



Compact 5000 I/O Serial Module

Catalog Number 5069-SERIAL



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.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this presentation 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.

	Preface	
	About This Publication	7
	Download Firmware, AOP, EDS, and Other Files	7
	Summary of Changes.....	7
	Appearance Change	8
	Before Change.	8
	After Change.	8
	Chapter 1	
Compact 5000 I/O Serial Module Overview	Controller and Software Compatibility.....	10
	Module Overview	11
	Local I/O or Remote I/O Modules	12
	Local I/O Modules	12
	Remote I/O Module.....	13
	Chapter 2	
Compact 5000 I/O Serial Module Operation	Ownership	15
	Construct a System	15
	Local I/O modules	15
	Remote I/O modules	16
	5069-ARM Address Reserve Module.....	17
	Power the Module.....	18
	5069-FPD Field Potential Distributor.....	19
	Configure Modules.....	20
	Connections	20
	Protected Operations.....	22
	Enter and Exit Protected Operations	22
	Chapter 3	
Compact 5000 I/O Serial Module Features	General	23
	Software Configurable	23
	Requested Packet Interval	24
	Fault and Status Reporting	25
	Module Inhibiting	25
	Electronic Keying.....	26
	Status Indicators	27
	Module Firmware.....	27
	Common Module Functions	28
	Control Line Menu.....	29

Generic ASCII Data Exchange 29
 Data Sent with the Serial Port 29
 Generic ASCII Transmit Functions 30
 Generic ASCII Transmit Methods 30
 Data Received from the Serial Port 30
 Generic ASCII Receive Functions 33
 Generic ASCII Receive Methods 34
 Modbus Master Data Exchange 35
 Modbus Master Functions 36
 Command Disable function 36
 Modbus Slave Data Exchange 37
 Modbus Slave Functions 38
 DF1 Protocols 39
 Using Modems that Support DF1 Communication Protocols 39
 Modem Control Line Operation 40
 Configuration Considerations for RTS Send and Off Delays 42
 DF1 Point-to-Point Protocol 42
 DF1 Master Protocol 43
 Configuring a Master Station for Standard Polling Mode 43
 Minimum DF1 Half-Duplex Master ACK Timeout 47
 Determining Minimum Master Serial Port ACK Timeout 47
 Create Polling Lists 48
 Monitor Active Stations 49
 Configuring a Message-Based Mode DF1 Half-Duplex
 Master Station 49
 Configuring a Master Station for Message-based Polling Mode 50
 DF1 Slave Protocol 53
 Configuring the Controller as a Slave Station 53
 DF1 Radio Modem Protocol 55
 Designing Communication for DF1 Radio Modem Protocol 55
 DF1 Data Exchange 58
 DF1 Point to Point Parameters 59
 DF1 Master Parameters 59
 DF1 Slave Parameters 59
 DF1 Radio Modem Parameter 60
 DH-485 Data Exchange 61
 DH-485 Functions 62

Chapter 4

**Compact 5000 I/O Serial Module
 Configuration**

Before You Begin 63
 Add a New Module 63
 Discover Local I/O Modules 64
 Discover Remote I/O Modules 66
 New Local I/O Modules 68
 New Remote I/O Module 69

Edit the Module Configuration	71
General Category	71
Connection Category	77
Module Info Category	88
View the Module Tags	89

Chapter 5

Troubleshoot Your Module

Compact 5000 I/O Serial Module Status Indicators	91
--	----

Appendix A

Module Tags

Name Conventions	93
Generic ASCII and Modbus Slave Name Conventions	93
Modbus Master Name Conventions	94
DF1 Name Conventions	94
DH-485 Name Conventions	94
Access the Tags	95
Channel Configured Generic ASCII Tags	96
Channel Configured for Generic ASCII	97
Input Tags	97
Output Tags	99
Channel Configured for Modbus Master	100
Input Tags	100
Output Tags	102
Channel Configured for Modbus Slave	103
Input Tags	104
Output Tags	105
Channel Configured for DF1—Point to Point	105
Input Tags	106
Output Tags	106
Channel Configured for DF1—Master	107
Input Tags	108
Output Tags	108
Channel Configured for DF1—Slave	109
Input Tags	109
Channel Configured for DF1—Radio Modem	110
Input Tags	110
Output Tags	110
Channel Configured for DH-485	111
Input Tags	111

Modbus Command Codes	Appendix B	
	Master Command List Function Codes	113
	Read Coil Status (Function Code 01)	113
	Read Input Status (Function Code 02)	114
	Read Holding Registers (Function Code 03)	115
	Read Input Registers (Function Code 04)	116
	Force Single Coil (Function Code 05)	117
	Preset Single Register (Function Code 06)	118
	Force Multiple Coils (Function Code 15)	119
	Preset Multiple Registers (Function Code 16)	120
Programming Example	Appendix C	
	Generic ASCII Sample Code	121
	Transmit and Receive Channel Configurations	121
	Sample Code	122
	Modbus Sample Code Configuration	124
	Modbus Master Command List	124
	Modbus Slave Address Mapping Table	124
	Modbus Sample Code Configuration Example	124
	Modbus Master Sample Code	125
	Modbus Slave Sample Code	125
	Example MSG	
	Control Blocks	127
ASCII Conversion Tables	Appendix D	
	ASCII Conversions	131
	Index	133
	Additional Resources	135

About This Publication

This manual describes how to use Compact 5000™ I/O serial module in Logix 5000™ systems.

Make sure that you are familiar with the following:

- Use of a Logix 5000 controller
- Use of an EtherNet/IP™ network, if the serial module is installed in a remote location that is accessible via the EtherNet/IP network.
- Studio 5000 Logix Designer® environment.

IMPORTANT The Compact 5000 I/O Serial module is only compatible with the following controllers:

- CompactLogix™ 5380
- Compact GuardLogix® 5380
- CompactLogix 5480
- ControlLogix® 5580
- GuardLogix® 5580

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this presentation 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.

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.

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 the Protected Operations section	22

Appearance Change

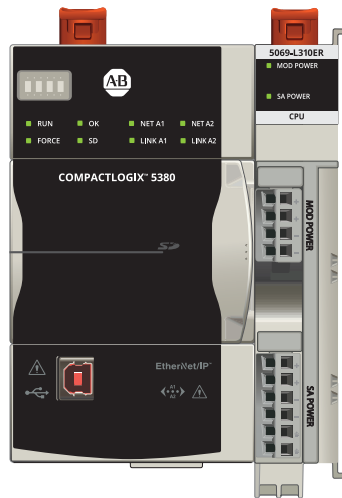
The appearance of the CompactLogix Series A Controllers has been updated.

Before Change.



CompactLogix 5380 Controller

After Change.



CompactLogix 5380 Controller

Compact 5000 I/O Serial Module Overview

Topic	Page
Controller and Software Compatibility	10
Module Overview	11
Local I/O or Remote I/O Modules	12

The Compact 5000™ I/O serial module provides two independent channels that function as network interface via RS232C, RS422 or RS-485 to serial devices communicating on Generic ASCII, Modbus RTU/ASCII, DF1 or DH-485 protocol.

The module has two channels that are independent of each other. The channels can transmit data to and receive data from serial devices using the following communication mode:

- Generic ASCII
- Modbus RTU/ASCII (Master/Slave)
- DF1
- DH-485

The serial module can reside locally in a CompactLogix™ 5380, Compact GuardLogix® 5380, or CompactLogix 5480 system. The module can also reside in a remote Compact 5000 I/O system accessible via an EtherNet/IP™ network.

IMPORTANT Use Studio 5000 Logix Designer® Version 31 or greater. You must install an Add-On Profile (AOP) to use the serial module. To find the Add-On Profile, go to the [Product Compatibility and Download Center](#) (PCDC).
DF1 and DH-485 must use Channel 1 and be local.
The serial module with firmware revision 2.011 and later with Add-On profile version 2.01 supports DH-485 and DF1 protocols DF1 Point to Point, DF1 Master, DF1 Slave, and DF1 Radio modem.

Controller and Software Compatibility

When you use Compact 5000 I/O serial module in Logix 5000™ controller control system, the Compact 5000 I/O Serial module is compatible with the following controllers:

- CompactLogix 5380
- Compact GuardLogix 5380
- CompactLogix 5480
- ControlLogix® 5580 (remote only)
- GuardLogix® 5580 (remote only)

IMPORTANT The serial module is not compatible with the 5069-AEN2TR. See the [Product Compatibility and Download Center](#) (PCDC) for more information.

Table 1 - Compact 5000 I/O Modules Controller and Software Compatibility Requirements

Location	Controllers		Logix Designer Application
	System	Cat. Nos.	
Local I/O modules	CompactLogix 5380	5069-L306ER, 5069-L306ERM, 5069-L310ER, 5069-L310ERM, 5069-L310ERMK, 5069-L310ER-NSE, 5069-L310ERS2, 5069-L320ER, 5069-L320ERM, 5069-L320ERMK, 5069-L320ERP, 5069-L330ER, 5069-L330ERM, 5069-L330ERMK, 5069-L340ER, 5069-L340ERM, 5069-L340ERP, 5069-L350ERM, 5069-L350ERMK, 5069-L380ERM, 5069-L3100ERM	Version 31.00.00 or later
	CompactLogix 5480	5069-L46ERMW	Version 32.00.00 or later
		5069-L430ERMW, 5069-L450ERMW, 5069-L4100ERMW, 5069-L4200ERMW	Version 32.01.00 or later
Compact GuardLogix 5380	5069-L306ERS2, 5069-L306ERMS2, 5069-L310ERS2, 5069-L310ERMS2, 5069-L320ERS2, 5069-L320ERS2K, 5069-L320ERMS2, 5069-L320ERMS2K, 5069-L330ERS2, 5069-5069-L330ERS2K, L330ERMS2, 5069-L330ERMS2K, 5069-L340ERS2, 5069-L340ERMS2, 5069-L350ERS2, 5069-L350ERS2K, 5069-L350ERMS2, 5069-L350ERMS2K, 5069-L380ERS2, 5069-L380ERMS2, 5069-L3100ERS2, 5069-L3100ERMS2	Version 31.00.00 or later	
Remote I/O modules ⁽¹⁾	CompactLogix 5380	5069-L306ER, 5069-L306ERM, 5069-L310ER, 5069-L310ERM, 5069-L310ER-NSE, 5069-L310ERS2, 5069-L310ERMK, 5069-L320ER, 5069-L320ERM, 5069-L320ERP, 5069-L330ER, 5069-L330ERM, 5069-L340ER, 5069-L340ERM, 5069-L340ERP, 5069-L350ERM, 5069-L380ERM, 5069-L3100ERM	Version 31.00.00 or later
	CompactLogix 5480	5069-L46ERMW	Version 32.00.00 or later
		5069-L430ERMW, 5069-L450ERMW, 5069-L4100ERMW, 5069-L4200ERMW	Version 32.01.00 or later
	Compact GuardLogix 5380	5069-L306ERS2, 5069-L306ERMS2, 5069-L310ERS2, 5069-L310ERMS2, 5069-L320ERS2, 5069-L320ERS2K, 5069-L320ERMS2, 5069-L320ERMS2K, 5069-L330ERS2, 5069-5069-L330ERS2K, L330ERMS2, 5069-L330ERMS2K, 5069-L340ERS2, 5069-L340ERMS2, 5069-L350ERS2, 5069-L350ERS2K, 5069-L350ERMS2, 5069-L350ERMS2K, 5069-L380ERS2, 5069-L380ERMS2, 5069-L3100ERS2, 5069-L3100ERMS2	Version 31.00.00 or later
	ControlLogix 5580	1756-L83E, 1756-L85E, 1756-L81E, 1756-L82E, 1756-L84E	Version 31.00.00 or later
GuardLogix 5580	1756-L81ES, 1756-L82ES, 1756-L83ES, 1756-L84ES	Version 31.00.00 or later	

(1) Serial module is not compatible with 5069-AEN2TR.

Module Overview

The module provides the communication connections to the serial devices.

