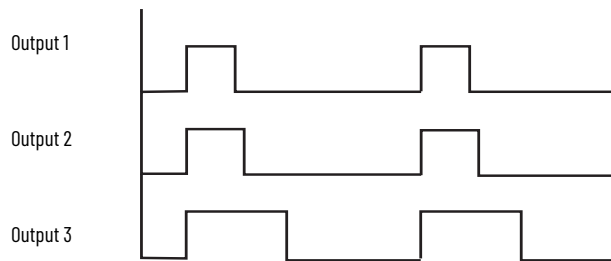
The stagger time for an output is calculated when the output turns On. If the On time and cycle times are changed by large amounts while the output is On, the stagger times may begin to overlap.

If the cumulative On time of staggered outputs is less than the cycle, each new On transition is staggered to begin 50 μs after the prior staggered output turns Off.

**Figure 15 - Outputs with Staggering**



**Figure 16 - Outputs without Staggering**



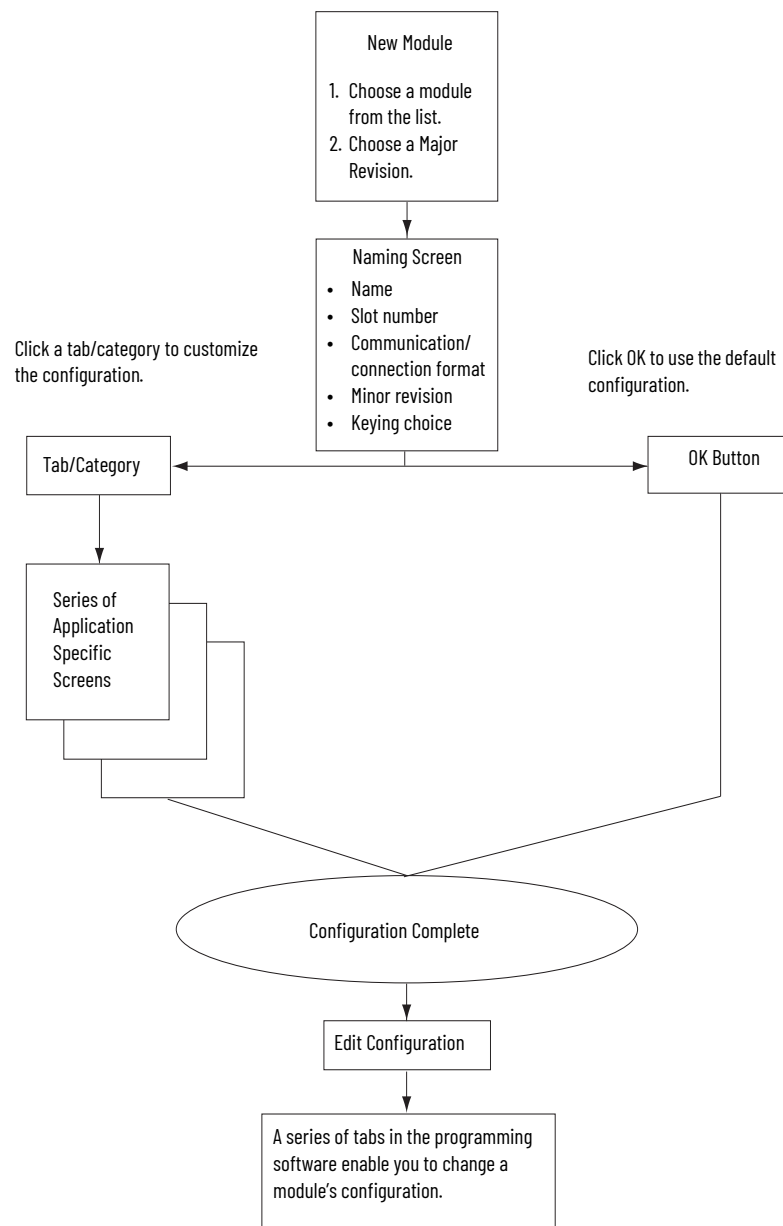
## Configure Digital I/O Modules

You must configure your module upon installation. The module does not work until it has been configured. In most cases, you use the Studio 5000 Logix Designer® application to complete the configuration. The application uses default configurations, such as RPI and filter times, to get your I/O module to communicate with its owner-controller. You can edit the default configuration as needed from the Module Properties dialog box.

### Configure ControlLogix Digital I/O Modules

Follow these steps to configure a ControlLogix® digital I/O module with the Logix Designer application.

1. Create a new module.
2. Accept or customize the default configuration for the module.
3. Edit the configuration as changes are needed.



## Create a New Module

Before creating a module, make sure you complete these procedures in the Logix Designer application:

- Create a controller project.
- If you plan to add the I/O module to a remote chassis, add EtherNet/IP™ communication modules to both the local and remote chassis in the I/O Configuration tree.
  - For more information on ControlLogix EtherNet/IP modules, see EtherNet/IP Modules in the Logix 5000® Control Systems User Manual, publication [ENET-UM001](#).

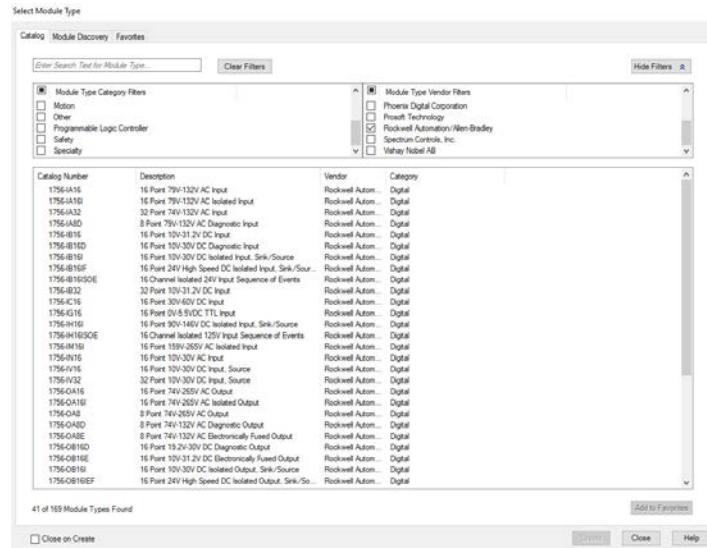
**IMPORTANT** The Studio 5000® environment lets you add I/O modules online. When using a previous version, you must be offline to create a module.

Follow these steps to add a local or remote I/O module.

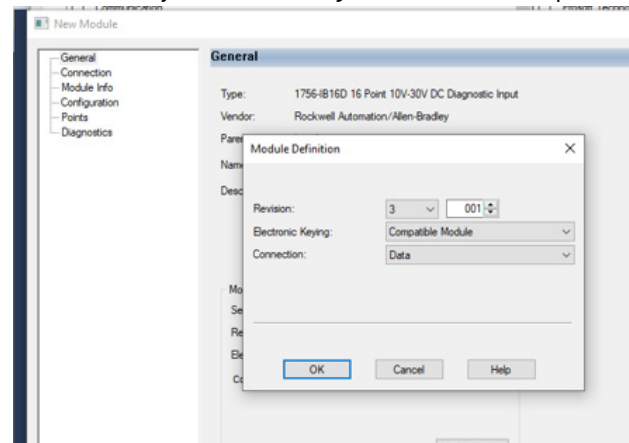
1. To add an I/O module to a local chassis, right-click the I/O Configuration folder and choose New Module.

To add an I/O module to a remote chassis, right-click the remote communication module, and choose New Module.

2. On the Select Module Type dialog box, select the digital module to create, and then click Create.



3. On the Select Major Revision dialog box, click OK to accept the default major revision.

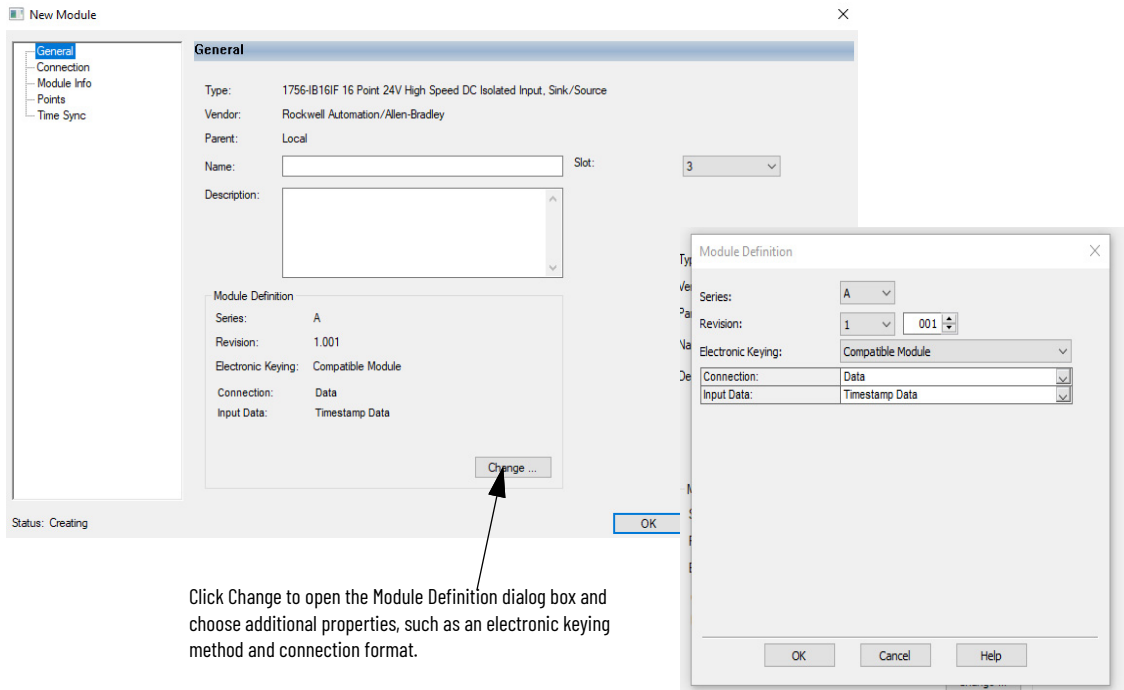
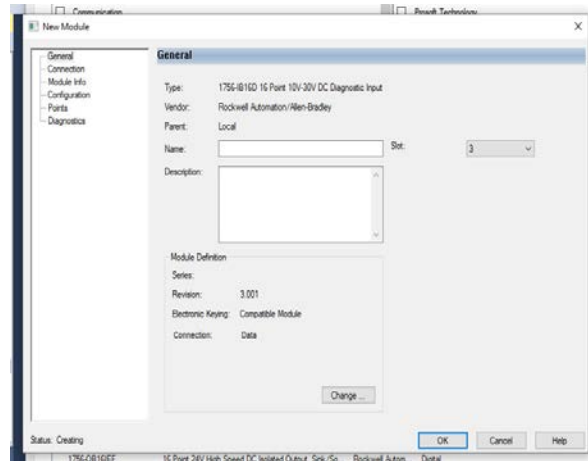


4. On the New Module dialog box, complete the fields and click OK.

For information about how to choose an electronic keying method, see [Electronic Keying on page 27](#).

For information about how to choose a communication format or connection type, see [Communication Formats, Connection Formats, and Module Definitions on page 64](#).

The fields on the New Module dialog box vary depending on the catalog number of your I/O module.



Click Change to open the Module Definition dialog box and choose additional properties, such as an electronic keying method and connection format.

## Communication Formats, Connection Formats, and Module Definitions

These are the communication formats and module definitions for Input and Output modules.

**Table 30 - Communication Formats – Input Module**

Communication Format	Data Return	Module
Input Data	Module returns only general fault and input data.	
CST Timestamped Input Data	Module returns input data with the value of the system clock from its local chassis when the input data changes.	1756-IA16, 1756-IA16K, 1756-IA16I, 1756-IA16IK, 1756-IA32, 1756-IA32K, 1756-IB16I, 1756-IB16IK, 1756-IB16, 1756-IB16K, 1756-IB16XT, 1756-IB32, 1756-IB32K, 1756-IB32XT, 1756-IC16, 1756-IG16, 1756-IG16K, 1756-IH16I, 1756-IH16IK, 1756-IM16I, 1756-IM16IK, 1756-IN16, 1756-IN16K, 1756-IV16, 1756-IV16K, 1756-IV32, 1756-IV32K
Rack Optimization	The Ethernet bridge module collects all digital input words in the remote chassis and sends them to the controller as a single rack image. This connection type limits the status and diagnostic information available.	
Listen Only—Input Data	These formats have the same definition as the similarly named options above except that they are Listen-only connections.	
Listen Only—CST Timestamped Input Data		
Listen Only—Rack Optimization		
Full Diagnostic Input Data	Module returns input data, the value of the system clock from its local chassis when the input data changes, and diagnostic data.	1756-IA8D, 1756-IB16D
Listen Only—Full Diagnostic Input Data	This format has the same definition as Full diagnostic input data except that it is a Listen-only connection.	1756-IA8D, 1756-IB16D

**Table 31 - Module Definitions – Input Module**

Connection	Input Data	Data Return	Module
Data	Timestamp Data	Module returns input data with COS timestamps in CIP Sync® system time. To configure per point timestamping, see <a href="#">page 80</a> .	1756-IB16IF
	Data	Module returns input data without COS timestamps, useful when the highest possible throughput is required.	
Data with Event	Timestamp Data	Results in two input connections: Connection to return input data with COS timestamps in CIP Sync system time. Connection to initiate event tasks. See <a href="#">Dedicated Connection for Event Tasks on page 48</a> .	
Listen Only	Timestamp Data	These formats have the same definition as those above except that they are Listen-only connections.	
	Data		
Listen Only with Event	Timestamp Data		

**Table 32 - Communication Formats – Output Module**

Communication Format	Data Return	Module	Communication Format
Output Data	The owner-controller sends the module only output data.	1756-OA8, 1756-OA16I, 1756-OB8, 1756-OB16I, 1756-OB16IS <sup>(1)</sup> , 1756-OB32, 1756-OC8, 1756-OG16, 1756-OH8I, 1756-ON8, 1756-OW16I, 1756-OX8I	Output Data
Scheduled Output Data	The owner-controller sends the module output data and a CST timestamp value		Scheduled Output Data
Rack Optimization	The owner-controller sends all digital output words to the remote chassis as a single rack image.		Rack Optimization
Listen Only—Output Data	These formats have the same definition as those above except that they are Listen-only connections.		Listen Only—Output Data
Listen Only—Rack Optimization			Listen Only—Rack Optimization
CST Timestamped Fuse Data—Output Data			The owner-controller sends the module only output data. The module returns Fuse Blown status with the value of the system clock (from its local chassis) when the fuse is either blown or reset.
CST Timestamped Fuse Data—Scheduled Output Data	The owner-controller sends the module output data and a CST timestamp value. The module returns Fuse Blown status with the value of the system clock (from its local chassis) when the fuse is either blown or reset.	1756-OA16, 1756-OA8E, 1756-OB16E, 1756-OB8EI, 1756-OV16E, 1756-OV32E	CST Timestamped Fuse Data—Scheduled Output Data
Listen Only - CST Timestamped Fuse Data—Output Data	This choice has the same definition as CST timestamped fuse data - output data except that it is a Listen-only connection.		Listen Only - CST Timestamped Fuse Data—Output Data
Full Diagnostics—Output Data	The owner-controller sends the module only output data. The module returns diagnostic data and a timestamp of diagnostics.	1756-OA8D, 1756-OB16D	Full Diagnostics—Output Data
Full Diagnostics—Scheduled Output Data	The owner-controller sends the module output data and a CST timestamp value. The module returns diagnostic data and a timestamp of diagnostics.		Full Diagnostics—Scheduled Output Data
Listen Only—Full Diagnostics—Output Data	This format has the same definition as Full diagnostics - output data except that it is a Listen-only connection.		Listen Only—Full Diagnostics—Output Data

(1) The 1756-OB16IS module does not support the Rack optimization, Listen only—rack optimization and Scheduled output data communication formats.

Table 33 - Module Definitions – Output Module

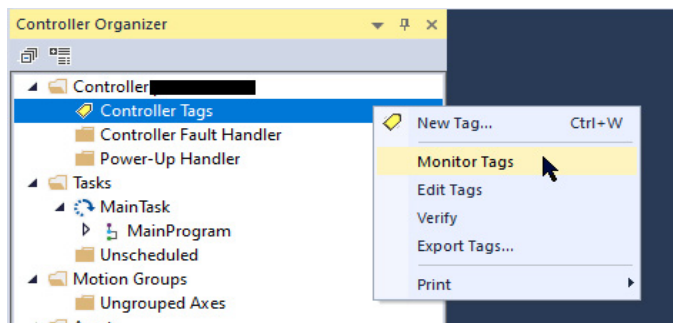
Connection	Output Data	Data Return	Module
Data	Data	The owner-controller sends the module only output data.	1756-OB16IEF, 1756-OB16IEFS
	Scheduled Per Module	The owner-controller sends the module output data and a CIP Sync timestamp value.	1756-OB16IEF
	Scheduled Per Point	The owner-controller sends output data and a CIP Sync timestamp value to points configured for scheduling.	1756-OB16IEFS
Listen Only	None	Establishes a Listen-only connection with no data.	1756-OB16IEF, 1756-OB16IEFS
Peer Input with Data	Data with Peer	Establishes a Listen-only connection to input peer modules. See the Peer Ownership Application Technique, publication <a href="#">1756-AT016</a> .	1756-OB16IEF

## View and Change Module Tags

When you create a module, a set of tags is created by the ControlLogix system that can be viewed in the Tag Editor of the Logix Designer application. Each configured feature on your module has a unique tag that can be used in the controller's program logic.

Follow these steps to access a module's tags.

1. On the Controller Organizer, expand the Controller folder, right-click Controller Tags, and choose Monitor Tags.



The Controller Tags dialog box appears with data.

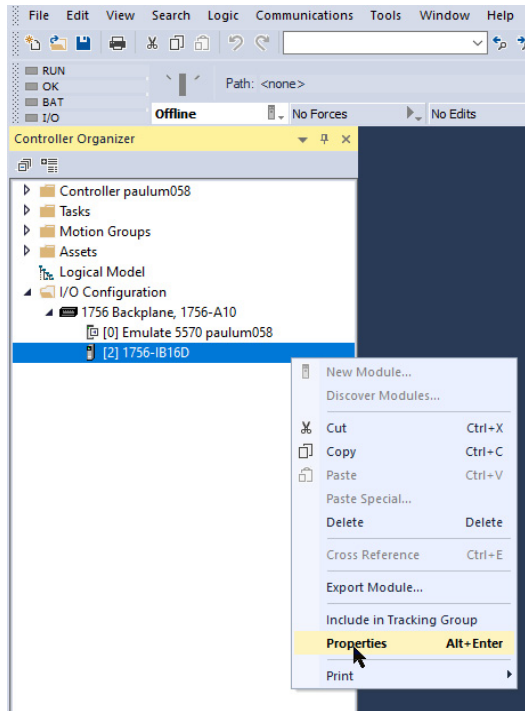
2. Expand the slot number of the module for which to view information. See [Appendix A](#) for details on viewing and changing a module's configuration tags.

## Edit the Configuration

After you add a module to the I/O configuration in the Logix Designer application, you can review and edit the configuration. You can also download the data to the controller while online. This is called dynamic reconfiguration.

Follow these steps to edit a module's configuration.

1. On the Controller Organizer, right-click an I/O module and choose Properties.



2. On the Module Properties dialog box, click the tab/category corresponding to the feature to modify, and then click OK:
  - To configure connection properties between the module and the controller, see [page 67](#).
  - To configure features common to all modules, see [Chapter 1](#).
  - To configure features specific to diagnostic modules, see [Chapter 2](#).
  - To configure features specific to fast modules, see [Chapter 3](#).

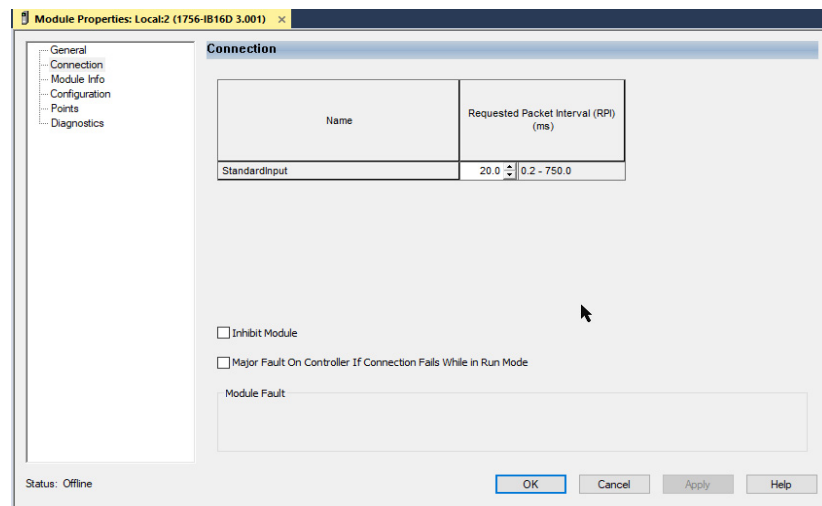
## Connection Properties

Connection properties define controller-to-module behavior. When defining connection properties, you can do the following:

- Select a requested packet interval (RPI) to set a defined, maximum period of time when data is transferred to the owner-controller
- Choose to inhibit the module
- Configure the controller so that a loss of connection to this module causes a major fault
- View information about the condition of the connection between the module and the controller

Follow these steps to configure connection properties.

1. On the Module Properties dialog box, click the Connection tab/category.

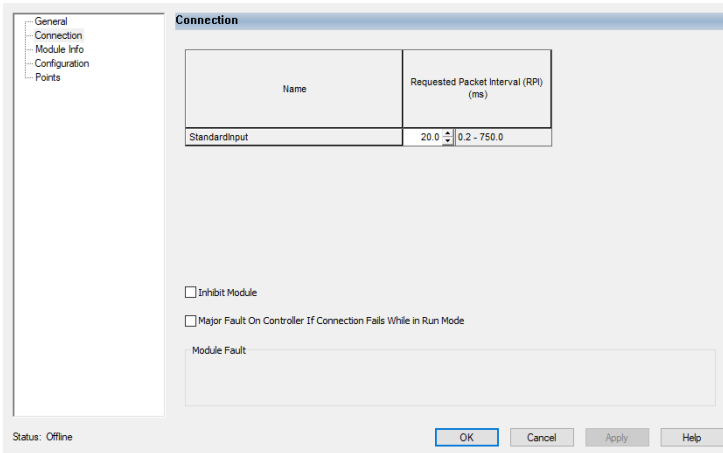


2. Complete the fields as described and click OK.

Field	Description
Requested Packet Interval (RPI)	Enter an RPI value or use the default. See <a href="#">RPI</a> for more information.
Inhibit module	Check the box to help prevent communication between the owner-controller and the module. This option enables maintenance of the module without faults being reported to the controller. See <a href="#">Module Inhibiting</a> for more information.
Major fault On Controller If Connection Fails While in Run Mode	Check the box to create a major fault if there is a connection failure with the module while in Run mode. For important information on this checkbox, see the Logix 5000 Controllers Information and Status Programming Manual, publication <a href="#">1756-PM015</a> .
Module Fault	The fault box is empty if you are offline. The type of connection fault appears in the text box if a fault occurs when the module is online.

## Configure Input Modules

Follow these steps to set an RPI value.

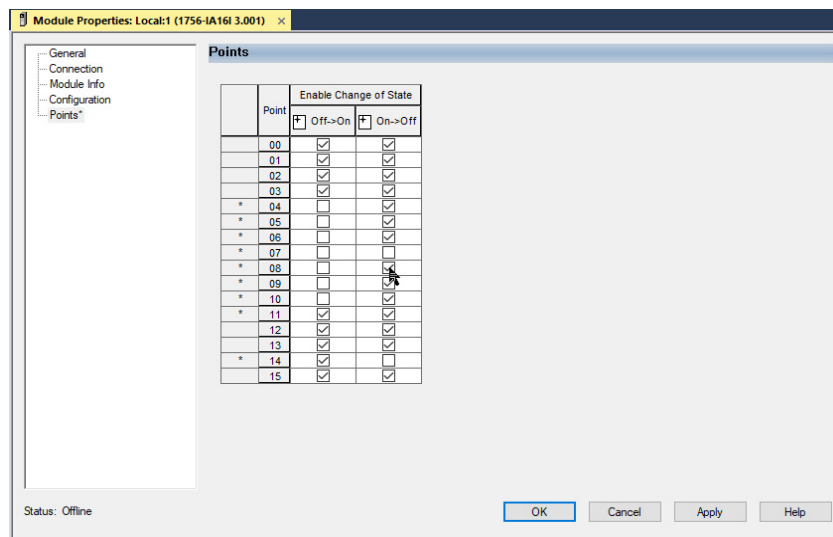


1. On the Module Properties dialog box, click the Connection tab.
2. In the Requested Packet Interval (RPI) field, enter an RPI value.
3. Click OK.

## Enable COS

Follow these steps to enable or disable COS.

1. On the Module Properties dialog box, click the Points tab.

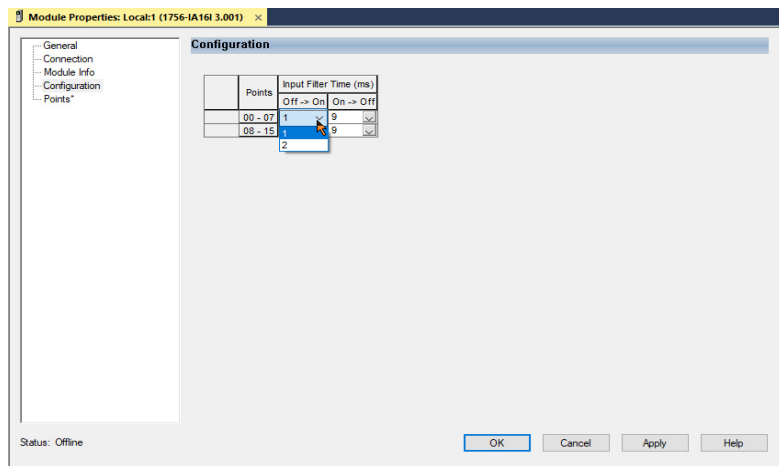


2. Do one of the following in the Enable Change of State columns:
  - To enable COS for a point, check the corresponding Off to On or On to Off checkbox.
  - To disable COS for a point, clear the corresponding Off to On or On to Off checkbox.
3. Click OK.

## Configure Input Filter Times

Follow these steps to configure the input filter time.

1. On the right side of the Configuration tab, choose the input filter times from the Off ? On and On ? Off dropdown menus.

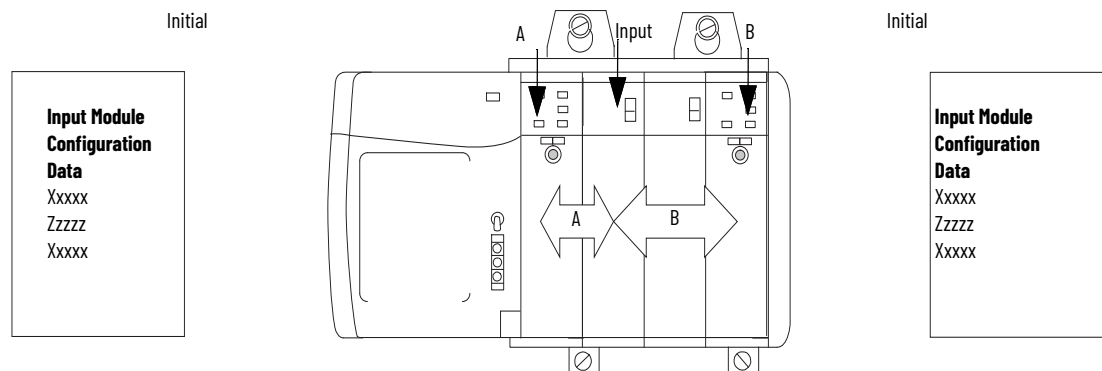


2. Click OK.

## Configure Input Module with Multiple Owners

Be careful when changing the configuration data of an input module in a multiple owner scenario. If the configuration data is changed in owner A and sent to the module, that data is accepted as the new configuration for the module. Owner B continues to listen unaware that any changes have been made in the behavior of the input module, as illustrated.

**Figure 17 - Module Configuration Changes with Multiple Owners**



### IMPORTANT

A message in the Logix Designer application alerts you to the possibility of a multiple owner-controller situation and lets you inhibit the connection before changing the module configuration. We recommend inhibiting the connection when changing the configuration of a module with multiple owners.

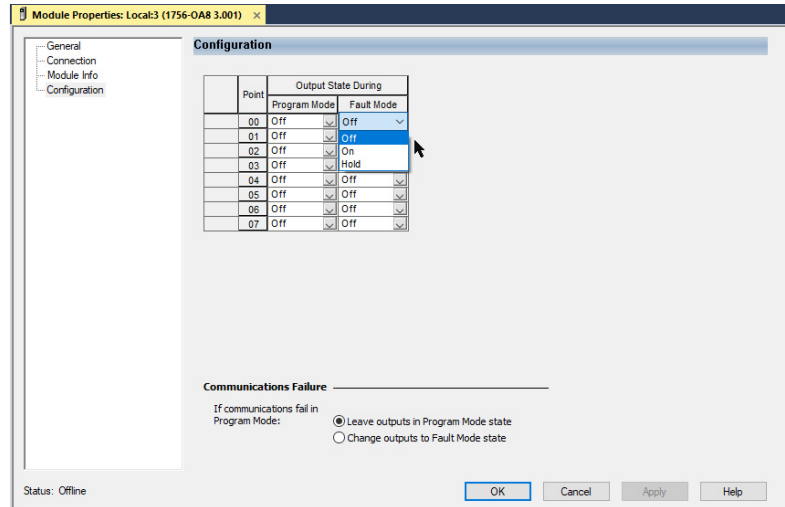
To help prevent other owner-controllers from receiving potentially erroneous data, use these steps when changing the configuration of a module in a multiple owner scenario while online.

1. For each owner-controller, inhibit the connection to the module either on the Connection tab or the message dialog box regarding the multiple owner condition.
2. Make the appropriate configuration data changes in the software. For more information about using the Logix Designer application to change the configuration, see [Chapter 4](#).
3. Repeat [step 1](#) and [step 2](#) for all owner-controllers, making the same changes in each.
4. Clear the Inhibit checkbox in each owner-controller configuration.

## Configure Output Modules

Follow these steps to configure an output state.

1. On the Module Properties dialog box, click the Configuration tab.

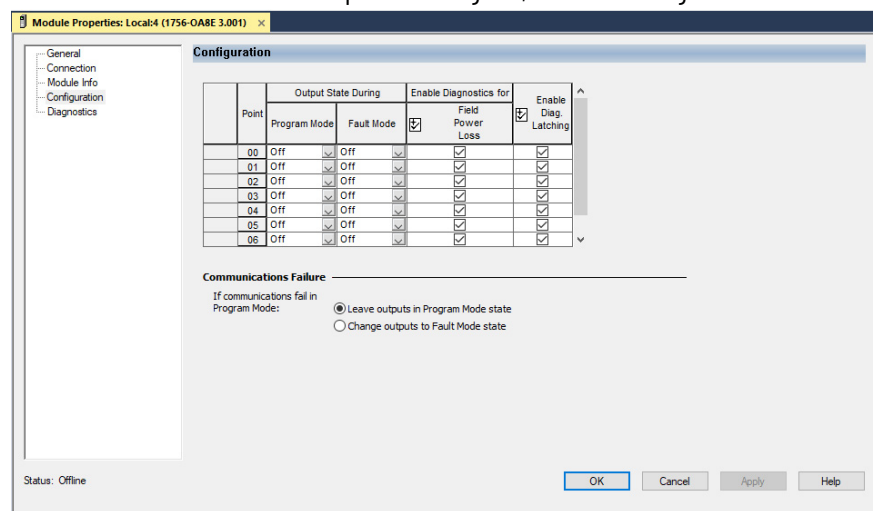


2. From the Program Mode dropdown menu, choose whether the output state of the module is On or Off during Program mode:
  - On
  - Off
  - Hold (Retain current output state)
3. From the Fault Mode dropdown menu, choose the output state of the module during Fault mode:
  - On
  - Off
  - Hold (Retain current output state)
4. Click OK.

## Enable or Disable Field Power Loss (1756-OA8E only)

Follow these steps to enable or disable diagnostics for field power loss.

1. On the Module Properties dialog box, click the Configuration tab.

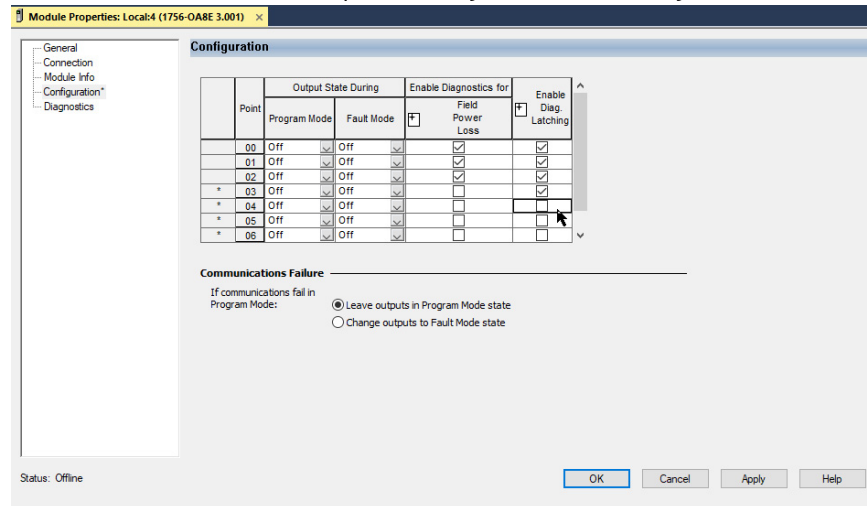


2. In the Enable Diagnostics for Field Power Loss column:
  - To enable detection for a specific point, select the corresponding checkbox.
  - To disable detection for a specific point, clear the corresponding checkbox.
3. Click OK.

## Enable or Disable Diagnostic Latch of Information (1756-0A8E only)

Follow these steps to enable diagnostic latch of information.

1. On the Module Properties dialog box, click the Configuration tab.



2. Do one of the following in Enable Diag. Latching column:

- To enable diagnostic latching for a specific point, check the corresponding checkbox.
- To disable diagnostic latching for a specific point, clear the corresponding checkbox.

3. Click OK.

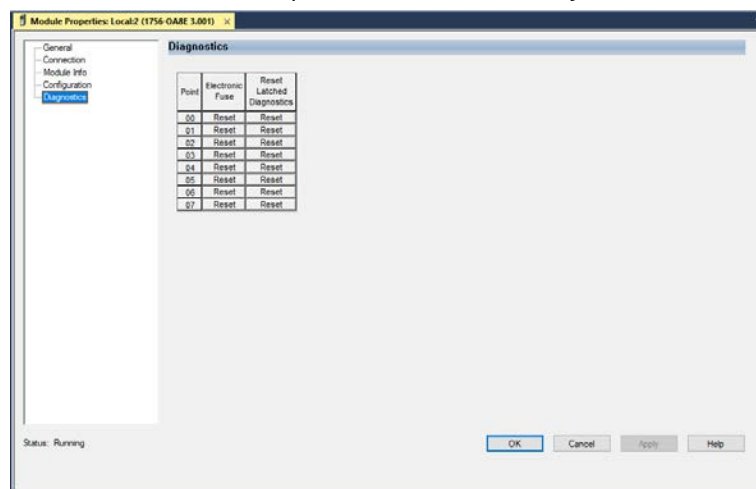
## Reset Latched Fault (0A8E only)

Latched diagnostic features can be cleared by using these methods:

- Reset Diagnostic Latch service
- Software reset during online monitoring
- Cycling power to the module

Follow these steps to a reset a latched fault through the Logix Designer application during online monitoring.

1. On the Modules Properties screen, click the Diagnostics tab.

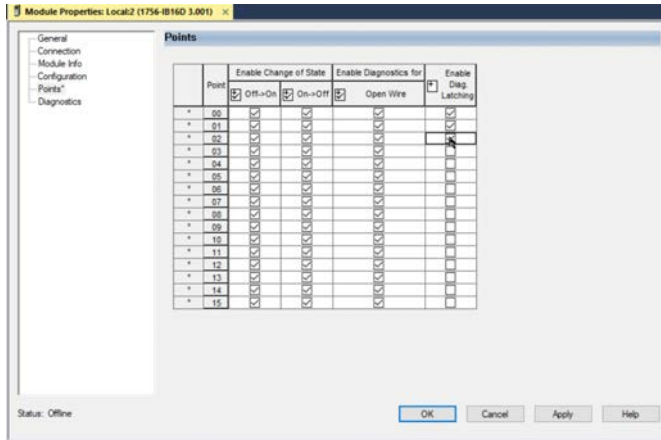


2. In the Reset Latched Diagnostics column, click Reset next to the output point for which to reset a latched fault.
3. Click OK.

## Configure Diagnostic Modules

Follow these steps to enable or disable diagnostic latching.

1. On the Module Properties dialog box, click the Points tab.



2. Do one of the following in the Enable Diag. Latching column:
  - To enable diagnostic latching for a specific point, check the corresponding checkbox.
  - To disable diagnostic latching for a specific point, clear the corresponding checkbox.
3. Click OK.

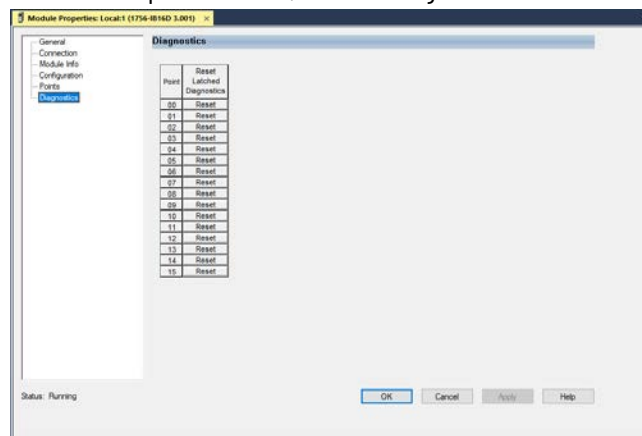
Latched diagnostic features can be cleared by using these methods:

- Reset Diagnostic Latch service
- Software reset during online monitoring
- Cycling power to the module

## Reset Latched Fault

Follow these steps to reset a latched fault through the Logix Designer application during online monitoring.

1. On the Modules Properties screen, click the Diagnostics tab.



2. Click Reset next to the point for which to reset a latched fault.

Click OK.

## Configure Diagnostic Input Modules

Follow these steps to enable COS for diagnostic input modules.

1. On the Module Properties dialog box, click the Points tab.

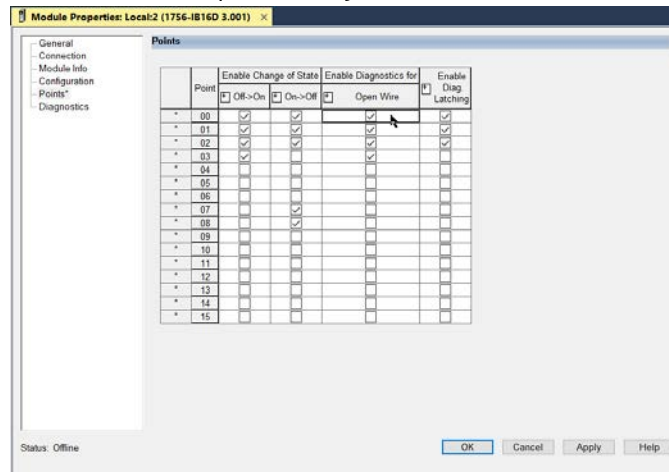


2. Do the following in the Enable Change of State column:
  - To enable the input module to send new data to the owner-controller at the RPI, on input COS if it is enabled, and if a diagnostic fault occurs, check the corresponding Off ? On or On ?Off checkbox for a point.
  - To disable the feature, clear the corresponding checkbox for a point.
  - Real-time data is not sent when a diagnostic fault occurs but is still sent at the specified RPI or on input COS if it is enabled.
3. Click OK.

## Configure Open Wire Detection

Follow these steps to configure open wire detection.

1. On the Module Properties dialog box, click the Points tab.

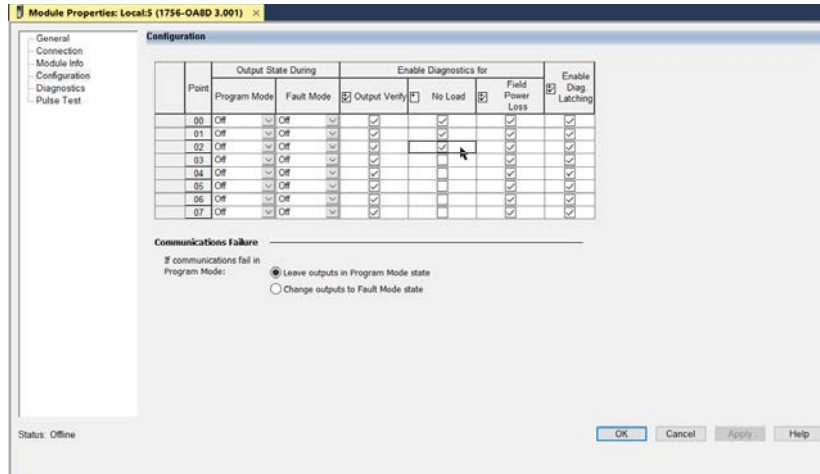


2. Do one of the following in the Open Wire (middle) column:
  - To enable the open wire detection for a specific point, check the corresponding checkbox.
  - To disable open wire detection for a specific point, clear the corresponding checkbox.
3. Click OK.

## Configure Diagnostic Output Modules

Follow these steps to enable no load detection.

1. On the Module Properties dialog box, click the Configuration tab.



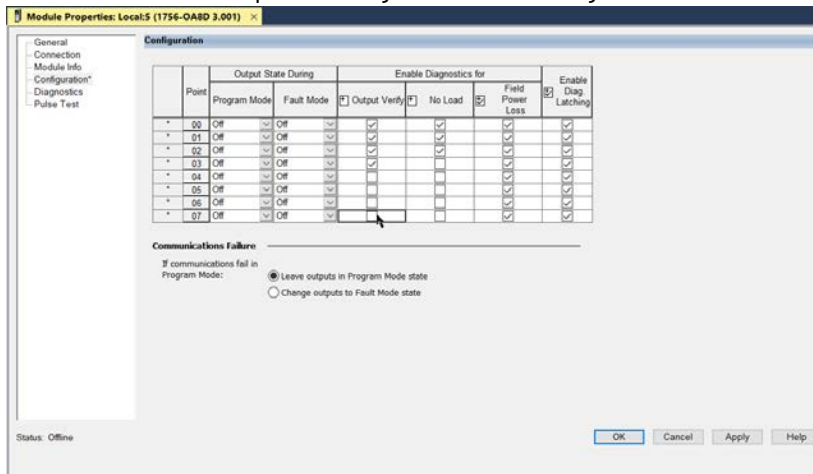
2. Do one of the following in the No Load column:
  - To enable the feature for a specific point, check the corresponding checkbox.
  - To disable the feature for a specific point, clear the corresponding checkbox.
3. Click OK.

This feature has a corresponding tag that can be examined in the user program in the event of a fault. For more information on these tags, see [Appendix A](#).

## Enable Field-side Output Verification

Follow these steps to enable the field-side output verification.

1. On the Module Properties dialog box, click the Configuration tab.



2. Do one of the following in the Output Verify column:
  - To enable the feature for a specific point, check the corresponding checkbox.
  - To disable the feature for a specific point, clear the corresponding checkbox.
3. Click OK.

## About Redundant Owners

Redundant ownership lets a single controller use redundant network paths to communicate with a single output or lets two separate controllers coordinate to control the output.

To enable Redundant Ownership mode in the output module, each connection to the output module is via a redundant owner module definition. In the module definition, you select between Redundant Output Owner A and Redundant Output Owner B. Each pair of connections must have an A and a B requested—that is, if the first is A, the second must be B.

If the second connection does not pair with the original connection, it is rejected with a general status = 0x01 and extended status = 0x031D. All other aspects of configuration between the two connections must be identical.

After the connection is established, a redundant owner uses the Claim Owner Output (COO) tag to indicate that it is ready for ownership of the outputs. The Ready Owner Output (ROO) tag indicates that an owner is ready to take ownership if needed. The ROO is implemented as a single bit, OwnerReady, as this implementation is intended for only two controllers.

When the OwnerClaim bit is set, it indicates that the controller wants to claim ownership of the redundant connection and have its outputs actively used. If both redundant owners have COO set, the last originator application that transitioned its COO flag from cleared to set is the owner.



Only a received 0-to-1 transition, and not simply the receipt of a new connection, is considered such a transition for ownership.

If neither redundant owner has OwnerClaim set, then the controller that has the OwnerReady bit set becomes the owner. If neither controller has OwnerClaim set and both have OwnerReady set, owner A becomes the owner. If both controllers have OwnerClaim=0 and OwnerReady=0, then the outputs go to IDLE (Program mode).

Finally, the Input data adds status fields that indicate if the output module has an active owner (OwnerActive) and for which controller, A or B (OwnerID). For each of owner A and B, three status fields indicate if the controller is connected (OwnerAConnected/OwnerBConnected) and echo the output values received for each owner via OwnerAClaim/OwnerBClaim and OwnerAReady/OwnerBReady, respectively.

## Requirements

You must use output modules that support Redundant Owner connections:

- 1756-OB16IEF output module Peer Ownership and Scheduled Output are **not** supported
- Any Allen-Bradley 1756 input module that supports multiple owners

Redundant owner supports these ControlLogix 1756 Ethernet modules:

- 1756-EN2T
- 1756-EN2TP
- 1756-EN2TR
- 1756-EN2F
- 1756-EN4TR

## Restrictions

The redundant owner solution uses two connections for every I/O module—one for each of the redundant connections. Each connection is shown in the Studio 5000 Logix Designer® I/O Configuration—two entries for each I/O module.

- The two I/O module configurations must match.
- Two connections are used for each I/O module.
- An output module's ideal RPI is 25 ms.
- This speed is for the fastest response to an event. Any output module RPI can be used, but may have increased response time to a fault.
- An input module's ideal RPI is 25 ms, though any input module RPI can be used.
- Direct I/O connections must be used.
- Rack-optimized connections are not supported.

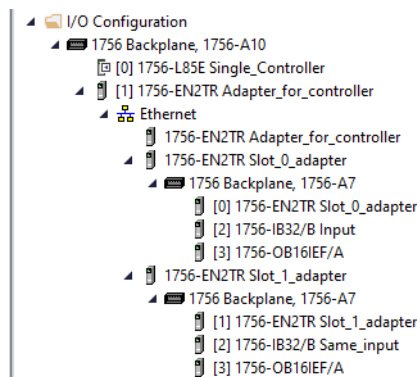
Also see [Redundant Owner Configuration Tags on page 143](#).