Figure 1 - Example Compact 5000 I/O Serial Module

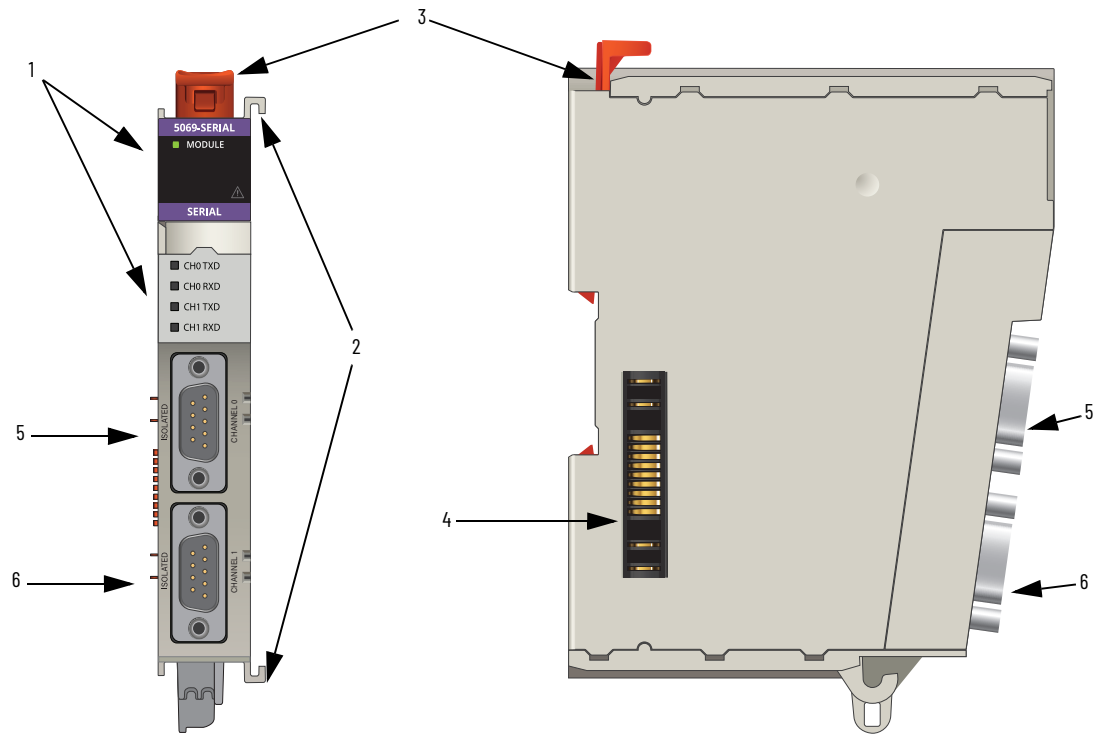


Table 2 - Compact 5000 I/O Serial Module Parts

Item	Item Name	Description
1	Status Indicators	Displays the status of communication, module health, and input/output devices. Indicators help with troubleshooting anomalies.
2	Interlocking side pieces	Securely installs Compact 5000 I/O serial modules in the system.
3	DIN rail latch	Secures the module on the DIN rail.
4	MOD power bus and SA power bus connectors	Pass system-side and field-side power across the internal circuitry of the I/O modules in a Compact 5000 I/O system. The connectors are isolated from each other.
5	Channel 0	Channel 0 isolated serial port.
6	Channel 1	Channel 1 isolated serial port.

Local I/O or Remote I/O Modules

You can use Compact 5000 I/O modules as local or remote I/O modules, with some restrictions that are based on the module and controller type. Compatibility requirements apply and are described in [Controller and Software Compatibility on page 10](#).

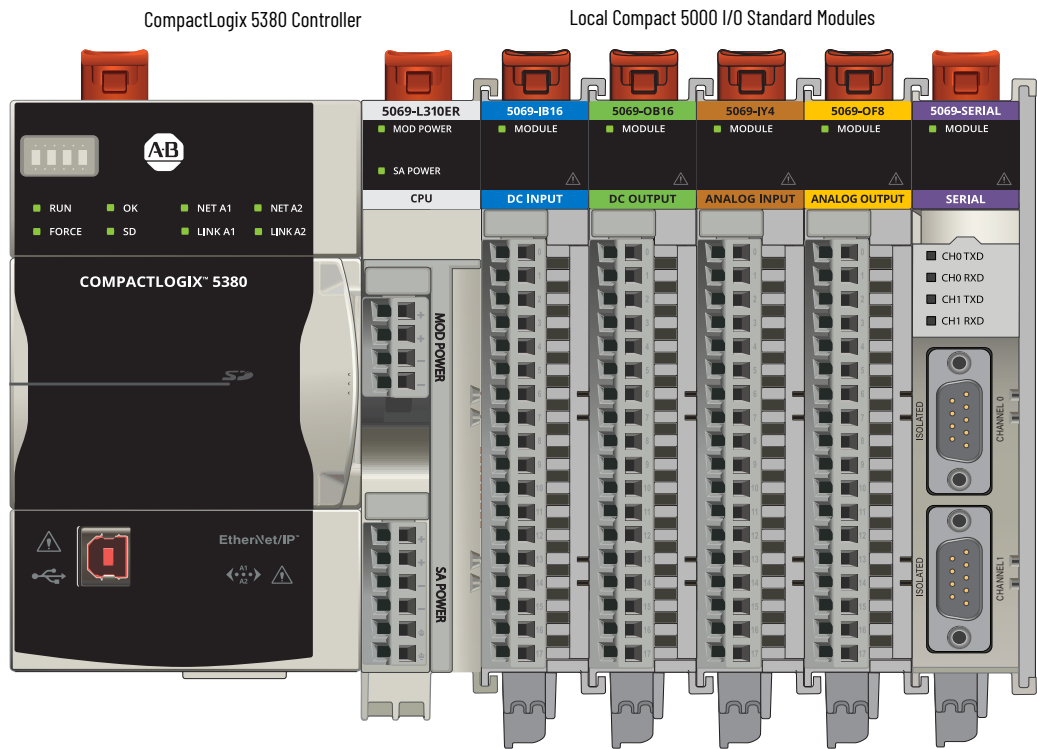
Local I/O Modules

When Compact 5000 I/O Serial module resides in the same system as the controller, it is a local I/O module. Local I/O modules are installed to the right of the controller and exchange data with the controller over the system backplane.

IMPORTANT Compact 5000 I/O Serial modules can function as local I/O modules in the following:

- CompactLogix 5380 control systems
- Compact GuardLogix 5380 control systems
- CompactLogix 5480 control systems

Figure 2 - Compact 5000 I/O Serial Module as Local I/O Module



Remote I/O Module

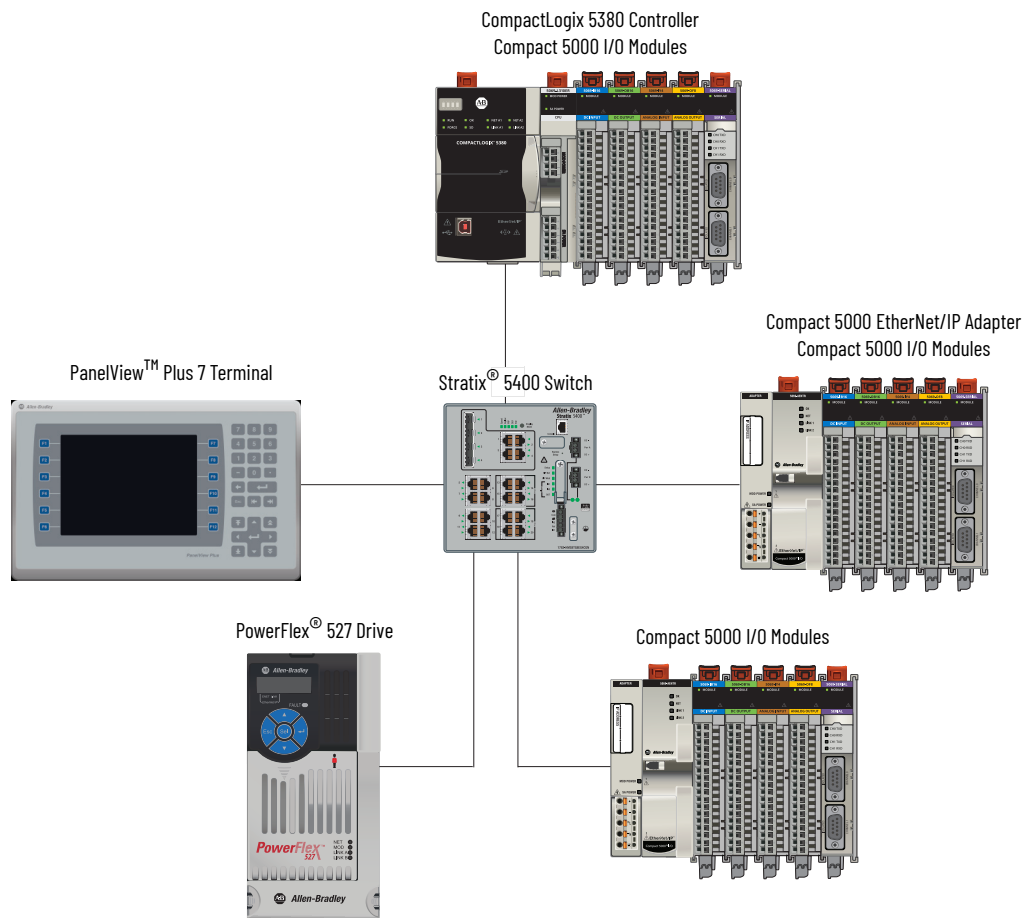
When a Compact 5000 I/O Serial module resides in a separate location from the controller, it is a remote I/O module. Remote Compact 5000 I/O modules are accessible over an EtherNet/IP network via a Compact 5000 EtherNet/IP adapter.

Remote Compact 5000 I/O modules are installed to the right of the adapter and exchange data across the remote system backplane. The data is exchanged with the controller over the EtherNet/IP network.

IMPORTANT Compact 5000 I/O Serial modules can function as remote I/O modules in the following:

- CompactLogix 5380 control systems
 - Compact GuardLogix 5380 control systems
 - CompactLogix 5480 control systems
 - ControlLogix 5580 control systems
 - GuardLogix 5580 control systems
- DF1 and DH-485 modes are not supported remotely.

Figure 3 - Compact 5000 I/O Serial Module as Remote I/O Module



Notes:

Compact 5000 I/O Serial Module Operation

Topic	Page
Ownership	15
Construct a System	15
5069-ARM Address Reserve Module	17
Power the Module	18
5069-FPD Field Potential Distributor	19
Configure Modules	20
Protected Operations	22

Ownership

Every I/O module in a Logix 5000™ control system must be owned by a controller, also known as the owner-controller. When the Compact 5000™ I/O modules are used in a Logix 5000 control system, the owner-controller performs the following:

- Stores configuration data for every module that it owns.
- Can reside in a location that differs from the Compact 5000 I/O modules.
- Sends the I/O module configuration data to define module behavior and begin operation in the control system.

Each Compact 5000 I/O serial module must continuously maintain communication with its owner-controller during normal operation.

Construct a System

Before you use your serial module, you must complete tasks based on whether the modules are local or remote I/O modules:

Local I/O modules

Complete the following steps.

1. Install a CompactLogix™ 5380, CompactLogix 5480, or Compact GuardLogix® 5380 controller.

IMPORTANT You must use a Compact GuardLogix 5380 controller if the local Compact 5000 I/O modules include safety modules. For more information on controller compatibility, see [Controller and Software Compatibility on page 10](#).

2. Install the modules to the right of the controller.

3. Install the end cap on the last module in the system.

IMPORTANT The end cap in a CompactLogix 5380, CompactLogix 5480, or Compact GuardLogix 5380 control system covers the exposed interconnection on the last module on the DIN rail.
If you do not install an end cap on the last module on the DIN rail, equipment damage or injury can occur.

Remote I/O modules

Complete the following:

1. Install a controller that is compatible with the remote Compact 5000 I/O modules.

IMPORTANT Remember, you must use a Compact GuardLogix 5380 controller or GuardLogix 5580 controller if the set of remote Compact 5000 I/O modules include safety modules.
For more information on controller compatibility, see [Controller and Software Compatibility on page 10](#).

2. Install an EtherNet/IP™ network.
3. Connect the controller to the network.
4. Install a Compact 5000 EtherNet/IP adapter.

IMPORTANT If serial module is included, you must use AENTR

5. Connect the adapter to the network.
6. Install the Compact 5000 I/O modules to the right of the adapter.
7. Install the end cap on the last module in the system.

IMPORTANT The end cap in a CompactLogix 5380, CompactLogix 5480, or Compact GuardLogix 5380 control system covers the exposed interconnection on the last module on the DIN rail.
If you do not install an end cap on the last module on the DIN rail, equipment damage or injury can occur.

5069-ARM Address Reserve Module

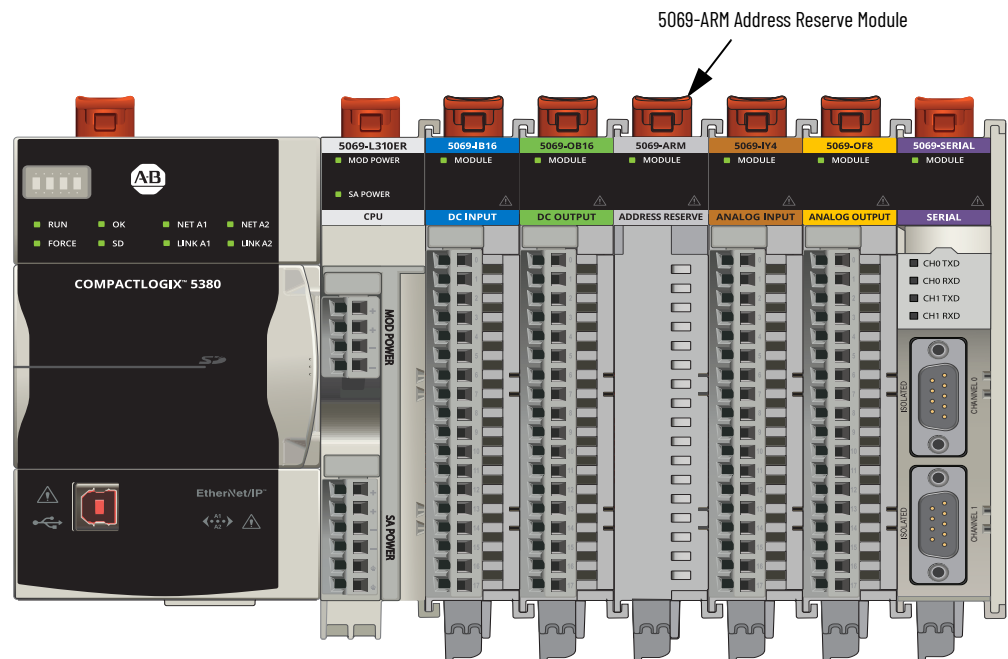
Every Compact 5000 I/O module has a unique node address in a system. As modules are installed, the node addresses increment. The Logix Designer application project includes modules in the I/O Configuration that correspond to the physical modules.

If a module is not available during initial system installation and operation, you can use a 5069-ARM address reserve module to reserve the slot in the system. That is, you install the 5069-ARM address reserve module to reserve the node address. The address reserve module remains installed until the I/O module is available.

When you install the address reserve module, you also make sure that the subsequently-installed modules are at the correct node address.

You use the corresponding entry in the Logix Designer application project to reserve the node address. That is, when you add modules to the I/O Configuration tree in the project, you add an address reserve module at the node address that matches the physical module location.

Figure 4 - Compact 5000 I/O System with 5069-ARM Address Reserve Module



Power the Module

Compact 5000 I/O modules receive the following power types from first component in the system, that is, the controller or adapter:

- System-side Power - Powers the system and lets modules transfer data and execute logic.

System-side power is provided through the Module (MOD) Power connector and is passed to each module as it is added to the system.

System-side power is also known as **MOD power**.

- Field-side Power - Powers field-side devices that are connected to some Compact 5000 I/O modules.

Field-side power is provided through the Sensor/Actuator (SA) Power connector and is passed to each module as it is added to the system.

Field-side power is also known as **SA power**.

IMPORTANT The Compact 5000 I/O Serial Module does not use SA power. However, verify that the module is installed in a position where SA power uses DC voltage.

Power begins at the leftmost device in the system and passes across the Compact 5000 I/O module internal circuitry via power buses, that is, a MOD power bus. The leftmost device is either a controller or an EtherNet/IP adapter, depending on whether the Serial module is a local or remote I/O module.

IMPORTANT Remember the following:

- A system uses only one MOD power bus that starts at a controller or adapter and passes across all modules that are installed in the system.
- A system can use multiple SA power bus. The first SA power bus typically starts at the controller or adapter, and 5069-FPD field potential distributors let you establish new SA power buses in the same system.

For more information on how to power local Compact 5000 I/O modules, see the following:

- CompactLogix 5380 and Compact GuardLogix 5380 Controllers User Manual, publication [5069-UM001](#)
- CompactLogix 5480 Controller User Manual, publication [5069-UM002](#)

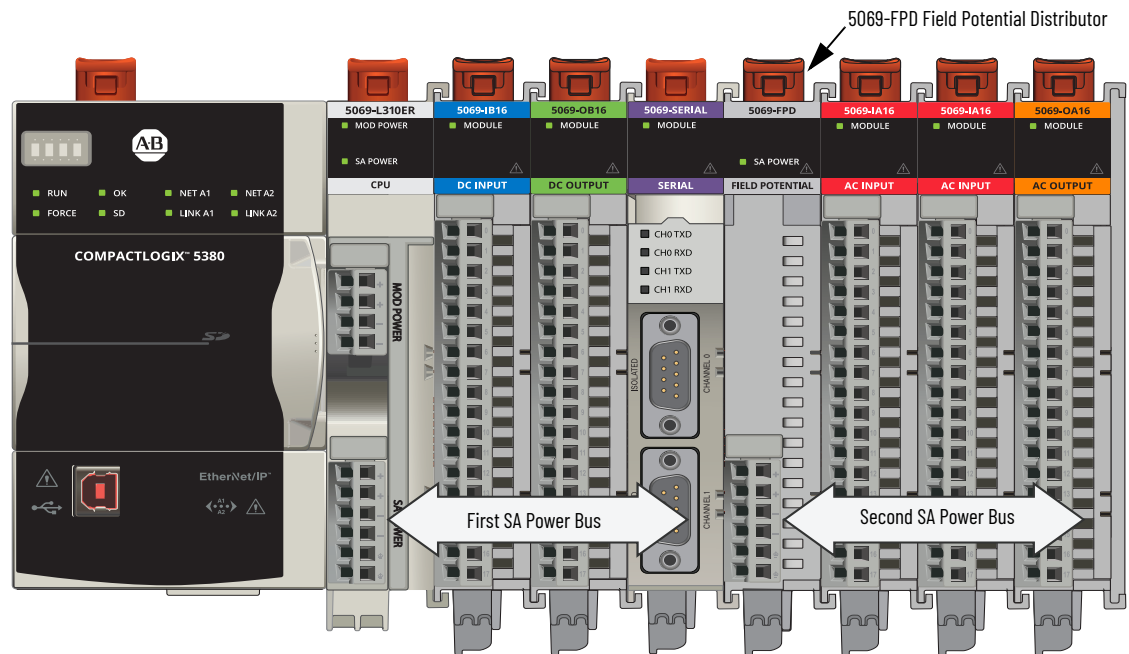
For more information on how to power remote Compact 5000 I/O modules, see the Compact 5000 EtherNet/IP Adapters User Manual, publication [5069-UM007](#).

5069-FPD Field Potential Distributor

The controller or adapter, based on whether the module is a local or remote I/O module, is the primary source of field-side power, that is, SA power, in the system. However, you can use a 5069-FPD field potential distributor to break field-side power distribution in a system and establish a new SA power bus.

Field-side power passes across the internal circuitry of the Compact 5000 I/O modules beginning with the controller or the adapter. The field potential distributor blocks the passage of field-side power from the left and functions as a new field-side power source for the modules to the right.

Figure 5 - Compact 5000 I/O System with 5069-FPD Field Potential Distributor



For more information on how to power local Compact 5000 I/O modules, see the following:

- CompactLogix 5380 and Compact GuardLogix 5380 Controllers User Manual, publication [5069-UM001](#)
- CompactLogix 5480 Controller User Manual, publication [5069-UM002](#)

For more information on how to power remote Compact 5000 I/O modules, see the Compact 5000 EtherNet/IP Adapters User Manual, publication [5069-UM007](#).

Configure Modules

You must create a Logix Designer application project for the Logix 5000 controller that owns the Compact 5000 I/O Serial modules. The project includes module configuration data for the Compact 5000 I/O Serial module.

The Logix Designer application transfers the project to the owner-controller during the program download. Data is then transferred to the Compact 5000 I/O Serial module either across the backplane or over an EtherNet/IP network.

The Compact 5000 I/O modules can operate immediately after receiving the configuration data.

Connections

During module configuration, you must define the module. In the Module Definition parameters for most Compact 5000 I/O modules, you must choose a Connection type. A connection is a real-time data transfer link between the owner-controller and the module that occupies the slot that the configuration references.

When you download module configuration to a controller, the controller attempts to establish a connection to each module in the configuration.

Because part of module configuration includes a slot number in the local or remote system, the owner-controller checks for the presence of a module there. If a module is detected, the owner-controller sends the configuration.

One of the following occurs:

- If the configuration is appropriate to the module detected, a connection is made and operation begins.
- If the configuration is not appropriate to the module detected, the data is rejected and the Logix Designer application indicates that an error occurred.

The configuration can be inappropriate for many reasons. For example, a mismatch in electronic keying that helps prevent normal operation.

The owner-controller monitors its connection with a module. Any break in the connection, for example, the loss of power to the system, causes a fault. The Logix Designer application monitors the fault status tags to indicate when a fault occurs on a module.

Multiple Connections to One Serial Module

Unlike other Compact 5000 I/O modules that make one connection to the owner-controller, the Compact 5000 I/O Serial module can have multiple connections based on how the module channels are configured.

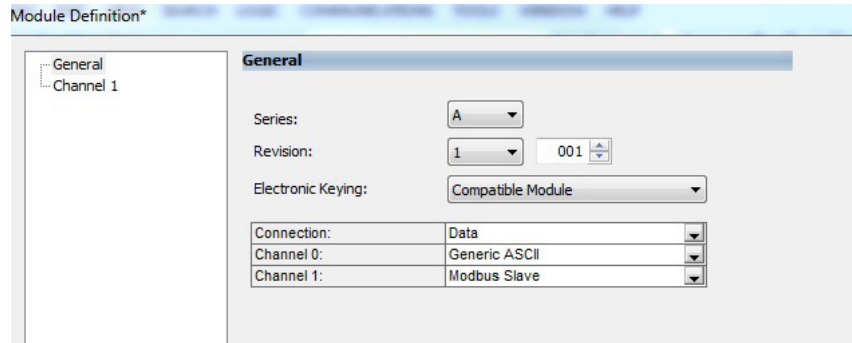
The Serial module is required to use the Data connection type in the Module Definition dialog box. However, the module has two channels that you must configure independently of each other. You can disable a channel or choose a mode, that is, the Generic ASCII, Modbus Master, Modbus Slave, DF1 or DH-485 mode.

The combination of channel configuration choices determines the number of connections that are made between the owner-controller and the Serial module.

Selected Protocol	Description
Generic ASCII	The module returns the following to the owner-controller: <ul style="list-style-type: none"> • General fault data • Input Data • Output Data
Modbus Master	The module returns the following to the owner-controller: <ul style="list-style-type: none"> • Generic fault data • Last executed command data • Input/Output • Operation status
Modbus Slave	The module returns the following to the owner-controller: <ul style="list-style-type: none"> • Generic fault • Slave input data • Slave output data
DF1	The module returns the following to the owner-controller: <ul style="list-style-type: none"> • General fault data • Input Data • Output Data
DH-485	The module returns the following to the owner-controller: <ul style="list-style-type: none"> • General fault data • Input Data • Output Data

For example, consider the following examples:

- Channel 0 = Generic ASCII, Channel 1 = Modbus Slave - Two connections are made between the owner-controller and the module.



- Channel 0 = Generic ASCII, Channel 1 = Modbus Master - Three connections are made between the owner-controller and the module

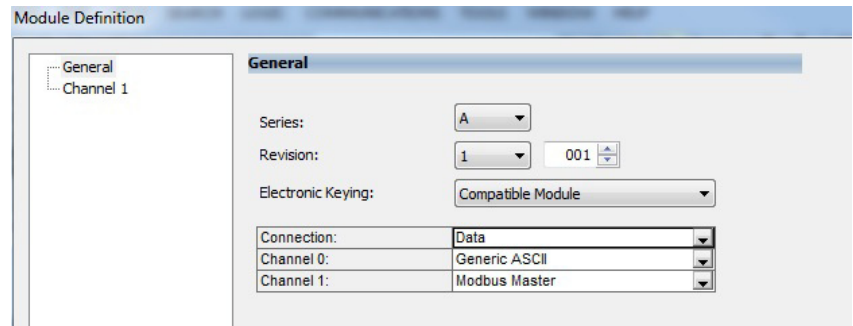


Table 3 shows the total number of connections for all protocol combinations.

Table 3 - I/O Connections For Each Protocol Combination

		CHO Protocol Choice (Cannot select DFI or DH-485)			
		Disabled	Generic ASCII	Modbus Slave	Modbus Master
CHI Protocol	Disabled	0 connection	1 connection	1 connection	2 connections
	Generic ASCII	1 connection	2 connections	2 connections	3 connections
	Modbus Slave	1 connection	2 connections	2 connections	3 connections
	Modbus Master	2 connections ⁽¹⁾	3 connections	3 connections	4 connections
	DFI Point to Point	1 connection	2 connections	2 connections	3 connections
	DFI Master	1 connection	2 connections	2 connections	3 connections
	DFI Slave	1 connection	2 connections	2 connections	3 connections
	DFI Radio Modem	1 connection	2 connections	2 connections	3 connections
	DH-485	1 connection	2 connections	2 connections	3 connections

(1) The Modbus Master can have 1 or 2 connections depending on the command list configuration.

Protected Operations

To maintain the secure operation of your Compact 5000 I/O Serial module, operations that can disrupt module operation are restricted based on the module operating mode. [Table 4](#) describes the restrictions.

Table 4 - Protected Operations on Compact 5000 I/O Serial Module

Current Module Operation	Activity						
	Firmware Update Request	Module Reset Request	Connection Request	Configuration Change	Connection or Data Format Change	Electronic Keying Change	RPI Change
Connection not running	Accepted						
Connection running	Rejected		Rejected	Accepted ⁽¹⁾	Not allowed ⁽²⁾		Accepted ⁽³⁾
Firmware update is in process	Rejected						

- (1) Configuration changes are made in the Module Properties dialog box and you click Apply.
- (2) The difference between Rejected and Not allowed is that rejected activities can be attempted in the Logix Designer application but do not take effect. The activities that are not allowed, that is, attempts to change the Connection or Data Format used, do not occur in the Logix Designer application. For example, if you attempt to reset a module that is connected to the owner-controller, the Logix Designer application executes the request and alerts you that it was rejected. If you attempt to change the data format on a module that is connected to an owner-controller, the Logix Designer application does not execute the attempted change. The application only alerts you that the change is not allowed. In the case, if the change is attempted online, the Module Definition dialog box field that changes the data format is disabled.
- (3) The change occurs after the connection is closed and reopened. You can close and reopen the connection in the following ways:
 - Change the project while it is offline and download the updated project before going online again.
 - Change the project while it is online and click Apply or OK in the Module Properties dialog box. In this case, a dialog box alerts you of the ramifications before the change is made.