Redundant Owner can work in single-controller and multiple-controller applications, with or without redundancy.

Your application determines which controller is the Claiming Owner and which is the Ready Owner. The relationship is similar to a primary and secondary relationship.

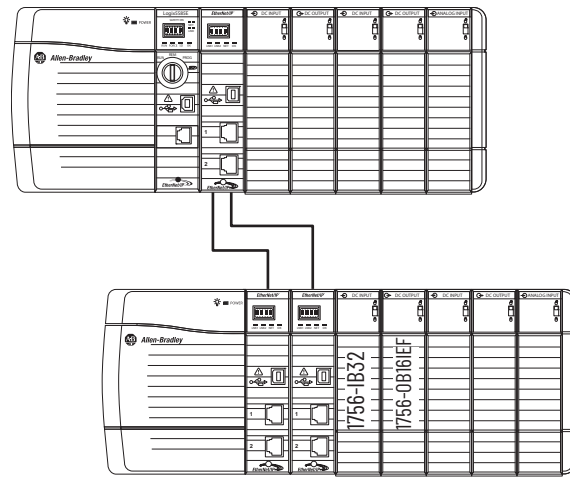
## Single Controller Behavior

A single controller case is the simpler one as both connections are owned by the same controller. Thus there is never a case where one connection is in Run mode and the other connection is in Program mode, so no need to worry about synchronizing.



This can also be achieved by using two communication modules in the Controller chassis.

Figure 18 - Single Controller



32763-M

1. Initialize I/O tags.
2. Set Claim in one connection.
3. Set Ready in the other connection.
4. Monitor which connection is the owner and update tags accordingly (Non Owner before Owner to make sure that they're always in sync).

That single controller behavior should result in the following:

1. Output is not owned until the Claim connection is made.
2. Any disruption in the Claim connection causes the I/O module to switch to the Ready path.
3. When the Claim connection recovers from the disruption, the I/O module switches back to Claim.

### Multiple Controller Behavior

For multiple controllers, the applications synchronize the behavior between the controllers. Note Ethernet modules A and B, and RedundantOutputA and RedundantOutputB.

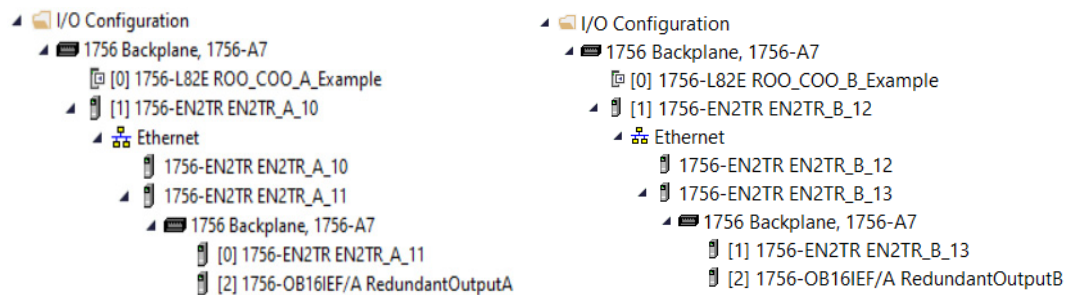
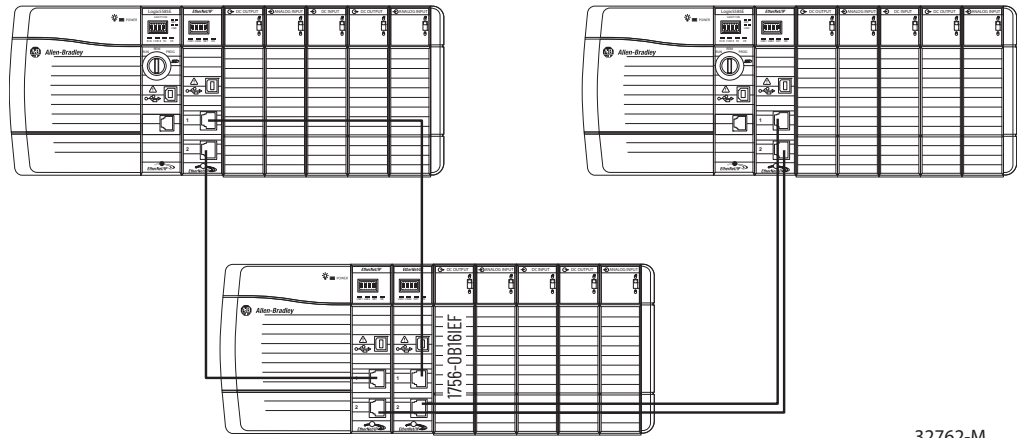


Figure 19 - Multiple Controller



32762-M

**IMPORTANT**

If you use ControlLogix Redundancy as the multiple controller system, then you do not have to manage the dual controllers. If you use any other method, then you must manage the primary and secondary controllers yourself. This application does not manage the controllers, just the I/O.

You can use Produced/Consumed tags for synchronization; however, the Redundant Owner Input tags provide the majority of information needed for synchronization.

- After both controllers decide it is safe, the primary controller sets Claim.
- After the secondary controller sees that the primary controller has taken ownership, the secondary controller sets Ready.
- The secondary controller monitors the I/O and the other controller to determine the current owner.
- If the secondary controller becomes the owner, it can set its Claim bit to take or hold ownership, or can just maintain Ready.
- Once the primary controller comes back and is in Run mode, that controller toggles its Claim bit, and the I/O switches back to that controller.
  - A "S:FS" XIC should be used to disable Claim/Ready bits until the controller has entered Run mode and can evaluate if it should make a Claim or Ready transition.
  - Failure to do this could lead to a 'claim' happening before the controller is back in Run mode. Thus, the output would transition briefly to its Program Mode Safe State before entering Run mode rather than making a smooth transition from secondary controller back to primary controller.
- When the secondary controller sees that the primary controller has retaken ownership, it clears its Claim bit and returns to Ready.
- Monitor for Connection Loss on controller. Add GSV to monitor connection status and clear Claim and Ready flags on Connection Loss to help prevent undesired behavior when connection is restored.
- Monitor for Major Fault on controller. Optionally, add Fault Routine to clear Claim and Ready owner flags if Major Fault is not to be cleared in faulted controller to allow secondary controller to take immediate control.

## Response Times for Output Modules

Response time to a change in ownership is from the time the event takes place until the time that the output modules use the connection data from the redundant connection.

Digital output response to a change in ownership in less than 250 ms. The average time is 220 ms. A faster RPI does not increase the response time. We recommend a digital output RPI of 25 ms.

These recommendations are for response performance:

- Faster RPIs have a longer response time.
- Slower RPIs can have a much longer response time.
- RPIs should be  $4 \times \text{RPI} = 200 \text{ ms}$ ,  $8 \times \text{RPI} = 200 \text{ ms}$ , or  $16 \times \text{RPI} = 200 \text{ ms}$ , so 25 is the recommended RPI.
- If the prior RPI criterion is not met, the CIP™ connection timeout delay minimum will be  $200 \text{ ms} + 1 \text{ RPI}$ . If the  $4/8/16 \times \text{RPI}$  does not equal 200 ms (or 100 ms), then the response will be longer.
- For non-redundant ControlLogix applications, the CIP connection timeout delay minimum is 100 ms.
- Slower RPIs greatly limit the effect of the response time on the application.

---

**IMPORTANT** This information pertains only to output modules.  
Use existing guidelines for setting input module RPIs.

---

## Input Modules

Many ControlLogix 1756 input modules support multiple owners.

The table below lists Bulletin number 1756 digital input modules that **do not** support multiple module owners.

---

**IMPORTANT** You must use input modules that support multiple owners.

---

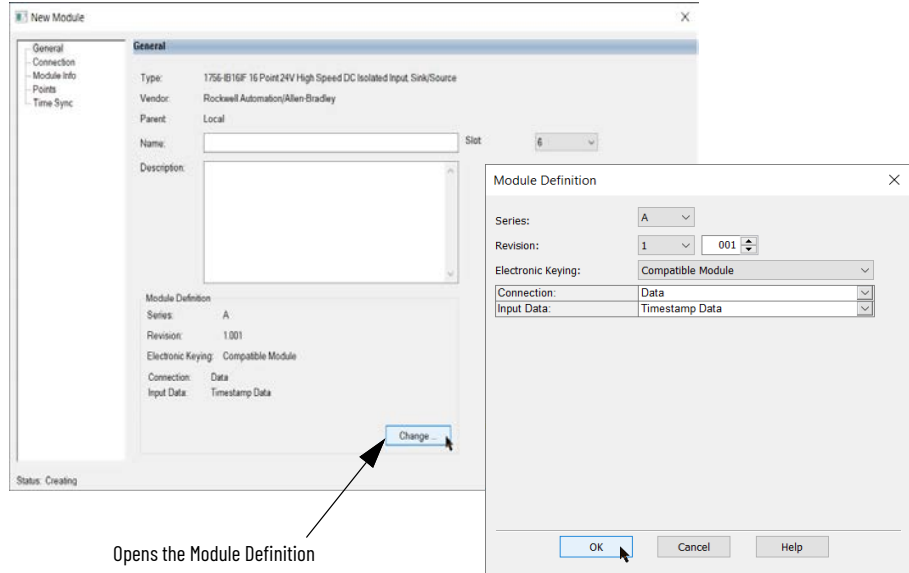
**Table 34 - Input Modules that Do not Support Multiple Owners**

Cat. No.	Module Type	Replacement Module
1756-IB16ISOE	Sequence of Events Digital Input	—

# Configure Fast Input Modules

Follow these steps to configure per-point timestamping and enable COS.

1. On the New Module dialog box, click Change to display the Module Definition dialog box.



2. Use this table to choose a connection format and input data type from the Connection and Input Data dropdown menus.

**IMPORTANT** To enable timestamping, choose Timestamp Data as the input data type.

**Table 35 - Timestamp Data**

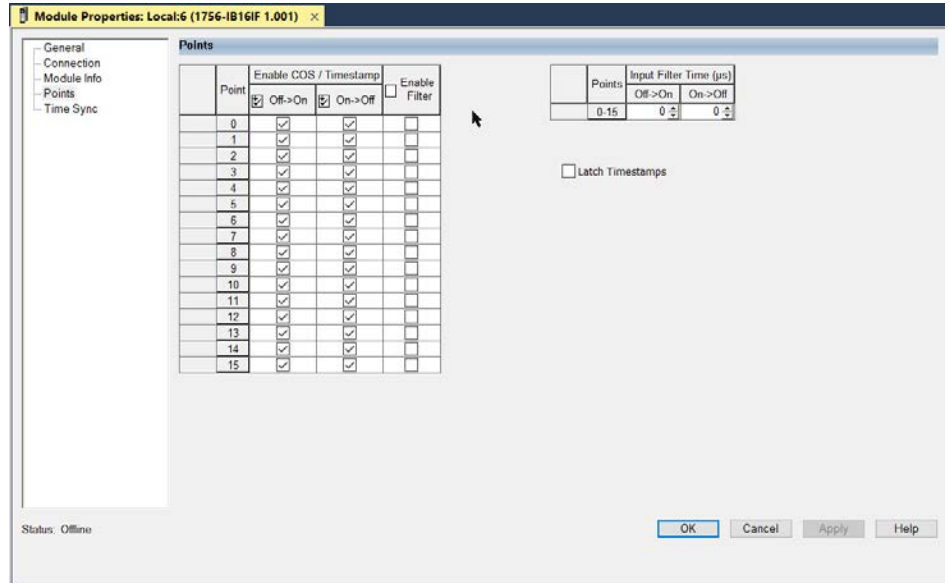
Connection Format	Input Data	Data Return
Data	Timestamp Data	Module returns input data with COS timestamps in CIP Sync system time.
	Data	Module returns input data <b>without</b> COS timestamps. This format is useful when the highest possible throughput is required, and timestamps are not required.
Data with Event	Timestamp Data	Results in two input connections: <ul style="list-style-type: none"> <li>• Connection to return input data with COS timestamps in CIP Sync system time.</li> <li>• Connection to initiate event tasks. See <a href="#">page 48</a>.</li> </ul>
Listen Only	Timestamp Data	These formats have the same definition as those above except that they are Listen-only connections.
	Data	
Listen Only with Event	Timestamp Data	



You can change the connection format at any time after creating a module except when you are online. The AOP applies all the configuration data and creates the tags that are required for the new connection format.

- On the New Module or Module Properties dialog box, click the Points tab.

Timestamp fields only appear on the Configuration tab when you choose Timestamp Data from the Input Data dropdown menu on the Module Definition dialog box.



- Complete the fields as described in the table and click OK.

Table 36 -

Field	Description	Configuration Tag
Enable COS/ Timestamps Off ? On	To enable COS and timestamping for an Off to On transition for a point, check the corresponding checkbox. To disable COS and timestamping for an Off to On transition for a point, clear the corresponding checkbox.	Pt[x].COSOffOnEn
Enable COS/ Timestamps On ? Off	To enable COS and timestamping for an On to Off transition for a point, check the corresponding checkbox. To disable COS and timestamping for an On to Off transition for a point, clear the corresponding checkbox.	Pt[x].COSOnOffEn
Latch Timestamps	Check the checkbox to latch a CIP Sync timestamp for a COS transition: <ul style="list-style-type: none"> <li>When an initial timestamp is latched, timestamps for subsequent COS transitions are dropped.</li> <li>Once a latched timestamp is acknowledged via the corresponding bit in the Pt[x].NewDataOffOnAck or Pt[x].NewDataOnOffAck tag, the timestamp is overridden upon the next COS transition.</li> </ul> <b>IMPORTANT:</b> Timestamps are latched only for points that are enabled for COS and timestamping.	LatchTimestamps

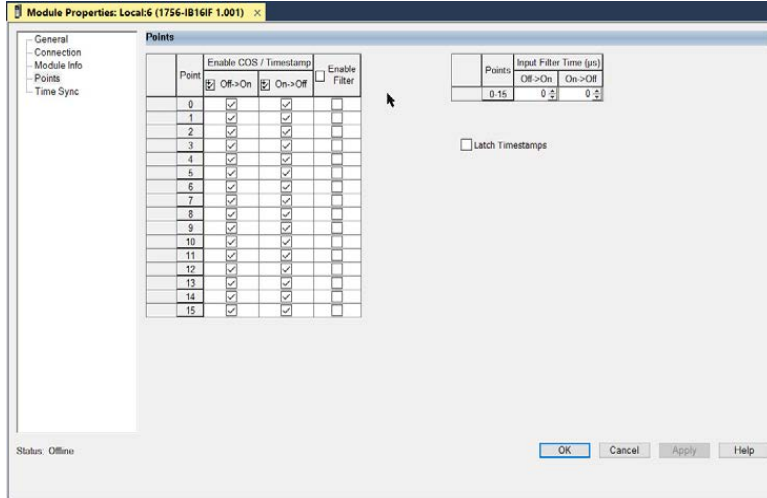
- If you checked the Latch Timestamps checkbox, use program logic or the Logix Designer application tag editor to acknowledge transitions and clear latched timestamps via the Pt[x].NewDataOffOnAck and Pt[x].NewDataOnOffAck output tags.

For more information about module tags, refer to [Appendix A](#).

## Configure Input Filter Times

Follow these steps to configure input filter times.

1. On the Module Properties dialog box, click the Points tab.
2. In the Input Filter Time column, enter Off to On and On to Off input filter times from 0...30,000  $\mu$ s and click OK.



3. Complete the fields as described in this table and click OK.

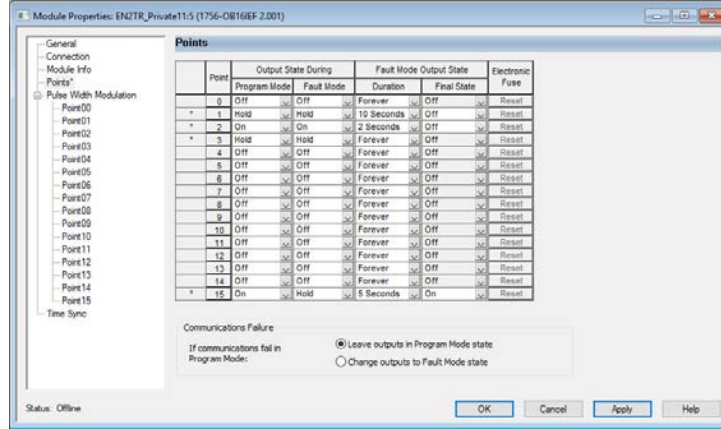
**Table 37 -**

Field	Description	Configuration Tag
Enable Filter	To enable filtering for a point, check the corresponding checkbox. To disable filtering for a point, clear the corresponding checkbox.	Pt[x].FilterEn
Input Filter Time Off ? On	Enter an Off to On input filter time from 0...30,000 $\mu$ s.	FilterOffOn
Input Filter Time On ? Off	Enter an On to Off input filter time from 0...30,000 $\mu$ s.	FilterOnOff

# Configure Fast Output Modules

Follow these steps to configure a fault state delay.

1. On the Module Properties dialog box, click the Points category.



Complete the fields as described in this table and click OK.

Table 38 -

Field	Description	1756-OB16IEF Configuration Tag	1756-OB16IEFS Configuration Tag
Fault Mode Output State Duration	Choose the length of time that you want the output to remain in Fault mode state before transitioning to the final state: <ul style="list-style-type: none"> <li>• 1 second</li> <li>• 2 seconds</li> <li>• 5 seconds</li> <li>• 10 seconds</li> <li>• Forever (default)</li> </ul> <b>IMPORTANT:</b> If you choose Forever, the output remains in the Fault mode state until a connection is re-established. For example, if the Fault mode is Hold, and you specify a duration of Forever, then the output retains its Hold state and does not transition to a Final state if a fault occurs.	Pt[x].FaultValueStateDuration	FaultValueStateDuration
Fault Mode Output State Final State	Choose whether you want the module to transition to an On or Off state after the Fault mode duration time elapses. The default final state is Off. If you chose Forever, you cannot choose a final state. The module retains its current Fault mode state.	Pt[x].FaultFinalState	FaultFinalState

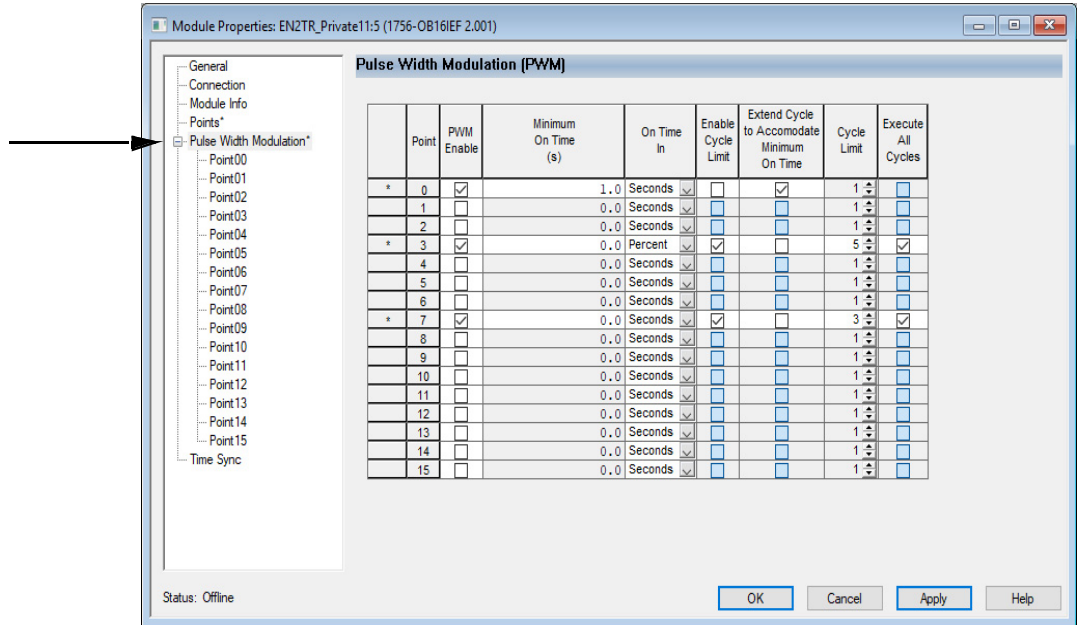
## Configure Pulse-width Modulation

Follow these steps to configure PWM.

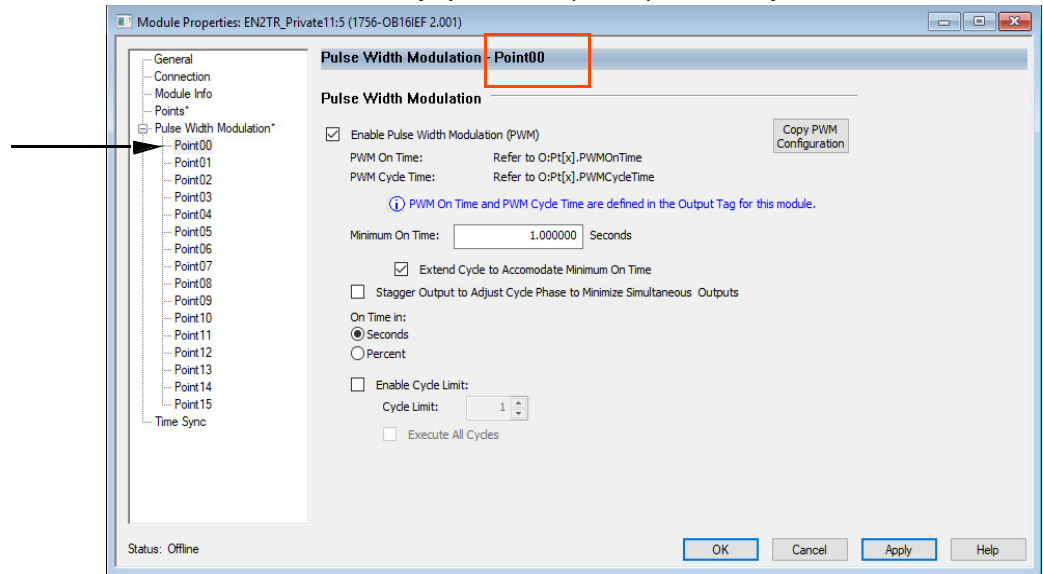
1. Use program logic or the Logix Designer application tag editor to define the Cycle time and On time for an output point via the PWMCycleTime and PWMOnTime output tags.

For more information about module tags, refer to [Appendix A](#).

- On the Module Properties dialog box, click the pulse-width modulation roll-up category for a high-level view.



- Click the PointXX category to view a specific point's configuration.

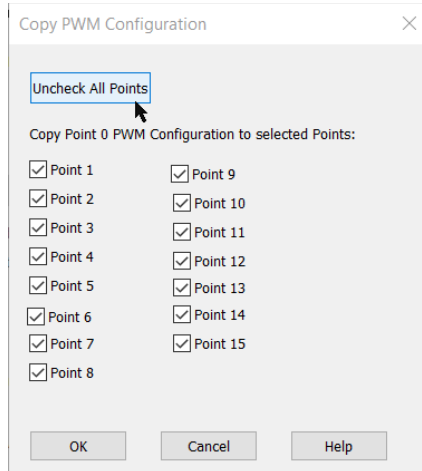


4. To modify that point's configuration, or to set its initial configuration, complete the fields in the PWM area as described in this table.

Field	Description	1756-OB16IEF Tag Name	1756-OB16IEFS Tag Name
Enable pulse-width modulation (PWM)	Check the checkbox to enable PWM. If this checkbox is cleared, all other PWM fields are unavailable, and the PWM On time and cycle time for the point are ignored. By default, PWM is disabled.	C:Pt[x].PWMEEnable	C:PWM.Enable
PWM On Time (view only)	Displays the length of time that a pulse is active, as defined in the PWMOnTime output tag. By default, this value is defined in seconds with a range of 0.0002...3600.0. However, you can define the value as 0...100% of the cycle time by clicking On Time in Percent. <b>IMPORTANT:</b> Before PWM functions, you must enable PWM during configuration and define the PWM cycle time and On time in the PWMCycleTime and PWMOnTime output tags. If PWM is enabled (C:PWMEEnable = 1) and the output is instructed to turn On (O:Data = 1), the output generates a PWM signal.	O:Pt[x].PWMOnTime	O:PWM.OnTime
PWM Cycle Time (view only)	Displays the duration of each pulse cycle, as defined in the PWMCycleTime output tag. This value is always displayed in seconds with a range of 0.001...3600.0 seconds. <b>IMPORTANT:</b> Before PWM functions, you must enable PWM during configuration and define the PWM cycle time and On time in the PWMCycleTime and PWMOnTime output tags. If PWM is enabled (C:PWMEEnable = 1) and the output is instructed to turn On (O:Data = 1), the output generates a PWM signal.	O:Pt[x].PWM CycleTime	O:PWM.CycleTime
Minimum On Time	Type the minimum length of time required for the output to turn On. This value must be defined in seconds. For example, if a heating coil requires a minimum of 2 seconds to heat up, and you enter a value of 2.000 in this field, the shortest pulse that is allowed is never less than 2.000 seconds. The default value of zero disables the feature.	C:Pt[x].PWMMinimumOnTime	C:PWM.MinimumOnTime
Extend Cycle to Accommodate Minimum On Time	Check or clear this checkbox to determine the output behavior when the On time is <b>less than</b> the minimum On time: <ul style="list-style-type: none"> <li>Check the checkbox to increase the duration of the pulse cycle to maintain the On time to Cycle time ratio while considering the minimum On time.</li> <li><b>Note:</b> Extending the cycle time is typically useful only when the On time is a result of a calculation.</li> <li>Clear the checkbox if you do <b>not</b> want to increase the duration of the pulse cycle. In this case, the output does not turn On if the On time is less than the minimum On time.</li> </ul> By default, the checkbox is cleared, and cycles do not extend.	C:Pt[x].PWMEExtendCycle	C:PWM.ExtendCycle
Stagger Output to Adjust Cycle Phase to Minimize Simultaneous Outputs	Check the checkbox to minimize the load on the power system by staggering output transitions. See <a href="#">Figure 15 on page 60</a> . By default, this checkbox is cleared and staggering is disabled. When staggering is disabled for an output point, the output always turns On at the beginning of a pulse cycle.	C:Pt[x].PWMStaggerOutput	C:PWM.StaggerOutput
On Time in Seconds or On Time in Percent	To define PWM On time in seconds, click On Time in Seconds. To define PWM On time as a percentage of the cycle time, click On Time in Percent. By default, the On time is defined in seconds.	C:Pt[x].PWMOnTimeInPercent	C:PWM.OnTimeInPercent
Enable Cycle Limit	Check the checkbox to let only a fixed number of pulse cycles occur. See <a href="#">Figure 12 on page 58</a> . By default, the Enable Cycle Limit checkbox is cleared, and pulse cycles continue to occur until the output turns Off.	C:Pt[x].PWMCycleLimitEnable	C:PWM.CycleLimitEnable
Cycle Limit	Enter the maximum number of pulse cycles that you want to occur on each output logic transition when Enable Cycle Limit is checked: <ul style="list-style-type: none"> <li>If you check the Execute All Cycles checkbox, the specified number of cycles occur even if the Data output tag turns Off before the completion of the specified number of cycles.</li> <li>If you clear the Execute All Cycles checkbox, the specified number of cycles occur only if the Data output tag remains On for enough time for the specified number of cycles. For example, if you specify a cycle limit of 4, and the output turns Off after 3 cycles, the fourth cycle does <b>not</b> occur.</li> </ul> This field is only available when the Enable Cycle Limit checkbox is checked. By default, the cycle limit is 10. Valid values are 1...27.	C:Pt[x].PWMCycleLimit	C:PWM.CycleLimit
Execute All Cycles	Check the checkbox to always execute the number of cycles that are specified in the Cycle Limit field even if the Data output tag turns Off. For example, if you specify a cycle limit of 2, and the output turns Off after 1 cycle, the second cycle still occurs despite the output turning Off. See <a href="#">Figure 13 on page 58</a> . If the output logic transitions multiple times before the cycle limit is reached, all subsequent transitions are ignored until the cycle limit is reached. Once the cycle limit is reached, a new cycle sequence begins. This field is only available when the Enable Cycle Limit checkbox is checked. By default, the Execute All Cycles checkbox is cleared.	C:Pt[x].PWMEExecuteAllCycles	C:PWM.ExecuteAllCycles

5. To copy the current configuration to one or more of the remaining output points, so that multiple outputs share the same PWM behavior, do the following:
  - a. Select the PointXX category to copy.
  - b. Click Copy PWM Configuration.
  - c. On the Copy PWM Configuration dialog box, check the points to which to apply the current configuration and click OK.

By default, all points are checked.

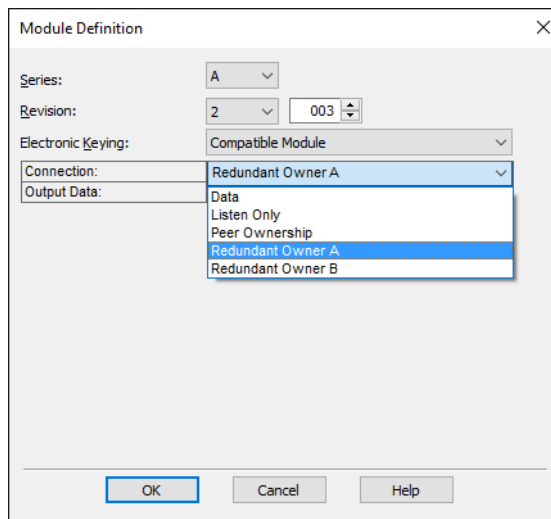


6. To save the configuration for each output point that you specified, click OK.

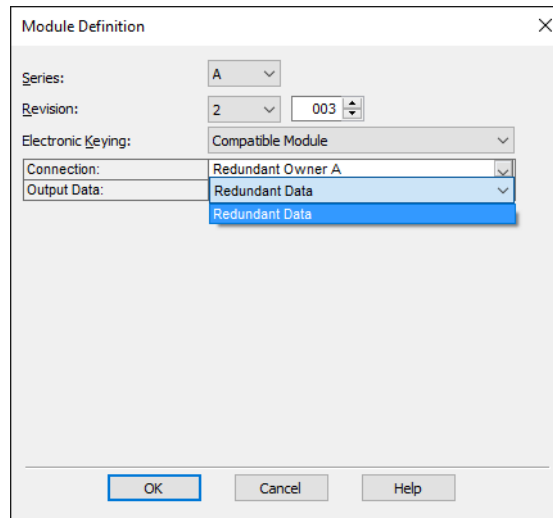
### Configure the Output Module as Redundant Owner

In a redundant owner application, configure one output module as Redundant Owner A and the second as Redundant Owner B. In our example project, Redundant Owner A is the module that is connected to Ethernet B.

1. In the Module Definition dialog box, in the Connection field, choose Redundant Owner A.



- In the Output Data field, choose Redundant Data.



- To configure the second output module as Redundant Owner B, repeat steps 1 and 2, choosing Redundant Owner B and Redundant Data.

The module definition is shown in the Module Properties dialog box.

## Implement Redundant Ethernet Modules

Follow these steps to implement Redundant Ethernet modules in your project.

- Add the additional I/O configuration to your project for the second Ethernet module. In the example project, this is EN2T\_B.



To simplify testing, you may want to create a project just for the redundant Ethernet modules and test that project before adding in your application code.

We recommend following the module naming examples that are used in the example project.

- Ethernet module in Slot 0: EN2T\_A\_<your name for the chassis>
  - Ethernet module in Slot 1: EN2T\_B\_<your name for the chassis>
  - Reference the example application for more naming examples
- Add in the task for I/O\_Mapping\_25ms.
  - Set the periodic rate of this task to 25 ms or a multiple of the fastest output module RPI.
  - Import the example Program into the I/O\_Mapping\_25ms task.
  - Modify the imported routines for each chassis that you have in your system. In the example project the routines are called R001, R002, and so on. One chassis per routine.
  - Add an I/O module to the chassis routines.
  - Insert an Add-On Instruction for each digital input module that you have in the chassis.
  - If the digital input module is only 16 channels, set the I16\_Ch tag to 1 (true). This limits input processing to the first 16 Booleans and reduce controller scan time. If you have a digital module that is not compatible with any of the Add-On Instructions from the example project, you can use the Redundant\_Digital\_Input Add-On Instruction to map each channel individually. You can also create your own Add-On Instruction for the module.
  - Add the default value for each channel to the Add-On Instruction. The default value is the value that the input is set to when communication through both Ethernet modules to the input module is faulted.

10. Insert a rung with two CPS (Synchronous Copy File) instructions for each Digital Output module that you have in the chassis.  
In the example project, the rung number was equal to the slot number of the module being referenced. We used rungs with only an NOP (No Operation) instruction to hold the rung number for slots without an analog module.
11. In each routine add rungs for the analog modules that you have.  
Most analog modules have their own Add-On Instructions (AOI).  
Be aware of the channel count of the AOI and module.
  - If the analog module is in Differential mode, the channel count is cut in half.
  - If you have an analog module that is not compatible with any of the Add-On Instructions from the example project, you can use the Redundant\_Analog\_Input Add-On Instruction to map each channel individually. You can also create your own Add-On Instruction for the module.
  - In the example project, the rung number was equal to the slot number of the module being referenced. We used rungs with only an NOP (No Operation) instruction to hold the rung number for slots without an analog module.
12. Keep the Add-On Instruction at Hold Last State or set Hold Last State to 0 (zero) to use the default value.
13. Add the default value for each channel to the Add-On Instruction.
14. The default value is the value that the input is set to when communication through both Ethernet modules to the input module is faulted.

## Response Times for Output Modules

Response time to a change in ownership is from the time the event takes place until the time that the output modules use the connection data from the redundant connection.

Digital output response to a change in ownership in less than 250 ms. The average time is 220 ms. A faster RPI does not increase the response time. We recommend a digital output RPI of 25 ms.

These recommendations are for response performance:

- Faster RPIs have a longer response time.
- Slower RPIs can have a much longer response time.
- RPIs should be  $4 \times \text{RPI} = 200 \text{ ms}$ ,  $8 \times \text{RPI} = 200 \text{ ms}$ , or  $16 \times \text{RPI} = 200 \text{ ms}$ , so 25 is the recommended RPI.
- If the prior RPI criterion is not met, the CIP connection timeout delay minimum will be  $200 \text{ ms} + 1 \text{ RPI}$ . If the  $4/8/16 \times \text{RPI}$  does not equal 200 ms (or 100 ms), then the response is longer.
- For non-redundant ControlLogix applications, the CIP connection timeout delay minimum is 100 ms.
- Slower RPIs greatly limit the effect of the response time on the application.

---

**IMPORTANT** This information pertains only to output modules.  
Use existing guidelines for setting input module RPIs.

---

## Troubleshoot Your Module

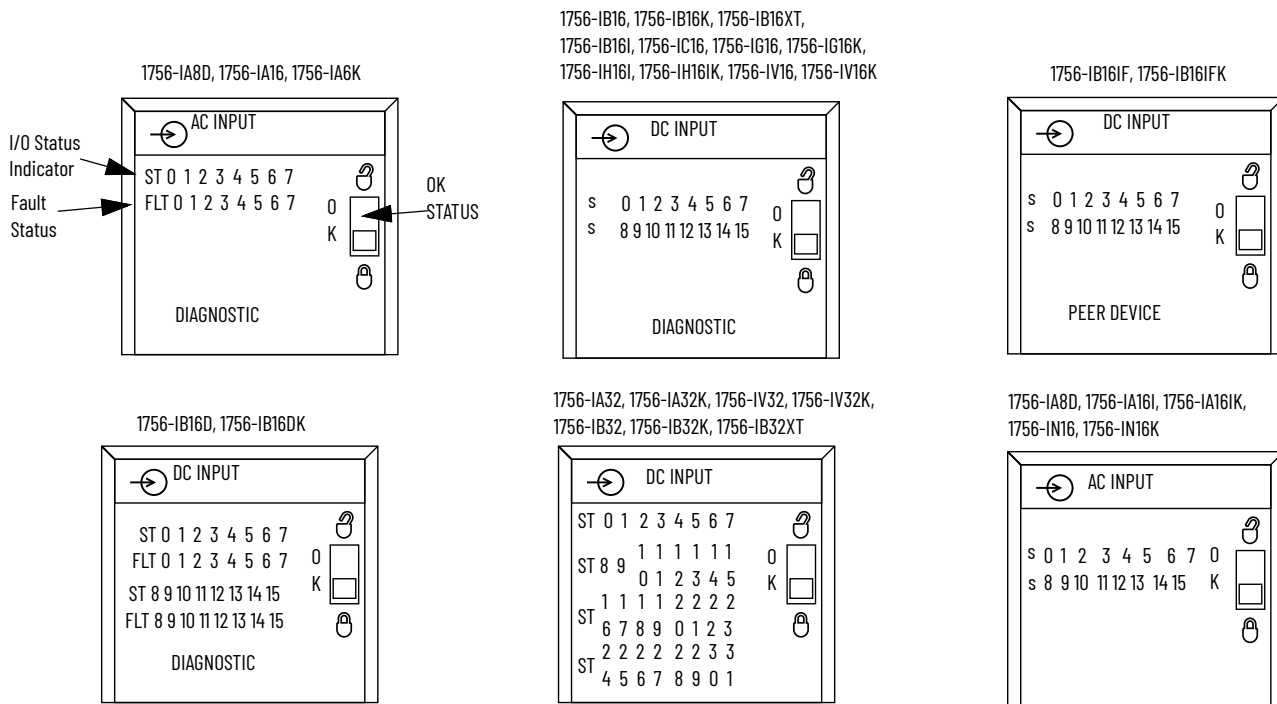
### Status Indicators

ControlLogix® input modules support the status indicators that are described in [Table 39](#). The available status indicators vary by module catalog number, as shown in [Figure 20 on page 89](#).

**Table 39 - Status Indicators for Input Modules**

Indicator	Status	Description
OK Status	Steady green	The inputs are being multicast and in a normal operating state.
	Flashing green	The module has passed internal diagnostics, but is not multicasting inputs or is inhibited. Uninhibit the connection or establish a connection to enable communication to the module.
	Steady red	The module must be replaced.
	Flashing red	Previously established communication has timed out. Check the controller and chassis communication.
I/O Status	Yellow	The input is On.
Fault Status	Red	The input has encountered a fault. Check the input point at the controller.

**Figure 20 - Input Module Status Indicators by Catalog Number**



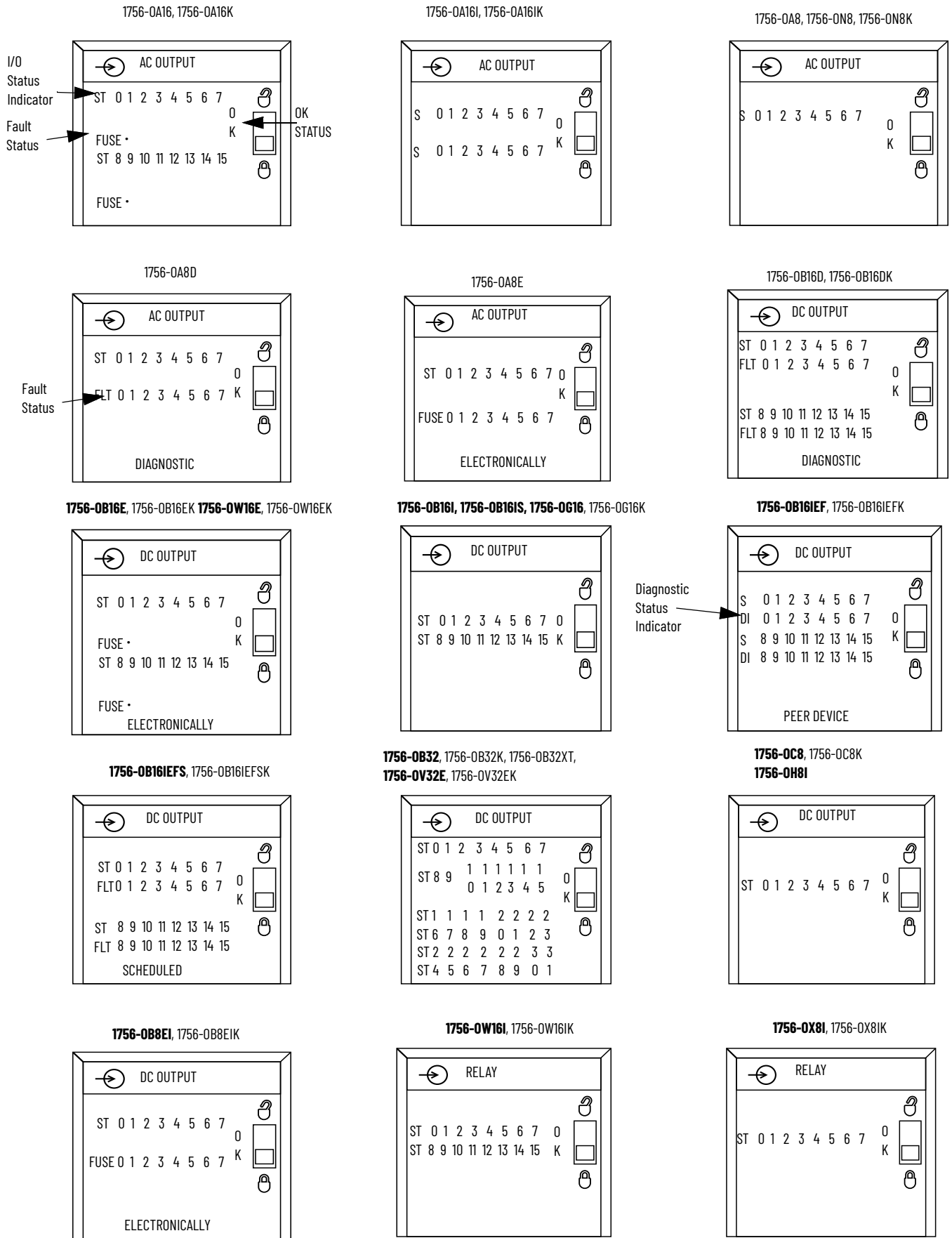
## Status Indicators for Output Modules

ControlLogix output modules support the status indicators that are described in this table. The available status indicators vary by module catalog number, as shown in [Figure 21 on page 91](#).

**Table 40 - Status Indicators for Output Modules**

Indicator	Status	Description
OK Status	Steady green	The outputs are actively being controlled by a system processor.
	Flashing green	The module has passed internal diagnostics but is not actively controlled or it is inhibited or controller is in Program mode. Uninhibit connection, establish connection or transition controller to Run mode to enable communication to module.
	Steady red	The module must be replaced.
	Flashing red	Previously established communication has timed out. Check the controller and chassis communication.
I/O Status	Yellow	The output is On.
Fuse Status	Red	A short overload fault has occurred for a point in this group. Check the wiring for a short-overload. Also check the Module Properties dialog box in the Logix Designer application and reset the fuse.
Fault Status	Red	The output has encountered a fault. Check the output point at the controller.
Diagnostic Status	Steady red	The output has encountered a fault. Check the output point at the controller.
	Flashing red	The output is listening for peer inputs and using the inputs to determine the state of the output point.

Figure 21 - Output Module Status Indicators by Catalog Number




## Use the Logix Designer Application for Troubleshooting

In addition to the status indicator display on the module, the Logix Designer application alerts you to fault conditions.

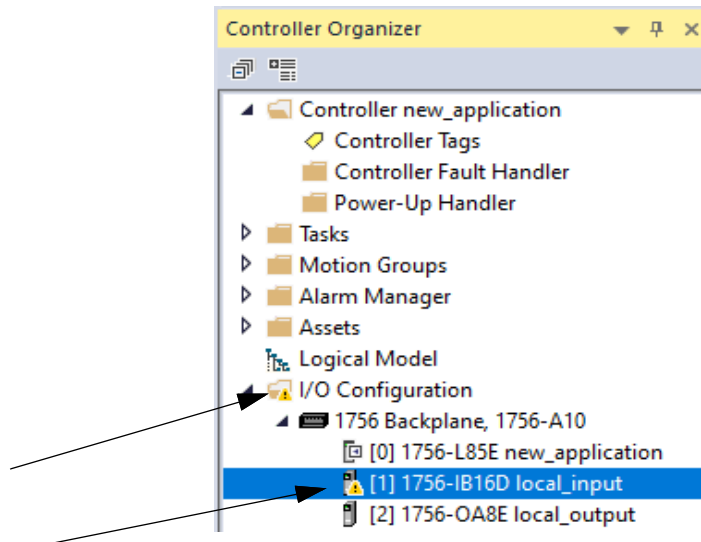
Fault conditions are reported in these ways:

- Warning signal on the main screen next to the module—This occurs when the connection to the module is broken.
- Message in a screen’s status line.
- Notification in the Tag Editor—General module faults are also reported in the Tag Editor. Diagnostic faults are reported only in the Tag Editor.
- Status on the Module Info tab.

These figures display fault notification in the Logix Designer application.

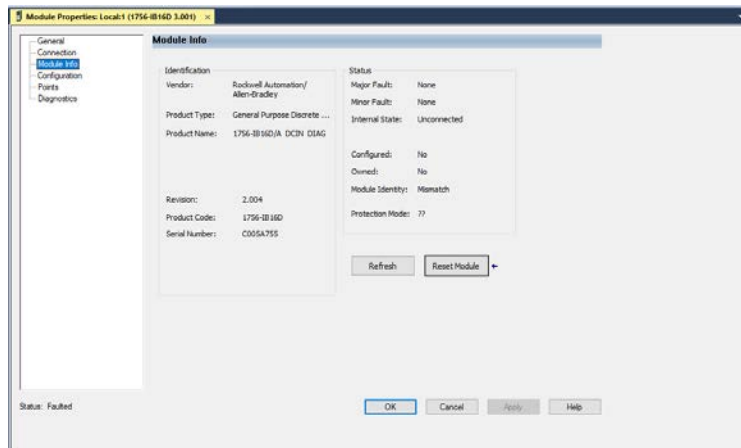
A warning icon  appears in the I/O Configuration tree when a communication fault occurs.

**Figure 22 - Warning Signal on Main Screen**



The major and minor faults are listed on the Module Info tab in the Status section.

**Figure 23 - Fault Message in Status Line**



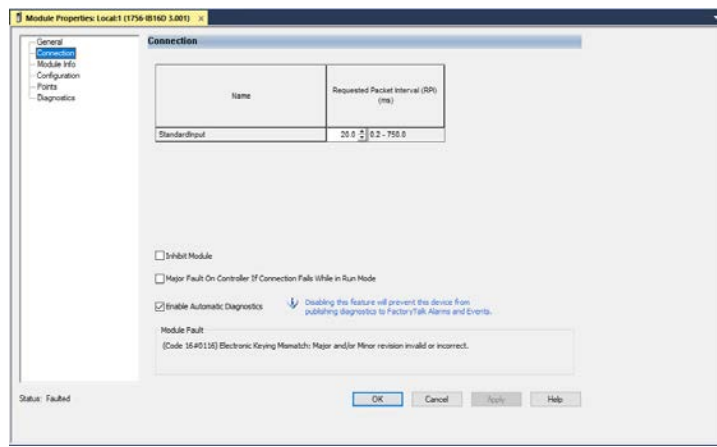
The Value field displays 65535 to indicate that the module connection has been broken.