Enter and Exit Protected Operations

The Compact 5000 I/O serial module enters Protected Mode as soon as I/O connection is established with the module. The Compact 5000 I/O Serial module exits Protected Mode as soon as I/O connection to the module is stopped.

Compact 5000 I/O Serial Module Features

Topic	Page
General	23
Generic ASCII Data Exchange	29
Modbus Master Data Exchange	35
Modbus Slave Data Exchange	37
DF1 Protocols	39
DF1 Data Exchange	58
DH-485 Data Exchange	61

General

The Compact 5000™ I/O serial module supports these module-wide features:

- [Software Configurable](#)
- [Requested Packet Interval](#)
- [Fault and Status Reporting](#)
- [Module Inhibiting](#)
- [Electronic Keying](#)

IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

Software Configurable

You use the Logix Designer application to configure the module, monitor system operation, and troubleshoot issues. You can also use the Logix Designer application to retrieve the following information from any module in the system:

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

By minimizing the need for tasks, such as setting hardware switches and jumpers, the software makes module configuration easier and offers excellent reliability.

Requested Packet Interval

The Requested Packet Interval (RPI) is a configurable parameter that defines a specific rate at which data is exchanged between the owner-controller and the module.

You set the RPI value during initial module configuration and can adjust it as necessary after module operation has begun.

-
- IMPORTANT** If you change the RPI while the project is online, the connection to the module is closed and reopened in one of the following ways:
- You inhibit the connection to the module, change the RPI value, and uninhibit the connection.
 - You change the RPI value. In this case, the connection is closed and reopened immediately after you apply the change to the module configuration.
-

For more information on guidelines for specifying RPI rates, see the Logix 5000™ Controllers Design Considerations Reference Manual, publication [1756-RM094](#).

Table 5 - RPI Range

Protocol	Setting Range of RPI
Generic ASCII	2 ms...750 ms (by 0.1 ms)
Modbus Master	6 ms...750 ms (by 0.1 ms)
Modbus Slave	6 ms...750 ms (by 0.1 ms)
DF1 Point to Point	2 ms...750 ms (by 0.1 ms)
DF1 Master	2 ms...750 ms (by 0.1 ms)
DF1 Slave	2 ms...750 ms (by 0.1 ms)
DF1 Radio Modem	2 ms...750 ms (by 0.1 ms)
DH-485	2 ms...750 ms (by 0.1 ms)

-
- IMPORTANT**
- If the RPI is not a multiple of 0.1 ms, round the number down to the closest multiple value of 0.1 ms. For example, if the RPI is 2.37 ms, round to 2.3 ms.
 - Depending on the software version, you can set each channel value separately.
 - When using Immediate Output (IOT) instruction, remember to add a 1 ms execution dwell time before each subsequent execution.
-

Fault and Status Reporting

The Compact 5000 I/O serial modules report fault and status data along with channel data. Fault and status data is reported in the following ways:

- Logix Designer application
- Module status indicators

For more information on fault reporting, see the full chapter, [Troubleshoot Your Module on page 91](#).

Module Inhibiting

Module inhibiting lets you indefinitely suspend a connection between an owner-controller and a serial module without removing the module from the configuration. This process lets you temporarily disable a module, such as to perform maintenance.

You can use module inhibiting in the following ways:

- You write a configuration for an I/O module but inhibit the module to help prevent it from communicating with the owner-controller. 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 connection to the module does not exist.

You can use module inhibiting in these instances:

- You want to update a serial I/O module, for example, update the module firmware revision. Use the following procedure.
 - a. Inhibit the module.
 - b. Perform the update.
 - c. Uninhibit the module.
- You use a program that includes a module that you do not physically possess yet. You do not want the controller to 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.

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.

Attribute	Description
Vendor	The device manufacturer.
Device Type	The general type of the product, for example, 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 following Electronic Keying options are available.

Keying Option	Description
Compatible Module	<p>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 the following characteristics:</p> <ul style="list-style-type: none"> • Same catalog number • Same or higher Major Revision • Minor Revision as follows: <ul style="list-style-type: none"> - If the Major Revision is the same, the Minor Revision must be the same or higher. - If the Major Revision is higher, the Minor Revision can be any number.
Disable Keying	<p>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.</p> <p>ATTENTION: Be cautious when using Disable Keying; if used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We strongly recommend that you do not use 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.</p>
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 | <ul style="list-style-type: none"> • 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. |
|------------------|---|

More Information

For more detailed information on Electronic Keying, see Electronic Keying in Logix 5000 Control Systems Application Technique, publication [LOGIX-ATool](#).

Status Indicators

Each Compact 5000 I/O serial 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.

For more information on status indicators, see Chapter 5, [Troubleshoot Your Module on page 91](#).

Module Firmware

The Compact 5000 I/O serial modules are manufactured with module firmware installed. If updated module firmware revisions are available in the future, you can update the firmware.

Updated firmware revisions are made available for various reasons, for example, to correct an anomaly that existed in previous module firmware revisions.

You access updated firmware files through the [Product Compatibility and Download Center](#) (PCDC). At the PCDC, you can use the module catalog number to check for firmware updates. If the catalog number is not available, no updates exist.

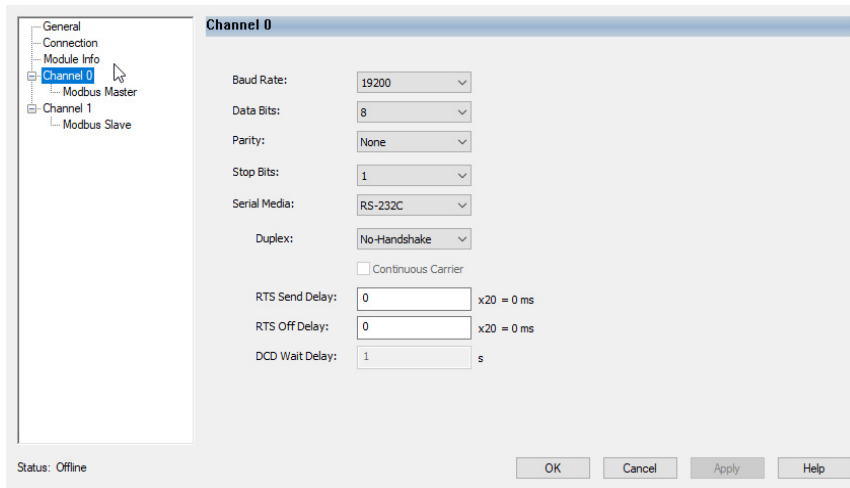
Common Module Functions

The module supports the following terms, definitions, and parameters.

Table 6 - Common Module Function

Function	Definition	Available Options
Baud rate	The communication speed of each channel.	<ul style="list-style-type: none"> • 9600 • 1200 • 2400 • 4800 • 19200 (default) • 38400 • 57600 • 115200
Data Bits	The number of data bits are used to represent one character of data.	<ul style="list-style-type: none"> • 7 bit • 8 bit (default)
Parity	Sets the parity of transmitted data for error detection. It is created with data files and used to check data integrity and help with data recovery.	<ul style="list-style-type: none"> • None (default) • Even • Odd
Stop Bits	This parameter sets the number of stop bits for each data value sent.	<ul style="list-style-type: none"> • 1 (default) • 2
Serial Media	Type of media that is connected to the channels communication ports.	<ul style="list-style-type: none"> • RS-232C (default) • RS-422 • RS-485
Duplex	The type of communication that is used by each channel.	<ul style="list-style-type: none"> • No Handshake (default) • Full-duplex • Half-duplex
Continuous Carrier	A carrier frequency that is transmitted even when data is not being sent. Continuous carrier is selected if you want to use it with half-duplex communication. The checkbox is unavailable if you have chosen something other than half-duplex communication, or if you have chosen Master as your protocol. The default option is cleared when enabled.	<ul style="list-style-type: none"> • On • Off
RTS Send Delay	Enter the time (x20 ms) to delay transmitting the first character of a message after turning on the RTS line. The default value is 0.	<ul style="list-style-type: none"> • 0...32767 (default = 0)
RTS Off Delay	Enter the time (x20 ms) to delay turning off the RTS line after the last character has been transmitted. The default value is 0.	<ul style="list-style-type: none"> • 0...32767 (default = 0)
DCD Wait Delay	The number of seconds to wait before lowering the DCD modem line. When DCD is high, the controller is in the middle of transmitting data. This delay may be needed because of the latency in the sending radio transmissions.	<ul style="list-style-type: none"> • 0...255 (default = 1)

Selecting Functions in Logix Designer Application



Control Line Menu

When you are required to connect to a dial-up modem, see [Table 7](#) that explains the duplex setting in the serial port.

Table 7 - Control Line Menu

Modem	Duplex Status	Controller	Your Function Choice	Continuous Carrier
Not using a modem	—		No Handshaking	
Using a modem	Modems in a point-to-point link are full-duplex	—	Full-duplex	—
	Master Modem is a full-duplex while slave modem is half-duplex.	Master Controller	Full-duplex	
	—	Slave Controller	Half-duplex	Select the continuous carrier checkbox.
	All modems in the system are half-duplex.	—	Half-duplex	Clear the continuous carrier checkbox (default).

Generic ASCII Data Exchange

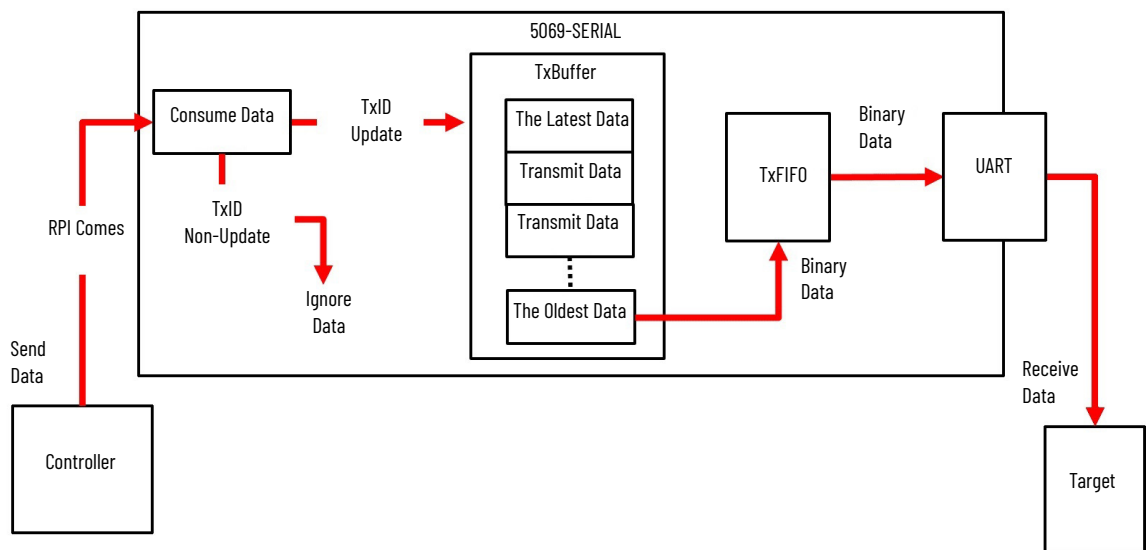
The following illustrations show different data exchanges using the serial module.

Data Sent with the Serial Port

Data is sent out of the serial port using the following steps:

1. The controller sends out the consumed packet with an incremented TxID.
2. The serial module puts the packet into the TxBuffer.
3. The serial module takes the oldest data from the TxBuffer and sends it out to the target.

Figure 6 - Data Sent Out to the Serial Port



Generic ASCII Transmit Functions

In Generic ASCII mode, you can define any kind of data to be transmitted to communicate to serial devices. Some examples are barcode scanners, dial up modems, serial printers, temperature controllers, and so on.

Table 8 - Generic ASCII Transmit Functions

Parameter	Definition	Available Options
Swap Mode	Select whether swapping will be done or what swapping method is to be used before the module sends output data to the Serial Port or after the module receives input data from the Serial Port.	<ul style="list-style-type: none"> No Change (default) Word Swap - After executing a word swap on 32-bit hex value 11112222, the result is 22221111. Byte Swap - After executing a byte swap on 32-bit hex value 11223344, the result is 22114433. Word and Byte Swap - After executing a word and byte swap on 32-bit hex value 11223344 the result is 44332211.
Termination Mode/ Delimiter	Selecting the termination mode of each channel. Choose to ignore or include the delimiter.	<ul style="list-style-type: none"> Ignore End Delimiter (default) - Transmits packet based on number of bytes specified only. Exclude Delimiter - Determines end of data when it finds the Termination Delimiter characters, but doesn't transmit the Termination Delimiter bytes with the packet. Include Delimiter - Determines end of data by Termination Delimiters and transmits them.
Termination Delimiter 1	Configure the channel's termination delimiter.	<ul style="list-style-type: none"> 7 bit \$00...\$7F 8 bit \$00...\$FF
Termination Delimiter 2	Configure the channel's termination delimiter.	<ul style="list-style-type: none"> 7 bit \$00...\$7F 8 bit \$00...\$FF (\$FF = disabled)

Generic ASCII Transmit Methods

To transmit the ASCII packet based on the number of characters, follow these steps:

- For this method, configure Termination Mode for Ignore End Delimiter.
- After copying the characters into the ASCII.TxData output tag array, write the number of characters into the ASCII.TxDataLength output tag, then increment the ASCII.TxID output tag.

To transmit the ASCII packet based on the termination delimiter characters, follow these steps:

- For this method, configure Termination Mode for either Include or Exclude.
- After copying the characters into the ASCII.TxData output tag array, copy the two configured termination delimiter characters as the next two characters in the array, then increment the ASCII.TxID output tag. While doing this, keep the TxDataLength output tag at 0.



Include transmits the packet with termination delimiter characters, and exclude does not.

Data Received from the Serial Port

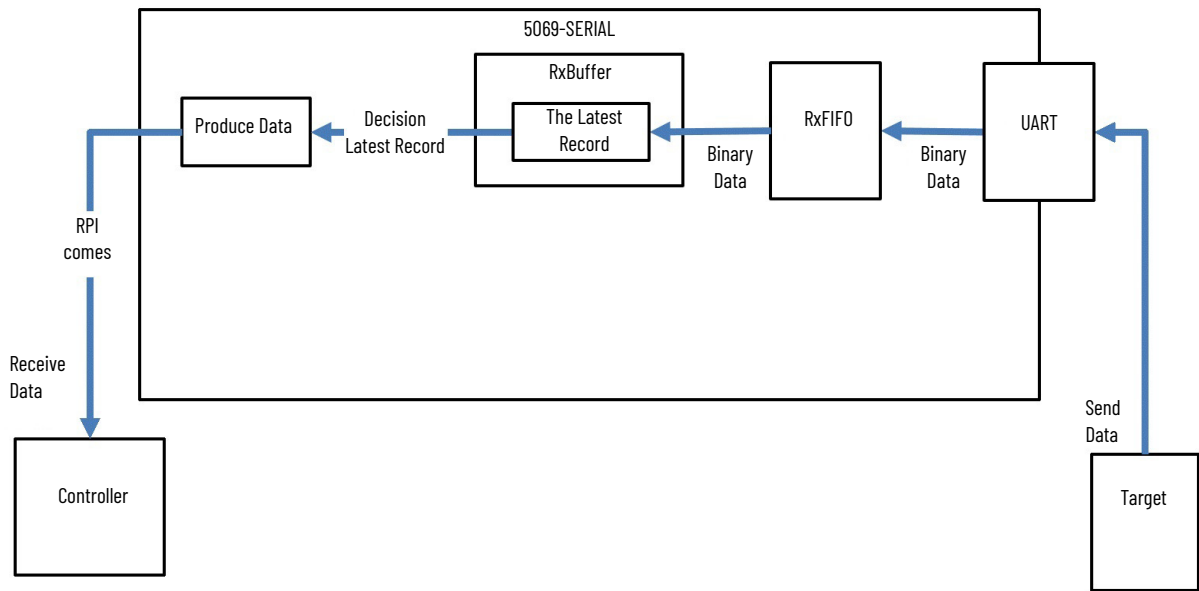
Immediate Mode

After the serial port receives data, it processes the data using the following steps.

- The serial module receives the packet.
- If any of the following conditions occur, the serial module creates a record.
 - Message Timeout timer expires.
 - The number of received bytes equals the configured Read Buffer Size.
 - Termination Delimiter bytes were received.

3. The serial module copies the data into the ASCII.RxData input tag, copies the number of characters that are received into the ASCII.RxDataLength input tag, increments the ASCII.RxID input tag and sends it to the controller.

Figure 7 - Data Received from the Serial Port in Immediate Mode

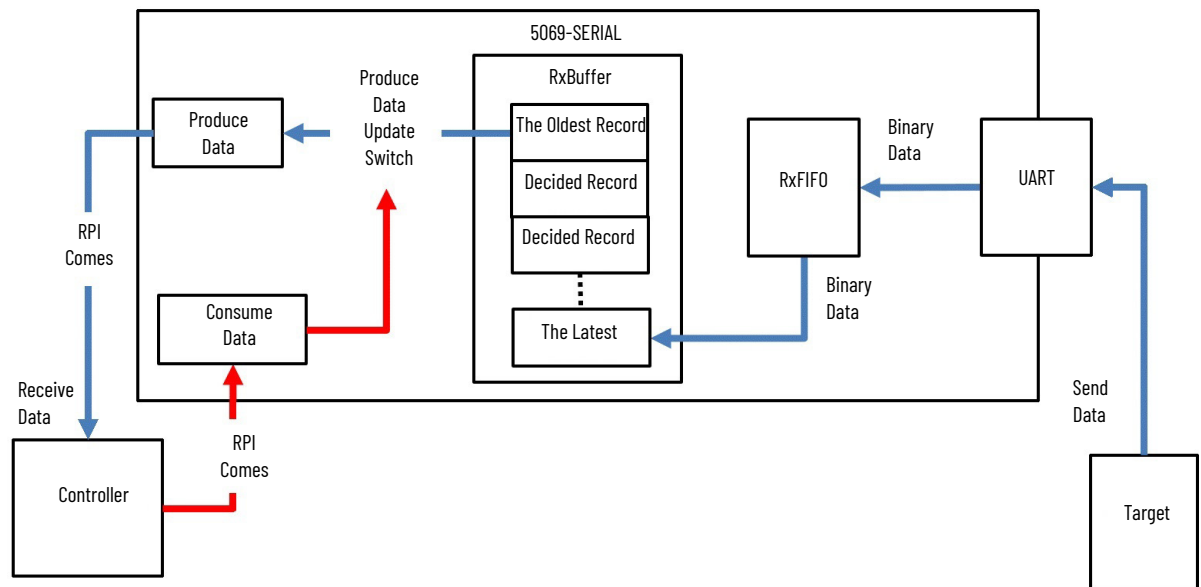


Master/Slave Handshake Mode

The serial port works in a handshake mode using the following steps.

1. The serial module receives the packet.
2. If any of the following conditions occur, the serial module creates a record.
 - a. The Message Timeout timer expires.
 - b. The number of received bytes equals the configured Read Buffer Size.
 - c. Termination Delimiter bytes were received.
3. The record is added into the RxBuffer.
4. Once the RxID (Consume Tag) is incremented by the user logic, the serial module takes the oldest record from RxBuffer, copies the data into the ASCII.RxData input tag, copies the number of characters that are received into the ASCII.RxDataLength input tag, and sends it to the controller.

Figure 8 - Serial Port Handshake Mode



IMPORTANT If Master/Slave Handshake is selected, dispose additional receiving data from the serial port for saved data in the receiving buffer of the firmware.

Generic ASCII Receive Functions

Table 9 - Generic ASCII Receive Functions

Swap Mode	Swapping is done before the module sends output data to the Serial Port or after the module receives input data from the Serial Port.	<ul style="list-style-type: none"> • No Change (default) • Word Swap - After executing a word swap on 32-bit hex value 11112222, the result is 22221111. • Byte Swap - After executing a byte swap on 32-bit hex value 11223344, the result is 22114433. • Word and Byte Swap - After executing a word and byte swap on 32-bit hex value 11223344 the result is 44332211.
Handshake Mode	Determines how the serial module passes the received data from the serial port to the controller.	<ul style="list-style-type: none"> • Master/Slave (default) - User logic must increment the ASCII.RxID output tag to receive the next packet of data into the ASCII.RxData input tag. • Immediate - ASCII.RxID input tag increments automatically indicating that the next packet of received data is available in the ASCII.RxData input tag.
Message Timeout	The timer resets every time that the module receives a new byte from the Serial Port. If a Timeout occurs, the Non-Delimited Flag is set, and Received Data is regarded as a new record to produce.	<ul style="list-style-type: none"> • 0 = Disabled (default) • 1...32,767 ms
Pad Character	Character that is used to fill the remainder of the ASCII.RxData array after the end of the received packet characters. Padding range is I.RxData[RxDataLength] to I.RxData[Read Buffer Size].	<ul style="list-style-type: none"> • 7 bit \$00...\$7F • 8 bit \$00...\$FF
Start Mode/ Delimiter	Select the usage of the Start Delimiter in the communication frame. Choose to ignore, exclude, or include the delimiter.	<ul style="list-style-type: none"> • Ignore Start Delimiter (default) - Start of received packet not based on the Start Delimiter. • Exclude - Start of packet is determined based on the Start Delimiter character, but this byte is not included in the ASCII.RxData input tag. • Include - Start Delimiter is always the first byte in the ASCII.RxData input tag.
Start Delimiter	Beginning of the message.	<ul style="list-style-type: none"> • 7 bit \$00...\$7F • Recommendation: When Delete Mode is enabled, do not configure Start Delimiter to DEL character. • 8 bit \$00...\$FF
Termination Mode/ Delimiter	Selecting the termination mode of each channel. Choose to ignore or include the delimiter.	<ul style="list-style-type: none"> • Ignore End Delimiter (default) - Receives packet based on number of bytes specified only. • Exclude Delimiter - Determines end of data when it finds the Termination Delimiter characters, but these bytes are not included in the ASCII.RxData input Tag. • Include Delimiter - Termination Delimiters bytes are included in the ASCII.RxData input tag.
Termination Delimiter 1	Configure the channel's termination delimiter.	<ul style="list-style-type: none"> • 7 bit \$00...\$7F • Recommendation: When Delete Mode is enabled, do not configure Termination Delimiter to DEL character. • 8 bit \$00...\$FF
Termination Delimiter 2	Configure the channel's termination delimiter.	<ul style="list-style-type: none"> • 7 bit \$00...\$7F • 8 bit \$00...\$FF (\$FF = disabled)
XON/XOFF	Selecting the flow control of each channel. Enables software handshaking.	<ul style="list-style-type: none"> • 0 = Disable (default) • 1 = Enable
Echo Mode	The module sends all bytes received from Serial Port immediately to the serial port by 1 byte, and sends produced data to the controller. Enables retransmission of all received characters.	<ul style="list-style-type: none"> • 0 = Disable (default) • 1 = Enable
Delete Mode	If the mode is ignored, it is handled as regular ASCII data. Choosing CRT means that the module does not send previous data and is replaced by three characters. Choosing printer means that the module does not send previous data and is replaced by one character.	<ul style="list-style-type: none"> • 0 = Ignore (default) - echoes DEL character same as any other character • 1 = CRT - receives DEL character, echoes backspace, space, backspace • 2 = Printer - receives DEL character, echoes '/' followed by previous character
Read Buffer Size	Max buffer length supported.	<ul style="list-style-type: none"> • 1...256 (default = 256)

Generic ASCII Receive Methods

When receiving the ASCII packet based on a fixed number of characters, the number of characters is configured in the Read Buffer Size parameter.

- For this method, configure Termination Mode for Ignore End Delimiter.

When receiving the ASCII packet based on timeout since the last character was received, the timeout in milliseconds is configured in the Message Timeout parameter.

- For this method, configure Termination Mode for Ignore End Delimiter.

When receiving the ASCII packet based on Termination delimiters at the end of the packet, if termination mode is “Include”, then the termination bytes remain appended to the end of the data copied into the ASCII.RxData input tag. If the termination mode is “Exclude”, then the termination bytes are stripped off.

For Generic ASCII Sample Code, see [page 121](#).

Modbus Master Data Exchange

For the Modbus Master data exchange, the following definitions apply:

- Write—the Modbus Master writes data to the Modbus Slave.
- Read—the Modbus Master reads data from the Modbus Slave.

The Compact 5000 I/O Serial Module can get Modbus Data from Produced/Consumed Data command, every RPI.