Figure 24 - Notification in Tag Editor

Name	Value	Data Type
Local:1:C	{...}	AB:1756_DI_DC_Diag:C:0
Local:1:I	{...}	AB:1756_DI_DC_Diag:I:0
Local:1:I.Fault	2#1111_111...	DINT
Local:1:I.Fault.0	1	BOOL
Local:1:I.Fault.1	1	BOOL

## Fault Type Determination

When you are monitoring a module's configuration properties in the Logix Designer application and receive a Communication fault message, the Connection tab lists the type of fault under Module Fault.



## Troubleshooting Fast Modules

**Issue:** I/O Module Connection Error or I/O Module Configuration Error (yellow triangle in the Logix Designer application I/O configuration)

1. Verify that both entries in the Logix Designer application I/O Configuration have the same RPI.
2. Verify that both entries in the Logix Designer application I/O Configuration have the same configuration.

You can toggle the Sync\_Module\_Config BOOL tag in the IO\_Mapping Program MainRoutine Rung 1. This triggers a copy of the module A configuration to the module B configuration, making them the same.

---

**IMPORTANT** This copy will not change the module RPI.  
The two I/O module entries must have the same configuration.

---

3. Remove the module from the chassis.
4. Reinsert the module.
5. If you still have the error, delete one of the module entries in the Logix Designer application I/O Configuration and add it into your project again.

These troubleshooting steps below assume that the system was commissioned and running without any errors before the issue occurred.

**Issue:** Output is ON in the user application code, but is not ON at the Output module.

1. Verify that at least one of the connections to the output module is active.
2. Verify that the output is ON in the output module tags.
3. Confirm whether all of the connections in the chassis are faulted, including the Ethernet modules.
4. Troubleshoot the Ethernet communication issue to the Ethernet modules.
5. Troubleshoot the error code displayed in Module Properties > Connection tab.

In the example project, this is EN2T\_B.

**Issue:** Incorrect output behavior on response

1. Verify that only one of the connections to the Output module has the .OwnerClaim output set to true (1).
2. Verify that only one of the connections to the Output module has the .OwnerReady output set to true (1).

Name	Value	Style	Data Type
ENET_A_R002:2:O.OwnerClaim	1	Decimal	BOOL
ENET_A_R002:2:O.OwnerReady	0	Decimal	BOOL

Name	Value	Style	Data Type
ENET_B_R002:2:O.OwnerClaim	0	Decimal	BOOL
ENET_B_R002:2:O.OwnerReady	1	Decimal	BOOL

For all of the issues described in this section, you can consult product manuals, Rockwell Automation® Knowledgebase, and Rockwell Automation Technical Support.

## Wiring Diagrams

This chapter provides wiring diagrams for all ControlLogix® digital modules.

**Table 41 - Types of Digital I/O Modules**

Digital I/O Type	Description
Diagnostic	These modules provide diagnostic features to the point level. These modules have a <b>D</b> at the end of the catalog number.
Electronic fusing	These modules have internal electronic fusing to prevent too much current from flowing through the module. These modules have an <b>E</b> at the end of the catalog number.
Individually isolated	These modules have individually isolated inputs or outputs. These modules have an <b>I</b> at the end of the catalog number.
Fast	These modules provide fast response times. These modules have an <b>F</b> at the end of the catalog number.
Conformal coated	These modules provide extended protection in harsh, corrosive environments. These modules have a <b>K</b> or an <b>XT</b> at the end of the catalog number.

**Table 42 - 1756 I/O Module Features**

Module Type	Features
1756 digital AC input modules	<ul style="list-style-type: none"> <li>• Change of state: Software configurable</li> <li>• Timestamp of inputs: ±200 µs</li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>
1756 digital AC output modules	<ul style="list-style-type: none"> <li>• Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>• Fault states per point: Hold last state, on or off (off is default)</li> <li>• States in Program mode per point: Hold last state, on or off (off is default)</li> <li>• Fusing: <ul style="list-style-type: none"> <li>- 1756-0A8D, 1756-0A8E: Electronically fused per point</li> <li>- 1756-0A16: Mechanically fused/group, 3.15 A @ 250V AC slow blow, 1500 A interruption current, Littelfuse p/n H2153.15</li> <li>- All other modules: Not protected. A fused IFM is recommended to protect outputs (see publication <a href="#">1492-TD008</a>)</li> </ul> </li> <li>• Module keying: Electronic, software configurable</li> <li>• RTB keying: User-defined mechanical</li> </ul>

Table 42 - 1756 I/O Module Features

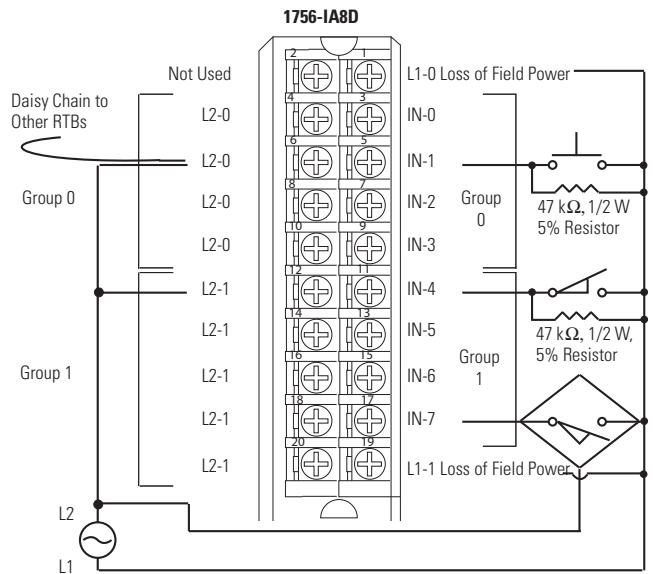
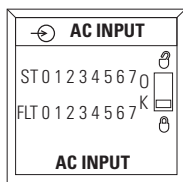
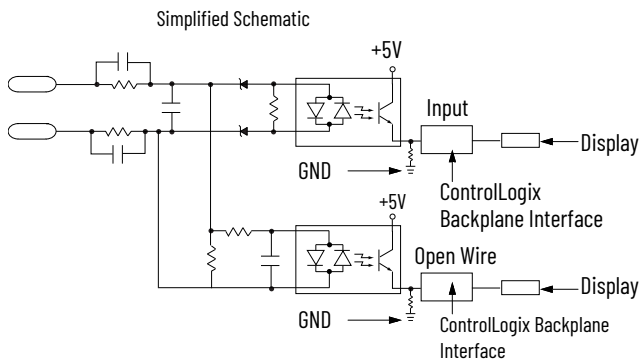
Module Type	Features
1756 digital DC input modules	<ul style="list-style-type: none"> <li>Reverse polarity protection: All modules except 1756-IG16 module</li> <li>Change of state: Software configurable</li> <li>Timestamp of inputs:                             <ul style="list-style-type: none"> <li>±100 µs for sequence of events modules<sup>(1)</sup></li> <li>±200 µs for all other modules</li> </ul> </li> <li>Module keying: Electronic, software configurable</li> <li>RTB Keying: User-defined mechanical</li> </ul>
1756 digital DC output modules	<ul style="list-style-type: none"> <li>Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>Fault states per point: Hold last state, on or off (off is default)</li> <li>States in Program mode per point: Hold last state, on or off (off is default)</li> <li>Fusing:                             <ul style="list-style-type: none"> <li>1756-OB8EI, 1756-OB16D, 1756-OB16E, 1756-OB16IEF, 1756-OB16IEFS, 1756-OV16E, 1756-OV32E: Electronically fused per point</li> <li>All other modules not protected. A fused IFM is recommended to protect outputs. See publication <a href="#">1492-TD008</a>.</li> </ul> </li> <li>Module keying: Electronic, software configurable</li> <li>RTB keying: User-defined mechanical</li> </ul>
1756 digital contact modules	<ul style="list-style-type: none"> <li>Scheduled outputs: Synchronization within 16.7 seconds maximum, reference to the Coordinated System Time</li> <li>Configurable fault states per point: Hold last state, on or off (off is default)</li> <li>Configurable states in Program mode per point: Hold last state, on or off (off is default)</li> <li>Fusing: Not protected. A fused IFM is recommended to protect outputs (See publication <a href="#">1492-TD008</a>)</li> <li>Module keying: Electronic, software configurable</li> <li>RTB keying: User-defined mechanical</li> </ul>

(1) For details, see the ControlLogix Sequence of Events Module Installation Instructions, publication [1756-IN592](#), and the ControlLogix Sequence of Events Module User Manual, publication [1756-UM528](#).

**IMPORTANT** For the latest I/O module specifications, see the 1756 ControlLogix I/O Modules Technical Specifications, publication [1756-TD002](#).

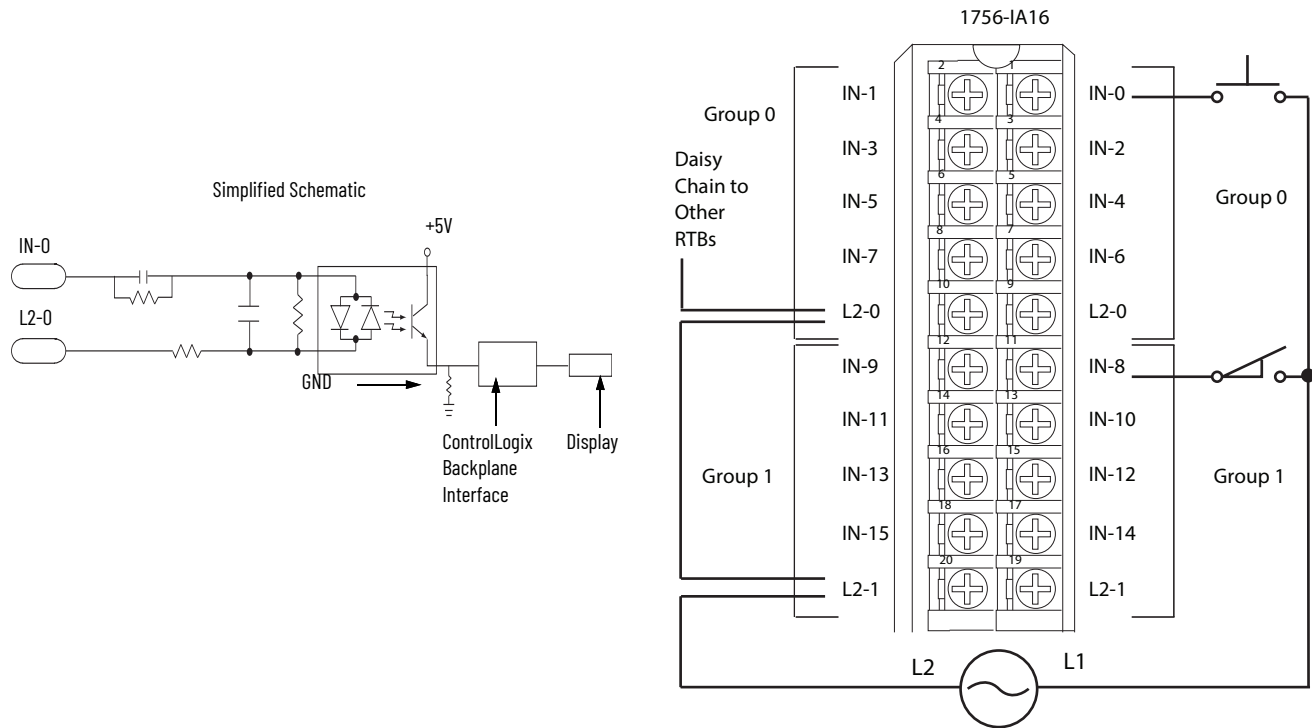
### 1756-IA8D

#### ControlLogix AC (79...132V) diagnostic input module



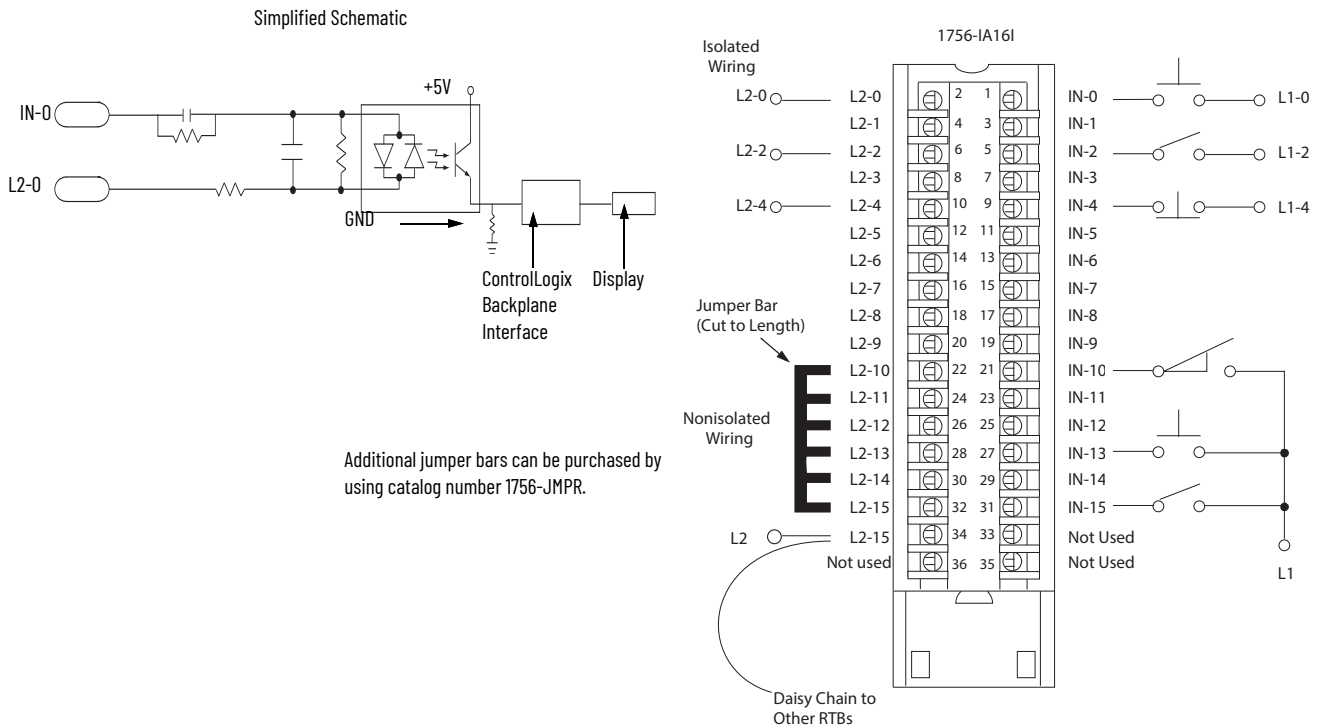
## 1756-IA16, 1756-IA16K

### ControlLogix AC (74...132V) input module



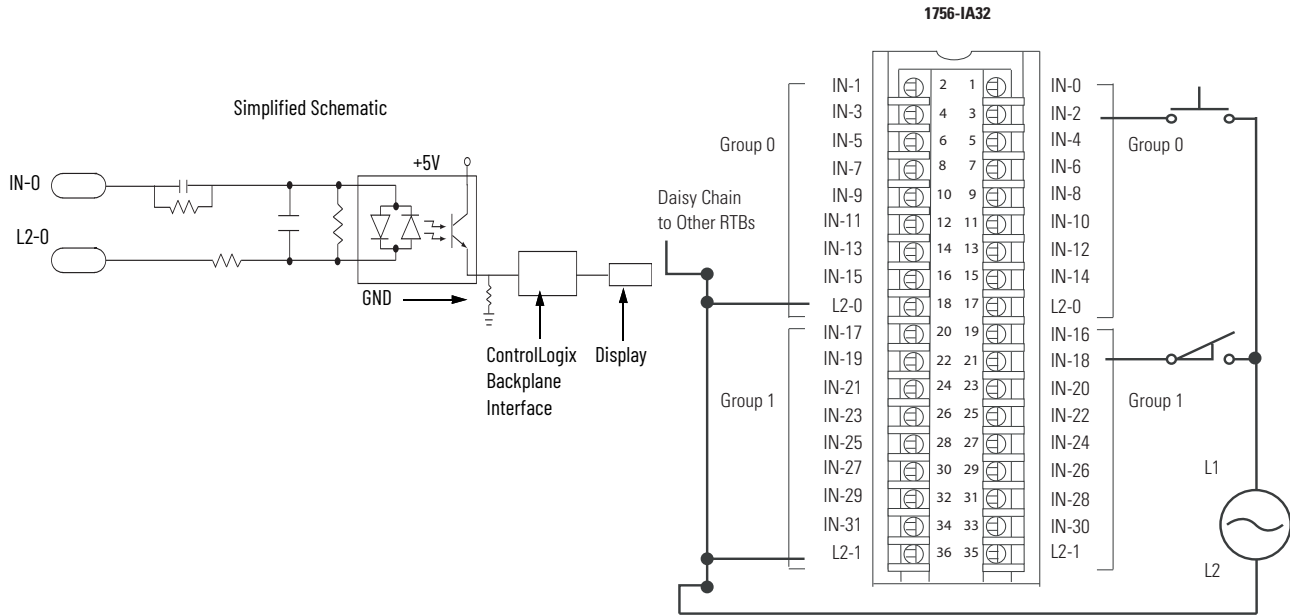
## 1756-IA16I, 1756-IA16IK

### ControlLogix AC (79...132V) isolated input module



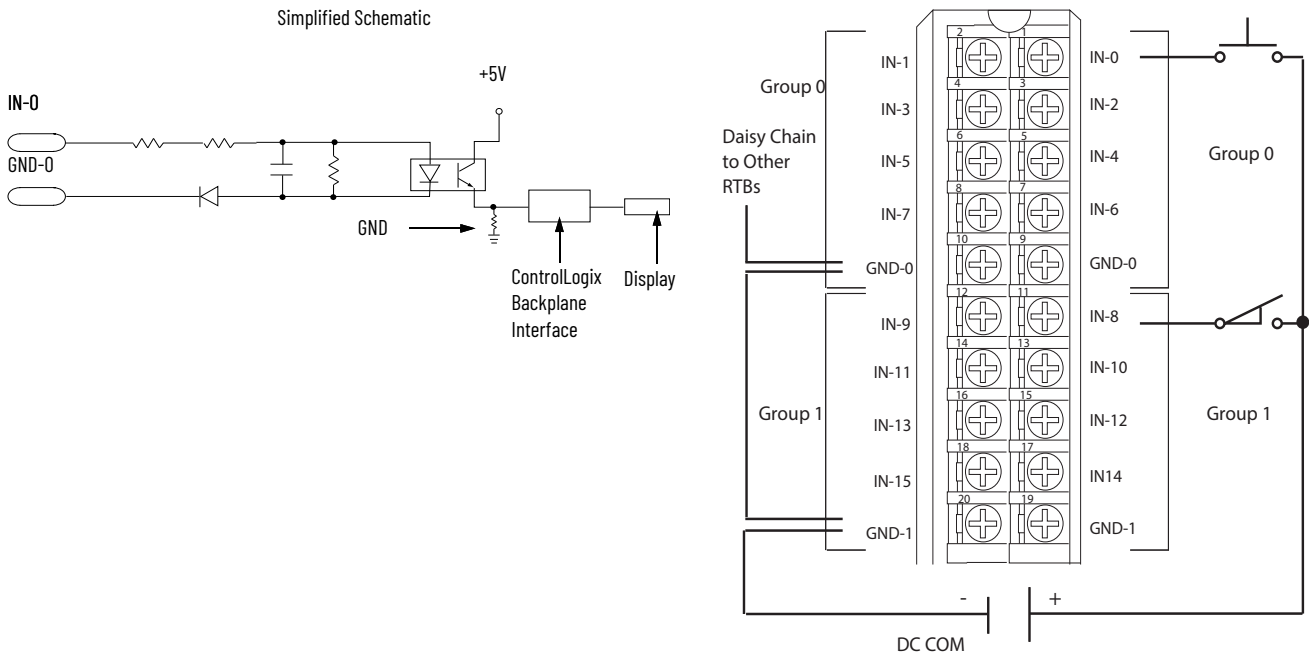
## 1756-IA32, 1756-IA32K

ControlLogix AC (74...132V) input module



## 1756-IB16, 1756-IB16K, 1756-IB16XT

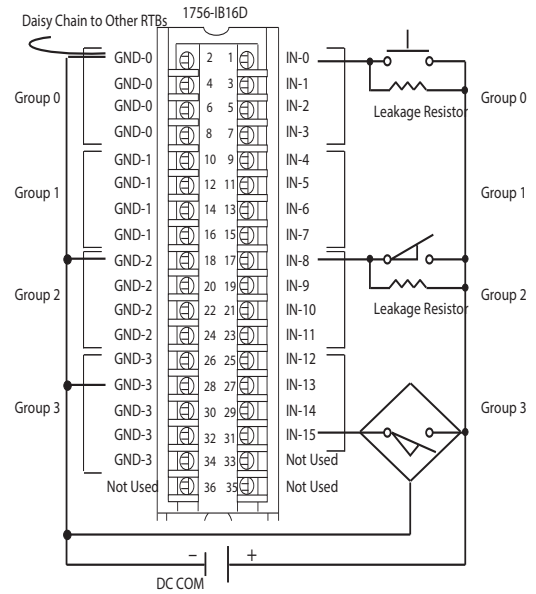
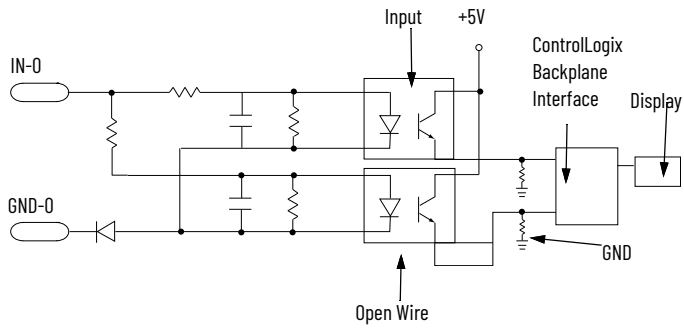
ControlLogix DC (10...31.2V) input module



# 1756-IB16D, 1756-IB16DK

## ControlLogix DC (10...30V) diagnostic input module

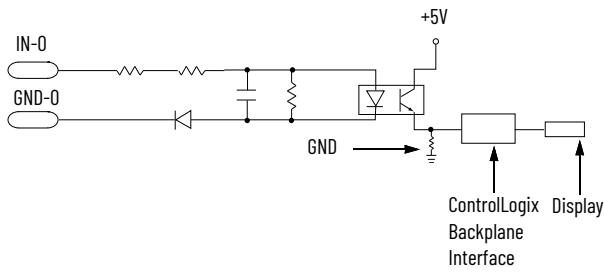
Simplified Schematic



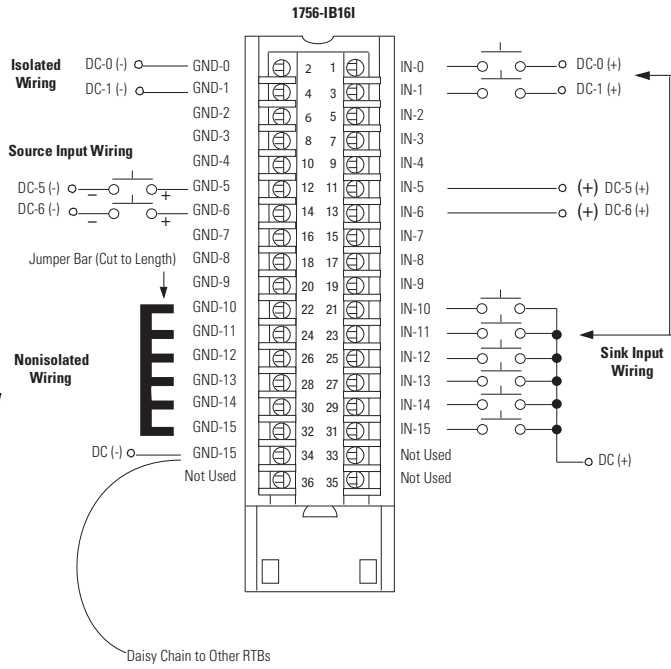
Recommended Leakage Resistor Size 1/4 W, 5%	Supply Voltage
3.9K	10V DC
5.6K	12V DC
15K	24V DC
20K	30V DC

# 1756-IB16I, 1756-IB16IK

## ControlLogix DC (10...30V) isolated input module

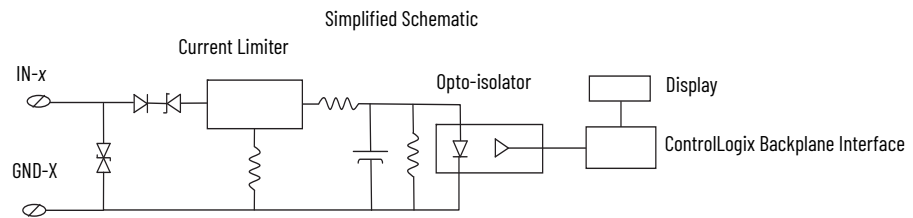
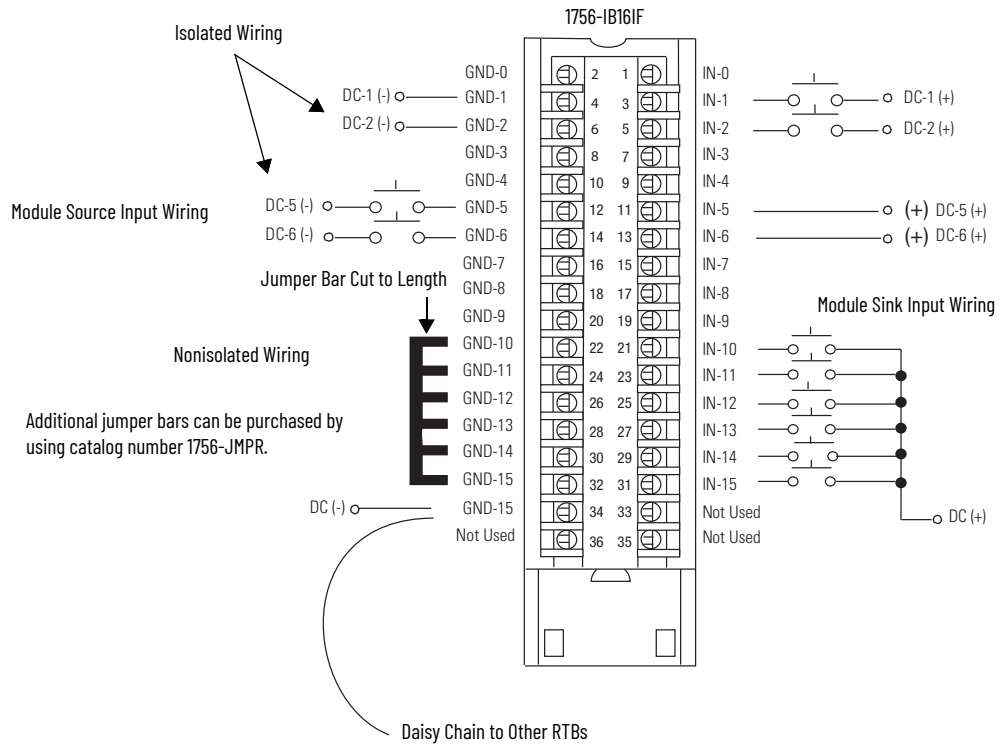


Additional jumper bars can be purchased by using catalog number 1756-JMPR.



# 1756-IB16IF, 1756-IB16IFK

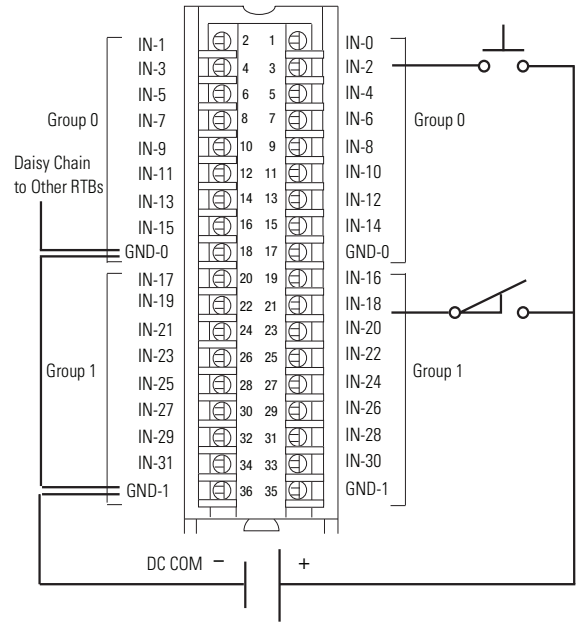
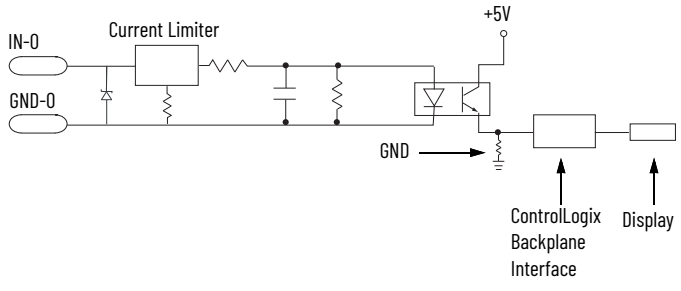
ControlLogix DC (10...30V) sinking or sourcing, isolated, fast input module



## 1756-IB32, 1756-IB32K, 1756-IB32XT

### ControlLogix DC (10...31.2V) input module

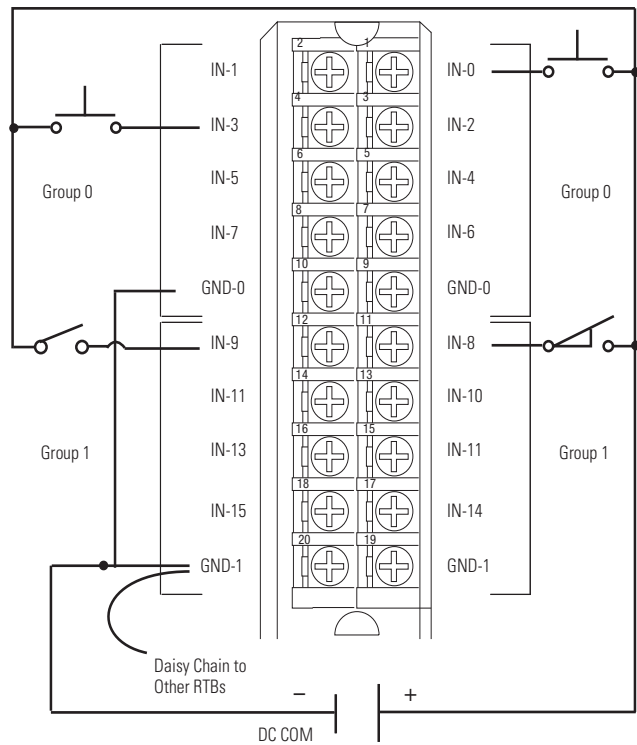
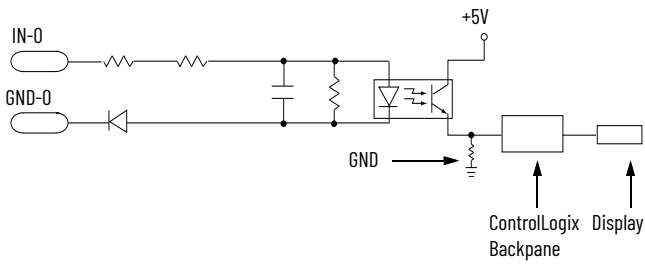
Simplified Schematic



## 1756-IC16

### ControlLogix DC (30...60V) input module

Simplified Schematic

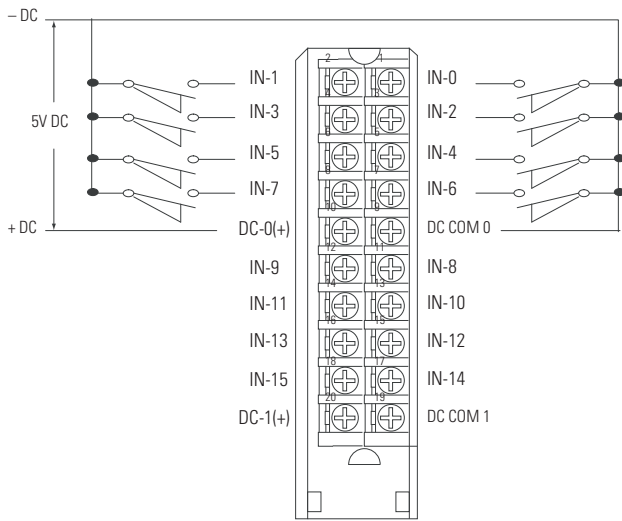


# 1756-IG16, 1756-IG16K

## ControlLogix TTL input module

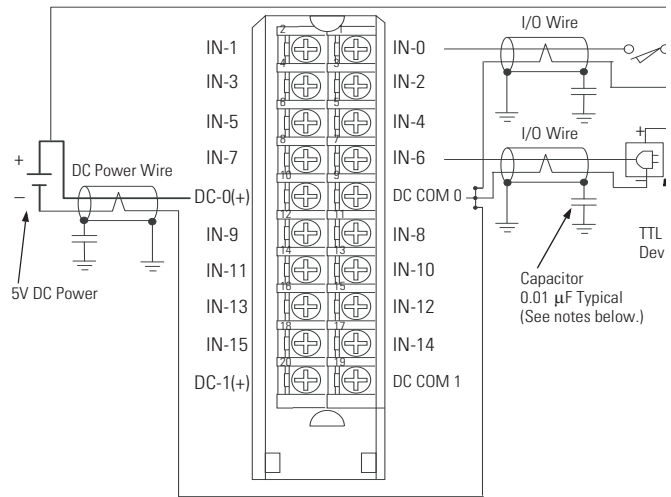
Standard Wiring

1756-IG16

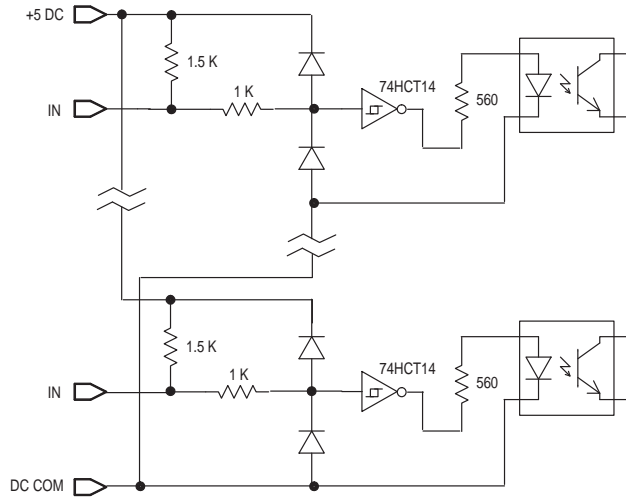


CE-Compliant Wiring

1756-IG16

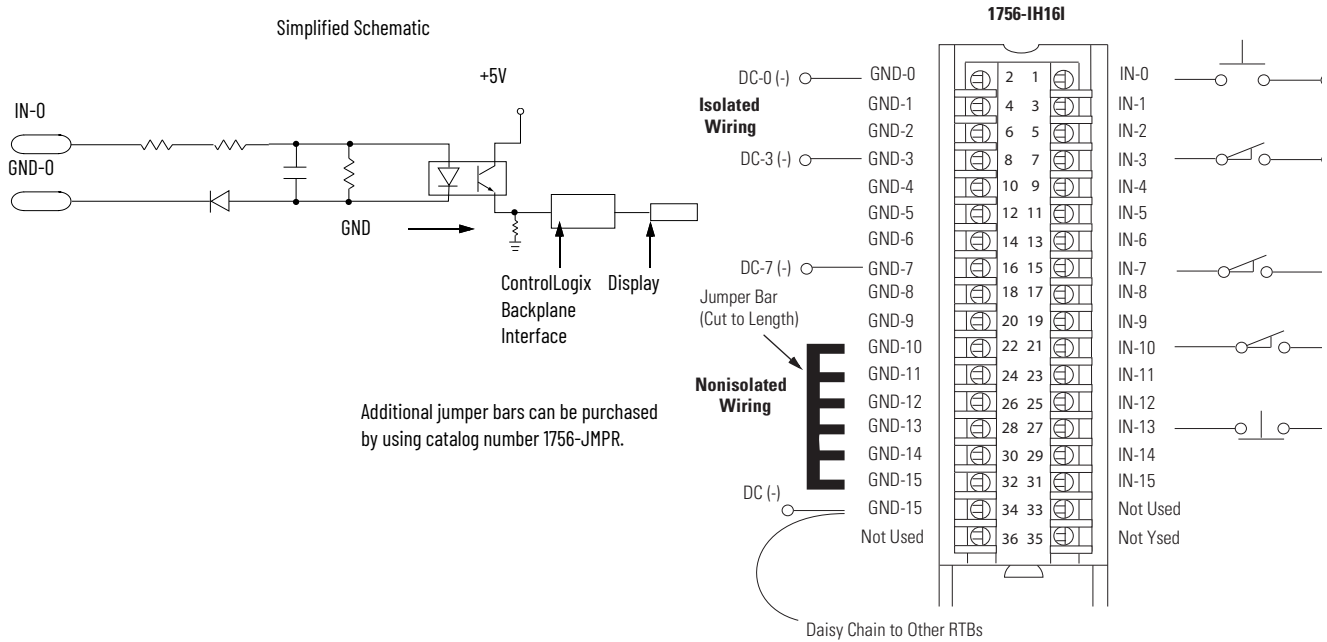


Simplified Schematic



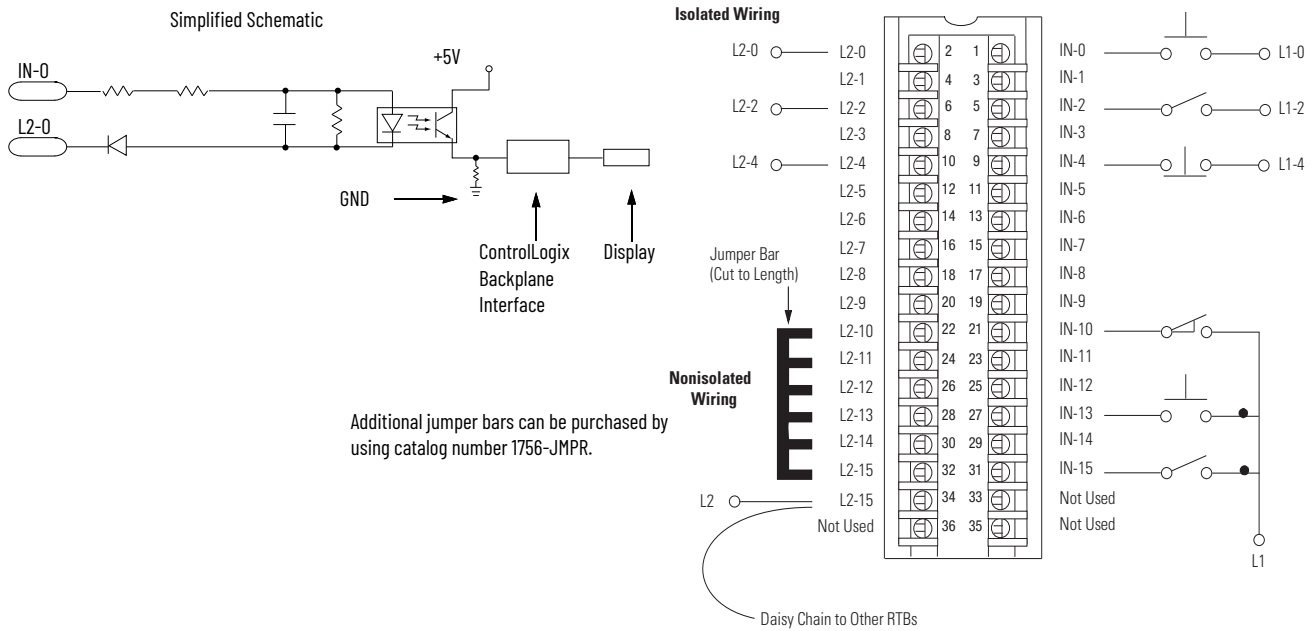
## 1756-IH16I, 1756-IH16IK

### ControlLogix DC (90...146V) isolated input module



## 1756-IM16I, 1756-IM16IK

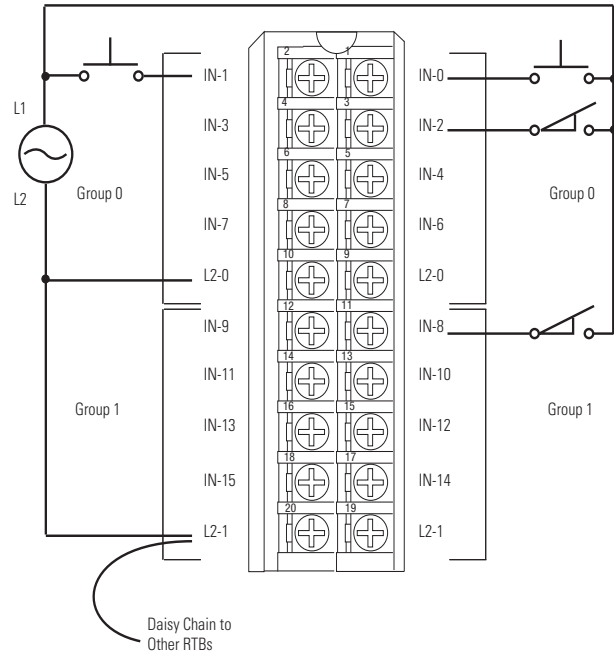
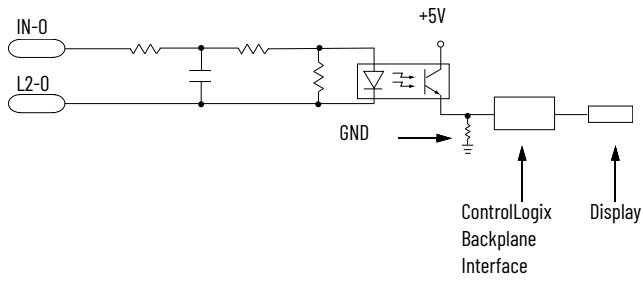
### ControlLogix AC (159...265V) input module



## 1756-IN16, 1756-IN16K

### ControlLogix AC (10...30V) input module

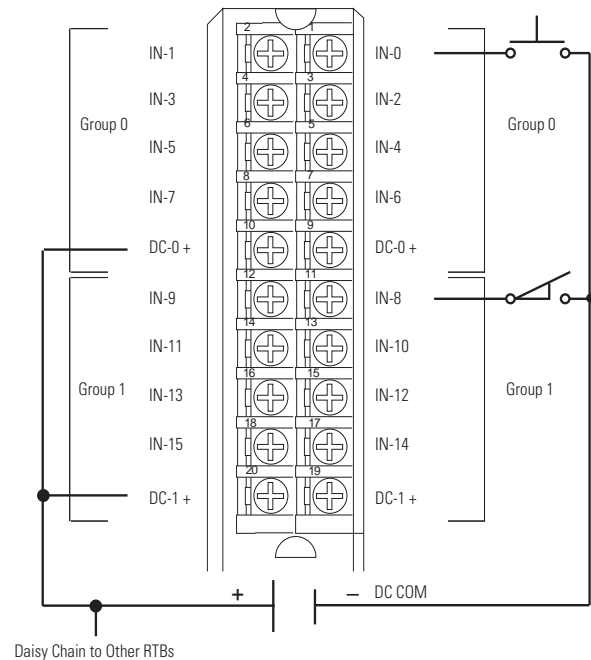
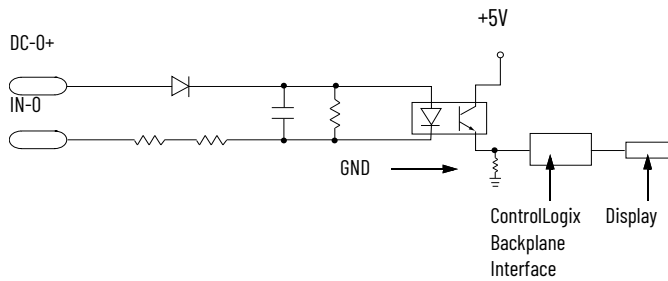
Simplified Schematic



## 1756-IV16, 1756-IV16K

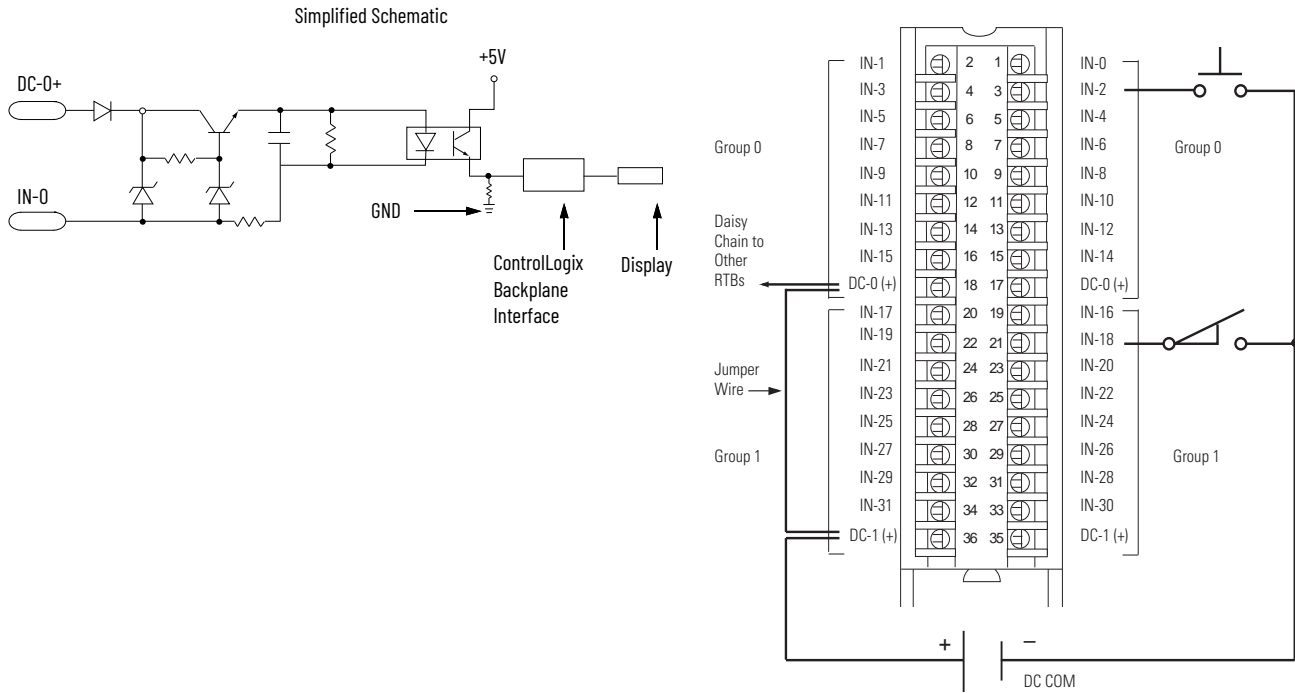
### ControlLogix DC (10...30V) sourcing input module