Figure 9 - Modbus Master Write Command

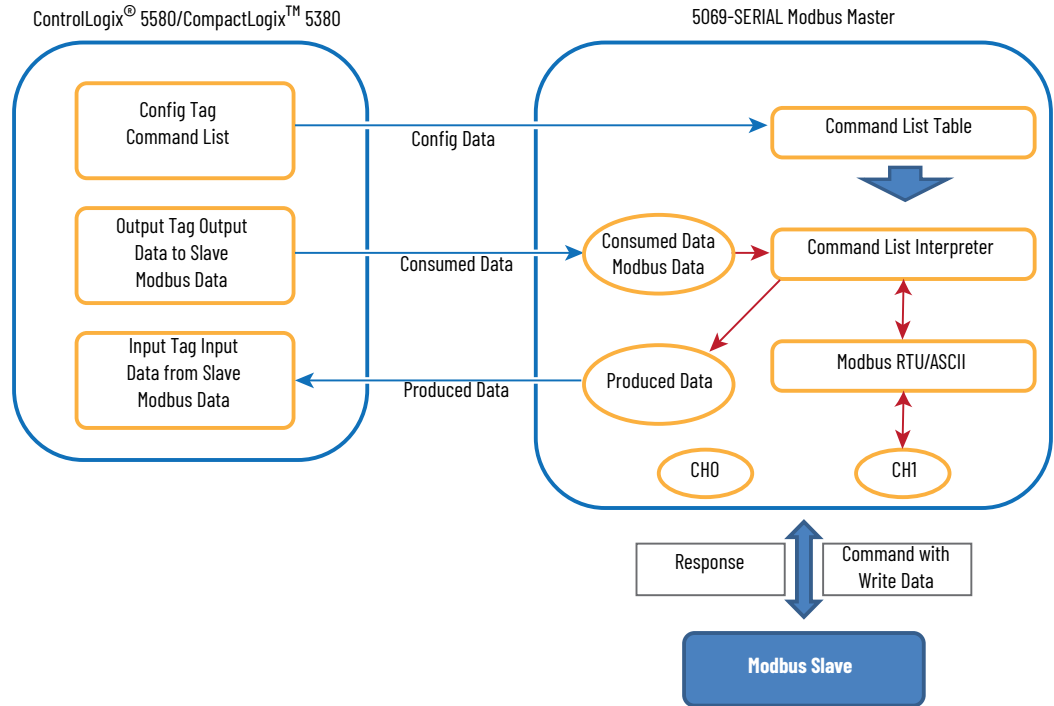
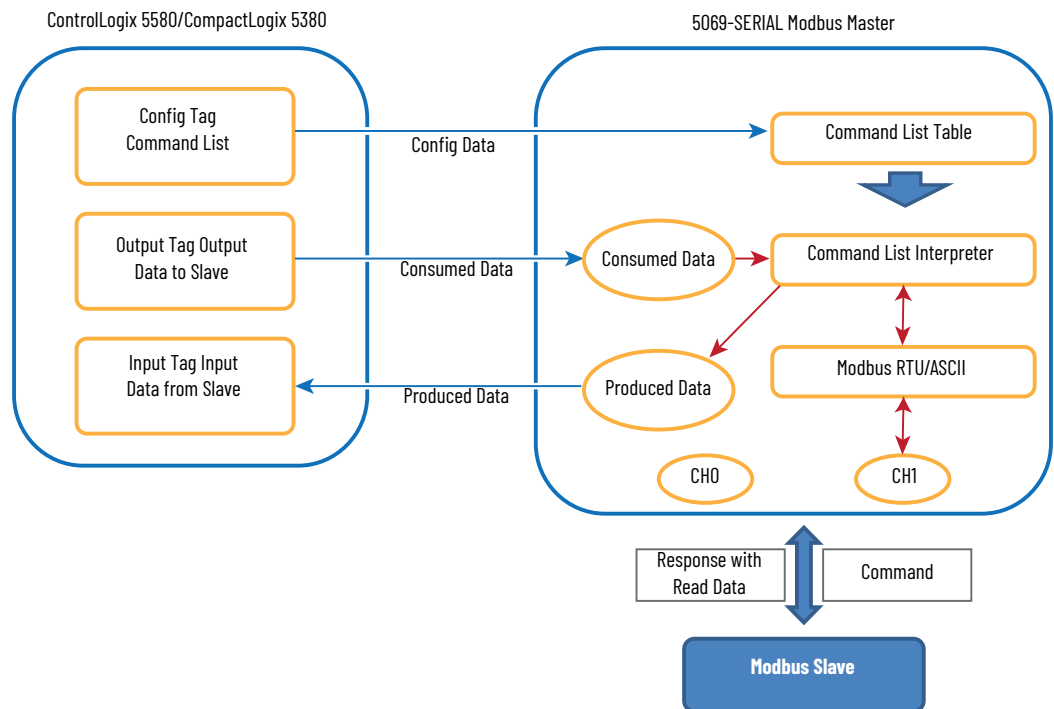


Figure 10 - Modbus Master Read Command



Modbus Master Functions

In Modbus mode, the serial module supports both Modbus ASCII and Modbus RTU to connect to Modbus enabled devices like flowmeters, power meters or remote RTU talking on Modbus RTU/ASCII.

Table 10 - Modbus Master Configuration Parameters

Parameter	Definition	Available Options
Modbus Format	Selecting communication method of Modbus of each channel.	<ul style="list-style-type: none"> 0 = RTU (default) 1 = ASCII Intervals of up to one second may elapse between characters within the message. Unless the user has configured a longer timeout, an interval greater than 1 second means that an error has occurred.
Response Timeout	The Maximum Delay time of each channel until it receives a response for Modbus Master Command from Modbus Slave. When the maximum time has passed, the Modbus Master determines the Modbus Slave did not respond.	<ul style="list-style-type: none"> 0...3600000 (x1 ms) (default = 200)
Broadcast Pause	The Wait time of each channel until the Modbus Slave finishes processing according to broadcasted command from the Modbus Master. Modbus Master does not send the next command until this time has passed.	<ul style="list-style-type: none"> 0...3600000 (x1 ms) (default = 200)
Inter-frame Timeout	Maximum delay time to receive Data of each channel	<ul style="list-style-type: none"> 0 is not a valid value. Use Table 11 and Table 12 to determine the minimum value.
Retry Count	This parameter specifies the number of times a command is retried if it fails. If Master Port does not receive a response after the last retry, the Slave devices communication is suspended on the port for Error Delay Counter scans.	<ul style="list-style-type: none"> 0...127 (default = 0)

Table 11 - Inter-frame Timeout Minimum Values (10 Bit)

Communication Rate	Default Value	1 Byte (10 Bit)	3.5 t (10 Bit)	Legal Range (10 Bit)
1200	58000	8333.333	29166.667	29000...65535000 (us)
2400	28000	4166.667	14583.333	14000...65535000 (us)
4800	14400	2083.333	7291.667	7200...65535000 (us)
9600	7200	1041.667	3645.833	3600...65535000 (us)
19200	3500	520.833	1822.917	1750...65535000 (us)
38400	3500	260.417	911.458	1750...65535000 (us)
57600	3500	173.611	607.639	1750...65535000 (us)
115200	3500	86.806	303.819	1750...65535000 (us)

Table 12 - Inter-frame Timeout Minimum Values (11 Bit)

Communication Rate	Default Value	1 Byte (11 Bit)	3.5 t (11 Bit)	Legal Range (11 Bit)
1200	64000	9166.667	32083.333	32000...65535000 (us)
2400	32000	4583.333	16041.667	16000...65535000 (us)
4800	16000	2291.667	8020.833	8000...65535000 (us)
9600	8000	1145.833	4010.417	4000...65535000 (us)
19200	3500	572.917	2005.208	1750...65535000 (us)
38400	3500	286.458	1002.604	1750...65535000 (us)
57600	3500	190.972	668.403	1750...65535000 (us)
115200	3500	95.486	334.201	1750...65535000 (us)

Command Disable function

In firmware revision 2.011 and later, the user is now able to disable a Modbus Master command at runtime by controlling the disable switch bit in the output array of the Modbus Master channel. A total of 50 bits are available with each bit representing a Modbus Master command. A disabled command would have the corresponding Ixx.CommandXX.Disabled bit set. To disable a Modbus

command, turn ON the disable corresponding bit in the output array.
 Example: Command o disable bit is referring to the Oxx.CommandoDisable bit.

Modbus Slave Data Exchange

For the Modbus Slave data exchange, the following definitions apply:

- Write—the Controller and Modbus Master can write data to the Modbus Slave.
- Read—the Modbus Master reads data from the Modbus Slave.

Figure 11 - Modbus Slave Write Command

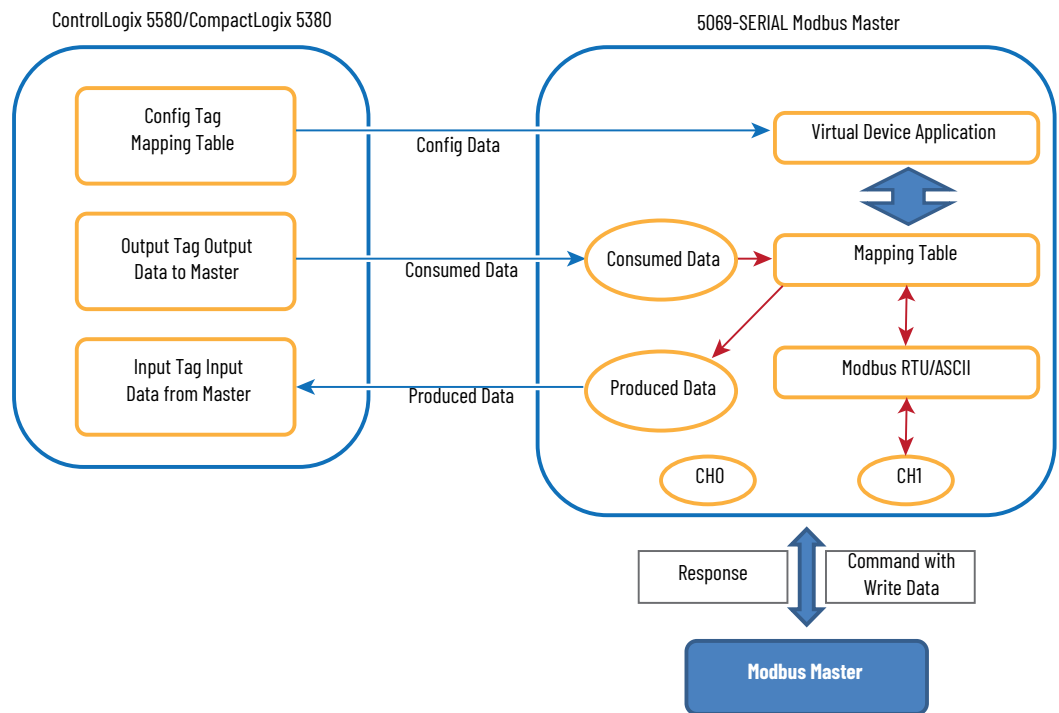
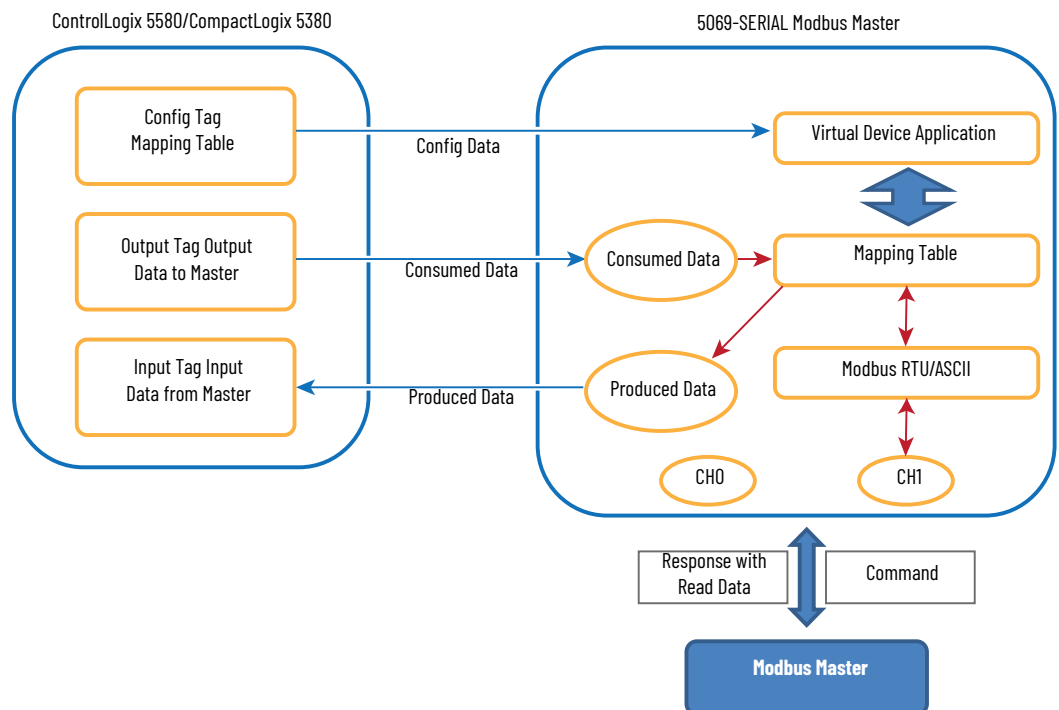


Figure 12 - Modbus Slave Read Command



Modbus Slave Functions

Table 13 - Modbus Slave Configuration Parameters

Parameter	Definition	Available Options
Modbus Format	Selecting communication method of Modbus of each channel.	<ul style="list-style-type: none"> • 0 = RTU (default) • 1 = ASCII - Intervals of up to one second may elapse between characters within the message. Unless the user has configured a longer timeout, an interval greater than 1 second means that an error has occurred.
Node Address	Numbers to identify all modules that are connected to each channel. You must set a number not equal to 0.	<ul style="list-style-type: none"> • 1..247 (default = 1)
Inter-frame Timeout	Maximum delay time to receive Data of each channel	<ul style="list-style-type: none"> • 0 is not a valid value. Use Table 11 and Table 12 to determine the minimum value.

Table 14 - Modbus Slave Supported Commands

Function Code as a Slave(in hex)	Function Name
0x01	Read Coil
0x02	Read Discrete Inputs
0x03	Read Holding Registers
0x04	Read Input Register
0x05	Write Single Coil
0x06	Write Single Register
0x16	Mask Write Register
0x17	Read/Write Multiple Registers
0x0F	Write Multiple Coil
0x10	Write Multiple Registers

DF1 Protocols

The serial module with firmware revision 2.011 and later with Add-On profile (AOP) version 2.01 supports DF1 protocols DF1 Point to Point, DF1 Master, DF1 Slave, and DF1 Radio modem. The combination of the revision and Add-On profile enables the serial module to connect to remote devices such as SLC™, MicroLogix™ 1400, and 1769-L32E.