Simplified Schematic



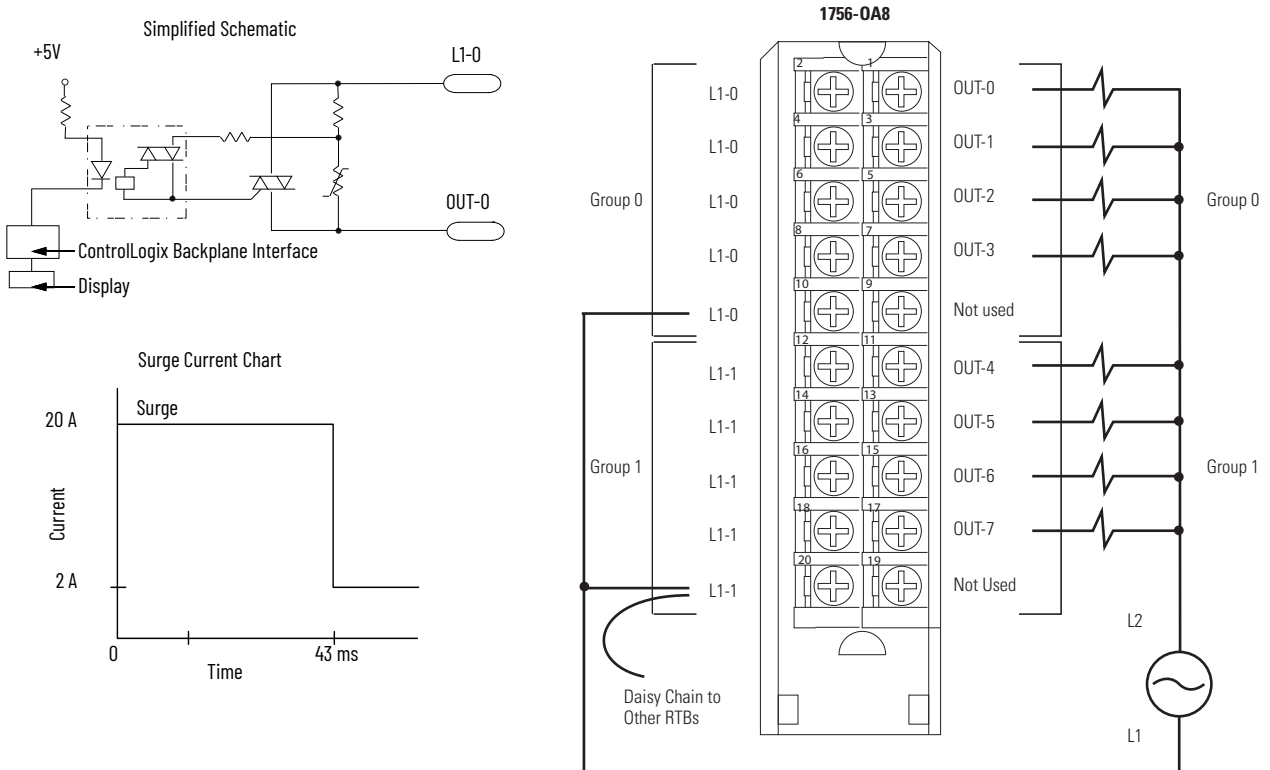
## 1756-IV32, 1756-IV32K

### ControlLogix DC (10...30V) sourcing input module



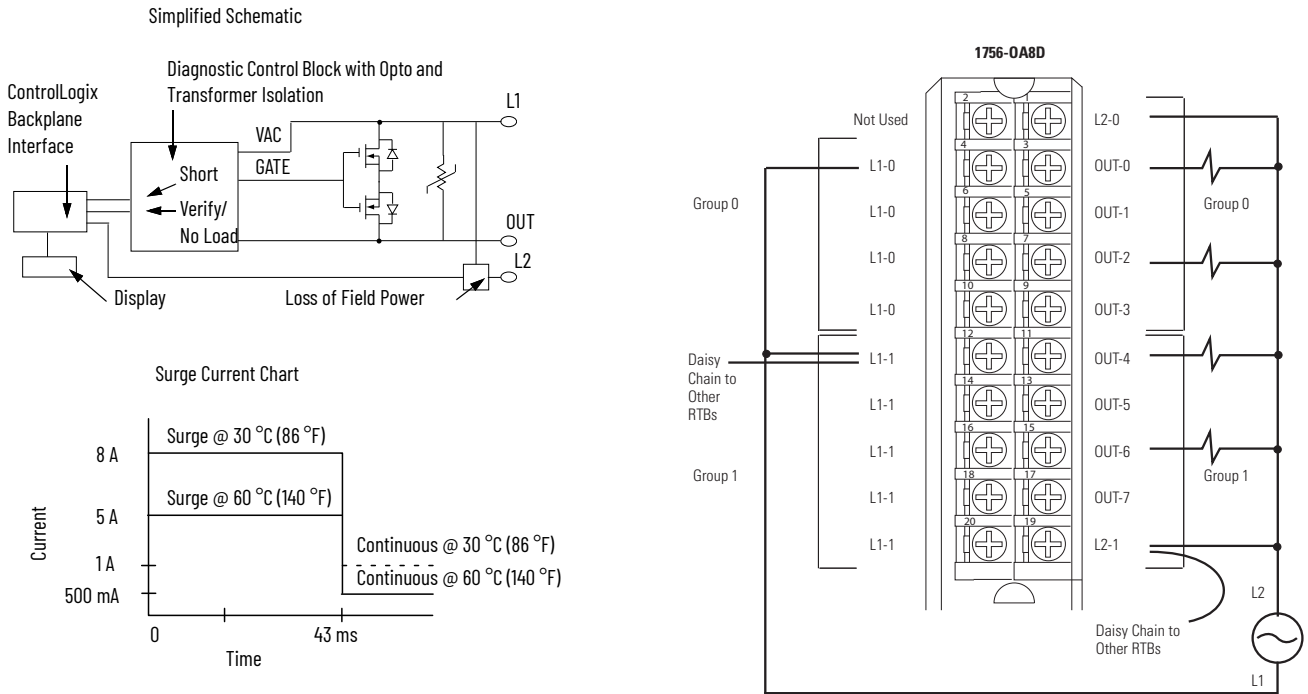
## 1756-0A8

### ControlLogix AC (74...265V) output module



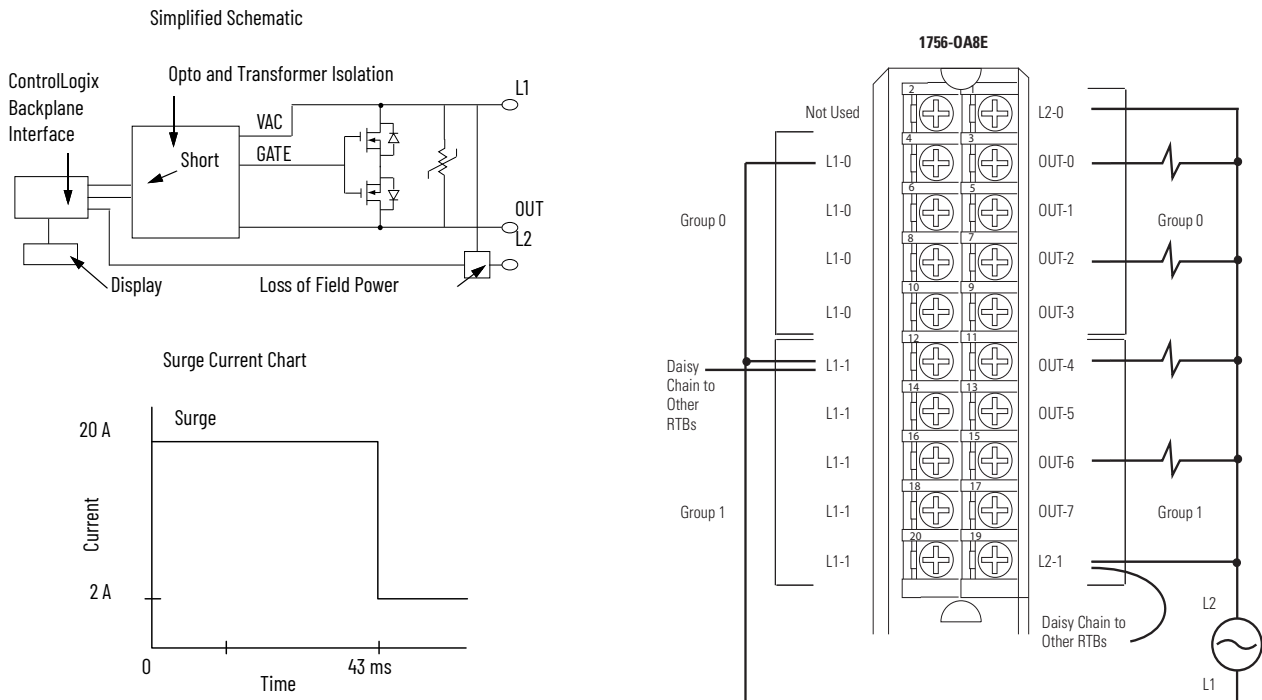
## 1756-0A8D

### ControlLogix AC (74...132V) diagnostic output module



## 1756-0A8E

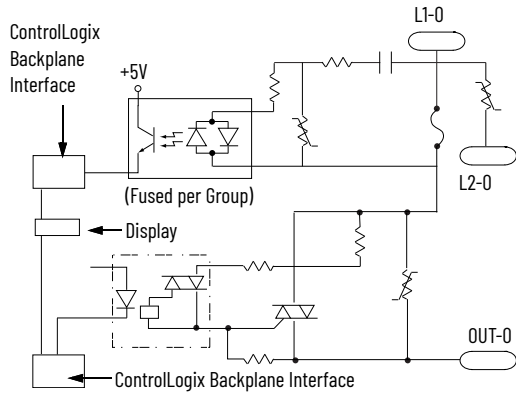
### ControlLogix AC (74...132V) electronically fused output module



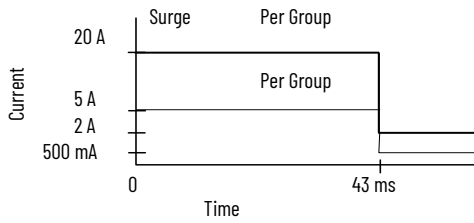
# 1756-0A16, 1756-0A16K

## ControlLogix AC (74...265V) output module

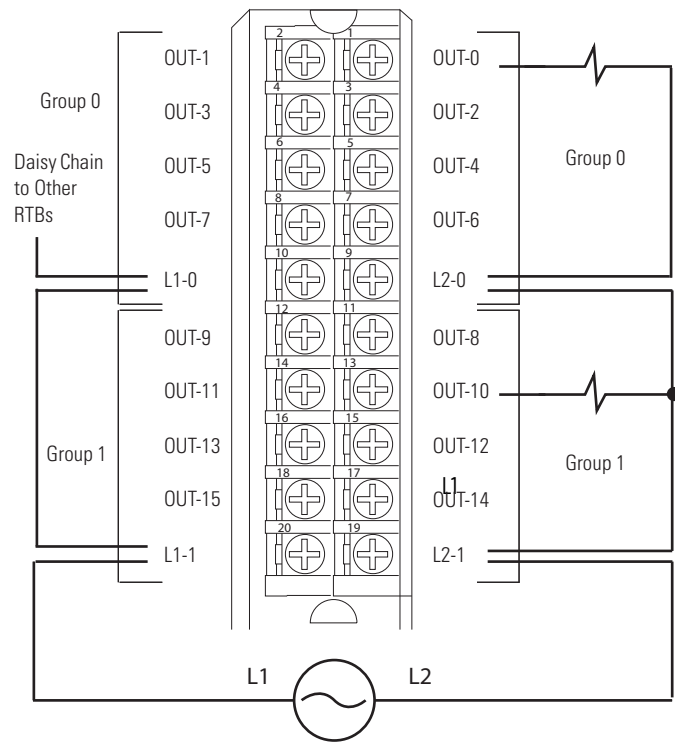
Simplified Schematic



Surge Current Chart



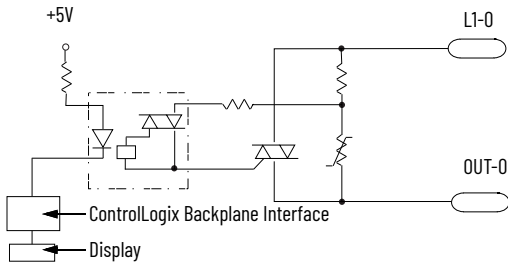
1756-0A16



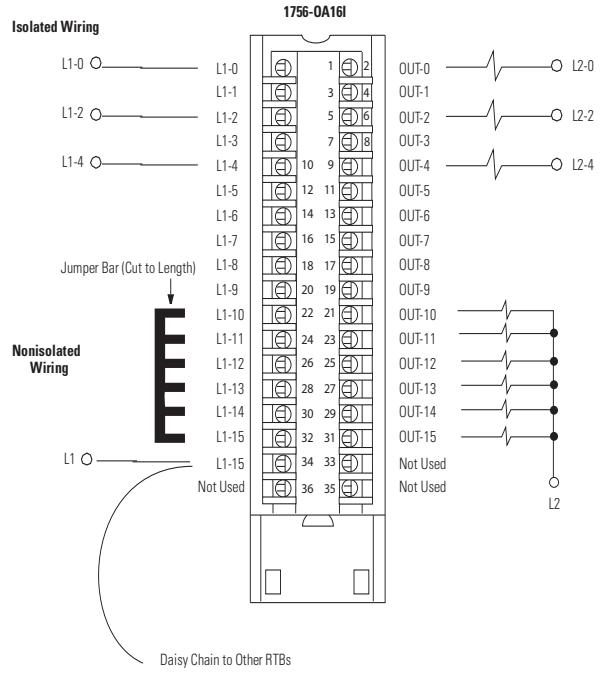
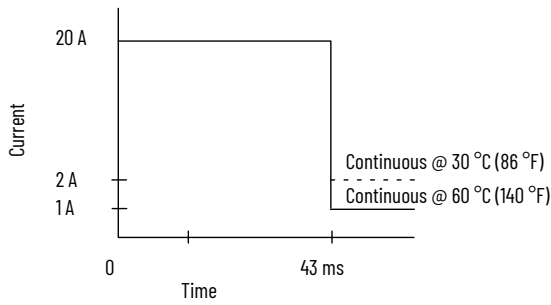
# 1756-0A16I, 1756-0A16IK

## ControlLogix AC (74...265V) isolated output module

Simplified Schematic



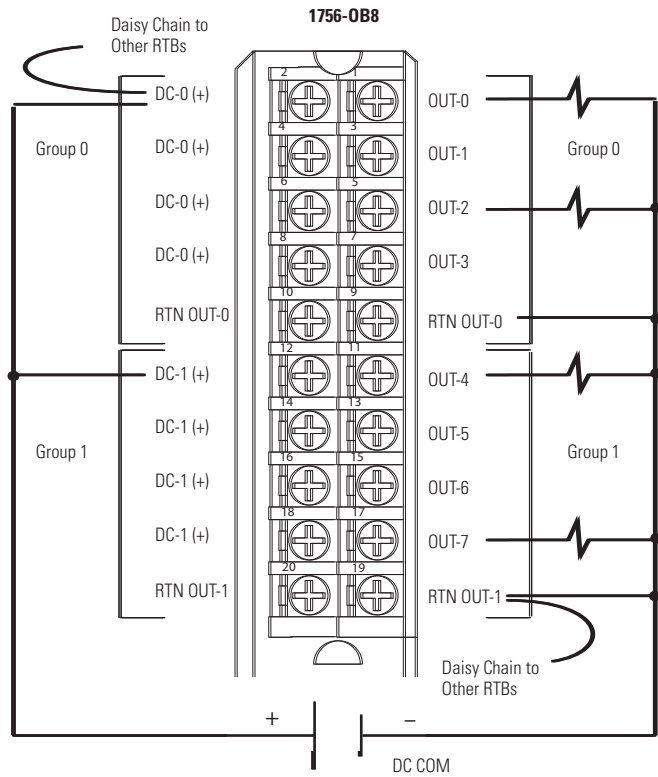
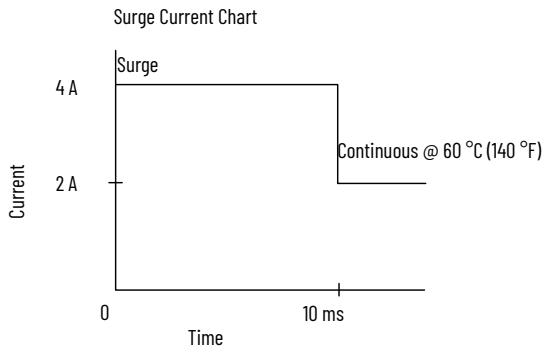
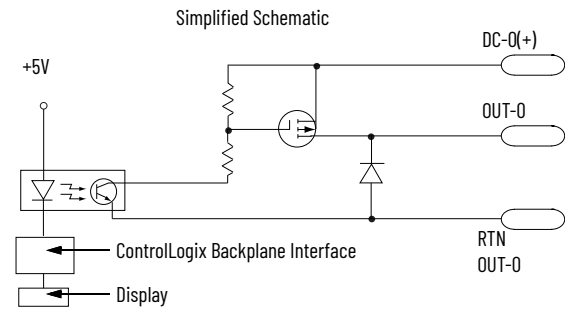
Surge Current Chart



Additional jumper bars can be purchased by using catalog number 1756-JMPR.

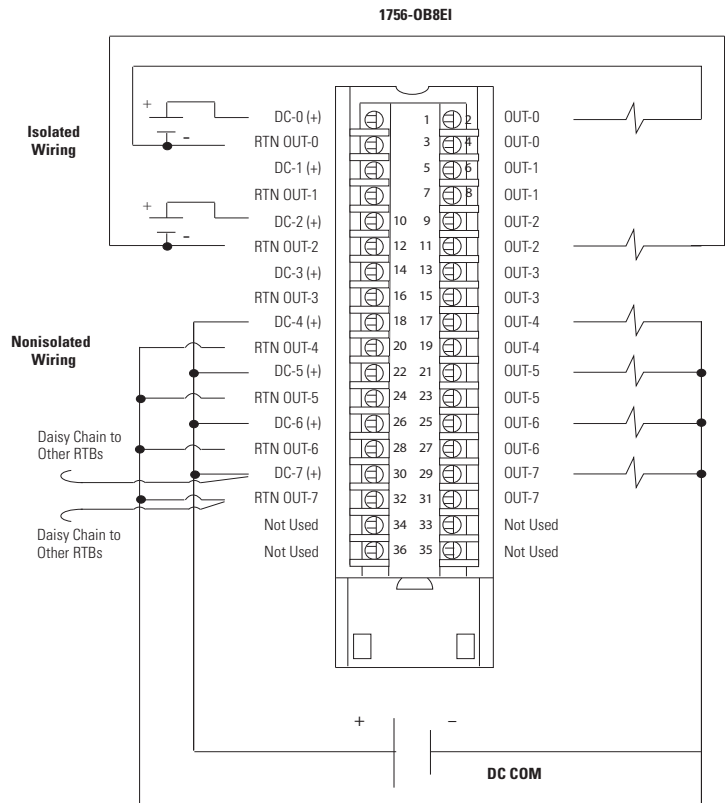
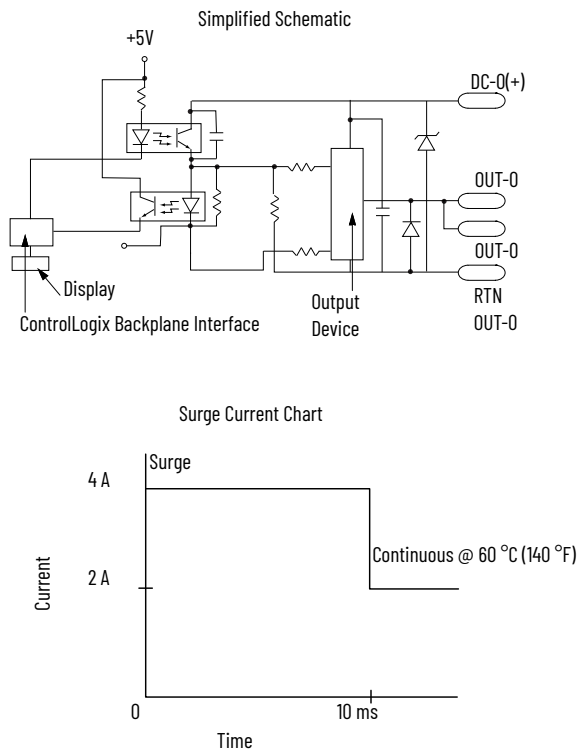
# 1756-OB8, 1756-OB8K

## ControlLogix DC (10...30V) output module



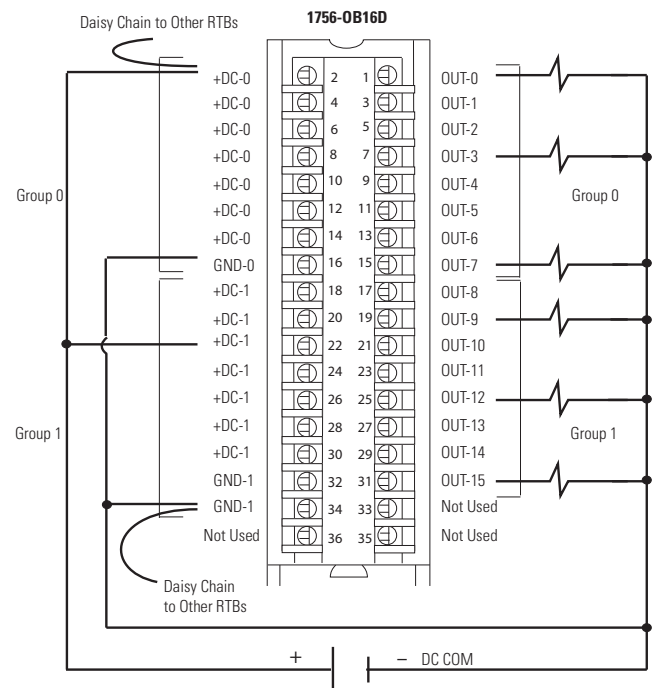
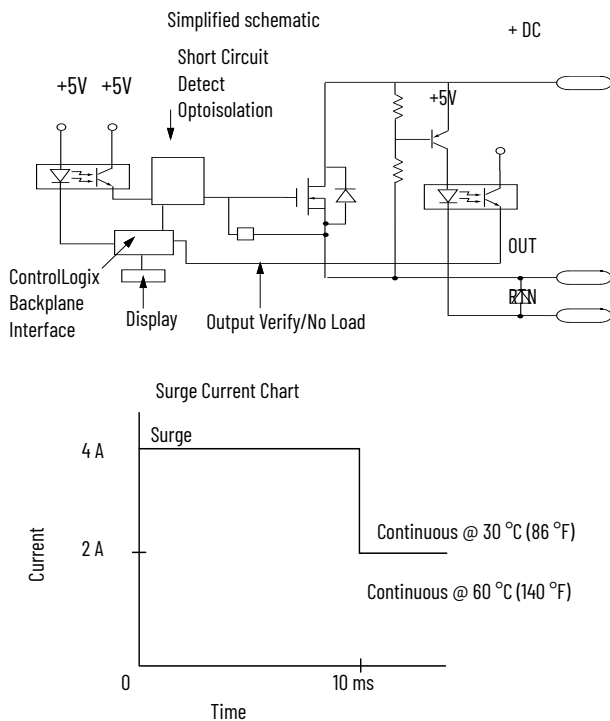
### 1756-OB8EI, 1756-OB8EK

ControlLogix DC (10...30V) electronically fused, isolated output module



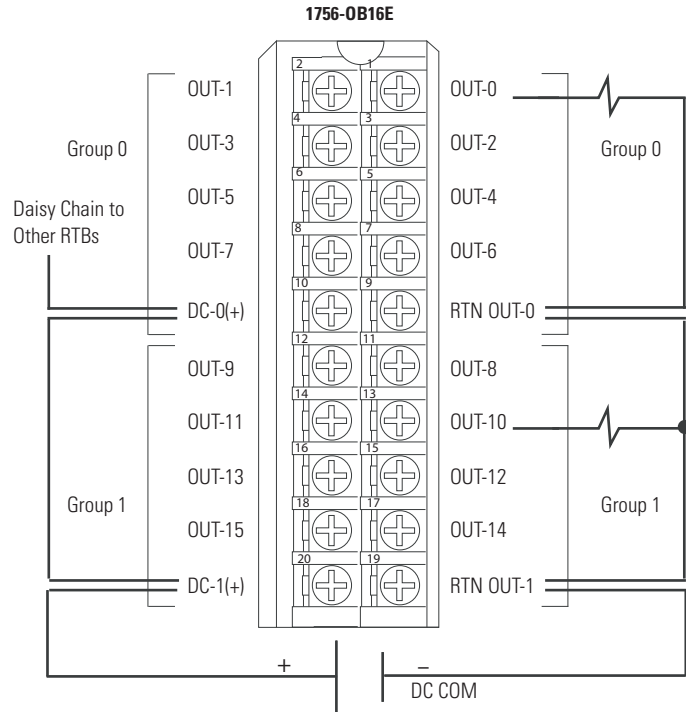
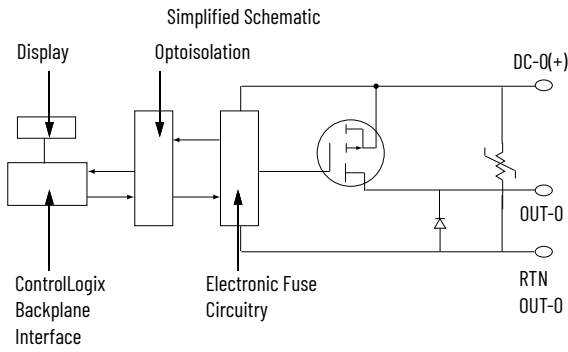
### 1756-OB16D, 1756-OB16DK

ControlLogix DC (19.2...30V) diagnostic output module



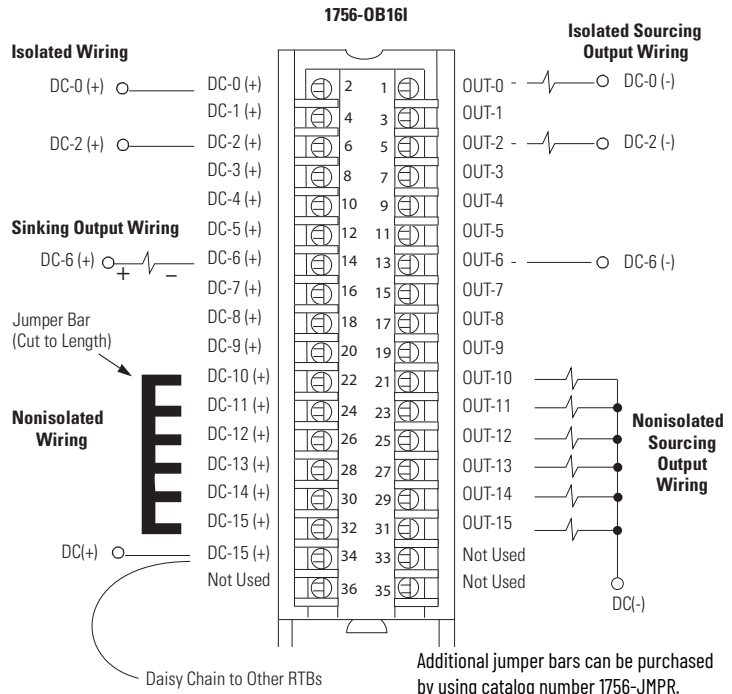
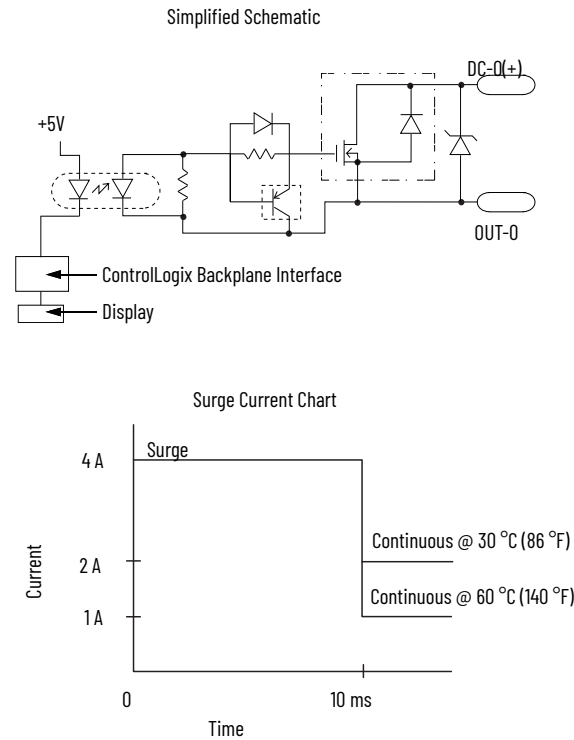
# 1756-OB16E, 1756-OB16EK

## ControlLogix DC (10...31.2V) electronically fused output module



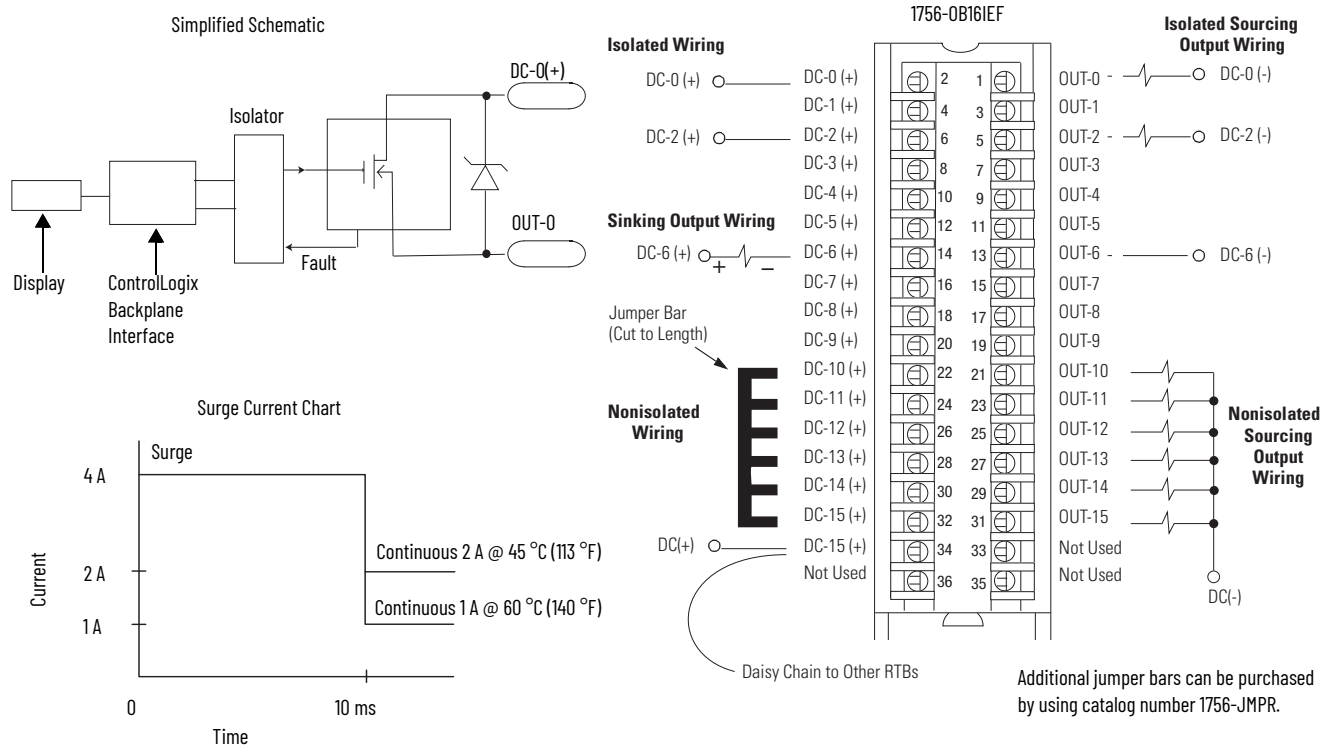
# 1756-OB16I

## ControlLogix DC (10...30V) isolated output module



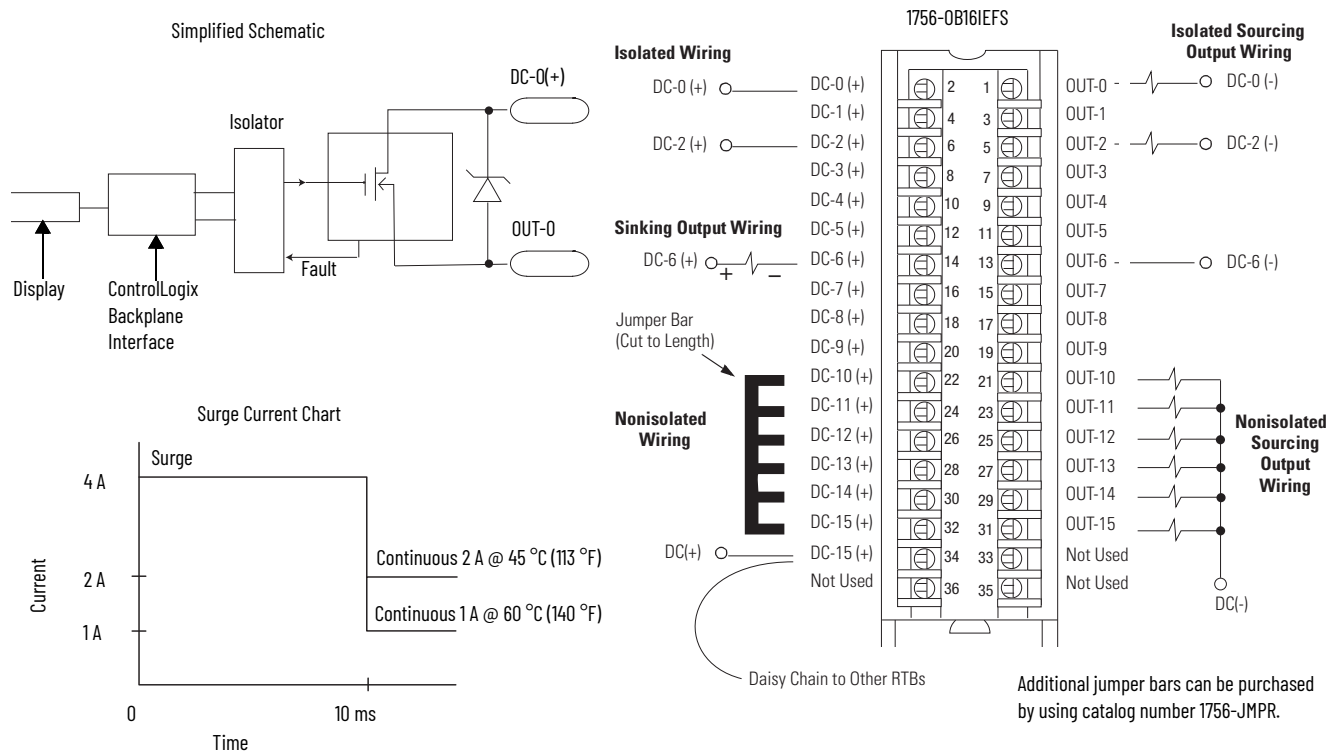
### 1756-OB16IEF, 1756-OB16IEFK

ControlLogix DC (10...30V) electronically protected, sinking or sourcing, isolated, fast output module



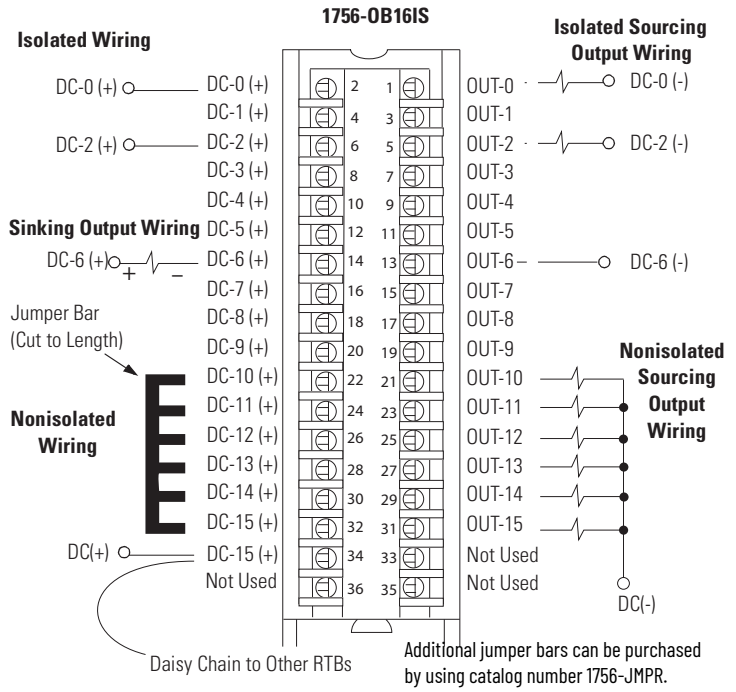
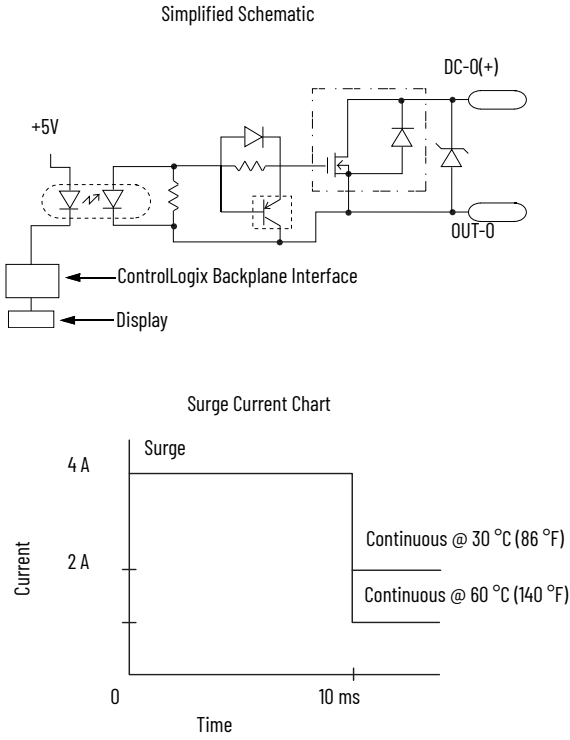
### 1756-OB16IEFS, 1756-OB16IEFSK

ControlLogix DC (10...30V) scheduled, electronically protected, sinking or sourcing, isolated, fast output module



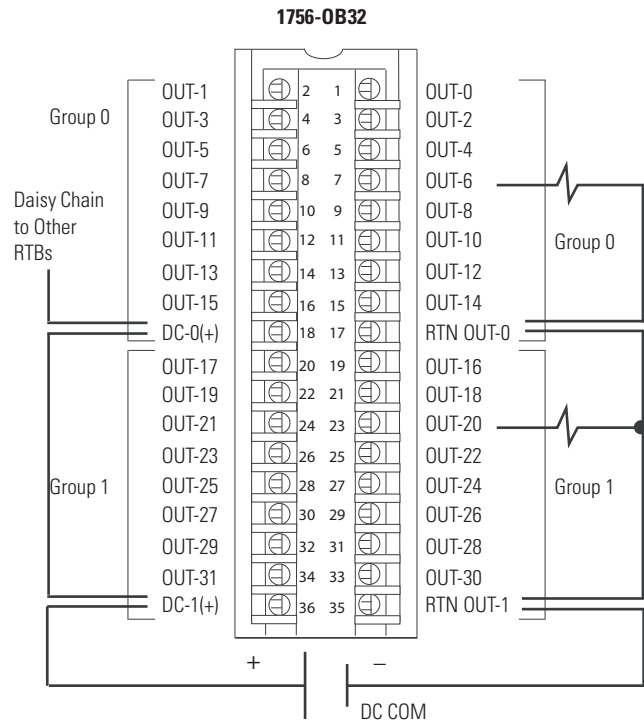
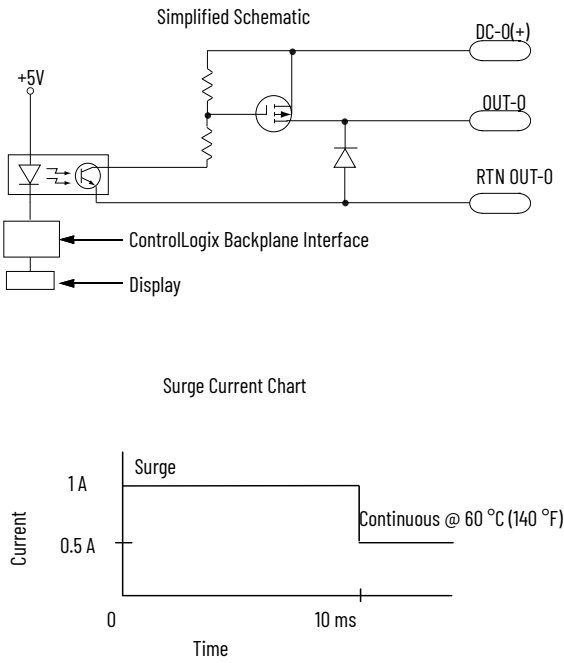
# 1756-OB16IS

## ControlLogix DC (10...30V) scheduled, isolated output module



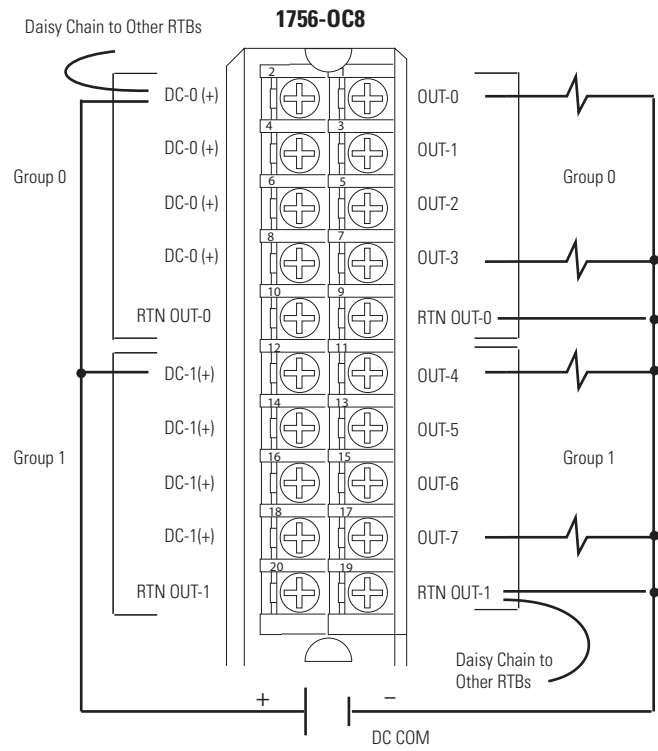
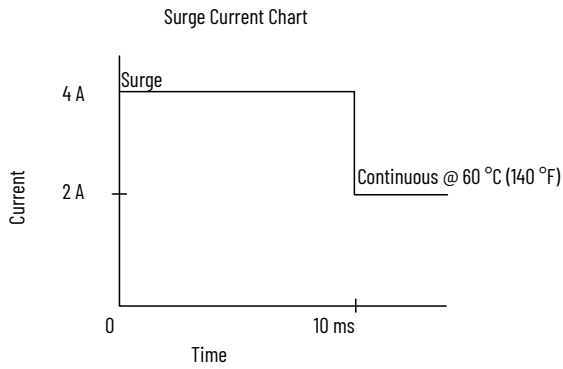
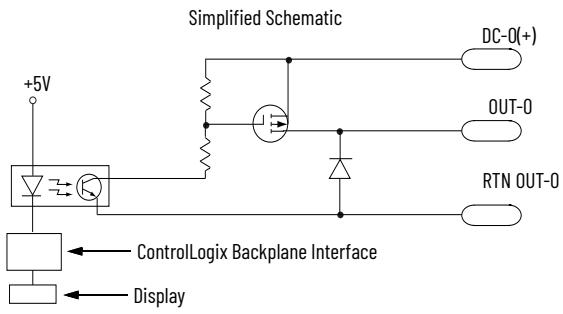
# 1756-OB32, 1756-OB32K, 1756-OB32XT

## ControlLogix DC (10...31.2V) output module



# 1756-OC8, 1756-OC8K

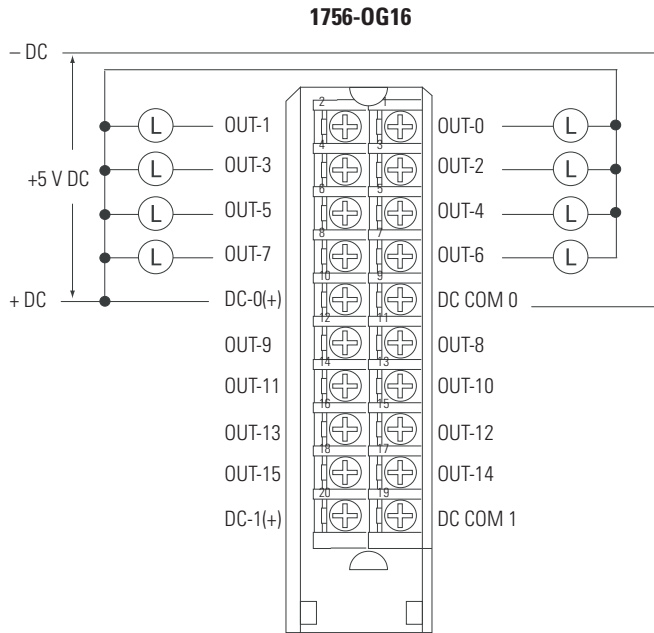
## ControlLogix DC (30...60V) output module