The details of these protocols can be found in the DF1 Protocol and Command Set Reference Manual, publication 1770-RM516.

IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

Using Modems that Support DF1 Communication Protocols

The types of modems that you can use with the serial module include dial-up phone modems, leased-line modems, radio modems, and line drivers. For point-to-point full-duplex modem connections, use DF1 full-duplex protocol. For point-to-multi-point modem connections, use DF1 half-duplex master and slave protocols. In this case, one (and only one) of the other devices must be configured for DF1 half-duplex master protocol.

IMPORTANT Do not attempt to use DH-485 protocol through modems under any circumstance.

Dial-up Phone Modems

Dial-up phone line modems support point-to-point full-duplex communications. Normally, a serial module on the initiating or receiving end of the dial-up connection, will be configured for DF1 full-duplex protocol with the control line parameter set for 'Full-Duplex Modem.' See [Modem Control Line Operation on page 40](#) for details on the operation of the RS-232 modem control signals when 'Full-Duplex Modem' is selected.

When the 5069-SERIAL is the initiator of the dial-up connection, use the native tag similar to the Generic ASCII mode to send out the 'AT' dial-up string (example: ATDT 555-1212). To hang up the connection, use the DTR signal in the output tag.

Leased-Line Modems

Leased-line modems are used with dedicated phone lines that are typically leased from the local phone company. The dedicated lines may be point-to-point topology supporting full-duplex communications between two modems or in a point-to-multi-point topology supporting half-duplex communications between three or more modems. In the point-to-point topology, configure the serial module for DF1 full-duplex protocol with the control line parameter set to "Full-Duplex Modem." In the point-to-multi-point topology, configure the serial module for DF1 half-duplex master or slave protocol with the control parameter set to "Half-Duplex Modem without Continuous Carrier." See [DF1 Half-Duplex Slave on page 41](#) for details on the operation of the RS-232 modem control signals when "Half-Duplex Modem without Continuous Carrier" is selected.

Radio Modems

Radio modems may be implemented in a point-to-point topology supporting either half-duplex or full-duplex communications, or in a point-to-multi-point topology supporting half-duplex communications between three or more modems. In the point-to-point topology using full-duplex radio modems, configure the serial module for DF1 full-duplex protocol. In the point-to-multi-point topology using half-duplex radio modems, configure the serial module for DF1 half-duplex master or slave protocol. If these radio modems require RTS/CTS handshaking, configure the control line parameter to “Half-Duplex Modem without Continuous Carrier.”

Line Drivers

Line drivers, also called short-haul modems, do not actually modulate the serial data, but rather condition the electrical signals to operate reliably over long transmission distances (up to several miles). The Allen-Bradley® AIC+ Advanced Interface Converter is a line driver that converts an RS-232 electrical signal into an RS-485 electrical signal, increasing the signal transmission distance from 50 to 4000 feet. In a point-to-point line driver topology, configure the serial module for DF1 full-duplex protocol. In a point-to-multipoint line driver topology, configure the serial module for DF1 half-duplex master or slave protocol. If these line drivers require RTS/CTS handshaking, configure the control line parameter to “Half-Duplex Modem without Continuous Carrier.”

Modem Control Line Operation

The following explains the operation of the serial module when you configure the RS-232 channel for the following applications.

DF1 Full-Duplex

When configured for DF1 full-duplex, the following control line operation takes effect:

- No Handshaking Selected—DTR is always active (high) and RTS is always inactive (low). Receptions and transmissions take place regardless of the states of DSR, CTS, or DCD inputs. Only make this selection when the serial module is directly connected to another device that does not require handshaking signals.
- Full-Duplex Modem Selected—DTR and RTS are always active except:
 - If DSR goes inactive, both DTR and RTS are dropped for 1 to 2 seconds, then reactivated. While DSR is inactive, the state of DCD is ignored. Neither receptions or transmissions are performed.
 - If DCD goes inactive while DSR is active, then receptions are not allowed. If DCD remains inactive for 9 to 10 seconds, DTR is set inactive. DTR is raised again in 5 to 6 seconds.

Reception requires DSR and DCD to be active. Transmission requires all three inputs (CTS, DCD, and DSR) to be active.

DF1 Half-Duplex Slave

When configured for DF1 half-duplex slave, the following control line operation takes effect:

- No Handshaking Selected—DTR is always active and RTS is always inactive. Receptions and transmissions take place regardless of the states of DSR, CTS, or DCD inputs. Only make this selection when the processor is directly connected to another device that does not require handshaking signals.
- Half-Duplex Modem with Continuous Carrier Selected—DTR is always active and RTS is only activated during transmissions (and any programmed delays before or after transmissions). The handling of DCD and DSR are the same as with Full-Duplex Modem. Reception requires DSR and DCD to be active. Transmissions require CTS, DCD, and DSR to be active.
- Half-Duplex Modem without Continuous Carrier Selected—This is the same as Half-Duplex Modem with Continuous Carrier except monitoring of DCD is not performed. DCD is still required for receptions, but is not required for transmissions. Transmissions still require CTS and DSR.

DF1 Half Duplex Master

When configuring for DF1 half-duplex master, the following control line operation takes effect:

- No Handshaking Selected—DTR is always active and RTS is always inactive. Receptions and transmissions take place regardless of the states of DSR, CTS, or DCD inputs. Only make this selection when the processor is directly connected to another device that does not require handshaking signals.
- Full-Duplex Modem Selected—DTR and RTS are always active except at the following times:
 - If DSR goes inactive, both DTR and RTS are dropped for 1 to 2 seconds then reactivated. While DSR is inactive, the state of DCD is ignored. Neither receptions or transmissions are performed.
 - If DCD goes inactive while DSR is active, then receptions are not allowed. If DCD remains inactive for 9 to 10 seconds, then DTR is set inactive. If DSR remains active, then DTR is raised again in 5 to 6 seconds.

Reception requires DSR and DCD to be active. Transmission requires all three inputs (CTS, DCD, and DSR) to be active.

- Half-Duplex Modem without Continuous Carrier Selected—DTR is always active and RTS is only active during transmissions (and any programmed delays before and after transmissions). The serial module does not monitor DCD.

If DSR goes inactive, RTS is dropped. While DSR is inactive, neither receptions or transmissions are performed.

Reception requires DSR to be active. Transmission requires two inputs, CTS and DSR, to be active.

Configuration Considerations for RTS Send and Off Delays

Through your programming software, the parameters RTS Send Delay and RTS Off Delay give you the ability to set how long RTS is on before transmission and how long to keep it on after transmission is complete. These parameters only apply when you select half-duplex modem with or without continuous carrier. For maximum communication throughput, leave these parameters at zero.

For use with half-duplex modems that require extra time to turnaround or key-up their transmitter even after they have activated CTS, the RTS Send Delay specifies (in 20 millisecond increments) the amount of delay time after activating RTS to wait before checking to see if CTS has been activated by the modem. If CTS is not yet active, RTS remains active, and as long as CTS is activated within one second, the transmission occurs. After one second, if CTS is still not activated, then RTS is set inactive and the transmission is aborted.

For modems that do not supply a CTS signal but still require RTS to be raised before transmission, jumper RTS to CTS and use the shortest delay possible without losing reliable operation.

If an RTS Send Delay of 0 is selected, then transmission starts as soon as CTS is activated. If CTS does not go active within one second after RTS is raised, RTS is set inactive and the transmission is aborted.

Certain modems will drop their carrier link when RTS is set inactive even though the transmission has not been finished. The RTS Off Delay parameter specifies in 20 millisecond increments the delay between when the last serial character is sent to the modem and when RTS is deactivated. This gives the modem extra time to transmit the last character of a packet.

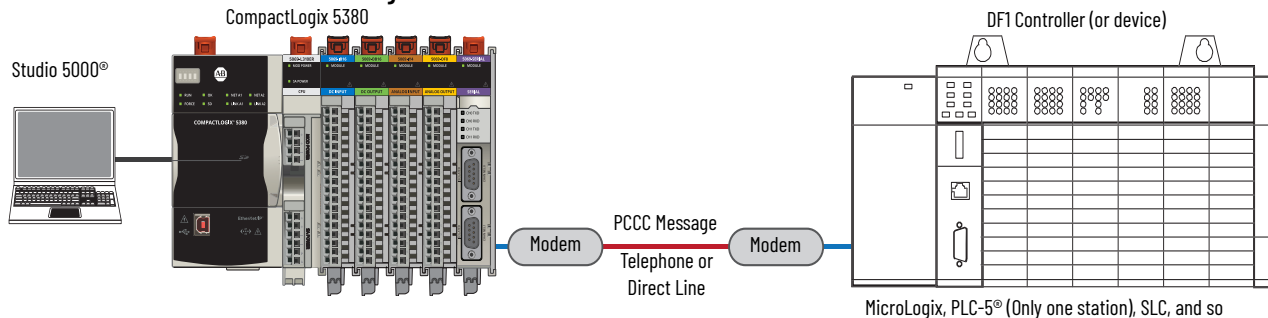
IMPORTANT For almost all modem applications, the RTS Off Delay should be left at 0. Never Select an RTS Off Delay that is greater than the RTS Send Delay in the other devices on the network, or you can incur two devices that try to transmit simultaneously

DF1 Point-to-Point Protocol

The DF1 Point to Point protocol is used when connecting from the controller to one DF1 device (or another controller).

This type of protocol supports simultaneous transmissions between two controllers (or devices) in both directions.

Figure 13 - DF1 Point-to-Point Protocol



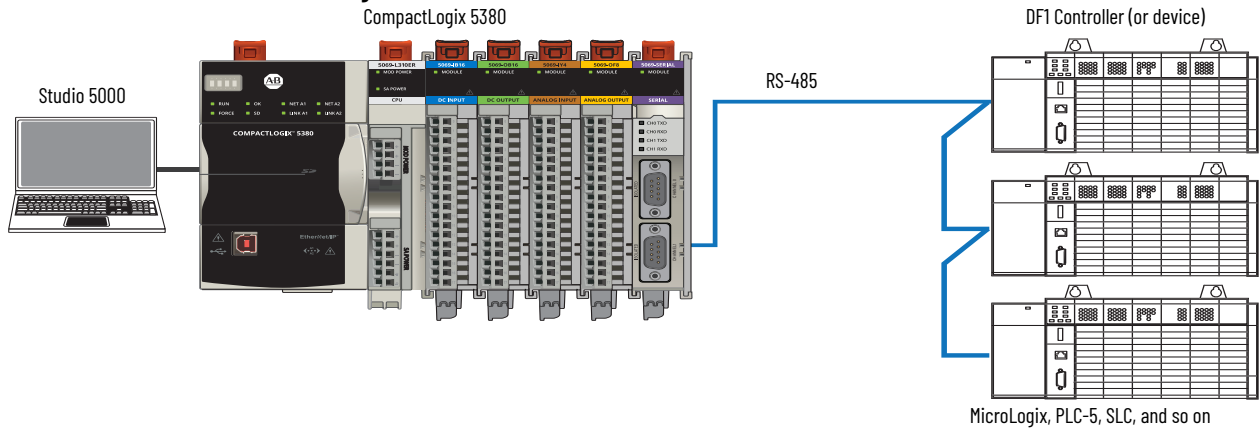
IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

DF1 Master Protocol

The DF1 Master protocol is used when connecting from the controller to multiple devices.

In DF1 Master protocol, RS-485 and Serial Radio Modem can be used.

Figure 14 - DF1 Master Protocol
CompactLogix 5380

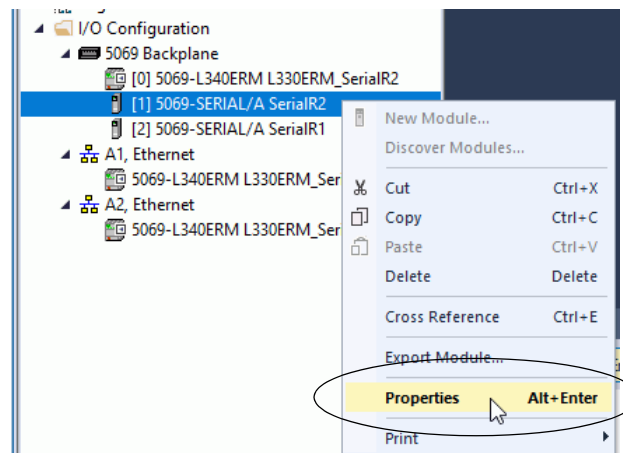


IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

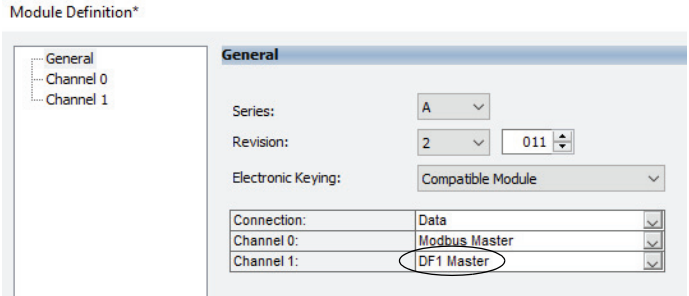
Configuring a Master Station for Standard Polling Mode

To configure a serial module as a master station using standard communication, complete the following steps.