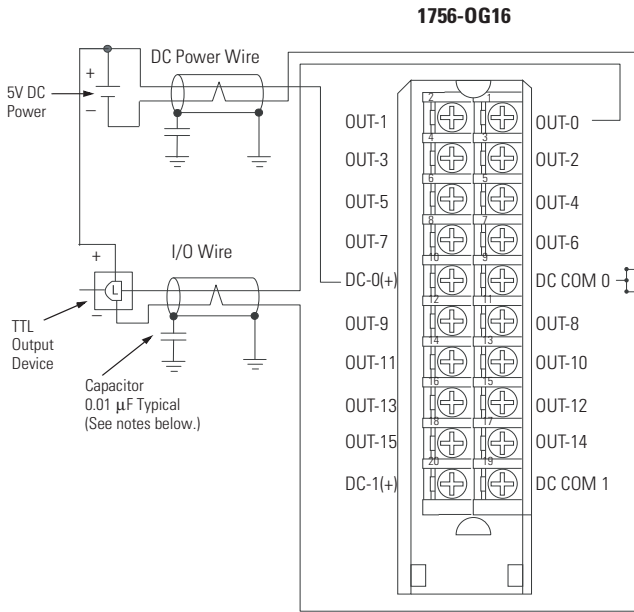
# 1756-0G16, 1756-0G16K

## ControlLogix TTL output module

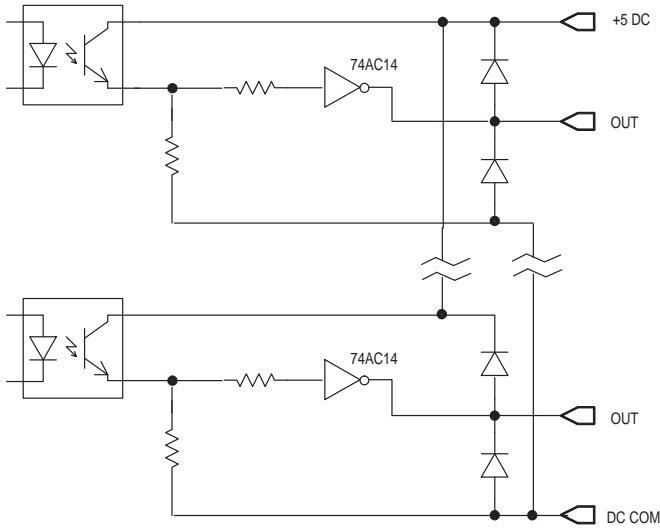
Standard Wiring



CE-Compliant Wiring

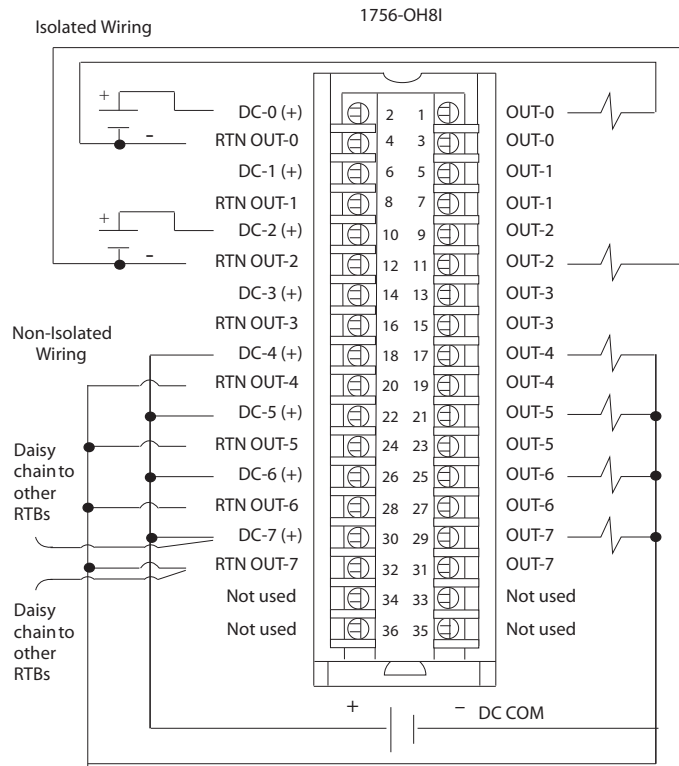
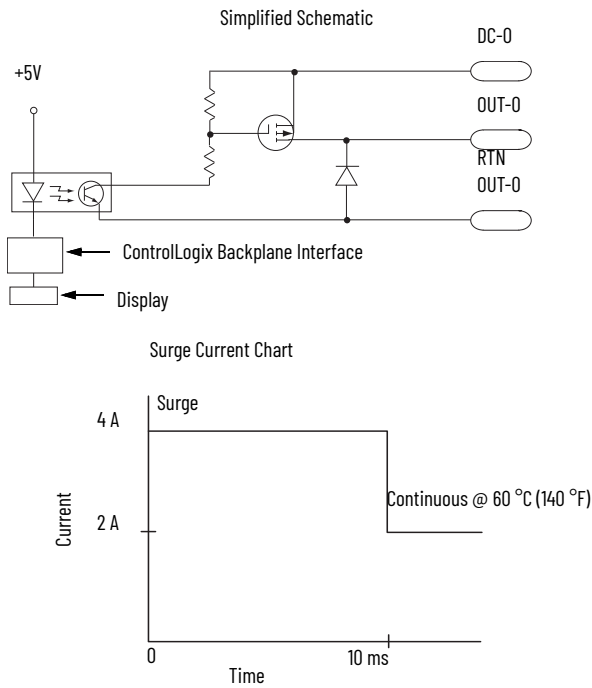


Simplified Schematic



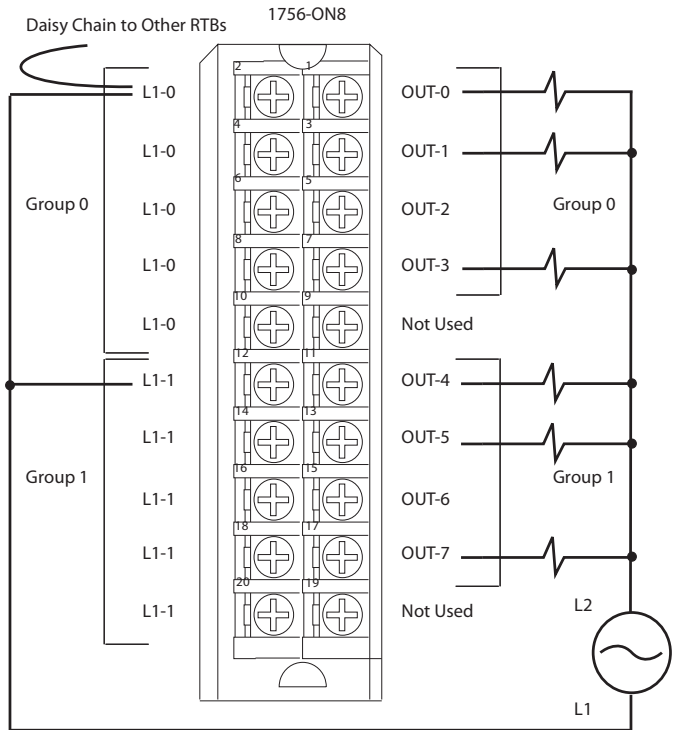
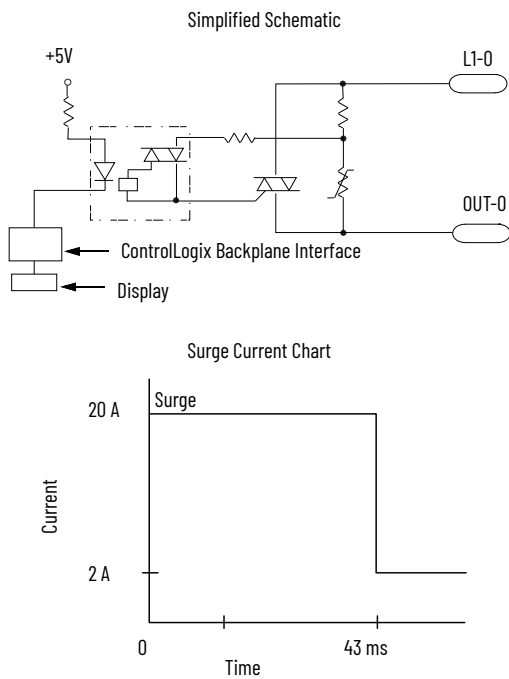
### 1756-OH8I

#### ControlLogix DC (90...146V) isolated output module



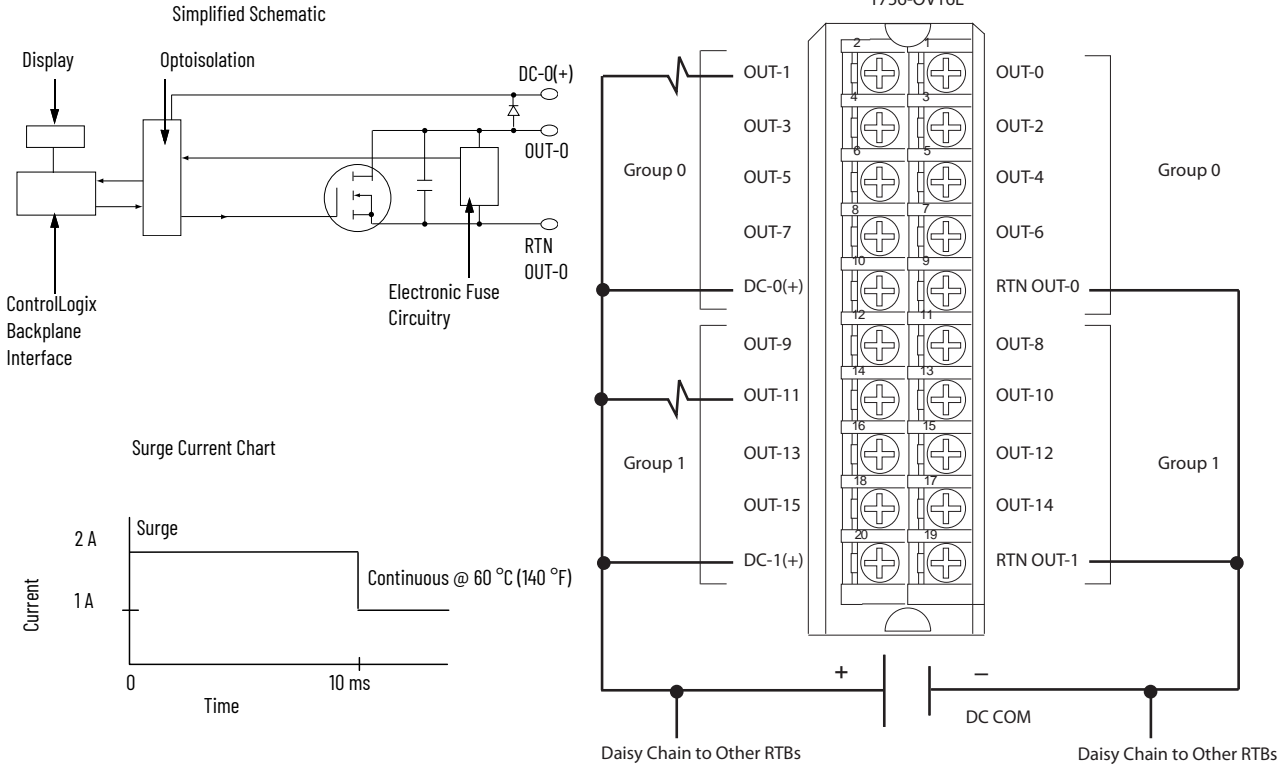
### 1756-ON8, 1756-ON8K

#### ControlLogix AC (10...30V) output module



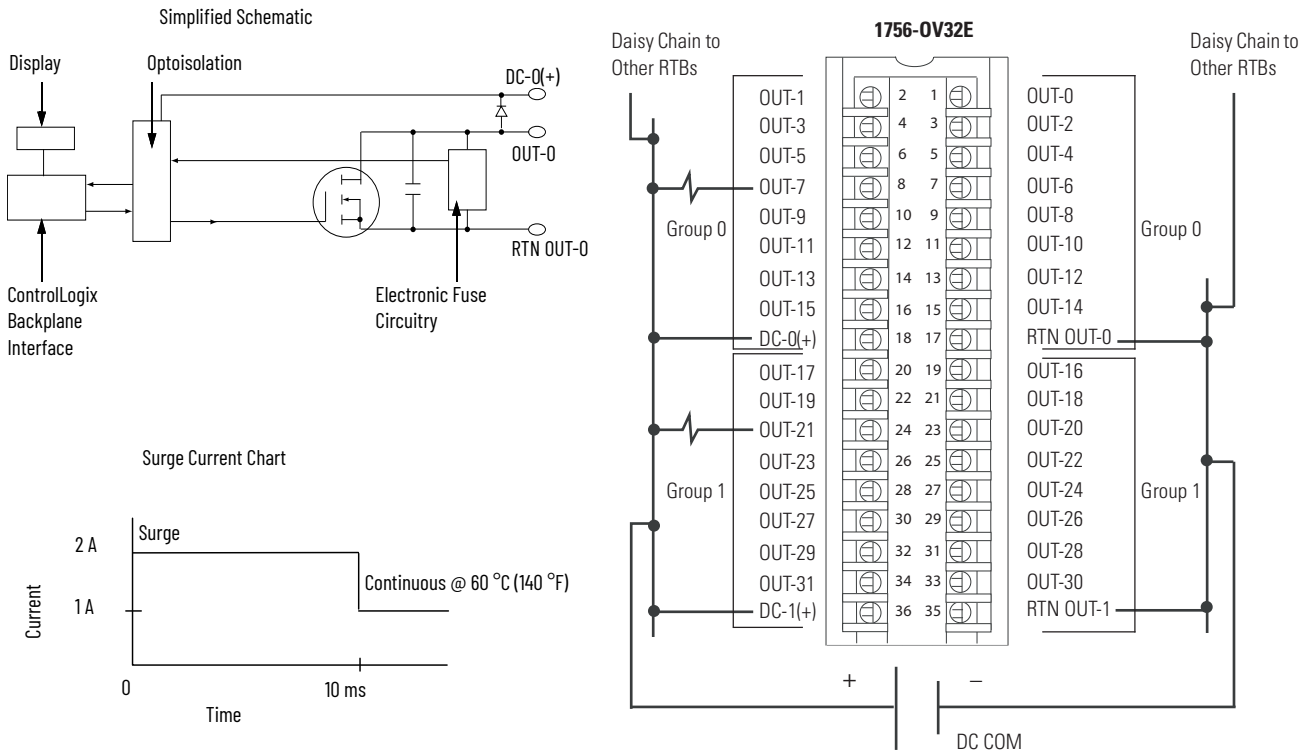
## 1756-OV16E

### ControlLogix DC (10...30V) electronically fused, sinking output module



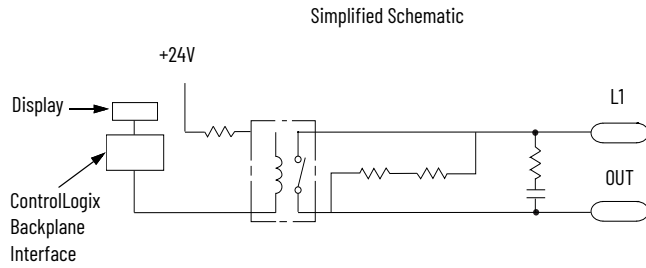
## 1756-OV32E, 1756-OV32EK

### ControlLogix DC (10...30V) electronically fused, sinking output module

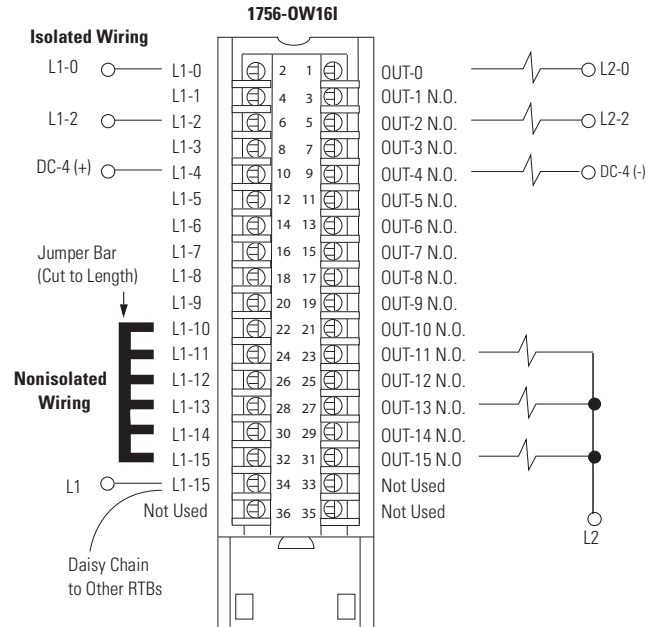


## 1756-0W16I, 1756-0W16IK

### ControlLogix AC (10...240V) DC (5...125V) isolated contact module

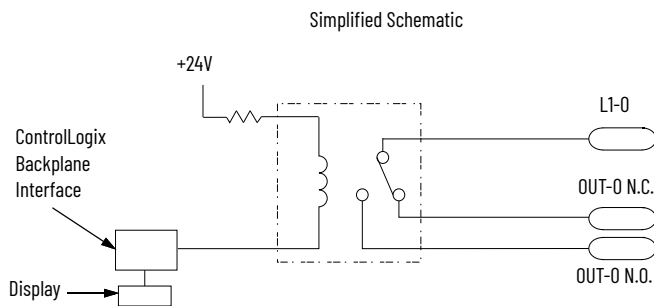


Additional jumper bars can be purchased by using catalog number 1756-JMPR.

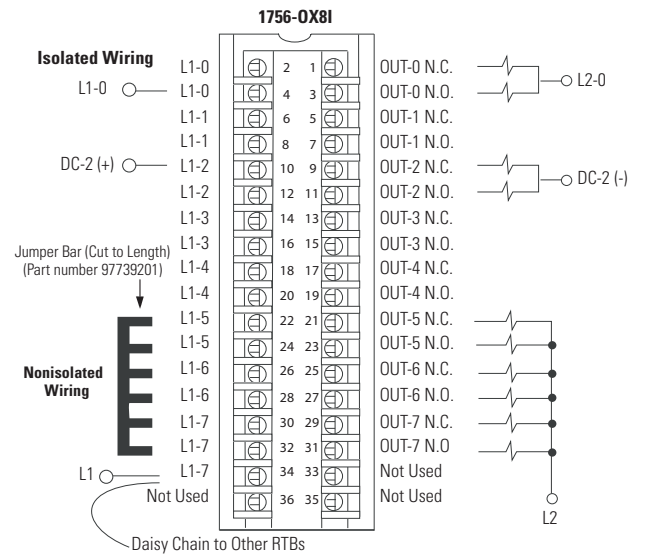


## 1756-0X8I, 1756-0X8IK

### ControlLogix AC (10...240V) DC (5...125V) isolated contact module



Additional jumper bars can be purchased by using catalog number 1756-JMPR.



**Notes:**

## Tag Definitions

This appendix describes the tags that are used for standard, diagnostic, and fast input and output modules.

Module-defined data types and tags are created when a module is initiated. The set of tags that are associated with any module depends on the type of module and the communication or connection format that is chosen during configuration.

### Standard and Diagnostic Input Module Tags

ControlLogix® standard and diagnostic input modules have two types of tags:

- **Configuration**—Structure of data that is sent from the controller to the I/O module upon powerup.
- **Input**—Structure of data continually sent from the I/O module to the controller containing the current operational status of the module.

---

**IMPORTANT** The tables below list all possible standard and diagnostic input module tags. In each application, the series of tags varies, depending on how the module is configured.

---

**Table 43 - Standard Input Module Configuration Tags**

Name	Data Type	Definition
COSOnOffEn (1 bit per point)	DINT	<b>Change of State On to Off</b> —Causes updated data to be sent to the controller immediately after an input for an On to Off transition of the masked input points. The CST timestamp is also updated. May be used to trigger an event task in the controller. 0 = Disable 1 = Enable
COSOffOnEn (1 bit per point)	DINT	<b>Change of State Off to On</b> —Causes updated data to be sent to the controller immediately after an input for an Off to On transition of the masked input points. The CST timestamp is also updated. May be used to trigger an event task in the controller. 0 = Disable 1 = Enable
FilterOnOff_0_7... (1 byte per group)	SINT	<b>Filter Times On to Off</b> —Filter time for digital filter in digital input modules for On to Off transition. Operates on groups of eight points. Valid DC filter times = 0, 1, 2, 9, 18 ms Valid AC filter times = 1, 2 ms
FilterOffOn_0_7... (1 byte per group)	SINT	<b>Filter Times Off to On</b> —Filter time for digital filter in digital input modules for Off to On transition. Operates on groups of eight points. Valid DC filter times = 0, 1, 2 ms Valid AC filter times = 1, 2 ms

**Table 44 - Standard Input Module Data Tags**

Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp can be configured to indicate the time that data changed (see COSOffOnEn, COSOnOffEn, COSStatus, DiagCOSDisable) and/or the time that a diagnostic fault occurred (see OpenWireEn, FieldPwrLossEn).
Data (1 bit per point)	DINT	<b>Off/On</b> —Status for each input point. 0 = Off 1 = On
Fault (1 bit per point)	DINT	<b>Fault Status</b> —An ordered status of faults that indicates a point is faulted and input data for that point may be incorrect. Check other diagnostic faults, if they are available, for further diagnosis of the root cause. If communication to the input module is lost, then all points for the module are faulted. 0 = No fault 1 = Fault (OpenWire or FieldPwrLoss or Comm Fault)

**Table 45 - Diagnostic Input Module Configuration Tags**

Name	Data Type	Definition
COSOnOffEn (1 bit per point)	DINT	<b>Change of State On to Off</b> —Triggers an event in the controller for On to Off transition of input point and causes the input module to update the data table as soon as possible. The CST timestamp is also updated. 0 = Disable 1 = Enable
COS OffOnEn (1 bit per point)	DINT	<b>Change of State Off to On</b> —Triggers an event in the controller for Off to On transition of input point and causes the input module to update the data table as soon as possible. The CST timestamp is also updated. 0 = Disable 1 = Enable
DiagCOSDisable (1 bit per point)	BOOL	<b>Diagnostic Change of State</b> —Triggers the module to transmit diagnostic status data with an updated timestamp as soon as the diagnostic data changes state.
FaultLatchEn (1 bit per point)	DINT	<b>Latch Fault</b> —If enabled for a point, any OpenWire or FieldPwrLoss stays latched in the faulted state even if the fault no longer exists until you clear the fault. 0 = Disable 1 = Enable latching
FieldPwrLossEn (1 bit per point)	DINT	<b>Field Power Loss</b> —Enables Field Power Loss diagnostic. 0 = Disable 1 = Enable
FilterOnOff_0_7... (1 byte per group)	SINT	<b>Filter Time On to Off</b> —Filter time for digital filter in digital input modules for On to Off transition. Operates on groups of eight points. Valid DC filter times = 0, 1, 2, 9, 18 ms. Valid AC filter times = 1, 2 ms.
FilterOffOn_0_7... (1 byte per group)	SINT	<b>Filter Time Off to On</b> —Filter time for digital filter in digital input modules for Off to On transition. Operates on groups of eight points. Valid DC filter times = 0, 1, 2 ms. Valid AC filter times = 1, 2 ms.
OpenWireEn (1 bit per point)	DINT	<b>Open Wire</b> —Enables Open Wire diagnostic. 0 = Disable. 1 = Enable.

**Table 46 - Diagnostic Input Module Data Tags**

Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp can be configured to indicate the time that data changed (see COSOffOnEn, COSOnOffEn, COSStatus, DiagCOSDisable) and/or the time that a diagnostic fault occurred (see OpenWireEn, FieldPwrLossEn).
Data (1 bit per point)	DINT	<b>Input Status</b> —On/Off status for each input point. 0 = Off 1 = On
Fault (1 bit per point)	DINT	<b>Fault Status</b> —An ordered status of faults that indicates a point is faulted and input data for that point may be incorrect. Check other diagnostic faults, if they are available, for further diagnosis of the root cause. If communication to the input module is lost or inhibited, then all points for the module are faulted by the processor. 0 = No fault 1 = Fault (OpenWire or FieldPwrLoss or Comm Fault)
FieldPwrLoss (1 bit per point)	DINT	<b>Field Power Loss</b> —AC input diagnostic detects that field power has failed or is disconnected from the module. Open Wire also is detected. 0 = No fault 1 = Fault
OpenWire (1 bit per point)	DINT	<b>Open Wire</b> —Diagnostic that detects that a wire has been disconnected from the input point. If a group of points shows this fault, then possibly the return (L1 or GND) is missing from the module. Also see FieldPwrLoss. 0 = No fault 1 = Fault

## Standard and Diagnostic Output Module Tags

ControlLogix standard and diagnostic digital output modules have three types of tags:

- **Configuration**—Structure of data that is sent from the controller to the I/O module upon powerup.
- **Input**—Structure of data continually sent from the I/O module to the controller containing the current operational status of the module.
- **Output**—Structure of data that is continually sent from the controller to the I/O module that can modify the module behavior.

**IMPORTANT** The tables below list all possible standard or diagnostic output module tags. In each application, the series of tags varies, depending on how the module is configured.

**Table 47 - Standard Output Module Configuration Tags**

Name	Data Type	Definition
FaultMode (1 bit per point)	DINT	<b>Fault Mode</b> —Used in conjunction with FaultValue to configure the state of outputs when a communication fault occurs. See FaultValue. 0 = Use FaultValue (Off or On) 1 = Hold Last State
FaultValue (1 bit per point)	DINT	<b>Fault Value</b> —Used in conjunction with FaultMode to configure the state of outputs when a communication fault occurs. See FaultMode. 0 = Off 1 = On
ProgMode (1 bit per point)	DINT	<b>Program Mode</b> —Used in conjunction with ProgValue to configure the state of outputs when the controller is in Program mode. See ProgValue. 0 = Use ProgValue (Off or On) 1 = Hold Last State
ProgValue (1 bit per point)	DINT	<b>Program Value</b> —Used in conjunction with ProgMode to configure output state when the controller is in Program mode. See ProgMode. 0 = Off 1 = On
ProgToFaultEn (1 byte per module)	BOOL	<b>Program to Fault Transition</b> —Diagnostic enables the transition of outputs to FaultMode if a communication failure occurs in Program mode. Otherwise outputs remain in Program mode. See ProgMode, ProgValue, FaultMode, FaultValue. 0 = Outputs stay in Program mode if a communication failure occurs. 1 = Outputs go to FaultMode if a communication failure occurs.

**Table 48 - Standard Output Module Input Data Tags**

Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp of diagnostic input data including fusing (see BlownFuse, NoLoad, OutputVerifyFault, FieldPwrLoss), which is updated whenever a diagnostic fault occurs or goes away.
Data (1 bit per point)	DINT	<b>Data</b> —Off/On status for the output point echoed back from the output module. This is used to verify proper communication only. No field side verification is done. For field side verification, see OutputVerifyFault. 0 = Off 1 = On
Fault (1 bit per point)	DINT	<b>Fault</b> —This is an ordered status of faults that indicates a point is faulted and I/O data for that point may be incorrect. Check other diagnostic faults, if they are available, for further diagnosis of the root cause. If communication to the input module is lost, then all points for the module are faulted. 0 = No fault 1 = Fault (FuseBlown, NoLoad, OutputVerifyFault, FieldPwrLoss, or CommFault)
FuseBlown (1 bit per point)	DINT	<b>Fuse Is Blown</b> —An electronic or mechanical fuse has detected a short or overload condition for an output point. All FuseBlown conditions are latched and must be reset by the user. 0 = No fault 1 = Fault

**Table 49 - Standard Output Module Output Data Tags**

Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp to be used with scheduled outputs and coordinated system time (CST). Used to synchronize outputs across the system by indicating the time (CST timestamp) at which the output module is to apply its outputs.
Data (1 bit per point)	DINT	<b>Output Status—On/Off</b> status of the output point originating from the controller. 0 = Off 1 = On

Table 50 - Diagnostic Output Module Configuration Tags

Name	Data Type	Definition
FaultLatchEn (1 bit per point)	DINT	<b>Latch Fault</b> —If enabled for a point, any NoLoad, OutputVerifyFault or FieldPwrLoss stays latched in the faulted state even if the fault no longer exists until the user clears the fault. This does not affect FuseBlown; it is always latched. 0 = Disable 1 = Enable latching
FaultMode (1 bit per point)	DINT	<b>Fault Mode</b> —Used in conjunction with FaultValue to configure the state of outputs when a communication fault occurs. See FaultValue. 0 = Use FaultValue (Off or On) 1 = Hold Last State
FaultValue (1 bit per point)	DINT	<b>Fault Value</b> —Used in conjunction with FaultMode to configure the state of outputs when a communication fault occurs. See FaultMode. 0 = Off 1 = On
FieldPwrLoss (1 bit per point)	DINT	<b>Field Power Loss</b> —Enables Field Power Loss diagnostic. 0 = Disable 1 = Enable
NoLoadEn (1 bit per point)	DINT	<b>No Load</b> —Enables No Load diagnostic. 0 = Disable 1 = Enable
OutputVerifyEn (1 bit per point)	DINT	<b>Output Verify</b> —Enables Output Verify diagnostic. 0 = Disable 1 = Enable
ProgMode (1 bit per point)	DINT	<b>Program Mode</b> —Used in conjunction with ProgValue to configure output state when the controller is in Program mode. See ProgValue. 0 = Use ProgValue (Off or On) 1 = Hold Last State
ProgValue (1 bit per point)	DINT	<b>Program Value</b> —Used in conjunction with ProgMode to configure output state when the controller is in Program mode. See ProgMode. 0 = Off 1 = On
ProgToFaultEn (1 byte per module)	BOOL	<b>Program to Fault Transition</b> —Diagnostic enables the transition of outputs to FaultMode if a communication failure occurs in Program mode. Otherwise outputs remain in Program mode. See ProgMode, ProgValue, FaultMode, FaultValue. 0 = Outputs stay in Program mode if a communication failure occurs. 1 = Outputs go to FaultMode if a communication failure occurs.

Table 51 - Diagnostic Output Module Input Data Tags

Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp of diagnostic input data including fusing (see BlownFuse, NoLoad, OutputVerifyFault, FieldPwrLoss), which is updated whenever a diagnostic fault occurs or goes away.
Data (1 bit per point)	DINT	<b>Output Echo Status</b> —Off/On status for the output point echoed back from the output module. This is used to verify proper communication only. No field side verification is done. For field side verification, see OutputVerifyFault. 0 = Off 1 = On
Fault (1 bit per point)	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. Faulted I/O data for that point may be incorrect. Check other diagnostic faults, if they are available, for further diagnosis of the root cause. If communication to the input module is lost or inhibited, then all points for the module are faulted by the processor. 0 = No fault 1 = Fault (FuseBlown, NoLoad, OutputVerifyFault, FieldPwrLoss, or CommFault)
FieldPwrLoss (1 bit per point)	DINT	<b>Field Power Loss</b> —AC output diagnostic detects that field power has failed or is disconnected from the module. No Load also is detected. 0 = No fault 1 = Fault
FuseBlown (1 bit per point)	DINT	<b>Fuse Is Blown</b> —An electronic or mechanical fuse has detected a short-circuit condition for an output point. All FuseBlown conditions are latched and must be reset by the user. 0 = No fault 1 = Fault
NoLoad (1 bit per group)	DINT	<b>No Load</b> —Indicates the absence of a load (such as, the wire is disconnected from the module). Operates only in the Off state 0 = No fault 1 = Fault
OutputVerifyFault (1 bit per point)	DINT	<b>Output Verify</b> —Diagnostic that indicates that the input has been commanded to the On state but the output has not been verified as On. 0 = No fault 1 = Fault (output is not On)

Table 52 - Diagnostic Output Module Output Data Tags

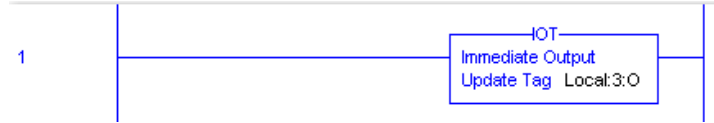
Name	Data Type	Definition
CSTTimestamp (8 bytes)	DINT[2]	<b>Coordinated System Time Timestamp</b> —Timestamp to be used with scheduled outputs and coordinated system time (CST). Used to synchronize outputs across the system by indicating the time (CST timestamp) at which the output module is to apply its outputs.
Data (1 bit per point)	DINT	<b>Output Status</b> —Status for the output point originating from the controller. 0 = Off 1 = On

## Fast Input Module Tags

The ControlLogix 1756-IB16IF fast input module has four types of tags:

- **Configuration**—Structure of data sent from the controller to the module upon powerup.
- **Input**—Structure of data continually sent from the I/O module to the controller or a listening peer module containing the current operational status of the module.
- **Output**—Structure of output data processed by the input module.

**IMPORTANT** Output tag information is sent to the 1756-IB16IF module only at the RPI rate defined during configuration. For optimal performance, use an Immediate Output (IOT) instruction. For example, the rung shown contains an IOT instruction for a fast input module in slot 3. Add a similar rung to your last routine within the Main Task to mimic normal output tag processing.



- **Event**—Structure of event data that is continually sent from the I/O module to the controller or a listening module containing the current operational status of the module.

Fast input modules use array data structures. Array data structures differ from the flat data structures of other digital I/O modules. For more information, see [Redundant Owner Configuration Tags on page 143](#).

**IMPORTANT** The Module Definition column in each table lists the connection type and input data type combinations that are required to create the corresponding tag. For more information, see [Create a New Module on page 62](#).

Table 53 - 1756-IB16IF Module Configuration Tags

Name	Data Type	Tag Definition	Module Definition
LatchTimestamps	BOOL	<p><b>Latch Timestamps</b>—Latches a CIP Sync™ timestamp for a COS transition:</p> <ul style="list-style-type: none"> <li>• When an initial timestamp is latched, timestamps for subsequent COS transitions are dropped.</li> <li>• Once a latched timestamp is acknowledged via the corresponding bit in the Pt[x].NewDataOffOnAck or Pt[x].NewDataOnOffAck tag, the timestamp is overridden upon the next COS transition.</li> </ul> <p>Requires COS to be enabled via the Pt[x].COSOffOnEn or Pt[x].COSOnOff tags            0 = Timestamps are overridden with each successive COS transition.            1 = Timestamps are latched until acknowledged.</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
FilterOffOn	INT	<p><b>Filter Time Off to On</b>—Defines how long an Off to On input transition must remain in the On state before the module considers the transition valid.            Valid filter time = 0...30,000 µs</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
FilterOnOff	INT	<p><b>Filter Time On to Off</b>—Defines how long an On to Off input transition must remain in the Off state before the module considers the transition valid.            Valid filter time = 0...30,000 µs</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
Pt[x].FilterEn	BOOL	<p><b>Filter</b>—If enabled for a point, input transitions must remain in the new state for a configured length of time before the module considers the transition valid.            0 = Filtering is disabled.            1 = Filtering is enabled.</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
Pt[x].COSOffOnEn	BOOL	<p><b>Change of State Off to On</b>—If enabled for a point, an Off to On transition triggers a timestamp recording and sends a COS message on the backplane.            0 = COS data is not produced upon an Off to On transition.            1 = COS data is produced upon an Off to On transition.</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
Pt[x].COSOnOffEn	BOOL	<p><b>Change of State On to Off</b>—If enabled for a point, an On to Off transition triggers a timestamp recording and sends a COS message on the backplane.            0 = COS data is not produced upon an On to Off transition.            1 = COS data is produced upon an On to Off transition.</p>	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data

Table 54 - 1756-IB16IF Module Input Tags

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the input module is lost, then all 32 bits are set. 0 = No fault 1 = Fault	Connection = Data or Listen Only Input Data = Data or Timestamp Data or Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
LocalClockOffset	DINT	<b>Local Clock Offset</b> —Indicates the offset in microseconds between the current CST and the CIP Sync value when a valid CIP Sync time is available.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
OffsetTimestamp	DINT	<b>Timestamp Offset</b> —Indicates when the CIP Sync time was last updated. The timestamp is in CIP Sync time.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
GrandMasterClockID	DINT	<b>Grandmaster Clock ID</b> —Indicates the ID of the CIP Sync Grandmaster to which the module is synced.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].Data	BOOL	<b>Input Status</b> —Indicates whether an input point is On or Off. 0 = The input point is Off. 1 = The input point is On.	Connection = Data or Listen Only Input Data = Data or Timestamp Data or Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Pt[x].Fault	BOOL	<b>Quality of Data after Fault</b> —Indicates whether input data for a faulted point is correct or incorrect. 0 = No fault 1 = Fault	Connection = Data or Listen Only Input Data = Data or Timestamp Data or Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Pt[x].NewDataOffOn	BOOL	<b>New Data Off to On</b> —Captures short duration pulses for Off to On transitions. A captured pulse remains latched until acknowledged via the Pt[x].NewDataOffOnAck output tag. 0 = No new Off to On transitions have occurred since the last acknowledgment. 1 = A new Off to On transition has occurred, but has not yet been acknowledged.	Connection = Data or Listen Only Input Data = Data or Timestamp Data or Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Pt[x].NewDataOnOff	BOOL	<b>New Data On to Off</b> —Captures short duration pulses for On to Off transitions. Pulse remains latched until acknowledged via the Pt[x].NewDataOnOffAck output tag. 0 = No new On to Off transitions have occurred since the last acknowledgment. 1 = A new On to Off transition has occurred, but has not yet been acknowledged.	Connection = Data or Listen Only Input Data = Data or Timestamp Data or Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Pt[x].TimestampDropped	BOOL	<b>Dropped Timestamp</b> —Indicates whether a timestamp was lost as a result of one of the following: <ul style="list-style-type: none"> <li>The corresponding bit was set in the LatchTimestamps configuration tag, so a new timestamp was not recorded because the previous timestamp was latched.</li> <li>The corresponding bit in the LatchTimestamps configuration tag was <b>not</b> set, but a timestamp was replaced by a new timestamp because the previous timestamp was not acknowledged via the Pt[x].NewDataOffOnAck or Pt[x].NewDataOnOffAck output tags.</li> </ul> 0 = A timestamp was not dropped. 1 = A timestamp was dropped.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].CIPSyncValid	BOOL	<b>CIP Sync Is Valid</b> —Indicates whether CIP Sync is available on the backplane. 0 = CIP Sync is not available. 1 = CIP Sync is available.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].CIPSyncTimeout	BOOL	<b>CIP Sync Timeout</b> —Indicates whether a valid time master on the backplane has timed out. 0 = A time master is either not detected on the backplane or is valid. See Pt[x].CIPSyncValid. 1 = A valid time master was detected on the backplane, but the time master has timed out.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].InputOverrideStatus	BOOL	<b>Input Override Status</b> —Indicates whether local inputs are being overridden by the value in the Pt[x].DataOverrideValue output tag because the corresponding bit in the Pt[x].DataOverrideEn output tag is set. 0 = Inputs are not being overridden. 1 = Inputs are being overridden.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].Timestamp.OffOn	DINT	<b>Off to On Timestamp</b> —Records a 64-bit timestamp for the input point's last transition to On. The timestamp is in CIP Sync time.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data
Pt[x].Timestamp.OnOff	DINT	<b>On to Off Timestamp</b> —Records a 64-bit timestamp for the input point's last transition to Off. The timestamp is in CIP Sync time.	Connection = Data, Data with Event, Listen Only, or Listen Only with Event Input Data = Timestamp Data

Table 55 - 1756-IB16IF Module Output Tags

Name	Data Type	Tag Definition	Module Definition
ResetTimestamps	BOOL	<b>Reset Timestamp</b> —When set, clears all timestamps when a rising edge occurs. 0 = Timestamps are not reset. 1 = Timestamps are reset when a rising edge occurs.	Connection = Data or Data with Event Input Data = Timestamp Data
ResetEvents	BOOL	<b>Reset Event</b> —When set, clears all events in the Event[x].NewEvent and Event[x].Timestamp tags when a rising edge occurs. 0 = Events are not cleared. 1 = Events are cleared when a rising edge occurs.	Connection = Data with Event Input Data = Timestamp Data
LatchEvents	BOOL	<b>Latch Event</b> —When set, latches an event until the event is acknowledged. Once acknowledged, the event is overridden by a new event. 0 = Events are overridden by new events. 1 = Events are latched until acknowledged, and new events are ignored.	Connection = Data with Event Input Data = Timestamp Data
Pt[x].NewDataOffOnAck	BOOL	<b>Acknowledge Off to On Transition</b> —A rising edge acknowledges Off to On transitions by clearing the corresponding bits in the Pt[x].Timestamp.OffOn[x] and Pt[x].NewDataOffOn input tags. 0 = Off to On transitions are not acknowledged. 1 = Off to On transitions are acknowledged on the initial transition to 1 of this bit.	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
Pt[x].NewDataOnOffAck	BOOL	<b>Acknowledge On to Off Transition</b> —A rising edge acknowledges On to Off transitions by clearing the corresponding bits in the Pt[x].Timestamp.OnOff[x] and Pt[x].NewDataOnOff input tags. 0 = On to Off transitions are not acknowledged. 1 = On to Off transitions are acknowledged on the initial transition to 1 of this bit.	Connection = Data Input Data = Data or Timestamp Data or Connection = Data with Event Input Data = Timestamp Data
Pt[x].DataOverrideEn	BOOL	<b>Override Data</b> —When set, simulates an input transition when in Run mode by overriding the actual input state with the value defined in the Pt[x].DataOverrideValue output tag. This function is useful for validating timestamping. 0 = The state of an input device is not being overridden. 1 = The state of an input device is being overridden by the value defined in the Pt[x].DataOverride output tag.	Connection = Data or Data with Event Input Data = Timestamp Data
Pt[x].DataOverrideValue	BOOL	<b>Override Data Value</b> —Defines the value to be applied to the input point when the corresponding bit in the Pt[x].DataOverrideEn tag is enabled. 0 = The input state is Off. A timestamp is recorded in the Pt[x].Timestamp.OnOff[x] input tag on a falling edge. 1 = The input state is On. A timestamp is recorded in the Pt[x].Timestamp.OffOn[x] input tag on a rising edge.	Connection = Data or Data with Event Input Data = Timestamp Data
Event[x].Mask	INT	<b>Event Mask</b> —When enabled for a point, an event is triggered when the state of the input matches the value of the corresponding bits in the Event[x].Value tag.	Connection = Data with Event Input Data = Timestamp Data
Event[x].Value	INT	<b>Event Value</b> —Defines whether an input point must be in the On or Off state before an event is triggered. An event is only triggered if the corresponding bits in the Event[x].Mask tag is enabled. 0 = The input must be in the Off state to trigger an event. 1 = The input must be in the On state to trigger an event.	Connection = Data with Event Input Data = Timestamp Data
Event[x].Disarm	BOOL	<b>Disarm Event</b> —Prevents events from being triggered for a point via the pattern defined in the Event[x].Mask and Event[x].Value tags. 0 = Events are triggered. 1 = Events are not triggered.	Connection = Data with Event Input Data = Timestamp Data
Event[x].NewEventAck	BOOL	<b>Acknowledge New Event</b> —When set, acknowledges a new event has occurred as indicated by the Event[x].NewEvent event tag. 0 = A new event has not been acknowledged. 1 = A new event has been acknowledged.	Connection = Data with Event Input Data = Timestamp Data

Table 56 - 1756-IB16IF Module Event Tags

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the input module is lost, then all 32 bits are set. 0 = No fault has occurred. 1 = A fault has occurred.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].NewEvent	BOOL	<b>New Event</b> —Indicates whether a new event has occurred. This bit is cleared only when acknowledged by the Event[x].NewEventAck output tag or reset by the ResetEvents output tag. 0 = No new event has occurred since the last acknowledged event. 1 = A new event has occurred since the last acknowledged event.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].EventDropped	BOOL	<b>Event Dropped</b> —Indicates whether an event has been dropped: <ul style="list-style-type: none"> <li>If the LatchEvents output tag is set, the last recorded event is retained until acknowledged, and a subsequent event is dropped.</li> <li>If the LatchEvents output tag is cleared, the last unacknowledged event is overwritten.</li> </ul> 0 = An event has not been dropped. 1 = An event has been dropped.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].CIPSyncValid	BOOL	<b>CIP Sync Valid</b> —Indicates whether a valid CIP Sync time master existed on the backplane at the time of an event. 0 = CIP Sync was not available on the backplane at the time of an event. 1 = CIP Sync was available on the backplane at the time of an event.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].CIPSyncTimeout	BOOL	<b>CIP Sync Timeout</b> —Indicates that a valid CIP Sync time master existed on the backplane at the time of an event, but has since timed out. 0 = CIP Sync has not timed out. 1 = CIP Sync was available on the backplane, but has since timed out prior to the event occurring.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].Data	INT	<b>Module Data</b> —Shows the input data for all 16 points on the module at the time an event occurs. Data for bits 0...15 is shown as a bit mask where bit 0 is Pt[0].Data and bit 15 is Pt[15].Data. 0 = On a per bit basis, indicates the corresponding bit in the Pt[x].Data input tag was Off when the event occurred. 1 = On a per bit basis, indicates the corresponding bit in the Pt[x].Data input tag was On when the event occurred.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data
Event[x].Timestamp	DINT	<b>Event Timestamp</b> —Records a 64-bit timestamp in CIP Sync format at the time an event occurs.	Connection = Data with Event or Listen Only with Event Input Data = Timestamp Data

## Fast Output Module Tags

ControlLogix fast output modules have three types of tags:

- **Configuration**—Structure of data sent from the controller to the I/O module upon powerup
- **Input**—Structure of data continually sent from the I/O module to the controller containing the current operational status of the module
- **Output**—Structure of data continually sent from the controller to the I/O module that can modify the module behavior

**IMPORTANT** The Module Definition column in each table lists the connection type and input data type combinations that are required to create the corresponding tag. For more information about defining connection and input data types, see [Create a New Module on page 62](#).

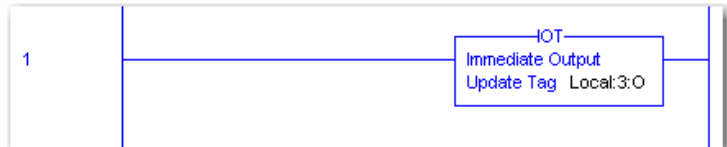
For description of module tags, see:

- [1756-OB16IEF Module](#)
- [1756-OB16IEFS Module](#)

### 1756-OB16IEF Module

**IMPORTANT** Output tag information is sent to the 1756-OB16IEF module only at the RPI rate defined during configuration. For optimal performance, use an Immediate Output (IOT) instruction.

For example, the rung shown contains an IOT instruction for a fast output module in slot 3. Add a similar rung to your last routine within the Main Task to mimic normal output tag processing.



The 1756-OB16IEF module uses array data structures. Array data structures differ from the flat data structures of other digital I/O modules. For more information, see [Redundant Owner Configuration Tags on page 143](#).

Table 57 - 1756-OB16IEF Module Configuration Tags

Name	Data Type	Tag Definition	Module Definition
ProgToFaultEn	BOOL	<b>Program to Fault Mode</b> —Enables the transition of outputs to Fault mode if a communication failure occurs in Program mode. Otherwise, outputs remain in Program mode. See Pt[x].FaultMode, Pt[x].FaultValue Pt[x].ProgMode, and Pt[x].ProgValue. 0 = Outputs stay in Program mode if communication fails. 1 = Outputs go to Fault mode if communication fails.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
InputPartnerSlot	SINT	<b>Peer Partner Slot</b> —Identifies the slot number of the local chassis where the peer input module resides. Valid values: • 0...16 • -1 = No input module has been identified as a peer.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerID	SINT	<b>Peer Partner ID</b> —Identifies the peer input module that controls outputs on the 1756-OB16IEF module. The type of module determines the connection type of format of input data. Valid values: 0 = None (default) 1 = 1756-IB16IF 2 = 1756-LSC8XIB8I	Connection = Peer Ownership Output Data = Data with Peer

Table 57 - 1756-OB16IEF Module Configuration Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
Pt[x].FaultMode	BOOL	<b>Fault Mode</b> —Used in conjunction with the Pt[x].FaultValue tag to determine the state of outputs when a communication failure occurs. 0 = Uses the output value defined in the Pt[x].FaultValue configuration tag (default). 1 = Holds the last state of the output for the length of time defined in the Pt[x].FaultValueStateDuration tag. If PWM is enabled for the output point and the output is currently On, the output continues PWM until the cycle limit is reached or a final fault state goes into effect via the Pt[x].FaultFinalState tag.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].FaultValue	BOOL	<b>Fault Value</b> —Defines the output value when a fault occurs. Holds the configured state of the output for the length of time defined in the Pt[x].FaultValueStateDuration tag. Requires the corresponding bit in the FaultMode tag to be cleared. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].FaultFinalState	BOOL	<b>Fault Final State</b> —Determines the final output state once the time in the Pt[x].FaultValueStateDuration tag elapses. 0 = Output turns Off once the time in the Pt[x].FaultValueStateDuration tag elapses, and module is still faulted. 1 = Output turns On once the time in the Pt[x].FaultValueStateDuration tag elapses, and module is still faulted.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].ProgMode	BOOL	<b>Program Mode</b> —Used in conjunction with the Pt[x].ProgValue tag to determine the state of outputs when the controller is in Program mode. 0 = Uses the output value defined in the Pt[x].ProgValue tag (default). 1 = Holds the last state of the output. If PWM is enabled for the output point and the output is currently On, the output continues to use PWM until the cycle limit is reached.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].ProgValue	BOOL	<b>Program Value</b> —Defines the output state during Program mode. Requires the corresponding bit for the Pt[x].ProgMode tag to be cleared. 0 = The output state is Off during Program mode. 1 = The output state is On during Program mode.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMEnable	BOOL	<b>Enable PWM</b> —When set, the pulse train for the output point is controlled by the current PWM configuration. 0 = PWM is disabled (default). 1 = PWM is enabled, and the output uses PWM when the output is On.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMEExtendCycle	BOOL	<b>Extend PWM Cycle</b> —Determines the output behavior when the value in the Pt[x].PWMOnTime output tag is less than the value in the Pt[x].PWMMinimumOnTime configuration tag. Requires PWM to be enabled via the Pt[x].PWMEnable tag. 0 = The duration of the pulse cycle is not extended (default). If the bit is cleared when the On time is less than the minimum On time, the output is never enabled. 1 = The duration of the pulse cycle is extended to maintain the On time to cycle time ratio while taking into account the minimum On time. <b>IMPORTANT:</b> An extension of the pulse cycle is limited to 10 times the cycle time. If the requested On time is less than 1/10 of the minimum On time, the output remains Off and the cycle does not extend.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMOnTimeInPercent	BOOL	<b>PWM On Time in Percent</b> —Determines whether PWM On time is defined as a percentage of the cycle time or is defined in seconds. Requires PWM to be enabled via the Pt[x].PWMEnable tag. 0 = Defines PWM On time in seconds (default). 1 = Defines PWM On time as a percentage.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMStaggerOutput	BOOL	<b>Stagger PWM Outputs</b> —When set, minimizes the load on the power system by staggering On transitions for outputs. Otherwise, outputs turn On immediately at the start of a cycle. Requires PWM to be enabled via the Pt[x].PWMEnable tag. 0 = Does not stagger output On transitions (default). Outputs turn On immediately when the Pt[x].Data tag is set to 1 beginning the PWM cycle with a rising edge. 1 = Staggers output On transitions. All outputs configured for PWM staggering turn On at different intervals to minimize a possible power surge if many outputs became energized simultaneously.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer

Table 57 - 1756-OB16IEF Module Configuration Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
Pt[x].PWMCycleLimitEnable	BOOL	<b>Enable PWM Cycle Limit</b> —Determines whether to let only a fixed number of pulse cycles occur. Requires PWM to be enabled via the Pt[x].PWMEnable tag. 0 = Pulse cycles continue to occur until the output turns Off (default). 1 = Lets only the number of pulse cycles defined via the Pt[x].PWMCycleLimit tag to occur.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMEecuteAllCycles	BOOL	<b>Execute All PWM Cycles</b> —Determines whether to execute the number of cycles defined via the Pt[x].PWMCycleLimit tag regardless of the output logic. Requires PWM to be enabled via the Pt[x].PWMEnable tag, and a cycle limit to be enabled via the Pt[x].PWMCycleLimitEnable tag. 0 = The output logic determines the number of cycles to produce (default). 1 = The Pt[x].PWMCycleLimit tag determines the number of cycles to produce regardless of output logic. For example, if you specify a cycle limit of 4, and the output turns Off after 3 cycles, all 4 cycles still occur despite the output being instructed to turn Off.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].FaultValueStateDuration	SINT	<b>Fault State Duration</b> —Defines the length of time that the output state remains in the Fault mode state before transitioning to a final state of On or Off. The Fault mode state is defined in the Pt[x].FaultValue tag. Valid values: • 0 = Hold forever (default). Output remains in Fault mode for as long as the fault condition persists. • 1, 2, 5, or 10 seconds	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMCycleLimit	SINT	<b>PWM Cycle Limit</b> —Defines the number of pulse cycles to occur when the output turns On: • If the corresponding bit in the Pt[x].PWMEecuteAllCycles tag is set, the configured number of cycles occur even if the output turns Off. • If the corresponding bit in the Pt[x].PWMEecuteAllCycles tag is cleared, the configured number of cycles occur only if the output remains On. For example, if the cycle limit is 4, and the output turns Off after 3 cycles, the 4th cycle does <b>not</b> occur. The default cycle limit is 10. Requires PWM to be enabled via the Pt[x].PWMEnable tag, and cycle limits to be enabled via the Pt[x].PWMCycleLimitEnable tag.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMinimumOnTime	REAL	<b>PWM Minimum On Time</b> —Defines the minimum length of time required for the output to turn On. Requires PWM to be enabled via the Pt[x].PWMEnable tag. Valid values: 0.0002...3600.0 seconds or 0...100 percent	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].AndToControllerData	INT	<b>Controller Data with AND Logic</b> —Determines the output state by applying AND logic to these sources: • Corresponding bits from the controller's output data (O:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].OrToControllerData	INT	<b>Controller Data with OR Logic</b> —Determines the output state by applying OR logic to these sources: • Corresponding bits from the controller's output data (O:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].AndToPeerInput	INT	<b>Peer Data with AND Logic</b> —Determines the output state by applying AND logic to these sources: • Corresponding bits from peer input data (I:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].OrToPeerInput	INT	<b>Peer Data with OR Logic</b> —Determines the output state by applying OR logic to these sources: • Corresponding bits from peer input data (I:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].AndToPeerWindow 0	SINT	<b>Peer Data with AND Logic</b> —Determines the output state by applying AND logic to these sources: • Corresponding bits from window 0 of the peer counter module (I:Counter[x].InputWindow0) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer

Table 57 - 1756-OB16IEF Module Configuration Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
OutputMap[x].OrToPeerWindow0	SINT	<b>Peer Data with OR Logic</b> —Determines the output state by applying OR logic to these sources: <ul style="list-style-type: none"> <li>Corresponding bits from window 0 of the peer counter module (I:Counter[x].InputWindow0)</li> <li>Other mapped bits specified in the output configuration</li> </ul>	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].AndToPeerWindow1	SINT	<b>Peer Data with AND Logic</b> —Determines the output state by applying AND logic to these sources: <ul style="list-style-type: none"> <li>Corresponding bits from window 1 of the peer counter module (I:Counter[x].InputWindow1)</li> <li>Other mapped bits specified in the output configuration</li> </ul>	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[x].OrToPeerWindow1	SINT	<b>Peer Data with OR Logic</b> —Determines the output state by applying OR logic to these sources: <ul style="list-style-type: none"> <li>Corresponding bits from window 1 of the peer counter module (I:Counter[x].InputWindow1)</li> <li>Other mapped bits specified in the output configuration</li> </ul>	Connection = Peer Ownership Output Data = Data with Peer

Table 58 - 1756-OB16IEF Module Input Data Tags

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the output module is lost, then all 32 bits of the Module Fault word are set. 0 = No fault 1 = Fault	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
GrandMasterClockID	DINT	<b>Grandmaster Clock ID</b> —Indicates the ID of the CIP Sync Grandmaster to which the module is synced.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
InputPartnerActive	BOOL	<b>Input Partner is Active</b> —Indicates whether the peer input module is actively producing input data to be consumed by a 1756-OB16IEF module. 0 = No input peer module is currently producing input data to be consumed by a 1756-OB16IEF module. 1 = The input peer module is actively producing input data to be consumed by a 1756-OB16IEF module for use in its peer logic.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerFault	BOOL	<b>Input Partner Fault</b> —Indicates whether the peer input module has faulted due to a connection loss. If the peer input module is faulted, the output module uses only controller data to determine the output state. 0 = The input peer module has not faulted. 1 = The input peer module has faulted and outputs transition to the configured Fault mode state.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerSlot	SINT	<b>Input Partner Slot</b> —Indicates the slot number of the peer input module. Valid values: <ul style="list-style-type: none"> <li>0...16</li> <li>-1 = No peer input module is defined.</li> </ul>	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerStatus	SINT	<b>Input Partner Status</b> —Indicates the status of the peer input module. Valid values: <ul style="list-style-type: none"> <li>2 = Communication Fault (Peer connection is lost)</li> <li>6 = Run (Peer connection open and in Run mode)</li> </ul>	Connection = Peer Ownership Output Data = Data with Peer
LocalClockOffset	DINT	<b>Local Clock Timestamp</b> —Indicates the offset between the current CST and the CIP Sync value when a valid CIP Sync time is available.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer

Table 58 - 1756-OB16IEF Module Input Data Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
OffsetTimestamp	DINT	<b>Timestamp Offset</b> —Indicates when the CIP Sync LocalClockOffset and GrandMasterID were last updated in CIP Sync format.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
OwnerActive	BOOL	<b>Owner Active</b> —Indicates that the output has a controlling owner.	B4
OwnerID	BOOL	<b>Owner ID</b> —Indicates which owner is active (controlling): 0 = Owner A 1 = Owner B	B5
OwnerAConnected	BOOL	<b>Owner A Connected</b> —Indicates that Owner A is connected.	B6
OwnerAClaim	BOOL	<b>Owner A Claim</b> —Indicates that Owner A is claiming outputs (COO).	B7
OwnerAReady	BOOL	<b>Owner A Ready</b> —Indicates that Owner A is Ready to claim outputs (ROO).	B8
OwnerBConnected	BOOL	<b>Owner B Connected</b> —Indicates that Owner B is connected.	B9
OwnerBClaim	BOOL	<b>Owner B Claim</b> —Indicates that Owner B is claiming outputs (COO).	B10
OwnerBReady	BOOL	<b>Owner B Ready</b> —Indicates that Owner B is Ready to claim outputs (ROO).	B11
Pt[x].Data	BOOL	<b>Data</b> —Indicates the current value to be sent to the corresponding output point. If PWM is enabled, this value transitions from 0 to 1 based on the PWM pulse train. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].Fault	BOOL	<b>Fault</b> —Indicates whether I/O data for the corresponding point may be incorrect due to a fault. 0 = No fault. 1 = A fault exists and I/O data may be incorrect. Any of these conditions set the bit for this tag: <ul style="list-style-type: none"> <li>Pt[x].FuseBlown = 1</li> <li>Pt[x].PWMCycleTime outside the valid range of 0.001...3600.0 seconds</li> <li>Pt[x].PWMOnTime outside the valid range of 0.0002...3600.0 seconds or 0...100 percent</li> <li>Pt[x].PWMCycleTime ≤ Pt[x].PWMOnTime</li> </ul>	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].FuseBlown	BOOL	<b>Fuse Is Blown</b> —Indicates whether a fuse has blown due to a short or overload condition for the corresponding point. All blown fuse conditions are latched and must be reset. 0 = Fuse is not blown. 1 = Fuse is blown and has not been reset.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMCycleLimitDone	BOOL	<b>PWM Cycle Limit Done</b> —Indicates whether the PWM pulse cycle limit defined in the Pt[x].PWMCycleLimit configuration tag has been reached. 0 = The PWM cycle limit has not yet been reached. The bit resets to 0 each time the output transitions to On to begin a new PWM cycle. 1 = The PWM cycle limit has been reached.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].CIPSyncValid	BOOL	<b>CIP Sync Is Valid</b> —Indicates whether the module has synchronized to a valid CIP Sync time master on the backplane. 0 = CIP Sync is not available. 1 = CIP Sync is available.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer

Table 58 - 1756-OB16IEF Module Input Data Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
Pt[x].CIPSyncTimeout	BOOL	<b>CIP Sync Timeout</b> —Indicates whether a valid time master on the backplane has timed out. 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. The module is currently using its local clock.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OutputOverrideStatus	BOOL	<b>Output Override Status</b> —Indicates whether local output data or logic point is set up to be overridden by the value in the Pt[x].OverrideOutputValue output tag. Requires the Pt[x].OverrideOutputEn output tag to be enabled. 0 = The override feature for the corresponding output is not enabled. 1 = The override feature for the corresponding output is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PeerInputOverrideStatus	BOOL	<b>Peer Input Override Status</b> —Indicates whether peer input data mapped to the corresponding output point is set up to be overridden by the value in the Pt[x].OverridePeerInputValue output tag. Requires the O:Pt[x].OverridePeerInputEn output tag to be enabled. 0 = The override feature for peer inputs is not enabled. 1 = The override feature for peer inputs is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PeerWindows0OverrideStatus	BOOL	<b>Peer Window 0 Override Status</b> —Indicates whether peer window 0 data mapped to the corresponding output point is set up to be overridden by the value in the Pt[x].OverridePeerWindow0Value output tag. Requires the O:Pt[x].OverridePeerWindow0En output tag to be enabled. 0 = The override feature for peer window 0 is not enabled. 1 = The override feature for peer window 0 is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PeerWindow1OverrideStatus	BOOL	<b>Peer Window 1 Override Status</b> —Indicates whether peer window 1 data mapped to the corresponding output point is set up to be overridden by the value in the Pt[x].OverridePeerWindow1Value output tag. Requires the O:Pt[x].OverridePeerWindow1En output tag to be enabled. 0 = The override feature for peer window 1 is not enabled. 1 = The override feature for peer window 1 is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Timestamp	DINT	<b>Timestamp</b> —A 64-bit CIP Sync timestamp of the last new output data or FuseBlown event.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer

Table 59 - 1756-OB16IEF Module Output Data Tags

Name	Data Type	Tag Definition	Module Definition
OwnerClaim	BOOL	B0 0 = Not Requesting Ownership 1 = Claim Ownership	When set, indicates that the controller wants to Claim Ownership of the Redundant connection and have its Outputs actively used. If both Redundant Owners have COO set, the last originator application that transitioned its COO flag from reset to set is the owner. Only an Rx'd 0 to a 1 and not just the receipt of a new connection is considered such a transition. If neither Redundant Owner has COO set, then the highest ROO becomes the owner.
OwnerReady	BOOL	B1 0 = Not Ready to Own 1 = Ready to Own	Indicates that the controller is Ready for Output Ownership. A value of 1 indicates ready while 0 indicates that the controller is not Ready to Own the Outputs. If neither controller has the COO bit set, the ROO determines the current Owner. If both controllers have COO=0 and ROO=0, then the Outputs go to IDLE (Program mode). If both controllers have COO=0 and ROO=1, OwnerA is Active.
Pt[x].Data	BOOL	<b>Data</b> —Indicates the On/Off state to apply to the output point. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].ResetFuseBlown	BOOL	<b>Reset Blown Fuse</b> —Attempts to clear a blown fuse status and apply output data when the bit transitions from Off to On.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverrideOutputEn	BOOL	<b>Override Output</b> —Overrides local output data for peer logic with the value defined in the Pt[x].OverrideOutputValue tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverrideOutputValue	BOOL	<b>Override Output Value</b> —Indicates the On/Off status to apply to all outputs mapped to the output point when the corresponding bit in the Pt[x].OverrideOutputEn tag is set. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverridePeerInputEn	BOOL	<b>Override Peer Input</b> —Overrides peer input data mapped to the output point with the value defined in the Pt[x].OverridePeerInputValue output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverridePeerInputValue	BOOL	<b>Override Peer Input Value</b> —Indicates the On/Off status to apply to all peer inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerInputEn output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverridePeerWindow0En	BOOL	<b>Override Peer Window 0</b> —Overrides peer window 0 inputs mapped to the output point with the value defined in the Pt[x].OverridePeerWindow0Value output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverridePeerWindow0Value	BOOL	<b>Override Peer Window 0 Value</b> —Indicates the On/Off status to apply to peer window 0 inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerWindow0En output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer

Table 59 - 1756-OB16IEF Module Output Data Tags (Continued)

Name	Data Type	Tag Definition	Module Definition
Pt[x].OverridePeerWindow1En	BOOL	<b>Override Peer Window 1</b> —Overrides peer window 1 inputs mapped to the output point with the value defined in the Pt[x].OverridePeerWindow1Value output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].OverridePeerWindow1Value	BOOL	<b>Override Peer Window 1 Value</b> —Indicates the On/Off status to apply to peer window 1 inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerWindow1En output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMCycleTime	REAL	<b>PWM Cycle Time</b> —Defines the duration of each pulse cycle. Requires PWM to be enabled via the Pt[x].PWMEnable configuration tag. Valid values: 0.001...3600.0 seconds	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[x].PWMOnTime	REAL	<b>PWM On Time</b> —Defines the length of time that a pulse is active. Requires PWM to be enabled via the Pt[x].PWMEnable configuration tag. Valid values: 0.0002...3600.0 seconds or 0...100.0 percent	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Timestamp	DINT	<b>Timestamp</b> —CIP Sync time at which to apply scheduled output data.	Connection = Data Output Data = Scheduled per Module
TimestampOffset	DINT	<b>Timestamp Offset</b> —Indicates the difference between the system time and the module's local time. The timestamp is in CIP Sync time. This value is typically set to zero but can be updated with the value of the SystemOffset in the controller's TIMESYNCHRONIZE object to enable Time Step Compensation in the module.	Connection = Data Output Data = Scheduled per Module

### 1756-OB16IEFS Module

The tag names and data structures for the 1756-OB16IEFS module vary based on the module definition:

- For Scheduled Per Point output, the module uses a flat data structure. See [Table 60](#), [Table 62](#), and [Table 64](#).
- For Data output or Listen Only connections, the module uses an array data structure. See [Table 61](#), [Table 63](#), and [Table 65](#). For more information about array data structures, see [Redundant Owner Configuration Tags on page 143](#).

Table 60 - 1756-OB16IEFS Module Configuration Tags—Scheduled per Point Output

Name	Data Type	Tag Definition	Module Definition
ProgToFaultEn	BOOL	<b>Program to Fault Mode</b> —Enables the transition of outputs to Fault mode if a communication failure occurs in Program mode. Otherwise, outputs remain in Program mode. See FaultMode, FaultValue, ProgMode, and ProgValue. 0 = Outputs stay in Program mode if communication fails. 1 = Outputs go to Fault mode if communication fails.	Connection = Data Output Data = Scheduled per Point
FaultMode	BOOL	<b>Fault Mode</b> —Used in conjunction with the FaultValue tag to determine the state of outputs when a communication failure occurs. 0 = Uses the output value defined in the Pt[x].FaultValue configuration tag (default). 1 = Holds the last state of the output for the length of time defined in the FaultValueStateDuration tag. If PWM is enabled for the output point and the output is currently On, the output continues PWM until the cycle limit is reached or a final fault state goes into effect via the FaultFinalState tag.	Connection = Data Output Data = Scheduled per Point
FaultValue	BOOL	<b>Fault Value</b> —Defines the output value when a fault occurs. Holds the configured state of the output for the length of time defined in the FaultValueStateDuration tag. Requires the corresponding bit in the FaultMode tag to be cleared. 0 = Off 1 = On	Connection = Data Output Data = Scheduled per Point

Table 60 - 1756-OB16IEFS Module Configuration Tags—Scheduled per Point Output (Continued)

Name	Data Type	Tag Definition	Module Definition
FaultFinalState	BOOL	<b>Fault Final State</b> —Determines the final output state once the time in the FaultValueStateDuration tag elapses. 0 = Output turns Off once the time in the FaultValueStateDuration tag elapses, and module is still faulted. 1 = Output turns On once the time in the FaultValueStateDuration tag elapses, and module is still faulted.	Connection = Data Output Data = Scheduled per Point
ProgMode	BOOL	<b>Program Mode</b> —Used in conjunction with the ProgValue tag to determine the state of outputs when the controller is in Program mode. 0 = Uses the output value defined in the ProgValue tag (default). 1 = Holds the last state of the output. If PWM is enabled for the output point and the output is currently On, the output continues to use PWM until the cycle limit is reached.	Connection = Data Output Data = Scheduled per Point
ProgValue	BOOL	<b>Program Value</b> —Defines the output state during Program mode. Requires the corresponding bit for the ProgMode tag to be cleared. 0 = The output state is Off during Program mode. 1 = The output state is On during Program mode.	Connection = Data Output Data = Scheduled per Point
FaultValueStateDuration	SINT	<b>Fault State Duration</b> —Defines the length of time that the output state remains in the Fault mode state before transitioning to a final state of On or Off. The Fault mode state is defined in the FaultValue tag. Valid values: • 0 = Hold forever (default). Output remains in Fault mode for as long as the fault condition persists. • 1, 2, 5, or 10 seconds	Connection = Data Output Data = Scheduled per Point
PWM[x].Enable	BOOL	<b>Enable PWM</b> —When set, the pulse train for the output point is controlled by the current PWM configuration. 0 = PWM is disabled (default). 1 = PWM is enabled, and the output uses PWM when the output is On.	Connection = Data Output Data = Scheduled per Point
PWM[x].ExtendCycle	BOOL	<b>Extend PWM Cycle</b> —Determines the output behavior when the value in the PWM.OnTime output tag is less than the value in the PWM.MinimumOnTime configuration tag. Requires PWM to be enabled via the PWM.Enable tag. 0 = The duration of the pulse cycle is not extended (default). If the bit is cleared when the On time is less than the minimum On time, the output is never enabled. 1 = The duration of the pulse cycle is extended to maintain the On time to cycle time ratio while taking into account the minimum On time. <b>IMPORTANT:</b> An extension of the pulse cycle is limited to 10 times the cycle time. If the requested On time is less than 1/10 of the minimum On time, the output remains Off and the cycle does not extend.	Connection = Data Output Data = Scheduled per Point
PWM[x].OnTimeInPercent	BOOL	<b>PWM On Time in Percent</b> —Determines whether PWM On time is defined as a percentage of the cycle time or is defined in seconds. Requires PWM to be enabled via the PWM.Enable tag. 0 = Defines PWM On time in seconds (default). 1 = Defines PWM On time as a percentage.	Connection = Data Output Data = Scheduled per Point
PWM[x].StaggerOutput	BOOL	<b>Stagger PWM Outputs</b> —When set, minimizes the load on the power system by staggering On transitions for outputs. Otherwise, outputs turn On immediately at the start of a cycle. Requires PWM to be enabled via the PWM.Enable tag. 0 = Does not stagger output On transitions (default). Outputs turn On immediately when the Data tag is set to 1 beginning the PWM cycle with a rising edge. 1 = Staggers output On transitions. All outputs configured for PWM staggering turns On at different intervals to minimize a possible power surge if many outputs became energized simultaneously.	Connection = Data Output Data = Scheduled per Point
PWM[x].CycleLimitEnable	BOOL	<b>Enable PWM Cycle Limit</b> —Determines whether to let only a fixed number of pulse cycles occur. Requires PWM to be enabled via the PWM.Enable tag. 0 = Pulse cycles continue to occur until the output turns Off (default). 1 = Lets only the number of pulse cycles defined via the PWM.CycleLimit tag to occur.	Connection = Data Output Data = Scheduled per Point

**Table 60 - 1756-OB16IEFS Module Configuration Tags—Scheduled per Point Output (Continued)**

Name	Data Type	Tag Definition	Module Definition
PWM[x].ExecuteAllCycles	BOOL	<b>Execute All PWM Cycles</b> —Determines whether to execute the number of cycles defined via the PWM.CycleLimit tag regardless of the output logic. Requires PWM to be enabled via the PWM.Enable tag, and a cycle limit to be enabled via the PWM.CycleLimitEnable tag. 0 = The output logic determines the number of cycles to produce (default). 1 = The PWM.CycleLimit tag determines the number of cycles to produce regardless of output logic. For example, if you specify a cycle limit of 4, and the output turns Off after 3 cycles, all 4 cycles still occur despite the output being instructed to turn Off.	Connection = Data Output Data = Scheduled per Point
PWM[x].CycleLimit	SINT	<b>PWM Cycle Limit</b> —Defines the number of pulse cycles to occur when the output turns On: <ul style="list-style-type: none"> <li>If the corresponding bit in the PWM.ExecuteAllCycles tag is set, the configured number of cycles occur even if the output turns Off.</li> <li>If the corresponding bit in the PWM.ExecuteAllCycles tag is cleared, the configured number of cycles occur only if the output remains On. For example, if the cycle limit is 4, and the output turns Off after 3 cycles, the 4th cycle does <b>not</b> occur.</li> </ul> The default cycle limit is 10. Requires PWM to be enabled via the PWM.Enable tag, and cycle limits to be enabled via the PWM.CycleLimitEnable tag.	Connection = Data Output Data = Scheduled per Point
PWM[x].MinimumOnTime	REAL	<b>PWM Minimum On Time</b> —Defines the minimum length of time required for the output to turn On. Requires PWM to be enabled via the PWM.Enable tag. Valid values: 0.0002...3600.0 seconds or 0...100 percent	Connection = Data Output Data = Scheduled per Point

**Table 61 - 1756-OB16IEFS Module Configuration Tags—Data Output**

Name	Data Type	Tag Definition	Module Definition
ProgToFaultEn	BOOL	<b>Program to Fault Mode</b> —Enables the transition of outputs to Fault mode if a communication failure occurs in Program mode. Otherwise, outputs remain in Program mode. See FaultMode, FaultValue, ProgMode, and ProgValue. 0 = Outputs stay in Program mode if communication fails. 1 = Outputs go to Fault mode if communication fails.	Connection = Data Output Data = Data
Pt[x].FaultMode	BOOL	<b>Fault Mode</b> —Used in conjunction with the FaultValue tag to determine the state of outputs when a communication failure occurs. 0 = Uses the output value defined in the Pt[x].FaultValue configuration tag (default). 1 = Holds the last state of the output for the length of time defined in the FaultValueStateDuration tag. If PWM is enabled for the output point and the output is currently On, the output continues PWM until the cycle limit is reached or a final fault state goes into effect via the FaultFinalState tag.	Connection = Data Output Data = Data
Pt[x].FaultValue	BOOL	<b>Fault Value</b> —Defines the output value when a fault occurs. Holds the configured state of the output for the length of time defined in the FaultValueStateDuration tag. Requires the corresponding bit in the FaultMode tag to be cleared. 0 = Off 1 = On	Connection = Data Output Data = Data
Pt[x].FaultFinalState	BOOL	<b>Fault Final State</b> —Determines the final output state once the time in the FaultValueStateDuration tag elapses. 0 = Output turns Off once the time in the FaultValueStateDuration tag elapses, and module is still faulted. 1 = Output turns On once the time in the FaultValueStateDuration tag elapses, and module is still faulted.	Connection = Data Output Data = Data
Pt[x].ProgMode	BOOL	<b>Program Mode</b> —Used in conjunction with the ProgValue tag to determine the state of outputs when the controller is in Program mode. 0 = Uses the output value defined in the ProgValue tag (default). 1 = Holds the last state of the output. If PWM is enabled for the output point and the output is currently On, the output continues to use PWM until the cycle limit is reached.	Connection = Data Output Data = Data
Pt[x].ProgValue	BOOL	<b>Program Value</b> —Defines the output state during Program mode. Requires the corresponding bit for the ProgMode tag to be cleared. 0 = The output state is Off during Program mode. 1 = The output state is On during Program mode.	Connection = Data Output Data = Data
Pt[x].PWMEnable	BOOL	<b>Enable PWM</b> —When set, the pulse train for the output point is controlled by the current PWM configuration. 0 = PWM is disabled (default). 1 = PWM is enabled, and the output uses PWM when the output is On.	Connection = Data Output Data = Data

Table 61 - 1756-OB16IEFS Module Configuration Tags—Data Output (Continued)

Name	Data Type	Tag Definition	Module Definition
Pt[x].PWMEExtendCycle	BOOL	<b>Extend PWM Cycle</b> —Determines the output behavior when the value in the PWM.OnTime output tag is less than the value in the PWM.MinimumOnTime configuration tag. Requires PWM to be enabled via the PWM.Enable tag. 0 = The duration of the pulse cycle is not extended (default). If the bit is cleared when the On time is less than the minimum On time, the output is never enabled. 1 = The duration of the pulse cycle is extended to maintain the On time to cycle time ratio while taking into account the minimum On time. <b>IMPORTANT:</b> An extension of the pulse cycle is limited to 10 times the cycle time. If the requested On time is less than 1/10 of the minimum On time, the output remains Off and the cycle does not extend.	Connection = Data Output Data = Data
Pt[x].PWMOnTimeInPercent	BOOL	<b>PWM On Time in Percent</b> —Determines whether PWM On time is defined as a percentage of the cycle time or is defined in seconds. Requires PWM to be enabled via the PWM.Enable tag. 0 = Does not stagger output On transitions (default). Outputs turn On immediately when the Data tag is set to 1 beginning the PWM cycle with a rising edge. 1 = Staggers output On transitions. All outputs configured for PWM staggering turn On at different intervals to minimize a possible power surge if many outputs became energized simultaneously.	Connection = Data Output Data = Data
Pt[x].PWMStaggerOutput	BOOL	<b>Stagger PWM Outputs</b> —When set, minimizes the load on the power system by staggering On transitions for outputs. Otherwise, outputs turn On immediately at the start of a cycle. Requires PWM to be enabled via the PWM.Enable tag. 0 = Does not stagger output On transitions (default). Outputs turn On immediately when the Data tag is set to 1 beginning the PWM cycle with a rising edge. 1 = Staggers output On transitions. All outputs configured for PWM staggering turn On at different intervals to minimize a possible power surge if many outputs became energized simultaneously.	Connection = Data Output Data = Data
Pt[x].PWMCycleLimitEnable	BOOL	<b>Enable PWM Cycle Limit</b> —Determines whether to let only a fixed number of pulse cycles occur. Requires PWM to be enabled via the PWM.Enable tag. 0 = Pulse cycles continue to occur until the output turns Off (default). 1 = Lets only the number of pulse cycles defined via the PWM.CycleLimit tag to occur.	Connection = Data Output Data = Data
Pt[x].PWMEExecuteAllCycles	BOOL	<b>Execute All PWM Cycles</b> —Determines whether to execute the number of cycles defined via the PWM.CycleLimit tag regardless of the output logic. Requires PWM to be enabled via the PWM.Enable tag, and a cycle limit to be enabled via the PWM.CycleLimitEnable tag. 0 = The output logic determines the number of cycles to produce (default). 1 = The PWM.CycleLimit tag determines the number of cycles to produce regardless of output logic. For example, if you specify a cycle limit of 4, and the output turns Off after 3 cycles, all 4 cycles still occur despite the output being instructed to turn Off.	Connection = Data Output Data = Data
Pt[x].PWMFaultValueStateDuration	SINT	<b>Fault State Duration</b> —Defines the length of time that the output state remains in the Fault mode state before transitioning to a final state of On or Off. The Fault mode state is defined in the FaultValue tag. Valid values: • 0 = Hold forever (default). Output remains in Fault mode for as long as the fault condition persists. • 1, 2, 5, or 10 seconds	Connection = Data Output Data = Data
Pt[x].PWMCycleLimit	SINT	<b>PWM Cycle Limit</b> —Defines the number of pulse cycles to occur when the output turns On: • If the corresponding bit in the PWM.ExecuteAllCycles tag is set, the configured number of cycles occur even if the output turns Off. • If the corresponding bit in the PWM.ExecuteAllCycles tag is cleared, the configured number of cycles occur only if the output remains On. For example, if the cycle limit is 4, and the output turns Off after 3 cycles, the 4th cycle does <b>not</b> occur. The default cycle limit is 10. Requires PWM to be enabled via the PWM.Enable tag, and cycle limits to be enabled via the PWM.CycleLimitEnable tag.	Connection = Data Output Data = Data
Pt[x].PWMMinimumOnTime	REAL	<b>PWM Minimum On Time</b> —Defines the minimum length of time required for the output to turn On. Requires PWM to be enabled via the PWM.Enable tag. Valid values: 0.0002...3600.0 seconds or 0...100 percent	Connection = Data Output Data = Data

Table 62 - 1756-OB16IEFS Module Input Data Tags—Scheduled per Point Output

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the output module is lost, then all 32 bits of the Module Fault word are set. 0 = No fault 1 = Fault	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
Data	BOOL	<b>Data</b> —Indicates the current value to be sent to the corresponding output point. If PWM is enabled, this value does transition from 0 to 1 based on the PWM pulse train. 0 = Off 1 = On	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
FuseBlown	BOOL	<b>Fuse Is Blown</b> —Indicates whether a fuse has blown due to a short or overload condition for the corresponding point. All blown fuse conditions are latched and must be reset. 0 = Fuse is not blown. 1 = Fuse is blown and has not been reset.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
CIPSyncValid	BOOL	<b>CIP Sync Is Valid</b> —Indicates whether the module has synchronized to a valid CIP Sync time master on the backplane. 0 = CIP Sync is not available. 1 = CIP Sync is available.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
CIPSyncTimeout	BOOL	<b>CIP Sync Timeout</b> —Indicates whether a valid time master on the backplane has timed out. 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. The module is currently using its local clock.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
LateScheduleCount	INT	<b>Late Schedule Count</b> —Increments each time a schedule is received late after its scheduled time. The counter rolls over every 65,535 late schedules. If a late schedule is the most recent schedule for a point, the output is still driven to new state. Monitoring the late schedule count may be useful to determine whether network delays or connection losses are impacting schedules.	Connection = Data Output Data = Scheduled per Point
LostScheduleCount	INT	<b>Lost Schedule Count</b> —Increments each time the Schedule.SequenceNumber output tag skips a value. A skipped sequence number may indicate a lost schedule. The counter rolls over every 65,535 lost schedules.	Connection = Data Output Data = Scheduled per Point
LocalClockOffset	DINT	<b>Local Clock Timestamp</b> —Indicates the offset between the current CST and the CIP Sync value when a valid CIP Sync time is available.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
OffsetTimestamp	DINT	<b>Timestamp Offset</b> —Indicates when the CIP Sync LocalClockOffset and GrandMasterID were last updated in CIP Sync format.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
GrandMasterClockID	DINT	<b>Grandmaster Clock ID</b> —Indicates the ID of the CIP Sync Grandmaster to which the module is synced.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
Timestamp	DINT	<b>Timestamp</b> —A 64-bit CIP Sync timestamp of the last new output data or FuseBlown event.	Connection = Data Output Data = Scheduled per Point or Connection = Listen Only Output Data = None
Schedule.State	SINT	<b>Schedule State</b> —State of the Current Schedule: <ul style="list-style-type: none"> <li>• 0 = Inactive</li> <li>• 1 = Active - schedule is active but not yet applied</li> <li>• 2 = Current - schedule is active and next to be applied</li> <li>• 3 = Expired - schedule has been applied</li> <li>• 4 = Schedule discarded - request in error</li> <li>• 5 = Late, but applied - schedule arrived after scheduled time but was applied since no more current schedule received</li> </ul>	Connection = Data Output Data = Scheduled per Point
Schedule.SequenceNumber	SINT	<b>Schedule Sequence Number</b> —The data echo indicating the sequence number of the schedule.	Connection = Data Output Data = Scheduled per Point

Table 63 - 1756-OB16IEFS Module Input Data Tags—Data Output or Listen Only Connections

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the output module is lost, then all 32 bits of the Fault word are set. 0 = No fault 1 = Fault	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].Data	BOOL	<b>Data</b> —Indicates the current value to be sent to the corresponding output point. If PWM is enabled, this value transitions from 0 to 1 based on the PWM pulse train. 0 = Off 1 = On	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].Fault	BOOL	<b>Fault Status</b> —Indicates whether a point is faulted. If communication to the output module is lost, then all 32 bits of the Fault word are set. 0 = No fault 1 = Fault	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].FuseBlown	BOOL	<b>Fuse Is Blown</b> —Indicates whether a fuse has blown due to a short or overload condition for the corresponding point. All blown fuse conditions are latched and must be reset. 0 = Fuse is not blown. 1 = Fuse is blown and has not been reset.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].PWMCycleLimitDone	BOOL	<b>PWM Cycle Limit Done</b> —Indicates whether the PWM pulse cycle limit defined in the Pt[x].PWMCycleLimit configuration tag has been reached. 0 = The PWM cycle limit has not yet been reached. The bit resets to 0 each time the output transitions to On to begin a new PWM cycle. 1 = The PWM cycle limit has been reached.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].CIPSyncValid	BOOL	<b>CIP Sync Is Valid</b> —Indicates whether the module has synchronized to a valid CIP Sync time master on the backplane. 0 = CIP Sync is not available. 1 = CIP Sync is available.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Pt[x].CIPSyncTimeout	BOOL	<b>CIP Sync Timeout</b> —Indicates whether a valid time master on the backplane has timed out. 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. The module is currently using its local clock.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
LocalClockOffset	DINT	<b>Local Clock Timestamp</b> —Indicates the offset between the current CST and the CIP Sync value when a valid CIP Sync time is available.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
OffsetTimestamp	DINT	<b>Timestamp Offset</b> —Indicates when the CIP Sync LocalClockOffset and GrandMasterID were last updated in CIP Sync format.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
GrandMasterClockID	DINT	<b>Grandmaster Clock ID</b> —Indicates the ID of the CIP Sync Grandmaster to which the module is synced.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None
Timestamp	DINT	<b>Timestamp</b> —A 64-bit CIP Sync timestamp of the last new output data or FuseBlown event.	Connection = Data Output Data = Data or Connection = Listen Only Output Data = None

**Table 64 - 1756-OB16IEFS Module Output Data Tags—Scheduled per Point Output**

Name	Data Type	Tag Definition	Module Definition
Data	BOOL	<b>Data</b> —Indicates the On/Off state to apply to a nonscheduled output point. 0 = Off 1 = On	Connection = Data Output Data = Scheduled per Point
ScheduleMask	BOOL	<b>Schedule Mask</b> —A mask indicating which output points are scheduled. 0 = The output point is unscheduled. The On/Off state is determined by the value in the Data output tag. 1 = The output point is scheduled. The On/Off state is determined by the Schedule[x].Data output tag.	Connection = Data Output Data = Scheduled per Point
ResetFuseBlown	BOOL	<b>Reset Blown Fuse</b> —Attempts to clear a blown fuse status and apply output data when the bit transitions from Off to On.	Connection = Data Output Data = Scheduled per Point
TimestampOffset	DINT	<b>Timestamp Offset</b> —Indicates the difference between the system time and the module's local time. The timestamp is in CIP Sync time. This value is typically set to zero but can be updated with the value of the SystemOffset in the controller's TIMESYNCHRONIZE object to enable Time Step Compensation in the module.	Connection = Data Output Data = Scheduled per Point
ScheduleTimestamp	DINT	<b>Schedule Timestamp</b> —The baseline CIP Sync time for all schedules. The module uses the baseline CIP Sync time combined with the offset value in the Schedule.Offset tag to calculate the absolute time a physical output turns On or Off.	Connection = Data Output Data = Scheduled per Point
Schedule[x].ID	SINT	<b>Schedule ID</b> —Identifies which schedule to apply to an output point. Valid schedules: 1...32 0= No schedule	Connection = Data Output Data = Scheduled per Point
Schedule[x].SequenceNumber	SINT	<b>Schedule Sequence Number</b> —Indicates the sequence count received with a schedule. The module recognizes a new schedule only when there is a change in sequence number. The first message received initializes the schedule.	Connection = Data Output Data = Scheduled per Point
Schedule[x].OutputPointSelect	SINT	<b>Schedule Output Point</b> —Indicates which physical output point is associated with a schedule. The module recognizes a new schedule only when there is a change in output point. The first message received initializes the schedule. Valid values: 0...15	Connection = Data Output Data = Scheduled per Point
Schedule[x].Data	SINT	<b>Schedule Data</b> —Indicates the On/Off state to apply to an output point at the scheduled time. 0 = Off 1 = On	Connection = Data Output Data = Scheduled per Point
Schedule[x].Offset	DINT	<b>Schedule Offset</b> —Indicates a schedule's offset value to be added to the baseline ScheduleTimestamp value to determine the absolute time at which a physical output turns On or Off. The offset value must be +/-35 minutes from the baseline ScheduleTimestamp value.	Connection = Data Output Data = Scheduled per Point
PWM.CycleTime	REAL	<b>PWM Cycle Time</b> —Defines the duration of each pulse cycle. Requires PWM to be enabled via the PWM.Enable configuration tag. Valid values: 0.001...3600.0 seconds	Connection = Data Output Data = Scheduled per Point
PWM.OnTime	REAL	<b>PWM On Time</b> —Defines the length of time that a pulse is active. Requires PWM to be enabled via the PWM.Enable configuration tag. Valid values: 0.0002...3600.0 seconds or 0...100.0 percent	Connection = Data Output Data = Scheduled per Point

**Table 65 - 1756-OB16IEFS Module Output Data Tags—Data Output**

Name	Data Type	Tag Definition	Module Definition
Pt[x].Data	BOOL	<b>Data</b> —Indicates the On/Off state to apply to a nonscheduled output point. 0 = Off 1 = On	Connection = Data Output Data = Data
Pt[x].ResetFuseBlown	BOOL	<b>Reset Blown Fuse</b> —Attempts to clear a blown fuse status and apply output data when the bit transitions from Off to On.	Connection = Data Output Data = Data
Pt[x].PWMCycleTime	REAL	<b>PWM Cycle Time</b> —Defines the duration of each pulse cycle. Requires PWM to be enabled via the PWM.Enable configuration tag. Valid values: 0.001...3600.0 seconds	Connection = Data Output Data = Data
Pt[x].PWMOnTime	REAL	<b>PWM On Time</b> —Defines the length of time that a pulse is active. Requires PWM to be enabled via the PWM.Enable configuration tag. Valid values: 0.0002...3600.0 seconds or 0...100.0 percent	Connection = Data Output Data = Data

## Redundant Owner Configuration Tags

The redundant owner configuration tags are identical to the existing 1756-OB16IEF configuration.

### Redundant Owner Input Tag Layout

We added eight Input tags to the existing Input data layout. The bit tag field, *PartnerBits*, increased from 2 bits to 10 bits to cover OwnerActive, Owner ID, and Active/Claim/Ready status for the A and B controllers.

**Table 66 - New Redundant Input Tag Description - 1756-OB16IEF**

Field	Data Type	Legal Values	Usage
OwnerActive	BOOL	B4	The Output has a Controlling Owner
OwnerID	BOOL	B5	Which Owner is Active (Controlling): 0 = Owner A 1 = Owner B
OwnerAConnected	BOOL	B6	Owner A is Connected.
OwnerAClaim	BOOL	B7	Owner A is claiming Outputs (COO).
OwnerAReady	BOOL	B8	Owner A is Ready to claim Outputs (ROO).
OwnerBConnected	BOOL	B9	Owner B is Connected.
OwnerBClaim	BOOL	B10	Owner B is claiming Outputs (COO).
OwnerBReady	BOOL	B11	Owner B is Ready to claim Outputs (ROO).

### Redundant Owner Output Tag Layout

We added two Input tags in the Output data: a BOOL to reflect the COO bit and another for the ROO bit.

**Table 67 - New Redundant Output Tag Description - 1756-OB16IEF**

Field	Data Type	Legal Values	Usage
OwnerClaim	BOOL	B0 0 = Not Requesting Ownership 1 = Claim Ownership	When set, indicates that the controller wants to Claim Ownership of the redundant connection and have its outputs actively used. <ul style="list-style-type: none"> <li>If both Redundant Owners have COO set, the last originator application that transitioned its COO flag from reset to set shall be the owner.</li> </ul> <b>NOTE:</b> Only an Rx'd 0 to a 1 and not just the receipt of a new connection is considered such a transition. <ul style="list-style-type: none"> <li>If neither Redundant Owner has COO set, then the highest ROO becomes Owner.</li> </ul>
OwnerReady	BOOL	B1 0 = Not Ready to Own 1 = Ready to Own	Indicates the controller's readiness for output ownership. A value of 1 indicates ready while 0 indicates the controller is not ready to own the outputs. <ul style="list-style-type: none"> <li>If neither controller has the COO bit set, the ROO determines the current owner.</li> <li>If both controllers have COO=0 and ROO=0, then the outputs go to IDLE (Program mode).</li> <li>If both controllers have COO=0 and ROO=1, OwnerA will be active.</li> </ul>

## Array Data Structures

Fast digital I/O modules use an array data structure. In this type of structure, all the tags for a particular point are organized under that point. For example, in [Figure](#), all of the tags that appear under point 0 also appear under points 1..15 for the input module in slot 1. With this structure, you can copy or access all of the data for a particular point by simply referencing or copying the point or alias for the point, such as Pt[3] or PressureValveTank3.

### Array Data Structure

[-] Local:1:I.Pt		{...}
[-] Local:1:I.Pt[0]		{...}
[-] Local:1:I.Pt[0].Data		0
[-] Local:1:I.Pt[0].Fault		0
[-] Local:1:I.Pt[0].NewDataOffOn		0
[-] Local:1:I.Pt[0].NewDataOnOff		0
[-] Local:1:I.Pt[0].TimestampDropped		0
[-] Local:1:I.Pt[0].CIPSyncValid		0
[-] Local:1:I.Pt[0].CIPSyncTimeout		0
[-] Local:1:I.Pt[0].InputOverrideStatus		0
[-] Local:1:I.Pt[0].Timestamp		{...}
[-] Local:1:I.Pt[0].Timestamp.OffOn		{...}
[-] Local:1:I.Pt[0].Timestamp.OnOff		{...}
[-] Local:1:I.Pt[1]		{...}
[-] Local:1:I.Pt[2]		{...}

Other digital I/O modules use a flat data structure. In this type of structure, only one instance of a tag exists for a module. For example, in [Figure](#), only one instance of each tag appears under the input module in slot 3. To reference or copy data for an individual point, you specify the tag name followed by a bit number, such as Data.0 or EventOverflow.3. Unlike an array structure where all the data for a point can be accessed via a single tag reference, a flat structure requires multiple tag references to access all the data for a point.

### Flat Data Structure

[-] Local:3:I		{...}
[+] Local:3:I.Fault	2#1111_1111_1111_1111_1111_1111_1111_1111	
[+] Local:3:I.Data	2#0000_0000_0000_0000_0000_0000_0000_0000	
[+] Local:3:I.NewData	2#0000_0000_0000_0000_0000_0000_0000_0000	
[+] Local:3:I.EventOverflow	2#0000_0000_0000_0000_0000_0000_0000_0000	
[-] Local:3:I.ReturningUCTTime		0
[-] Local:3:I.ReturningCIPSyncTime		0
[-] Local:3:I.CIPSyncValid		0
[-] Local:3:I.CIPSyncTimeout		0
[+] Local:3:I.EventNumber		0
[+] Local:3:I.LocalClockOffset		{...}
[+] Local:3:I.Offsettimestamp		{...}
[+] Local:3:I.Timestamp		{...}

The 1756-0B16IEFS module uses either type of data structure depending on how you configure the module. For more information, see [page 136](#).

## Use Ladder Logic To Perform Run Time Services and Reconfiguration

You can use ladder logic to perform run-time services on your module. For example, [page 154](#) shows how to reset an electronic fuse module by using the Studio 5000 Logix Designer<sup>®</sup> application. This appendix provides an example of how to reset the same fuse without using the Logix Designer application.

In addition to performing run-time services, you can use ladder logic to change configuration. [Chapter 4](#) explained how to use the Logix Designer application to set configuration parameters in your ControlLogix<sup>®</sup> digital I/O module. Some of those parameters may also be changed through ladder logic.

### Using Message Instructions

In ladder logic, you can use message instructions to send occasional services to any ControlLogix I/O module. Message instructions send an explicit service to the module, causing specific behavior to occur. For example, unlatching a high alarm can be performed by a message instruction.

Message instructions maintain these characteristics:

- Messages use unscheduled portions of system communication bandwidth
- One service is performed per instruction
- Performing module services does not impede module functionality, such as sampling inputs or applying new outputs

### Processing Real-time Control and Module Services

Services that are sent through message instructions are not as time critical as the module behavior defined during configuration and maintained by a real-time connection. Therefore, the module processes messaging services only after the needs of the I/O connection have been met.

For example, you may want to unlatch all process alarms on the module, but real-time control of your process is still occurring by using the input value from that same channel. Because the input value is critical to your application, the module prioritizes the sampling of inputs ahead of the unlatch service request.

This prioritization lets input channels be sampled at the same frequency and the process alarms to be unlatched in the time between sampling and producing the real-time input data.

### One Service Performed Per Instruction

A message instruction causes a module service to be performed only once per execution. For example, if a message instruction sends a service to the module to unlatch the high high alarm on a particular channel, that channel's high high alarm unlatches, but may be set on a subsequent channel sample. The message instruction must then be re-executed to unlatch the alarm a second time.

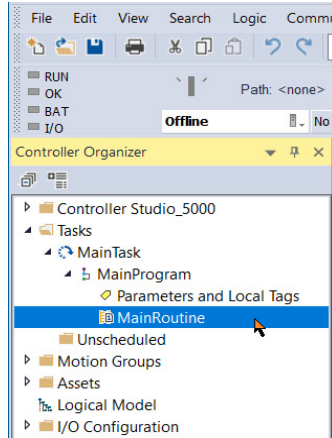
## Create a New Tag

This section shows how to create a tag in ladder logic when adding a message instruction. Ladder logic is in the main routine within the Logix Designer application.

Follow these steps to create a tag.

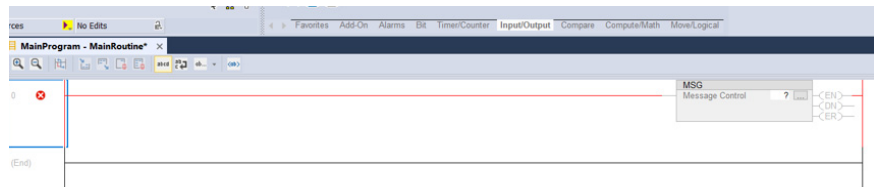
1. Start the Logix Designer application and open an existing I/O project or create a new one.
2. On the Controller Organizer, double-click MainTask.

Expand MainProgram to see Main Routine as a submenu item.

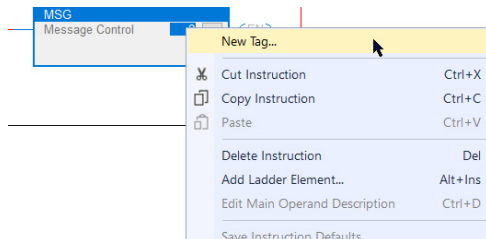


A graphic that looks like a ladder, with rungs, appears in the right side of the Logix Designer application. You attach run-time service, such as a message instruction, to the rungs and then download the information to a controller.

You can tell that the rung is in Edit mode because of the 'e' at the left side of the rung.



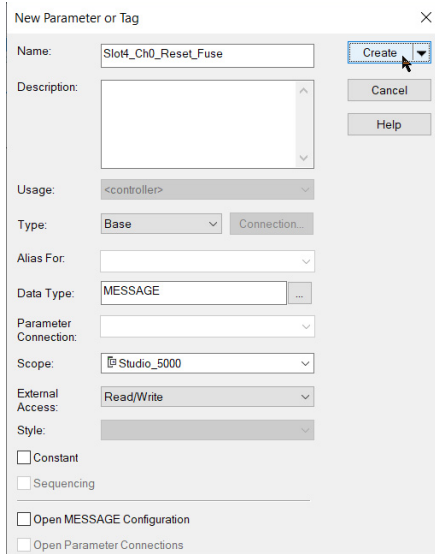
3. Find, then click MSG (message) instruction on the instruction toolbar.  
The MSG icon is among the formats on the Input/Output tab of the instruction toolbar. You can also drag-and-drop an instruction icon onto a rung. A green dot appears when a valid location is detected for the instruction on the rung.
4. Inside the message box in the Message Control field, right-click the question mark to access a dropdown menu.



5. Choose New Tag.

The New Tag dialog box appears with the cursor in the Name field.

**IMPORTANT** We suggest you name the tag to indicate what module service the message instruction is sending. For example, if a message instruction is to reset an electronic fuse, then name the tag, 'reset fuse', to reflect this.



6. Complete the fields on the New Tag dialog box.

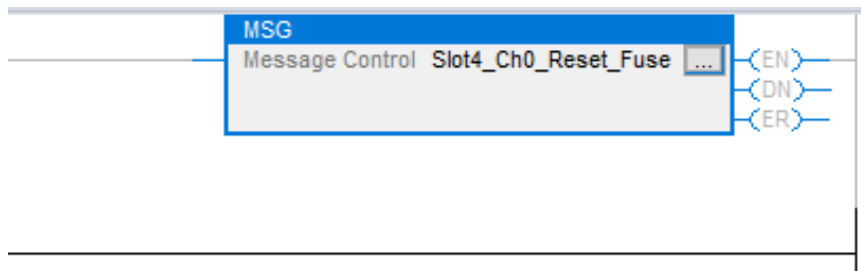
Field	Description
Name	Type the tag name, including the slot number in the module.
Description	Type an option tag description.
Usage	Use the default.
Type	Use the default.
Alias for	Leave blank.
Data Type	Choose MESSAGE.
Scope	Choose the Controller scope. <b>Note:</b> Message tags can be created only with the Controller scope.
External Access	Use the default.
Style	Leave blank.
Constant	Leave blank.
Open MESSAGE Configuration	Leave the box blank if you do NOT want to automatically access the Message Configuration screen when OK is clicked. You can still access the Message Configuration screen later by following the procedures on <a href="#">page 148</a> .

7. Click OK.

## Enter Message Configuration

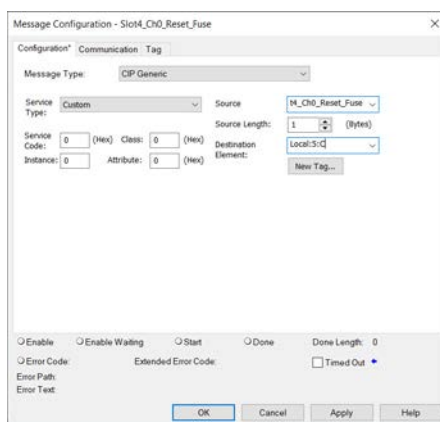
After creating a tag, you must enter certain parameters for the message configuration. This information is entered on the Configuration and Communication tabs of the Message Configuration dialog box.

The Message Configuration dialog box is accessed by clicking the box with the ellipses (in the Message Control field).



## Configuration Tab

The Configuration tab provides information on what module service to perform and where to perform it.



This table explains the relationship of the fields in the above dialog boxes. For example, despite different entry fields, both screen examples are configured to send a message to reset an electronic fuse (module service) on Channel 0 of a 1756-0A8D module (where to perform the service).

**Table 68 - Relationship of Message Configuration Parameters**

Parameter	Description
Service Type	Defines the type of module service to be performed. For example, a reset. <b>Note:</b> In version 10.07.00 or later, you can use a dropdown menu to choose the Service Type. The application defaults the Service Code, Instance, Class, and Attribute parameters based on the Service Type that you choose. All values are in Hex.
Class	Object that you are sending a message to, such as the device object or a discrete output point.
Instance	Each object can have multiple instances. For example, a discrete output can have 16 points or instances of where a message can be sent. This specifies the instance.
Attribute	Further identifies the exact address for the message. An analog input can have multiple alarms so this attribute acknowledges a specific alarm and not the other alarms. If an attribute is not specified (default to 0) the Service applies to all attributes of the Class/Instance.

This table lists tags that are used in the Source and Destination fields of the message instructions.

**Table 69 - Source and Destination Field Tags**

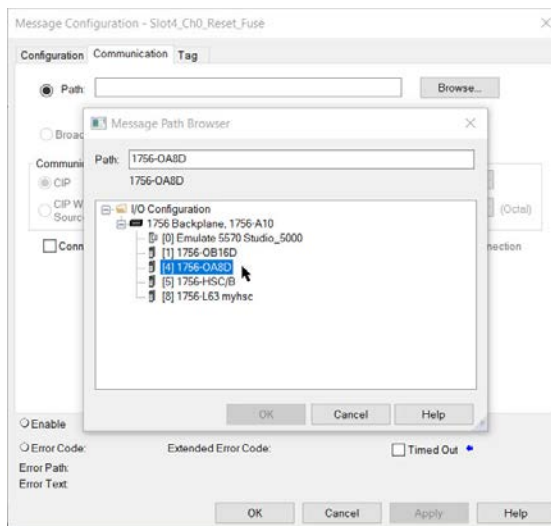
Source Tag	Description
Enable_32_Points DINT	Parameter used to determine which points are enabled for the service. That is, if bit 0 = 1 for Reset Fuse, then point 0 has its electronic fuse reset.
Results_32_Points DINT	Pass (0)/ Fail (1) result for the service. That is, if bit 0 = 1 for the results of the Reset Fuse, then the Reset Fuse failed for point 0.

Choose the physical location, slot number, and data type in the Source Element and Destination fields.

### Communication Tab

The Communication tab provides information on the path of the message instruction. For example, the slot number of a 1756-OA8D module distinguishes exactly which module a message is designated for.

**IMPORTANT** Use the Browse button to see a list of the I/O modules in the system. You choose a path when you choose a module from the list. You must name an I/O module during initial module configuration to choose a path for your message instruction. Click OK to set the path.



### Timestamped Inputs and Scheduled Outputs - Standard or Diagnostic Modules

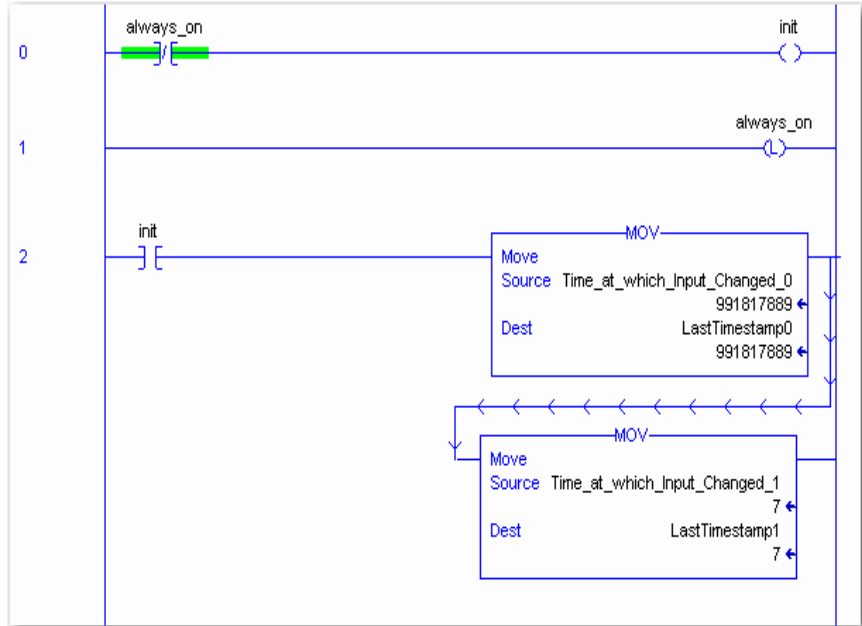
This section demonstrates the use of timestamped inputs and scheduled outputs for standard and diagnostic digital I/O modules. The Change of State timestamp can be used to synchronize the output turning On or Off based on the time that the input transitions. The program can be extended to include synchronizing multiple output modules by sending the same timestamp to all output modules.

In the illustration below, the output follows the state of input 0, but it is delayed by exactly 10 ms. The advantage of using CST over timers is that the synchronization is performed at the I/O module, which eliminates any jitter due to controller or communication delays.

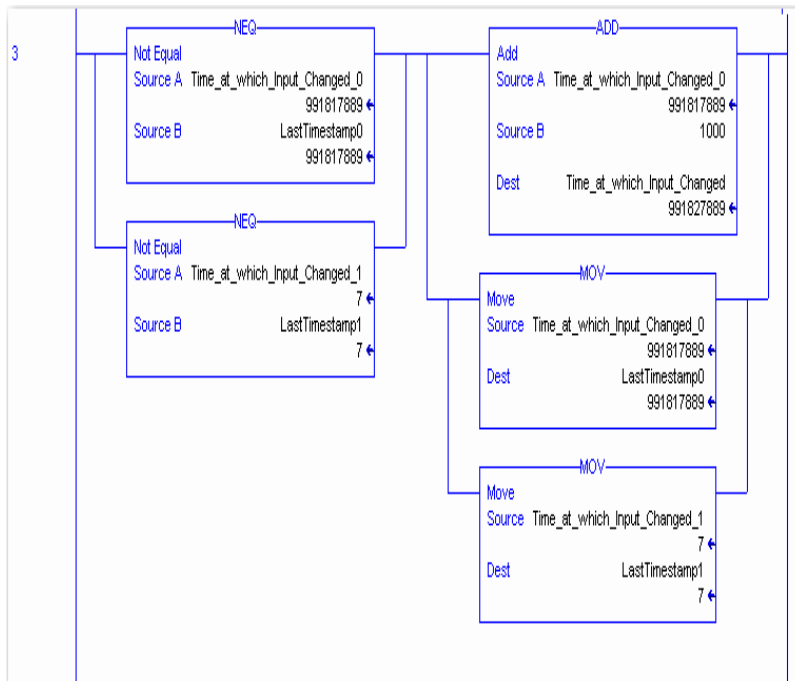
Your control becomes much more deterministic even under changing loads. For this synchronization to work properly, the 10 ms delay must be long enough to account for any controller, backplane, and network delays. The input and output modules must reside in the same rack as a Time Master (controller). Timestamp units are microseconds.

The following illustrations show the ladder instructions the program uses. The rungs perform these tasks:

- Rungs 0 and 1 detect the transition from Program to Run mode. This is used to turn On 'init', which causes the program to initialize its tags.
- Rung 2 only executes once and initializes the LastTimestamp. LastTimestamp is used to detect a Change of State on the input point by checking to see if the timestamp of the input data has changed.



- Rung 3 is the main rung that checks for Change of State on the input point by comparing the current input timestamp (Time\_at\_which\_Input\_Changed) with the last timestamp (LastTimestamp).

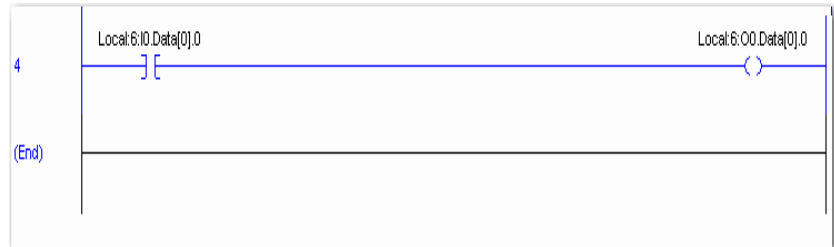


The input point (point 0) must have Change of State enabled or the timestamp does not update when the point transitions. Once Change of State has been detected, 10 ms is added to the input timestamp and sent to the output module's timestamp. This causes the output module to apply its output exactly 10 ms (10,000 μs) after the input changed state.

The MOVE instructions update LastTimestamp in preparation for the next change of state.

**IMPORTANT** Timestamps are 8 bytes in size, two DINTs, but only the lower 4 bytes of the output timestamp (Time\_at\_which\_Output\_Will\_Change) are used to schedule the outputs into the future (to a max of 16.7 s or 16,700,000 μs).

- Rung 4 is the standard XIC-OTE rung that controls the output point based on the input point.



The only difference is the output module is configured for scheduled outputs. The outputs are not applied until the scheduled time has occurred.

The Controller Tags dialog box shows examples of the tags created in ladder logic.

P	Tag Name	Alias For	Base Tag	Type	Style	Description
	always_on			BOOL	Decim	
	init			BOOL	Decim	
<input checked="" type="checkbox"/>	-LastTimestamp			DINT[2]	Decim	
	-LastTimestamp[0]			DINT	Decim	
	-LastTimestamp[1]			DINT	Decim	
<input checked="" type="checkbox"/>	Local:0:C			AB:175		
<input checked="" type="checkbox"/>	Local:0:I			AB:175		
<input checked="" type="checkbox"/>	Local:1:C			AB:175		
<input checked="" type="checkbox"/>	Local:1:I			AB:175		
<input checked="" type="checkbox"/>	Local:1:O			AB:175		
<input type="checkbox"/>	Time_at_which_Input_Changed_0	Local:0:I.CSTTimestamp[0]	Local:0:I.C	DINT	Decim	
<input type="checkbox"/>	Time_at_which_Input_Changed_1	Local:0:I.CSTTimestamp[1]	Local:0:I.C	DINT	Decim	
<input type="checkbox"/>	Time_at_which_Output_Will_Change	Local:1:O.CSTTimestamp[0]	Local:1:O	DINT	Decim	

### Timestamped Inputs and Scheduled Outputs - Fast Modules

This section demonstrates the use of timestamped inputs and scheduled outputs for fast digital I/O modules. The Change of State timestamp can be used to synchronize the output turning On or Off based on the time that the input transitions. The program can be extended to include synchronizing multiple output modules by sending the same timestamp to all output modules.

In the example on [page 152](#), the output follows the state of input 0, but it is delayed by the amount of time in the Delay tag. The advantage of using CIP Sync™ over timers is that the synchronization is performed at the I/O module, which eliminates any jitter due to controller or communication delays.

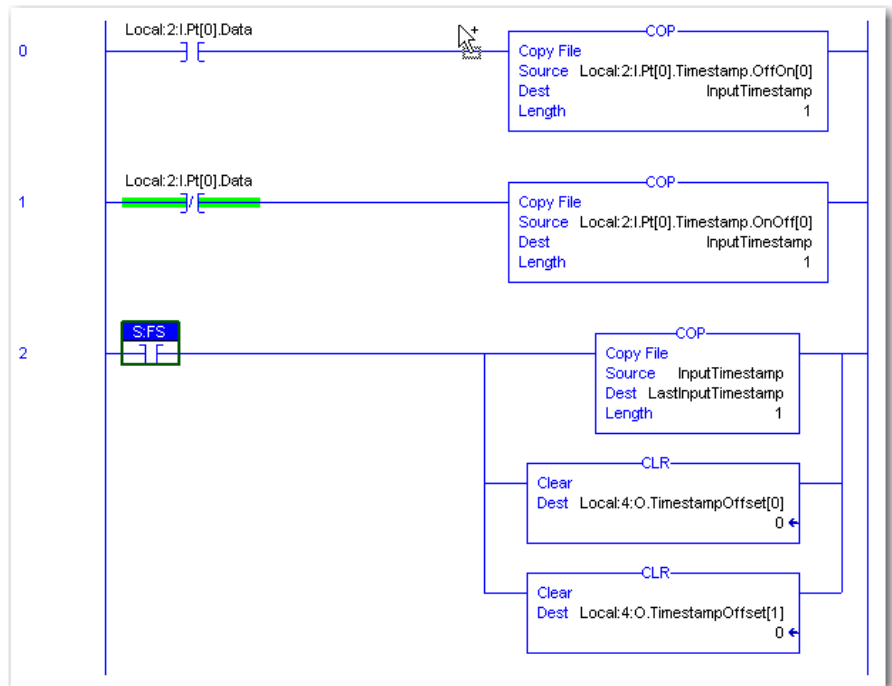
Your control becomes much more deterministic even under changing loads. For this synchronization to work properly, the value in the Delay tag must be long enough to account for any controller, backplane, and network delays.

In this example, the controller, input, and output modules all reside in the same chassis, but they can reside in separate chassis as long as they are all part of the same synchronized CIP Sync system. Time-stamp units are microseconds.

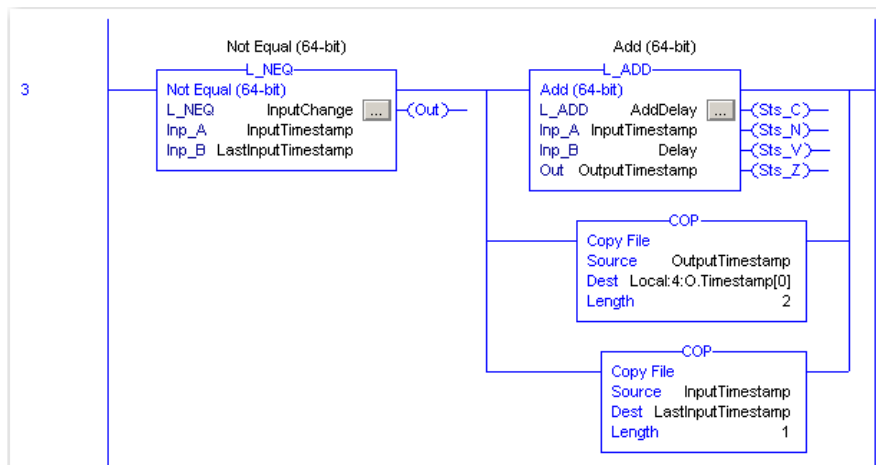
**IMPORTANT** Unlike standard and diagnostic I/O modules that use CST for timestamps, fast I/O modules use CIP Sync timestamps, which are a full 64 bits in width. Manipulation of CIP Sync time values requires the use of 64-bit math. The following example uses 64-bit Add-On Instructions contained in the LINT (64-bit signed 2's complement integer) Math Library at <https://www.rockwellautomation.com/en-us/support/product/product-downloads/application-code-library/sample-code.html>.

The following illustrations show the ladder instructions the program uses. The rungs perform these tasks:

- Rungs 0 and 1 capture the rising or falling timestamps for input 0 of a 1756-IB16IF module.
- Rung 2 executes only once at the transition from Program to Run mode. It initializes LastInputTimestamp, which is used to detect a change of state on the input point by checking to see if the timestamp of the input data has changed. This rung also clears the output module's TimestampOffset bit to disable its Time Step Compensation algorithm.



- Rung 3 is the main rung that checks for a change of state on the input point by comparing the current input timestamp with the last timestamp (LastInputTimestamp).

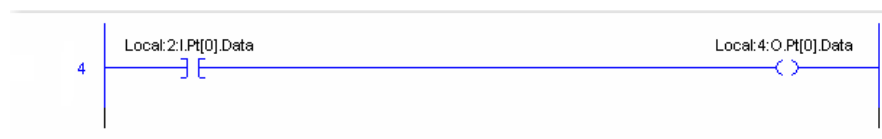


The input point (point 0) must have Change of State enabled. Otherwise, the timestamp does not update when the point transitions.

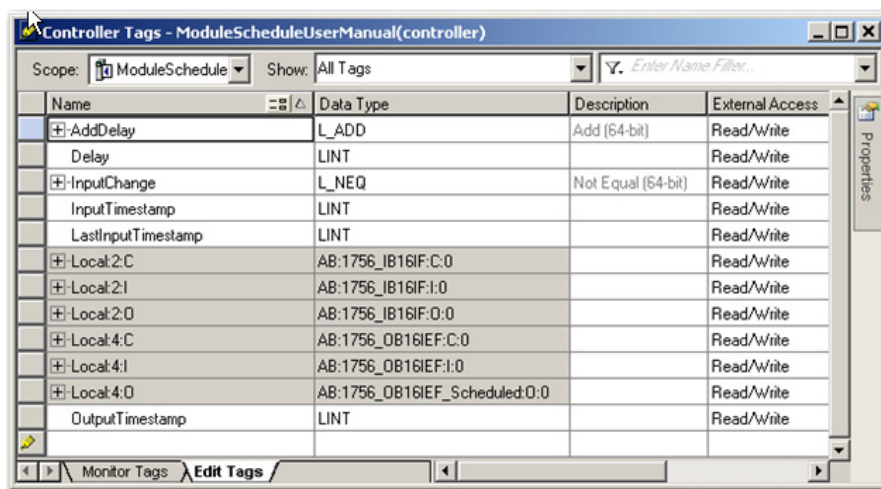
Once Change of State has been detected, the value in the Delay tag is added to the input timestamp and sent to the output module's timestamp using a COP instruction. This causes the output module to apply its output at a time equal to the time that the input changed state plus the Delay time.

The final COP instruction updates LastInputTimestamp in preparation for the next change of state.

- Rung 4 is the standard XIC-OTE rung that controls the output point based on the input point. The only difference is that the output module is configured for scheduled outputs. The outputs are not applied until the scheduled time has occurred.



The Controller Tags dialog box shows examples of the tags created in ladder logic.

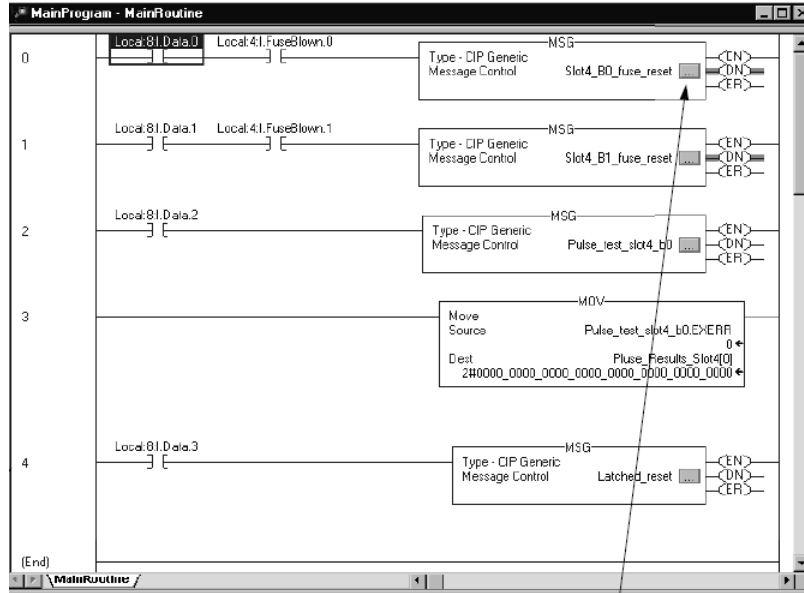


## Reset a Fuse, Perform Pulse Test, and Reset Latched Diagnostics

If an electronic fuse in a digital I/O module is shorted, a red LED appears on the front of the module and the FuseBlown bits for the output group, in the local input tag, will become true.

Once the short has been fixed, you can cycle power to the module with an electronic fuse via turning off the chassis power or re-seating the module itself, or you can use ladder logic to reset the electronic fuse.

This ladder logic program shows how to use ladder logic to reset an electronic fuse for a faulted point, perform a pulse test, and to reset latched diagnostics.



Click the box in each rung to see the associated configuration and communication.

The rungs perform these functions:

- Rungs 0 and 1 are used to perform a reset fuse service on Bits 0 and 1, respectively. The example is of a 1756-0A8D module in slot 4.
- Rung 2 performs a pulse test service to slot 4.
- Rung 3 moves the results of the pulse test to a data storage location. (The actual results appear in the message instruction tags under the tag name EXERR).
- Rung 4 performs a reset latched diagnostics service to slot 4. This example shows an output module.

The Controller Tags dialog box shows examples of the tags created in the ladder logic, as displayed in the tag editor.

P	Tag Name	Alias For	Base Tag	Type	Style	Description
	Slot4_B1_fuse_reset			MESSAGE		
	Slot4_B0_fuse_reset			MESSAGE		
<input type="checkbox"/>	reset_sl4_b1			DINT	Binary	
<input type="checkbox"/>	reset_sl4_b0			DINT	Decimal	
	Pulse_test_slot4_b0			MESSAGE		
<input type="checkbox"/>	pulse_sl4_b0			DINT[5]	Binary	
<input type="checkbox"/>	pulse_results_sl4			DINT[1]	Decimal	
<input type="checkbox"/>	Pulse_Results_Slot4			DINT[1]	Binary	
	Local:9:I			AB:1756_DI_DC_...		
	Local:9:C			AB:1756_DI_DC_...		
	Local:8:I			AB:1756_DI_Time...		
	Local:8:C			AB:1756_DI:C:0		
	Local:7:O			AB:1756_DO:0:0		

## Perform a WHO to Retrieve Module Identification and Status

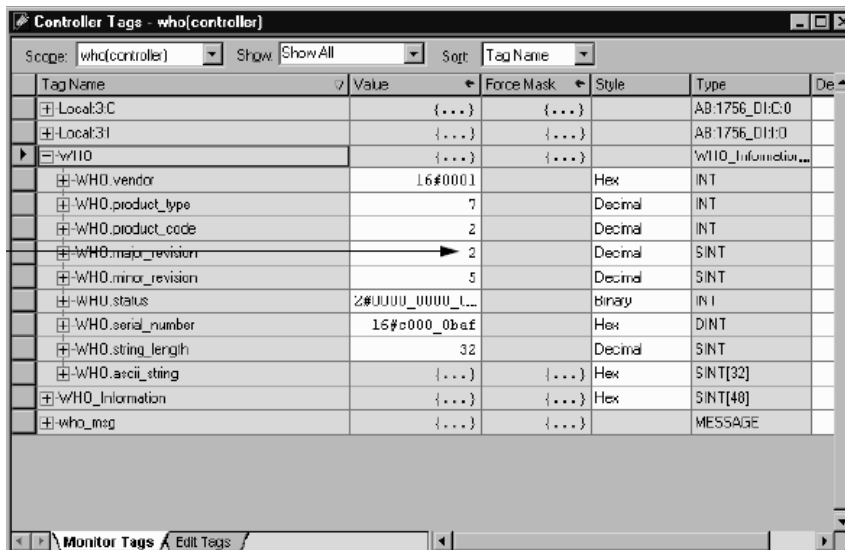
This ladder logic example shows how to retrieve module identification and status through a WHO service. A message instruction retrieves this module identification information:

- Product type
- Product code
- Major revision
- Minor revision
- Status
- Vendor
- Serial number
- String length
- ASCII string

An explanation of each identification category is provided after the ladder logic application.

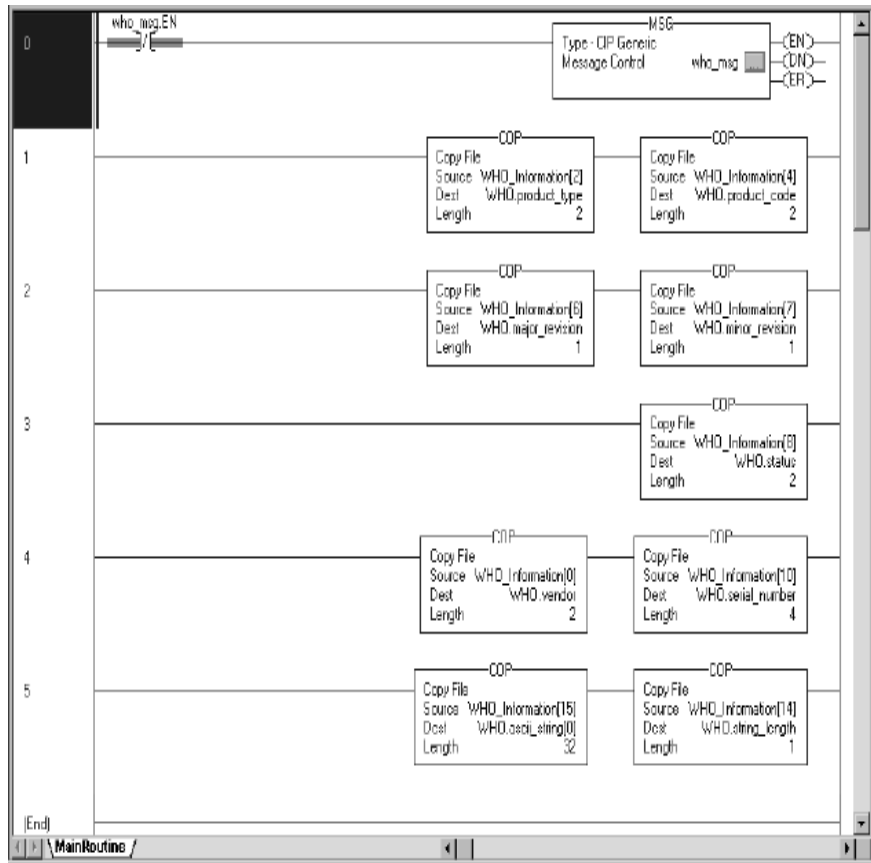
**IMPORTANT** The ladder logic example in this section uses a user-defined WHO data structure and a series of Copy instructions (following the Message instruction in the screen capture) to make the module identification information more easily understood.

The user-defined WHO data structure displays module identification information. For example, the Controller Tags dialog box shows the module's major revision is 2.



You do not have to create the user-defined data structure. If you choose not to create this structure, you can use the ASCII string and String length to retrieve and understand module identification through some interface excluding the Logix Designer application.

The illustration shows an example WHO ladder logic application.



The rungs perform these functions:

- Rung 0 constantly polls the module for WHO status. To conserve bandwidth, only poll for status when necessary.
- Rung 1 extracts the product type and catalog code.
- Rung 2 extracts the module’s major and minor revisions.
- Rung 3 extracts the module’s status information.
- Rung 4 extracts the vendor ID and serial number.
- Rung 5 extracts the module’s ASCII text string and the length of the text string in bytes.

This table defines the values returned for each rung.

**Table 70 - Rung Values**

Rung	Module ID Retrieved	Description
1	Product Type Catalog Code	Module's product type, 7=Digital I/O, 10=Analog I/O Module's catalog number
2	Major Revision Minor Revision	Module's major revision Module's minor revision
3	Status	Module's status. Multiple bits listed. Bit 0: 0 = Unowned, 1 = Owned Bit 1: Reserved Bit 2: 0 = Unconfigured, 1 = Configured Bit 3: Reserved Bits 7-4: Forms a 4-bit number indicating Device-Specific Status. 0 = Self-Test 1 = Flash update in progress 2 = Communications fault 3 = Not owned (outputs in Program mode) 4 = Unused 5 = Internal fault (need flash update) 6 = Run mode 7 = Program mode (output mods only) Bit 8: 0 = No fault, 1 = Minor recoverable fault Bit 9: 0 = No fault, 1 = Minor recoverable fault Bit 10: 0 = No fault, 1 = Minor recoverable fault Bit 11: 0 = No fault, 1 = major nonrecoverable fault Bits 15...12: Unused
4	Vendor ID Serial Number	Module manufacturer vendor, 1 = Allen-Bradley Module serial number
5	Length of ASCII Text String ASCII Text String	Number of characters in module's text string Module's ASCII text string description

## Review of Tags in Ladder Logic

When you use tags in ladder logic applications, remember these guidelines:

- Ladder logic tags represent the module on a **point per bit basis**. For example, point 0 = bit 0 on the module.
- If you are **performing a service** through the tags, a value of 0 helps prevent the action from occurring, and a value of 1 causes the action to occur. For example, if you want to reset the electronic fuse on a particular bit, enter 1 in the tags.
- If you are checking the **response of a service** through the tags, a value of 0 means the bit passed the service, and a value of 1 means the bit failed the service. For example, if you perform a pulse test and the response displays a 0 for a particular bit, the bit passed the test.

**Notes:**

## Choose a Correct Power Supply

Use the chart to determine the power your ControlLogix® chassis is using to help prevent an inadequate power supply. We recommend that you use this worksheet to check the power supply of each ControlLogix chassis used.

Slot Number	Module Cat. No.	Current @ 5.1V DC (mA)		Power @ 5.1V DC (Watts)	Current @ 24V DC (mA)		Power @ 24V DC (Watts)	Current @ 3.3V DC (mA)		Power @ 3.3V DC (Watts)
0			x 5.1V =			x 24V =			x 3.3V =	
1			x 5.1V =			x 24V =			x 3.3V =	
2			x 5.1V =			x 24V =			x 3.3V =	
3			x 5.1V =			x 24V =			x 3.3V =	
4			x 5.1V =			x 24V =			x 3.3V =	
5			x 5.1V =			x 24V =			x 3.3V =	
6			x 5.1V =			x 24V =			x 3.3V =	
7			x 5.1V =			x 24V =			x 3.3V =	
8			x 5.1V =			x 24V =			x 3.3V =	
9			x 5.1V =			x 24V =			x 3.3V =	
10			x 5.1V =			x 24V =			x 3.3V =	
11			x 5.1V =			x 24V =			x 3.3V =	
12			x 5.1V =			x 24V =			x 3.3V =	
13			x 5.1V =			x 24V =			x 3.3V =	
14			x 5.1V =			x 24V =			x 3.3V =	
15			x 5.1V =			x 24V =			x 3.3V =	
16			x 5.1V =			x 24V =			x 3.3V =	
	Totals	mA		W (1)	mA		W (2)	mA		W (3)
		This number cannot exceed the following: 10,000 mA for 1756-PA72, 1756-PB72 13,000 mA for 1756-PA75, 1756PB75, 1756-PC75, 1756-PH75			This number cannot exceed 2800 mA			This number cannot exceed 4000 mA		
				These three wattage values (1, 2, 3), added together, cannot exceed 75 W @ 60 °C (140 °F) for any power supply.						

**Notes:**

## Motor Starters for Digital I/O Modules

This appendix provides data to help you choose a ControlLogix® digital I/O module to drive Bulletin 500 series motor starters in your application. The tables list the number of motor starters (five sizes are listed for each module) that a particular digital I/O module can drive.

**IMPORTANT** When using the tables, remember that the supply voltage for each module must not drop below the minimum state motor starter supply voltage.

**Table 71 - Maximum Allowed 2-3 Pole Motor Starters (120V AC/60 Hz)**

Cat. No.	Motor Starters				
	Size 0...1	Size 2	Size 3	Size 4	Size 5
1756-OA16I	16	15 @ 30 °C (86 °F) 12 @ 60 °C (140 °F)	13 @ 30 °C (86 °F) 10 @ 60 °C (140 °F)	8 @ 30 °C (86 °F) 6 @ 60 °C (140 °F)	5 @ 30 °C (86 °F) 4 @ 60 °C (140 °F)
1756-OA16	16	14 (only 7 per group)	4 (Only 2 per group)	None	None
1756-OA8	8	8	8	8 @ 30 °C (86 °F) 6 @ 60 °C (140 °F)	5 @ 30 °C (86 °F) 4 @ 60 °C (140 °F)
1756-OA8D	8	8	8	None	None
1756-OA8E	8	8	8	6 (only 3 per group)	6 @ 30 °C (86 °F) (only 3 per group) 4 @ 60 °C (140 °F) (only 2 per group)

**Table 72 - Maximum Allowed 2-3 Pole Motor Starters (230V AC/60 Hz)**

Cat. No.	Motor Starters				
	Size 0-1	Size 2	Size 3	Size 4	Size 5
1756-OA16I	16	16	16	16 @ 30 °C (86 °F) 13 @ 60 °C (140 °F)	11 @ 30 °C (86 °F) 9 @ 60 °C (140 °F)
1756-OA16	16	16	16	4 (only 2 per group)	2 (only 1 per group)
1756-OA8	8	8	8	8	8

**Table 73 - Maximum Allowed 2-3 Pole Motor Starters (24V AC/60 Hz)**

Cat. No.	Motor Starters				
	Size 0-1	Size 2	Size 3	Size 4	Size 5
1756-ON8	4 @ 30 °C (86 °F) 3 @ 60 °C (140 °F)	4 @ 30 °C (86 °F) 3 @ 60 °C (140 °F)	None	None	None

## Determine the Maximum Number of Motor Starters

To determine the maximum number of motor starters that can be used by any 1756 digital I/O module, refer to this example.

Table 74 - Number of Motor Starters to be Used

Step	Value used in this example
1. Choose your motor starter.	Allen-Bradley® Bulletin 500 Size 3 120V AC/60 Hz/2-3 Poles. Inrush 1225VA, Sealed=45VA
2. Determine the number of motor starters required for your application.	11 size 3 motor starters
3. Choose a ControlLogix digital output module.	1756-0A16I <ul style="list-style-type: none"> <li>• Output voltage = 74...265V AC</li> <li>• Output steady state current per point = 2 A maximum @ 30 °C (86 °F) &amp; 1 A maximum @ 60 °C (140 °F) Linear derating</li> <li>• Output steady state current per module = 5 A maximum @ 30 °C (86 °F) &amp; 4 A maximum @ 60 °C (linear derating)</li> <li>• Output surge current p= 20 A maximum for 43 ms repeatable every 2 s @ 60 °C (140 °F)</li> </ul>
4. Determine the maximum environmental operating temperature.	50 °C (122 °F)
5. Confirm that the voltage range is within the motor starter range.	Motor starter uses 120V AC 1756-0A16I operates in a 74...120V AC voltage range
6. Confirm the inrush current per point.	Inrush of motor starter - Line voltage = Inrush current = 1225VA/120V AC = 10.2 A Inrush
7. Confirm the steady state point current of the module can drive the motor starter.	Sealed/Line voltage = Steady state current = 45VA/120V AC = 0.375 A @ 50 °C (122 °F) Output point current can drive: 2 A - (.033 A x 20 °C) = 2 A - 0.66 A = 1.34 A @ 50 °C (122 °F) Above 30 °C (86 °F), output point derates to 0.033 mA/°C (point derating) The 1756-0A16I output point current (1.34 A) can drive the motor starter (0.375 A @ 50 °C (122 °F))
8. Confirm that the 1756-0A16I/A total module current can drive 11 size 3 motor starters @ 50 °C (122 °F).	Motor starter steady state current x 11 motor starters = 0.375 x 11 = 4.125 A @ 50 °C (122 °F) The output total module current can drive: 5 A - (.033 A x 20 °C) = 5 A - 0.66 A = 4.34 A @ 50 °C (122 °F) Above 30 °C (86 °F) total output current derates to 0.033 mA/°C (Module derating) The 1756-0A16I total output current (4.34 A) can drive the 11 motor starters (4.125 A) @ 50 °C (122 °F)

## Major Revision Upgrades

Topic	Page
If Using a Compatible or Disabled Keying I/O Configuration	163
If Using an Exact Match Keying Configuration	163

Except for fast digital I/O modules (catalog numbers 1756-IB16IF, 1756-OB16IEF, and 1756-OB16IEFS), ControlLogix® 1756 digital I/O modules are transitioning to use a new, internal backplane Application Specific Integrated Circuits (ASIC) chip. As a result, the Major Revision number for these modules has also been upgraded. Digital I/O modules with the new ASIC have Major Revision 3.x.

**IMPORTANT** Do not downgrade your module's firmware from firmware revision 3.x to 2.x. Attempting to downgrade or downgrade a module's firmware from 3.x to 2.x will irreversibly damage the module. You must return modules that are damaged by an attempt to downgrade to firmware 2.x to Rockwell Automation.

Modules with the new internal backplane ASIC are form-fit, functional equivalents to the 2.x modules.

You can use Major Revision 3.x modules as direct replacements for Major Revision 2.x modules in these cases:

- The electronic keying of the module is specified as Compatible or Disabled Keying.
- The electronic keying of the module is Exact Keying, then additional steps are required. See [page 163](#) for details.

The use of the upgraded ASIC also impacts the firmware revisions that can be update upgraded to the module. Digital I/O modules at Major Revision 3.x cannot be backflashed to any 2.x firmware revision. Digital I/O modules at firmware revision 2.x cannot be update upgraded to any firmware revision 3.x.

### If Using a Compatible or Disabled Keying I/O Configuration

If you are replacing a 2.x module with a 3.x module and have configured the 2.x module to use Compatible or Disabled Keying, further steps are not required.

If you use Compatible or Disabled Keying configurations, 3.x modules can be used as a direct replacement for 2.x modules.

### If Using an Exact Match Keying Configuration

If you are currently using a 2.x module that is configured at Exact Match keying, consider changing the module's electronic keying in the I/O configuration to Compatible or Disabled Keying.

If you are replacing a 2.x module with a 3.x module and must use Exact Match keying in the I/O configuration, take additional action depending on your version of the programming software.

If you use Exact Match Keying and	Then Do this
RSLogix 5000® software, version 13.04.00 and later	<ol style="list-style-type: none"> <li>1. Delete the 2.x module from the I/O configuration in the project.</li> <li>2. Add a new 3.x revision module to the I/O configuration.</li> </ol>
RSLogix 5000 software, version 12.06.00 and earlier	Do one of the following: <ul style="list-style-type: none"> <li>• Change the module's configuration to Disable Keying.</li> <li>• Upgrade the software to version 13.04.00 or later and complete the steps that are listed for RSLogix™ software, version 13.04.00 or later.</li> </ul>

**Notes:**

## 1492 IFMs for Digital I/O Modules

### Cable Overview

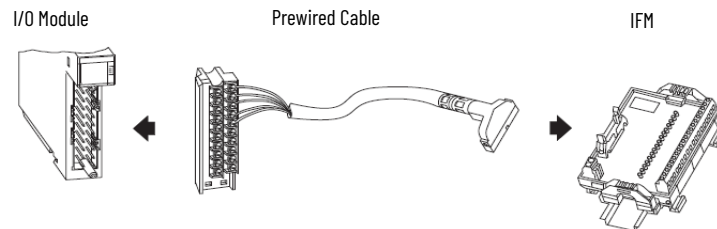
As an alternative to buying RTBs and connecting the wires yourself, you can buy a wiring system that connects to I/O modules through prewired and pretested cables.

#### IMPORTANT

The ControlLogix® system has been agency certified using only the ControlLogix RTB catalog numbers 1756-TBCH, 1756-TBCHXT, 1756-TBNH, 1756-TBNHXT, 1756-TBSH, 1756-TBSHXT, 1756-TBS6H, and 1756-TBS6HXT. Any application that requires agency certification of the ControlLogix system using other wiring termination methods may require application-specific approval by the certifying agency.

The combinations include the following:

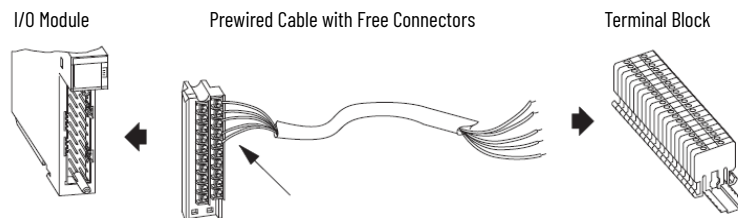
- **Interface modules (IFMs)** mount on DIN rails to provide the output terminal blocks for the I/O module. Use the IFMs with the prewired cables that match the I/O module to the interface module.



- **Prewired cables** are individually color-coded conductors that connect to a standard terminal block. The other end of the cable assembly is an RTB that plugs into the front of the I/O module. All of the prewired cables use  $0.326 \text{ mm}^2$  (22 AWG) wire.

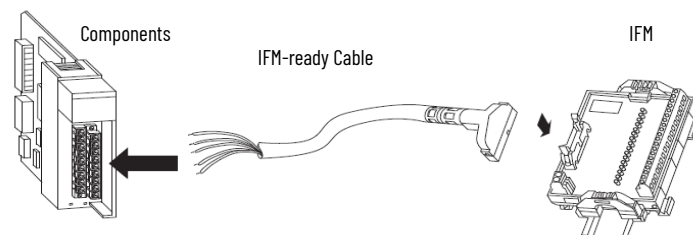
Additional prewired cable combinations include the following:

- **Digital I/O module-ready cables with free connectors** wire into standard terminal blocks or other type of connectors. The other end of the cable assembly is an RTB that plugs into the front of the I/O module.



Most of the I/O module-ready cables use  $0.823 \text{ mm}^2$  (18 AWG) conductors for higher current applications or longer cable runs.

- **IFM-ready cables** have a cable connection to attach to the IFM prewired to one end. The other end has free connectors to wire to I/O modules or other components.



The IFM-ready cables use  $0.326 \text{ mm}^2$  (22 AWG) wire.

This table lists the IFMs and prewired cables to be used with ControlLogix digital I/O modules.

**IMPORTANT** For the latest list, see the Digital/Analog Programmable Controller Wiring Systems Technical Data, publication [1492-TD008](#).

**IMPORTANT** IFMs and prewired cables are not rated for harsh environments (corrosive atmosphere, extended temperature, etc.). When they are used in a system with products that are rated for harsh environments, the system is derated to the specification values of the IFMs and cables.

**Table 75 - IFMs and Prewired Cables**

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable
1756-IA8D, 1756-IA8DK	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20D120	Status-indicating	Standard with 120V AC/DC status indicators <sup>(1)</sup>	
	1492-IFM20D120N		Narrow standard with 120V AC status indicators	
	1492-IFM20D120A-2		120V AC with extra terminals for inputs	
	1492-IFM20F-FS120A-4	Fusible	Two 4-point isolated groups with four terminals per input and 120V AC/DC blown fuse indicators	
1756-IA16, 1756-IA16K	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20F-3		3-wire sensor type input devices	
	1492-IFM20D120	Status-indicating	Standard with 120V AC/DC status indicators <sup>(1)</sup>	
	1492-IFM20D120N		Narrow standard with 120V AC status indicators	
	1492-IFM20D120A-2		120V AC with extra terminals for inputs	
1492-IFM20F-F120A-2	Fusible	Extra terminals with 120V AC/DC blown fuse status indicators.		
1756-IA16I, 1756-IA16IK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS120A-4	Fusible	Isolated with 120V AC status indicators and four terminals per input	
	1492-IFM40F-FSA-4		Isolated 120V AC/DC with four terminals per input	
	1492-IFM40F-FS120A-4		Isolated with 120V AC/DC blown fuse indicators and four terminals per input.	
1756-IA32, 1756-IA32K	1492-IFM40F	Feed-through	Standard	1492-CABLExZ (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40D120A-2	Status-indicating	120V AC status indicators and extra terminals for inputs	
1756-IB16, 1756-IB16K	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20F-3		3-wire sensor type input devices	
	1492-IFM20D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM20D24N		Narrow standard with 24V AC/DC status indicators	
	1492-IFM20D24A-2		24V AC/DC status indicators and extra terminals for inputs	
	1492-IFM20D24-3		3-wire sensor with 24V AC/DC status indicators	
1492-IFM20F-F24A-2	Fusible	Extra terminals with 24V AC/DC blown fuse indicators for inputs		
1756-IB16D, 1756-IB16DK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40DS24A-4	Status-indicating	Isolated with 24V AC/DC status indicators and four terminals per input	
	1492-IFM40F-F24AD-4	Fusible	Fused with 24V DC blown fuse low leakage indicators, four isolated groups, and four terminals per input	
	1492-IFM40F-FS24A-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per input <sup>(2)</sup>	
	1492-IFM40F-FSA-4		Isolated with 120V AC/DC with four terminals per input	

Table 75 - IFMs and Prewired Cables (Continued)

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable
1756-IB16I, 1756-IB16IK, 1756-IB16IF, 1756-IB16IFK, 1756-IB16ISOE, 1756-IB16ISOEK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24A-4	Status-indicating	Isolated with 24V AC/DC status indicators and four terminals per input	
	1492-IFM40F-FS24A-4	Fusible	Isolated with 24V AC/DC blown fuse indicators and four terminals per input	
	1492-IFM40F-FSA-4		Isolated with 120V AC/DC with four terminals per input	
1756-IB32, 1756-IB32K	1492-IFM40F	Feed-through	Standard	1492-CABLExZ (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40F-3		3-wire sensor type input devices	
	1492-IFM40D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM40D24A-2		24V AC/DC status indicators and extra terminals for inputs	
	1492-IFM40D24-3		3-wire sensor with 24V AC/DC status indicators for inputs	
1756-IC16	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20F-3		3-wire sensor type input devices	
1756-IG16, 1756-IG16K	N/A			
1756-IH16I, 1756-IH16IK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40F-FSA-4	Fusible	Isolated with 120V AC/DC with four terminals per input	
	1492-IFM40F-FS120A-4		Isolated with 120V AC/DC blown fuse indicators with four terminals per input	
1756-IM16I, 1756-IM16IK	1492-IFM40DS240A-4	Status-indicating	Isolated with 240V AC status indicators and four terminals per input	1492-CABLExY (x=cable length)
	1492-IFM40F-FS240A-4	Fusible	Isolated with 240V AC/DC blown fuse indicators and four terminals per input	
1756-IN16, 1756-IN16K	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20F-3		3-wire sensor type input devices	
	1492-IFM20D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM20D24N		Narrow standard with 24V AC/DC status indicators	
	1492-IFM20D24A-2		24V AC/DC status indicators and extra terminals for inputs	
	1492-IFM20D24-3		3-wire sensor with 24V AC/DC status indicators	
1492-IFM20F-F24A-2	Fusible	Extra terminals with 24V AC/DC blown fuse indicators for inputs		
1756-IV16, 1756-IV16K	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20F-3		3-wire sensor type input devices	
	1492-IFM20D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM20D24N		Narrow standard with 24V AC/DC status indicators	
	1492-IFM20D24A-2		24V AC/DC status indicators and extra terminals for inputs	
	1492-IFM20D24-3		3-wire sensor with 24V AC/DC status indicators	
1756-IV32, 1756-IV32K	1492-IFM40F	Feed-through	Standard	1492-CABLExZ (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40F-3		3-wire sensor type input devices	
	1492-IFM40D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM40D24A-2		24V AC/DC status indicators	
	1492-IFM20D24-2		24V AC/DC status indicators and extra terminals for inputs	
1492-IFM20D24-3		3-wire sensor with 24V AC/DC status indicators		

Table 75 - IFMs and Prewired Cables (Continued)

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable	
1756-0A8, 1756-0A8K	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)	
	1492-IFM20FN		Narrow standard		
	1492-IFM20F-2		Extra terminals		
	1756-0A8D	1492-IFM20DS120-4	Status-indicating	Isolated with 120V AC status indicators and four terminals per output	1492-CABLExW (x=cable length)
		1492-IFM20F-FS-2	Fusible	Isolated with 120V AC/DC with extra terminals for outputs	
		1492-IFM20F-FS120-2		Isolated with extra terminals with 120V AC/DC blown fuse indicators for outputs	
		1492-IFM20F-FS120-4		Isolated with four terminals with 120V AC blown fuse indicators for outputs	
1492-IFM20F-FS240-4	Isolated with four terminals with 240V AC/DC blown fuse indicators for outputs				
1756-0A8E, 1756-0A8EK	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)	
	1492-IFM20FN		Narrow standard		
	1492-IFM20F-2		Extra terminals		
	1756-0A16, 1756-0A16K	1492-IFM20DS120-4	Status-indicating	Isolated with 120V AC status indicators and four terminals per output	1492-CABLExV (x=cable length)
		1492-IFM20F-FS-2	Fusible	Isolated 120V AC/DC with extra terminals for outputs	
		1492-IFM20F-FS120-2		Isolated with extra terminals with 120V AC/DC blown fuse indicators	
		1492-IFM20F-FS120-4		Isolated with four terminals per output and 120V AC/DC blown fuse indicators	
1756-0A16, 1756-0A16K	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)	
	1492-IFM20FN		Narrow standard		
	1492-IFM20F-2		Extra terminals		
	1492-IFM20D120N	Status-indicating	Narrow standard with 120V AC status indicators		
	1492-IFM20D120-2		120V AC/DC status indicators and extra terminals for outputs		
	1492-IFM20F-F2	Fusible	Extra terminals for outputs		
	1492-IFM20F-F120-2		Extra terminals with 120V AC blown fuse indicators for outputs		
	1492-IFM20F-F240-2		Extra terminals with 240V AC blown fuse indicators for outputs		
	1492-XIM20120-8R	Relay Master	20-pin master with eight, 24V DC relays <sup>(3)</sup>		
	1492-XIM20120-16R		20-pin master with 16, 120V AC relays		
	1492-XIM20120-16RF		20-pin master with 16, 120V AC relays with fusing		
	1492-XIM120-8R	Relay Expander	Expander with eight, 120V AC relays <sup>(4)</sup>		
	1492-XIMF-F120-2	Fusible Expander	Expander with eight, 120V channels with blown fuse indicators <sup>(4)</sup>		
1492-XIMF-2	Feed-through Expander	Expander with eight feed-through channels <sup>(4)</sup>			
1756-0A16I, 1756-0A16IK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)	
	1492-IFM40DS120-4	Status-indicating	Isolated with 120V AC status indicators and four terminals per output		
	1492-IFM40-FS-2	Fusible	Isolated with extra terminals for outputs		
	1492-IFM40-FS-4		Isolated 240V AC/DC with four terminals per output		
	1492-IFM40F-FS120-2		Isolated with extra terminals and 120V AC/DC blown fuse indicators		
	1492-IFM40F-FS120-4		Isolated with 120V AC/DC blown fuse indicators and four terminals per output		
1492-IFM40F-FS240-4	Isolated with 240V AC/DC blown fuse indicators and four terminals per output				
1756-0B8, 1756-0B8K	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)	
	1492-IFM20FN		Narrow standard		
	1492-IFM20F-2		Extra terminals		
	1492-IFM20DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	1492-CABLExW (x=cable length)	
	1492-IFM20F-FS-2	Fusible	Isolated 120V AC/DC with extra terminals for outputs		
1492-IFM20F-FS24-2	Isolated with extra terminals per output and 24V AC/DC blown fuse indicators				

Table 75 - IFMs and Prewired Cables (Continued)

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable
1756-OB8EI, 1756-OB8EIK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs	
	1492-IFM40F-FS24-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per output	
1492-IFM40F-FS-4	Isolated 240V AC/DC with four terminals per output			
1756-OB16D, 1756-OB16DK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40F-2	Feed-through	Extra terminals	
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output <sup>(5)</sup>	
	1492-IFM40F-F24D-2	Fusible	Fused with 24V DC blown fuse low leakage status indicator circuit with four isolated groups and four terminals per output	
	1492-IFM40F-FS-2		Isolated with extra terminals for 120V AC/DC outputs	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs <sup>(6)</sup>	
	1492-IFM40F-FS24-4		Isolated with extra terminals and 24V AC/DC blown fuse indicators and four terminals per output <sup>(6)</sup>	
1492-IFM40F-FS-4		Isolated 240V AC/DC with four terminals per output		
1756-OB16E, 1756-OB16EK	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM20D24N		Narrow standard with 24V AC/DC status indicators	
	1492-IFM20D24-2		24V AC/DC status indicators and extra terminals for outputs	
	1492-IFM20F-F2	Fusible	120V AC/DC with extra terminals for outputs	
	1492-IFM20F-F24-2		Extra terminals with 24V AC/DC blown fuse indicators	
	1492-XIM2024-8R	Relay Master	20-pin master with eight, 24V DC relays <sup>(7)</sup>	
	1492-XIM2024-16R		20-pin master with 16, 24V DC relays	
	1492-XIM2024-16RF		20-pin master with 16, 24V DC relays with fusing	
	1492-XIM24-8R	Relay Expander	Expander with eight, 24V DC relays <sup>(4)</sup>	
	1492-XIMF-F24-2	Fusible Expander	Expander with eight, 24V channels with blown fuse indicators <sup>(4)</sup>	
1492-XIMF-2	Feed-through Expander	Expander with eight feed-through channels <sup>(4)</sup>		
1756-OB16I, 1756-OB16IK, 1756-OB16IEF, 1756-OB16IEFK, 1756-OB16IEFS, 1756-OB16IEFSK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs <sup>(8)</sup>	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs <sup>(8)</sup>	
	1492-IFM40F-FS24-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per output <sup>(8)</sup>	
1492-IFM40F-FS-4	Isolated with 240V AC/DC and four terminals per output <sup>(8)</sup>			
1756-OB16IS	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs <sup>(8)</sup>	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs <sup>(8)</sup>	
	1492-IFM40F-FS24-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per output <sup>(8)</sup>	
1492-IFM40F-FS-4	Isolated with 240V AC/DC and four terminals per output <sup>(8)</sup>			

Table 75 - IFMs and Prewired Cables (Continued)

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable
1756-OB32, 1756-OB32K	1492-IFM40F	Feed-through	Standard	1492-CABLExZ (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM40D24-2		24V AC/DC status indicators and extra terminals for outputs	
	1492-IFM40F-F2	Fusible	120V AC/DC with extra terminals for outputs	
	1492-IFM40F-F24-2		Extra terminals with 24V AC/DC blown fuse indicators for outputs	
	1492-XIM4024-8R	Relay Master	40-pin master with eight, 24V DC relays	
	1492-XIM4024-16R		40-pin master with 16, 24V DC relays	
	1492-XIM4024-16RF		40-pin master with 16, 24V DC relays with fusing	
	1492-XIM24-8R	Relay Expander	Expander with eight, 24V DC relays <sup>(4)</sup>	
	1492-XIMF-F24-2	Fusible Expander	Eight-channel expander with 24V AC blown fuse indicators <sup>(4)</sup>	
	1492-XIM24-16RF		Expander with 16, 24V DC relays with fusing <sup>(9)</sup>	
1492-XIMF-2	Feed-through Expander	Expander with eight feed-through channels <sup>(4)</sup>		
1756-OC8, 1756-OC8K	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	1492-CABLExW (x=cable length)
	1492-IFM20F-FS2	Fusible	Isolated 120V AC/DC with extra terminals for outputs	
1492-IFM20F-FS24-2	Isolated with extra terminals per output and 24V AC/DC blown fuse indicators			
1756-OG16, 1756-OG16K	N/A			
1756-OH81	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs	
	1492-IFM40F-FS120-2		Isolated with extra terminals and 120V AC/DC blown fuse indicators	
1756-ON8, 1756-ON8K	1492-IFM20F	Feed-through	Standard	1492-CABLExU (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	1492-CABLExW (x=cable length)
	1492-IFM20F-FS2	Fusible	Isolated 120V AC/DC with extra terminals for output	
1492-IFM20F-FS24-2	Isolated with extra terminals per output and 24V AC/DC blown fuse indicators			
1756-OV16E	1492-IFM20F	Feed-through	Standard	1492-CABLExX (x=cable length)
	1492-IFM20FN		Narrow standard	
	1492-IFM20F-2		Extra terminals	
	1492-IFM20D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM20D24-2		24V AC/DC status indicators and extra terminals for outputs	
	1492-IFM20F-F2	Fusible	120V AC/DC with extra terminals for outputs	
1492-IFM20F-F24-2	Extra terminals with 24V AC/DC blown fuse indicators			
1756-OV32E, 1756-OV32EK	1492-IFM40F	Feed-through	Standard	1492-CABLExZ (x=cable length)
	1492-IFM40F-2		Extra terminals	
	1492-IFM40D24	Status-indicating	Standard with 24V AC/DC status indicators	
	1492-IFM40D24-2		24V AC/DC status indicators and extra terminals for outputs	
	1492-IFM40F-F2	Fusible	120V AC/DC with extra terminals for outputs	
1492-IFM40F-F24-2	Extra terminals with 24V AC/DC blown fuse indicators for outputs			

Table 75 - IFMs and Prewired Cables (Continued)

I/O Cat. No.	IFM Cat. No.	IFM Type	IFM Description	Prewired Cable
1756-0W16I, 1756-0W16IK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	
	1492-IFM40DS120-4		Isolated with 120V AC status indicators and four terminals per output	
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs	
	1492-IMF40F-FS24-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per output	
	1492-IFM40F-FS-4		Isolated 240V AC/DC with four terminals per output	
	1492-IMF40F-FS120-2		Isolated with extra terminals and 120V AC blown fuse indicators	
	1492-IMF40F-FS120-4		Isolated with 120V AC/DC blown fuse indicators and four terminals per output	
1492-IMF40F-FS240-4	Isolated with 240V AC/DC blown fuse indicators and four terminals per output			
1756-0X8I, 1756-0X8IK	1492-IFM40F	Feed-through	Standard	1492-CABLExY (x=cable length)
	1492-IFM40DS24-4	Status-indicating	Isolated with 24/48V AC/DC status indicators and four terminals per output	
	1492-IFM40DS120-4		Isolated with 120V AC status indicators and four terminals per output	
	1492-IFM40F-FS-2	Fusible	Isolated with extra terminals for 120V AC/DC outputs	
	1492-IFM40F-FS24-2		Isolated with extra terminals and 24V AC/DC blown fuse indicators for outputs	
	1492-IMF40F-FS24-4		Isolated with 24V AC/DC blown fuse indicators and four terminals per output	
	1492-IFM40F-FS-4		Isolated 240V AC/DC with four terminals per output	
	1492-IMF40F-FS120-2		Isolated with extra terminals and 120V AC blown fuse indicators	
	1492-IMF40F-FS120-4		Isolated with 120V AC/DC blown fuse indicators and four terminals per output	
1492-IMF40F-FS240-4	Isolated with 240V AC/DC blown fuse indicators and four terminals per output			

- (1) This IFM is not recommended for use with I/O modules that have an off-state leakage current exceeding 0.5 mA. Use a 1492-IFM20D120N or 1492-IFM20D120A-2 module for inputs. Use a 1492-IFM20D120-2 module for outputs.
- (2) The 1492-IFM40F-FS24A-4 module and the 1492-CABLExY cable can be used with the 1756-IB16D module. However, due to the 1492-IFM40F-FS24A-4 module's blown fuse leakage current rating, the wire off diagnostic function of the 1756-IB16D module does not indicate a blown or removed fuse as a wire off condition. If you require this diagnostic to function for a blown or removed fuse, you must use a 1492-IFM40F-F24AD-4 module.
- (3) Expandable to 16 by using a XIM120-BR or XIMF-24-2 module.
- (4) Can have up to 1 expandable module depending upon master used (total 16 pts or less). Extender cable is provided.
- (5) IFMs status indicator provides output On/Off indication. Due to the magnitude of current through the status indicator, the 1756-OB16D module no load diagnostic function does not work. If this function is required, use the 1492-IFM40F-2 module.
- (6) The 1492-IFM40F-FS24-2 and 1492-IFM40F-FS24-4 modules and the 1492-CABLExY cable can be used with the 1756-OB16D module. However, due to the 1492-IFM40F-FS24-2 and 1492-IFM40F-FS24-4 module's blown fuse leakage current rating, the no load diagnostic function of the 1756-OB16D module does not indicate a blown or removed fuse as a no load condition. If you require this diagnostic to function for a blown or removed fuse, you must use a 1492-IFM40F-F24D-2 module.
- (7) Expandable to 16 by using a XIM24-8R or XIMF-24-2 module.
- (8) Do not use this module in Output Sinking mode with fused IFM modules. The IFM module fuses do not properly protect the circuit.
- (9) One 1492-XIM24-16RF module is to be used with one 1492-XIM4024-16R or 1492-XIM4024-16RF master (32 pt. only).

Tables 76 and 77 describe the prewired, module-ready cables and connectors available for your ControlLogix digital I/O modules.

Table 76 - Module-Ready Cables

Cat. No. <sup>(1)</sup>	No. of Conductors	Conductor Size	Nominal Outer Diameter	RTB at the I/O Module End
1492-CABLExU	20	0.326 mm <sup>2</sup> (22 AWG)	9.0 mm (0.36 in.)	1756-TBNH
1492-CABLExV				
1492-CABLExW				
1492-CABLExX				
1492-CABLExY	40		11.7 mm (0.46 in.)	1756-TBCH
1492-CABLExZ				

- (1) Cables are available in lengths of 0.5 m (1.6 ft), 1.0 m (3.3 ft), 2.5 m (8.2 ft), and 5.0 m (16.4 ft). To order, insert the code for the desired cable length into the catalog number in place of the x: 005=0.5 m (1.6 ft), 010=1.0 m (3.3 ft), 025=2.5 m (8.2 ft), 050=5 m (16.4 ft). Build-to-order cable lengths are also available.

Table 77 - Module Connectors

Cat. No. <sup>(1)</sup>	No. of Conductors	Conductor Size	Nominal Outer Diameter	RTB at the I/O Module End
1492-CABLExTBNH	20	0.823 mm <sup>2</sup> (18 AWG)	11.4 mm (0.45 in.)	1756-TBNH
1492-CABLExTBCH	40 <sup>(2)</sup>		14.1 mm (0.55 in.)	1756-TBCH

- (1) Cables are available in lengths of 0.5 m (1.6 ft), 1.0 m (3.3 ft), 2.5 m (8.2 ft), and 5.0 m (16.4 ft). To order, insert the code for the desired cable length into the catalog number in place of the x: 005=0.5 m (1.6 ft), 010=1.0 m (3.3 ft), 25=2.5 m (8.2 ft), 050=5 m (16.4 ft). Build-to-order cable lengths are also available.
- (2) Four conductors are not connected to the RTB.

**Notes:**

## Rack-optimized Connections

# ControlNet Communication

When a digital I/O module is in a remote chassis with respect to its owner-controller, you may be able to choose Rack Optimization or Listen-only Rack Optimization during module configuration. The option that you choose depends on the communication module configuration. If the communication module uses Listen-only Rack Optimization, then the I/O module must also use Listen-only Rack Optimization.

A rack-optimized connection economizes bandwidth between owner-controllers and digital I/O modules in the remote chassis. Rather than having several direct connections with individual RPI values, an owner-controller has a single rack connection with a single RPI value. That RPI value accommodates all digital I/O modules in the remote chassis.

---

**IMPORTANT**

Because rack-optimized connections are applicable only in applications that use a remote chassis, you must configure the communication format, as described in [Chapter 4](#), for both the remote I/O module and the remote 1756-CNB module or EtherNet/IP™ module.

Make sure you configure both modules for rack optimization. If you choose another communication format for each module, the controller makes two connections to the same chassis (one for each format) and the same data travels across the ControlNet® network.

If you use rack optimization for both modules, you preserve bandwidth and configure your system to operate more efficiently.

---

The input, or data echo, information is limited to general faults and data. No additional status, such as diagnostic information, is available.

---

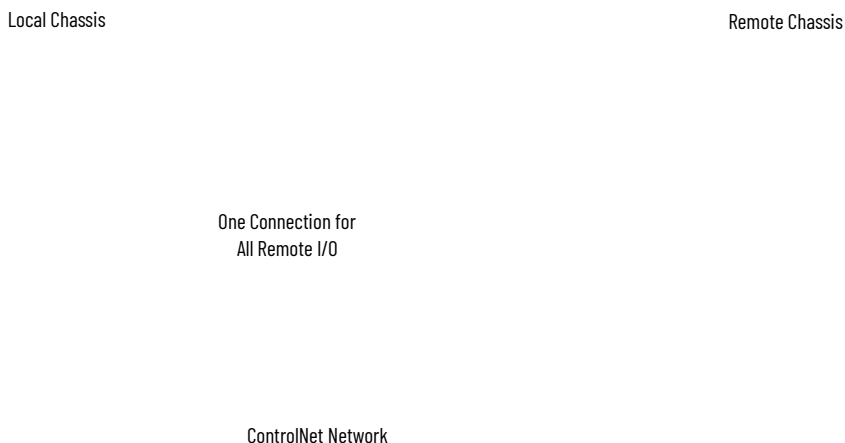
**IMPORTANT**

Each controller can establish connections, in any combination of direct or rack-optimized. In other words, you can use a rack-optimized connection between an owner-controller and multiple remote I/O modules while simultaneously using a direct connection between that same controller and any other I/O modules in the same remote chassis.

---

This figure shows how a rack-optimized connection eliminates the need for three separate connections. The owner-controller in the local chassis communicates with all the I/O modules in the remote chassis but uses only one connection. The ControlNet communication module sends data from the modules simultaneously at the RPI.

**Figure 25 - Rack-optimized Connection**



### Recommendations for Rack-optimized Connections

We recommend that you use a rack-optimized connection for these applications:

- Standard digital I/O modules
- Non-fused digital output modules
- Owner-controllers running low on connections

---

**IMPORTANT** Rack-optimized connections are available only to digital I/O modules. However, do not use a rack-optimized connection for diagnostic I/O modules or fused output modules. Diagnostic and fused output data is not transferred over a rack-optimized connection. This defeats the purpose of using those modules.

---

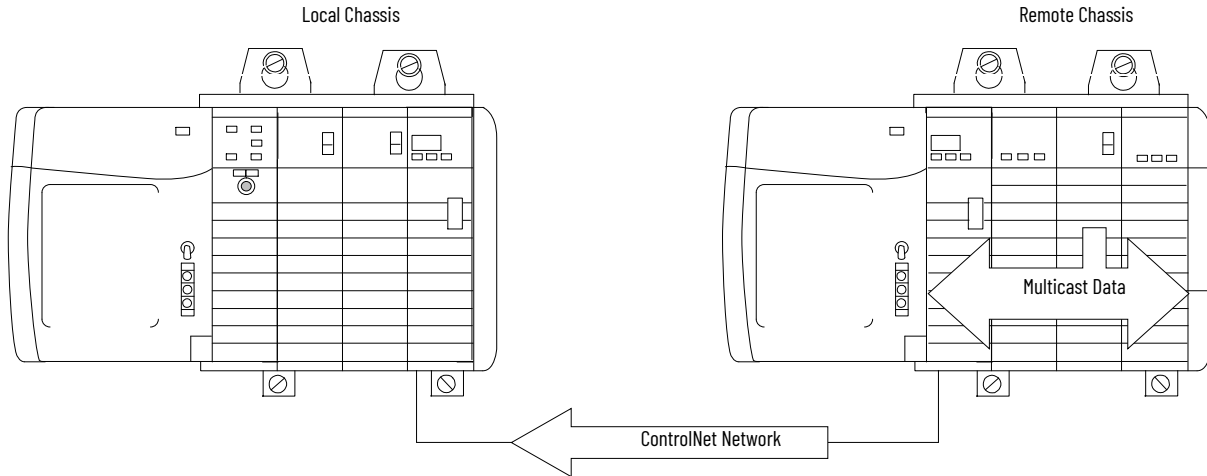
## Remote Input Modules Connected Via the ControlNet Network

When an RPI value is specified for an input module in a remote chassis that is connected by a scheduled ControlNet network, in addition to instructing the module to multicast data within its own chassis, the RPI also reserves a spot in the stream of data flowing across the ControlNet network.

The timing of this reserved spot may or may not coincide with the exact value of the RPI. But, the control system guarantees that the owner-controller receives data at least as often as the specified RPI.

As shown in this figure, the input data within the remote chassis is multicast at the configured RPI. The ControlNet communication module sends input data back to the owner-controller at least as often as the RPI.

Figure 26 - Remote Input Modules on ControlNet Network



The module RPI and reserved spot on the network are asynchronous to each other. This means that there are best and worst case scenarios as to when the owner-controller receives updated data from the module in a remote chassis.

### Best Case RPI Multicast Scenario

In the best case scenario, the module performs an RPI multicast with updated channel data just before the reserved network spot is made available. In this case, the remotely located owner receives the data almost immediately.

### Worst Case RPI Multicast Scenario

In the worst case scenario, the module performs an RPI multicast just after the reserved network slot has passed. In this case, the owner-controller does not receive data until the next available network slot.

**IMPORTANT** Enabling the COS feature on an input module in a remote chassis lets the module multicast data at both the RPI rate and when the input changes state. This helps to reduce the worst case time.

When selecting values for the remote module RPI, system throughput is optimized when its RPI value is a power of two times the current NUT running on the ControlNet network.

This table shows recommended RPI values for a system by using a NUT of 5 ms.

Table 78 - Recommended RPI Values for System by Using NUT of 5 ms

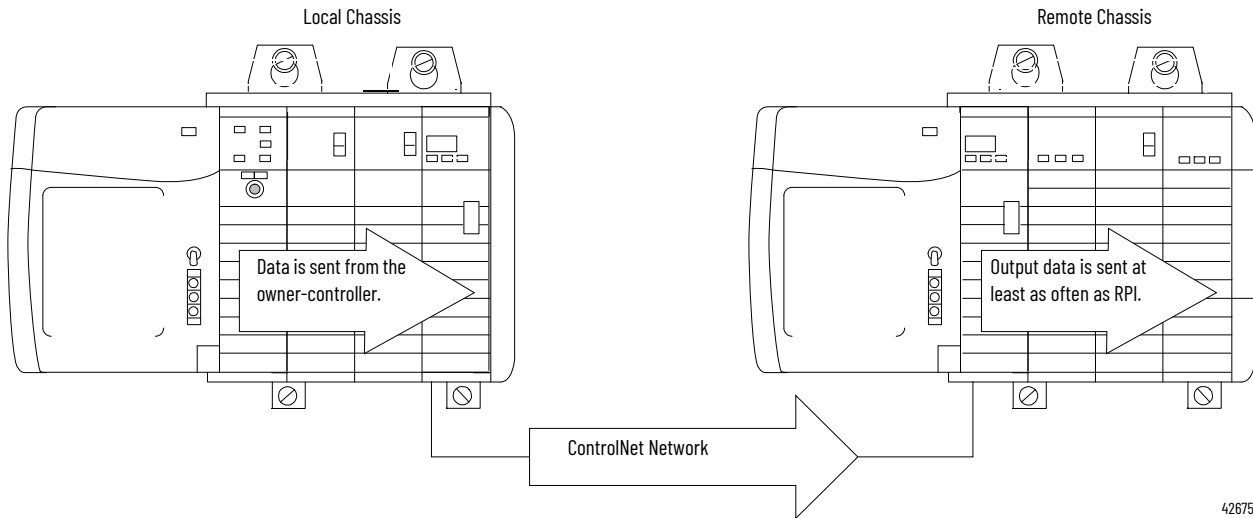
NUT=5 ms	x2 <sup>0</sup>	x2 <sup>1</sup>	x2 <sup>2</sup>	x2 <sup>3</sup>	x2 <sup>4</sup>	x2 <sup>5</sup>	x2 <sup>6</sup>	x2 <sup>7</sup>
Optimal RPI Values (ms)	5 ms	10 ms	20 ms	40 ms	80 ms	160 ms	320 ms	640 ms

## Remote Output Modules Connected Via the ControlNet Network

When an RPI value is specified for an output module in a remote chassis that is connected to the owner-controller by a scheduled ControlNet network, in addition to instructing the owner-controller to multicast the output data within its own chassis, the RPI also reserves a spot in the stream of data flowing across the ControlNet network.

The timing of this reserved spot may or may not coincide with the exact value of the RPI. But, the control system guarantees that the output module receives data **at least as often** as the specified RPI, as shown in this figure.

**Table 79 - Remote Output Modules on ControlNet Network**



The reserved spot on the network and the output data sent by the controller are asynchronous to each other. This means that there are best and worst-case scenarios as to when the owner-controller receives updated data from the module in a remote chassis.

### Best Case RPI Multicast Scenario

In the best case scenario, the owner-controller sends the output data just before the reserved network slot is made available. In this case, the remote output module receives the data almost immediately.

### Worst Case RPI Multicast Scenario

In the worst case scenario, the owner-controller sends the output data just after the reserved network slot has passed. In this case, the output module does not receive data until the next available network slot.

---

#### **IMPORTANT**

These best and worst-case scenarios indicate the time that is required for output data to transfer from the owner-controller to the module once the owner-controller has produced it. They do not take into account the user program time in the owner-controller.

The receipt of new data is a function of the length of the user program and its asynchronous relationship with the RPI.

The owner-controller updates remote output modules at the end of each task and at the RPI, as described earlier in this section, if your application uses these components:

- 1756-CNB/D or 1756-CNBR/D modules
  - Studio 5000 Logix Designer® application
-

## Use the System Clock to Timestamp Inputs and Schedule Outputs

The CST is a chassis-specific time that is not synchronized with, or in any way that is connected to, the time that is generated over the ControlNet network to establish a network update time (NUT).

## Create a New Module

If you plan to add the I/O module to a remote chassis, add ControlNet modules to both the local and remote chassis in the I/O Configuration tree.

- For more information on ControlLogix® ControlNet modules, see ControlNet Modules in Logix5000™ Control Systems User Manual, publication CNET-UM001.

**Notes:**

## History of Changes

This appendix contains the new or updated information for each revision of this publication. These lists include substantive updates only and are not intended to reflect all changes. Translated versions are not always available for each revision.

### Change Log

#### 1756-UM058J-EN-P, September 2023

---

##### Change

Moved all ControlNet<sup>®</sup>-related content to Appendix G

Added Inclusive Language Statement

Removed Catalog Number 1756-OB8I

Added Catalog Numbers:

1756-IB16K, 1756-IB16DK, 1756-IB16IK, 1756-IB16IFK, 1756-IB32K, 1756-IG16K, 1756-IH16IK, 1756-IN16K, 1756-IV16K, 1756-IV32K, 1756-OA16K, 1756-OA16IK, 1756-OB16DK, 1756-OB16EK, 1756-OB16IEFSK, 1756-OB8EIK, 1756-OB8K, 1756-OB32K, 1756-OC8K, 1756-OG16K, 1756-ON8K, 1756-OV32EK, 1756-OW16IK, 1756-OX8IK

This publication was reformatted and updated to the most current template.

---

#### 1756-UM058I-EN-P, December 2017

---

##### Change

Information added about how to configure the 1756-OB16IEF redundant owner

---

#### 1756-UM058H-EN-P, May 2015

---

##### Change

Updated the Electronic Keying section

Updated the Attention text on RIUP support in the Install the Module section

Updated the MainTask tag name in Create a New Tag

Updated the use of the Browse button in the Communication Tab section

Updated Number of Motor Starters to be Used table

---

#### 1756-UM058G-EN-P, November 2012

---

##### Change

Studio 5000 Logix Designer<sup>®</sup> application is the rebranding of RSLogix 5000<sup>®</sup> software

Added the 1756-OB16IEFS module to the list of I/O modules

Added content to describe when output data is sent to the 1756-OB16IEFS module in motion applications

Added the 1756-OB16IEFS module to the CIP Sync<sup>™</sup> time section

Added the 1756-OB16IEFS module to sections about electronic fusing, diagnostic latching, and time-scheduled output control

Added the 1756-OB16IEFS module to the list of fast I/O modules

Added software version requirements for the 1756-OB16IEFS module

Added the 1756-OB16IEFS module to the table of connection formats

Added the wiring diagram for the 1756-OB16IEFS module

Added status indicators for the 1756-OB16IEFS module

Added tag definitions for the 1756-OB16IEFS module

Added the 1756-OB16IEFS module to the list of IFMs

---

**1756-UM058F-EN-P, April 2012****Change**

Added sections about using CIP Sync time

Added 1756-OB16IEF module to the list of modules with electronic fusing

Added a chapter to describe features of the 1756-IB16IF and 1756-OB16IEF modules

Added connection formats for the 1756-IB16IF and 1756-OB16IEF modules

Added leakage resistor sizing and supply voltage chart for the 1756-IB16D modules

Added wiring diagrams for the 1756-IB16IF and 1756-OB16IEF modules

Added new tags for the 1756-IB16IF and 1756-OB16IEF modules

Added a section about timestamped inputs and scheduled outputs for fast I/O modules

**1756-UM058E-EN-P, August 2010****Change**

Added Information for scheduling I/O modules on the ControlNet® network and setting up modules to trigger event-based tasks

Features and module-specific information for mosuled 1756-IA32, 1756-IG16, 1756-OB8I, 1756-OB16IS, 1756-OG16, 1756-OV32E

Added Using electronic keying with examples of Exact Match, Compatible, and Disabled Keying

Added new digital I/O specifications

Requirements for firmware updates for Major Revision 3.x

Updated information on Interface Modules (IFMs) and pre-wired cables that are available with digital I/O modules

**1756-UM058D-EN-P, October 2004****Change**

Added Event-based tasks information to Chapter 2

Features description and module-specific information (wiring diagrams) for modules 1756-IA32, 1756-IG16, 1756-OB16IS, 1756-OG16, 1756-OV32E added to Chapter 3

Added Using the Motion Axis Output Cam (MAOC) Instructions with Time Scheduled Output Control

Added Scheduled output data per point communications format - module 1756-OB16IS only

Revised Updated Environmental Condition Specifications

Added Using 1492 Wiring Systems with Your Digital I/O Module

New and revised terms in Glossary

**1756-UM058C-EN-P, March 2001****Change**

New section added – Internal Module Operations

Chapter 2 revised for Connections

Chapters 3 and 4 revised for Electronic Keying and Output Data Echo

New information about 1756-IV16 Module added in Chapter 3 and 7

New information about 1756-IV32 Module added in Chapter 3 and 7

New information about 1756-OV16E Module added in Chapter 3 and 7

Index updated with new terms

**1756-UM058B-EN-P, July 1999****Change**

Chapter 7 contains updated backplane current and simplified schematic information for some I/O modules where noted

Appendix D contains information on driving motor starters with ControlLogix® I/O modules

Chapter arrangement modified; Chapter 4 diagnostic modules added

**A****array data structure** 144**C****Change of State (COS)**

data transmissions 22

**CIP Sync time** 30, 136, 142, 151**Claim Owner Output** 75**communication**

format 15

producer/consumer model 24

**communication format**

about 15

CST Timestamped Fuse Data 64

CST Timestamped Input Data 64

Full Diagnostic Input Data 64

Full Diagnostics 64

Input Data 64

Listen Only 64

Output Data 64

Rack Optimization 64

Scheduled Output Data 64

**configure**

fault state delay 83

input filter time 69

input filter times 52

modules 27

peer ownership 47

per point timestamping 51

point-level output states 35

pulse width modulation 83

redundant owner

output module 86

**connection**

direct 18

format 15

rack-optimized 18

**connection format**

about 15

Data 64, 65

Data with Event 50, 64

Listen Only 64, 65

Listen Only with Event 64

Peer Input with Data 65

**ControlNet network**

input modules in remote chassis 23

output modules in remote chassis 25

tip on conserving bandwidth 22

**COO. See Claim Owner Output (COO)****coordinated system time (CST)** 29, 149**create**

event tags for fast module 50

new module 62

**CST Timestamped Data communication**

format 64

**CST Timestamped Fuse Data communication**

format 64

**D****data exchange**

peer ownership 47

producer/consumer model 11, 24

**data structure**

array 144

flat 144

**Data with Event connection format** 50**diagnostic**

features 39 - 45

latching 37

**direct connection** 18**disable**

change of state 68, 81

diagnostic latching 71

diagnostics for field power loss 70

filtering 82

module communication 28

timestamp latching 81

timestamps 81

**dynamic reconfiguration** 66**E****electronic fusing** 35**enable**

change of state 68, 81

diagnostic latching 71

diagnostics for field power loss 70

filtering 82

timestamp latching 81

timestamps 81

**Ethernet modules**

redundant owner 75

**event task trigger** 23, 48 - 50**F****fast I/O module**

array data structure 144

CIP Sync time 30, 151

event task trigger 48 - 50

fault and status reporting 50 - 54

output module compatibility 54

per point timestamping 51 - 81

pulse capture 48

pulse width modulation 55 - 86

response time 47

software configurable filter times 52 - 82

**fault**

latch 37

type 93

**features**

diagnostic 39 - 45

digital I/O modules 26

fast 47 - 54

**field power loss detection**

1756-0A8E module 37

**flat data structure** 144**Full Diagnostic Input Data communication**

format 64

**Full Diagnostics communication format** 64  
**fusing, electronic** 35

## I

**IFM. See interface module**  
**Input Data communication format** 64  
**input module**  
     support multiple owners 79  
**interface module** 11  
**internal module operation** 20

## K

**keying**  
     mechanical 13

## L

**latch**  
     fault 37  
     pulse 48  
     timestamps 81  
**Listen Only communication format** 64  
**local chassis**  
     output modules 25  
**locking tab** 13  
**loss of field power** 33

## M

**major revision** 61  
**mechanical**  
     fusing 35  
     keying 13  
**minor revision** 61  
**module**  
     1756-OA16 108  
     1756-OA16I 109  
     1756-OA8 106  
     1756-OA8D 107  
     1756-OA8E 107  
     1756-OB16D 111  
     1756-OB16E 112  
     1756-OB16I 112  
     1756-OB16IEF 113  
     1756-OB16IEFS 113  
     1756-OB16IS 114  
     1756-OB32 114  
     1756-OB8 110  
     1756-OB8E1 111  
     1756-OC8 115  
     1756-OG16 116  
     1756-OH8I 117  
     1756-ON8 117  
     1756-OV16E 118  
     1756-OV32E 118  
     1756-OW16I 119  
     1756-OX8I 119  
**module compatibility**  
     fast  
         output modules 54  
**module fault bits, Redundant Owner** 33

**module identification information** 14  
     ASCII text string 14  
     major revision 14  
     minor revision 14  
     product code 14  
     product type 14  
     serial number 14  
     status 14  
     vendor ID 14  
**module status** 14

## O

**output**  
     data echo 24, 33  
**Output Data communication format** 64  
**output module**  
     response time 79, 88  
     troubleshooting 94  
**ownership** 16  
     controller-I/O module relationship 16  
     direct connection 18  
     input remote connections 23  
     Listen-only 19, 173  
     output remote connections 25  
     rack  
         optimization 19, 20, 173, 174

## P

**peer ownership** 47  
**producer/consumer model** 11, 24  
**pulse**  
     capture 48  
     latch 48  
**pulse width modulation**  
     cycle time 55  
     On time 55

## R

**Rack Optimization communication format** 64  
**rack-optimized connection** 18, 19, 20, 173, 174  
**Ready Owner Output** 75  
**redundant owner**  
     configure  
         output module 86  
     Ethernet modules 75  
     requirements 75, 79  
     restrictions 76  
     troubleshooting  
         I/O Module Configuration Error 93  
         I/O Module Connection Error 93  
         output module 94  
**Redundant Owner, module fault bits** 33  
**remote chassis**  
     input modules 22, 23  
     output modules 25  
**removable terminal block** 11  
     parts illustration 13  
**Removal and Insertion Under Power** 14  
**requested packet interval** 22, 44  
**requirements for redundant owner** 75, 79

**response time**

output module 79, 88

**restrictions**

redundant owner 76

**ROO. See Ready Owner Output (ROO)****RPI. See requested packet interval****RSNetWorx software**

transfer configuration data 20

**RTB. See removable terminal block****S****scheduled output data**

fast I/O modules 30, 65, 151

standard and diagnostic modules 29, 149

**Scheduled Output Data communication format**

64

**specifications 9****status indicators 13****T****task, event 23, 48 - 50****timestamps**

CIP Sync 30, 136, 142, 151

CST 29, 149

latch 81

standard and diagnostic I/O modules 149

**tips**

conserving ControlNet bandwidth 22

pulse test 43

**trigger**

event task 23, 48 - 50

**troubleshooting**

module status indicators 13

**W****wiring connections**

field wiring options 34

interface module 11

removable terminal block 11





# Rockwell Automation Support

Use these resources to access support information.

<b>Technical Support Center</b>	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	<a href="http://rok.auto/support">rok.auto/support</a>
<b>Local Technical Support Phone Numbers</b>	Locate the telephone number for your country.	<a href="http://rok.auto/phonesupport">rok.auto/phonesupport</a>
<b>Technical Documentation Center</b>	Quickly access and download technical specifications, installation instructions, and user manuals.	<a href="http://rok.auto/techdocs">rok.auto/techdocs</a>
<b>Literature Library</b>	Find installation instructions, manuals, brochures, and technical data publications.	<a href="http://rok.auto/literature">rok.auto/literature</a>
<b>Product Compatibility and Download Center (PCDC)</b>	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	<a href="http://rok.auto/pcdc">rok.auto/pcdc</a>

## Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at [rok.auto/docfeedback](http://rok.auto/docfeedback).

## Waste Electrical and Electronic Equipment (WEEE)



At the end of life, this equipment should be collected separately from any unsorted municipal waste.



Rockwell Automation maintains current product environmental compliance information on its website at [rok.auto/pec](http://rok.auto/pec).

Allen-Bradley, ControlLogix, ControlLogix-XT, expanding human possibility, Integrated Architecture, Logix 5000, Rockwell Automation, RSLogix, RSLogix 5000, Studio 5000, and Studio 5000 Logix Designer are trademarks of Rockwell Automation, Inc.

CIP, CIP Sync, ControlNet, and EtherNet/IP are trademarks of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

**rockwellautomation.com** ————— **expanding human possibility®**

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2663 0600

ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608

UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800

Publication 1756-UM058K - January 2025

Supersedes Publication 1756-UM058J - September 2023

Copyright © 2025 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.