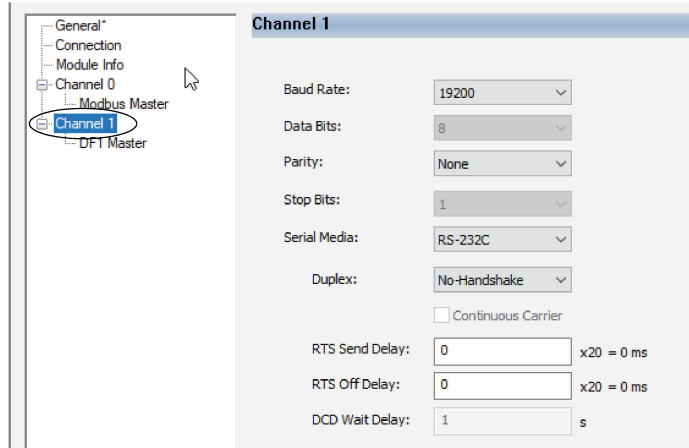
1. Open the Logix Designer application project that includes the serial module.
2. Right-click the module and choose Properties. Configure Channel 1 to DF1 Master if it is not configured to DF1 Master



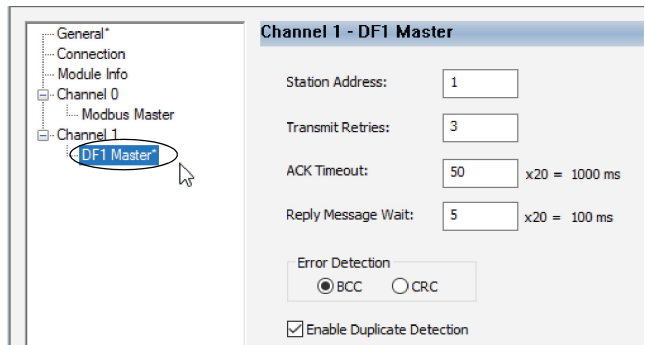
- Verify that DF1 Master is selected under channel one. Configure Channel 1 to DF1 Master if it is not configured to DF1 Master.



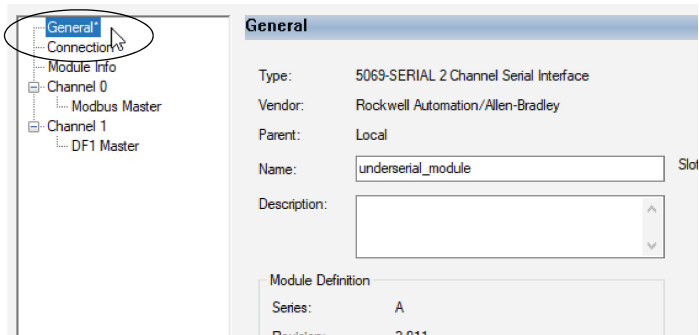
- Click Channel 1 and configure the parameters as necessary.



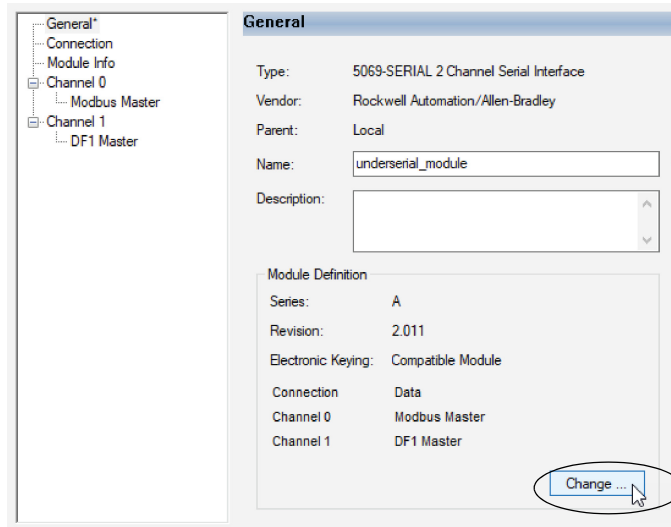
- Click DF1 Master and configure the parameters as necessary.



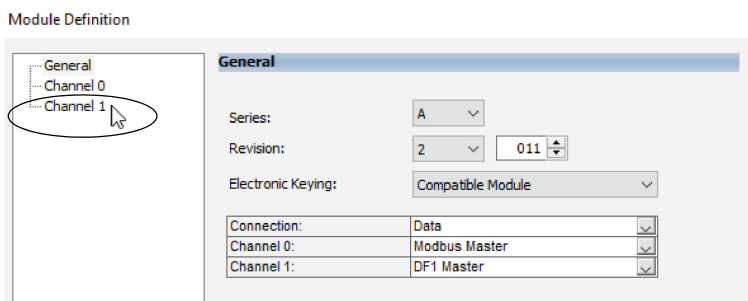
- Click General in Module properties.



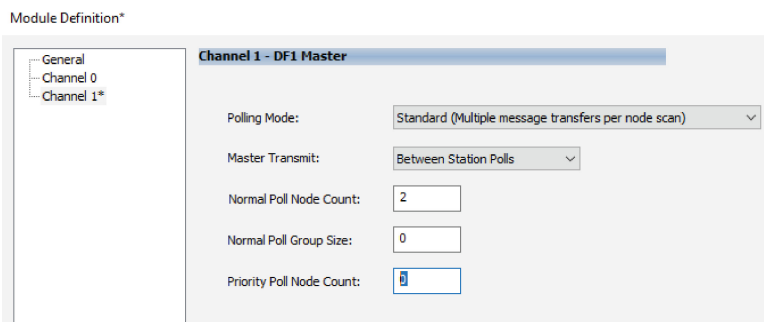
7. Click Change under General Module Properties.



8. Click Channel 1 under Module Definition.



9. Configure the polling settings as necessary.



10. Click OK to accept the changes and click Yes to close the dialog box.

Use the following table to understand the communication parameters needed for specification on the Channel Configuration screen for standard-communication mode.

Define the parameters in the following table when configuring a serial module as a master station using standard-communication mode to talk to slave stations.

Field	Description	Default
Station Address	The station address for the serial port on the DF1 master. Enter a valid DF1 address (0...254).	0
Transmit Retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value 0...127.	3
ACK Timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0...32767. Limits are defined in 20 ms intervals.	50(1000 ms)
Polling Mode	<ul style="list-style-type: none"> • Message based—Do not allow the slave to initiate messages • Message based—Allow the slave to initiate messages • Standard—Multiple message transfers per node scan • Standard—Single message transfer per node scan 	Default is Message-Based (Allow the Slave to initiate message)
Master Transmit	<p>If you want the master station to:</p> <ul style="list-style-type: none"> • Send all master station-initiated MSG instructions to the slave stations before polling the next slave station in the poll list, choose Between Station Polls. This method verifies that master station-initiated messages are sent in a timely and regular manner (after every slave station poll). • Only send master station-initiated MSG instructions when the master's station number appears in the polling sequence; choose In Poll Sequence. With this method, sending master station-initiated messages are dependent upon where and how often the master station appears in the poll list. To achieve the same goal as the Between Station Polls method, the master-station's address would have to appear after every slave-station's address. 	Between Station Polls
Normal Poll Node Count	<p>Contains the total number of stations to poll under Normal Poll that will allow the serial module to auto-create the following tags for control and monitoring.</p> <ul style="list-style-type: none"> • Latest Normal Poll Node in Local:x:1:DF1Master.LatestNormalPollNode • Maximum Normal poll node in Local:x:01:DF1Master.MaxNormalPollNode • Slave station address tags (depending on the number of stations that are entered in the Normal Poll Node Count) in Local:x:01:DF1Master.NormalPollNode[y] 	
Normal Poll Group Size	The number of normal stations the master station polls after polling all stations in the priority poll array.	0
Priority Poll Node Count	<p>Contains the total number of stations to poll under Priority Poll that will allow the serial module to auto-create the following tags for control and monitoring.</p> <ul style="list-style-type: none"> • Latest Priority Poll Node in Local:x:1:DF1Master.LatestPriorityPollNode • Maximum Priority poll node in Local:x:01:DF1Master.MaxPriorityPollNode • Slave station address tags (depending on the number of stations that are entered in the Normal Poll Node Count) in Local:x:01:DF1Master.PriorityPollNode[y] 	-
Error Detection	<p>Select BCC or CRC error detection. Configure all stations to use the same type of error checking.</p> <p>BCC—The serial module sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement, but does not detect as many errors as CRC</p> <p>CRC—the serial module sends and accepts messages with a 2-byte CRC for error checking. CRC is a more effective error detection algorithm.</p>	BCC
Enable Duplicate Detection	Select whether the serial module should detect and ignore duplicate messages.	Enabled

Minimum DF1 Half-Duplex Master ACK Timeout

The governing timeout parameter to configure for a DF1 Half-Duplex Master is the ACK Timeout. The ACK Timeout is the amount of time you want the serial module to wait for an acknowledgment of its message transmissions. Set in 20 millisecond intervals, the value is the amount of time the master will wait for:

- An ACK to be returned by a slave when the master has just sent it a message, or
- A poll response or message to be returned by a slave when the master has sent it a poll packet.

The timeout must be long enough that after the master has transmitted the last character of the poll packet, there is enough time for a slave to transmit (and the master receive) a maximum sized packet before the time expires.

To calculate the minimum ACK timeout, you must know:

- The modem communication rate
- Maximum sized data packet (the maximum number of data words that a slave write command or read reply packet might contain)
- The RTS/CTS or 'turnaround' delay of the slave modem
- The configured RTS Send Delay in the slave
- The program scan time of the slave

Determining Minimum Master Serial Port ACK Timeout

To determine the minimum ACK Timeout, you must first calculate the transmission time by multiplying the maximum sized data packet for your serial module by the modem rate in ms/byte. For an example, we will assume communications with SLC 5/03 slaves (103 data words or 224 ms. For approximate modem transmission rates, see the following table.

Next, you must determine the average slave program scan time. For this example, let's assume an average slave program scan time of 20 ms. Remember, program scan time will vary by application.

Table 15 - Approximate modem transmission rates

Modem bps	Approximate ms/byte
4800	2 ms/byte
9600	1 ms/byte
19200	0.5 ms/byte

Finally, you must determine the larger of two values, either the configured slave RTS Send Delay or the turnaround time of the slave modem. To find the RTS Send Delay time, navigate to the module definition page and select channel 1 to open the configuration screen. The RTS Send Delay time is in intervals of 20 ms, so with a value of 3 in the box, the RTS Send Delay time would be 20 ms multiplied by 3. Using this value (60 ms) for our example, and assuming that the turnaround time of the modem is 50 ms (which will vary by modem) you would choose to use the RTS Send Delay time of 60 ms for your calculation.

Having determined the maximum message transmission time (224 ms), the average slave program scan time (20 ms) and the largest of either RTS Send Delay (60 ms) or the modem turnaround time, the minimum ACK timeout is simply the sum of these values.

Parameter	Example Values (in ms)	
Max message transmission time	224	
Average program scan time	20	
RTS Send Delay	60	Use only the largest of these two values
modem turnaround time	50	
calculated ACK Timeout	304	
round up to nearest 20 ms	320	

Create Polling Lists

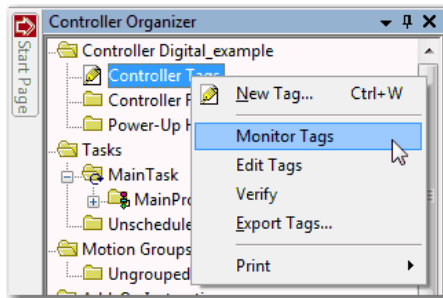
After defining your polling tags and group size, create polling lists by entering the station address of each slave station into either the normal poll node tag (Local:x:O1:DF1Master.NormalPollNode[y]) or priority poll node tag (Local:x:O1:DF1Master.PriorityPollNode[y]). Place each station address in an individual word in a poll tag (normal or priority).

The normal and priority poll file layout is as follows:

Poll Tag	Contains
Local:x:O1:DF1Master.MaxNormalPollNode Local:x:O1:DF1Master.MaxPriorityPollNode	Total number of stations to be polled (for a list)
Local:x:11:DF1Master.LatestNormalPollNode Local:x:11:DF1Master.LatestPriorityPollNode	The address of the station currently being polled This word is automatically updated by the master station as a new slave station is polled.
Local:x:O1:DF1Master.NormalPollNode[y] Local:x:O1:DF1Master.PriorityPollNode[y] Where y is 0 to xx	The slave station address in the order that the stations should be polled Store one station address in each word.

To place a station address in a poll file:

1. Right-click Controller Tags and choose Monitor Tags.



2. Expand the Local:x:O1:DF1Master.NormalPollNode and/or Local:x:O1:DF1Master.PriorityPollNode tag.
3. Enter the station addresses of the slave stations you want in the poll list. Put them in the order you want them polled. Enter the total number of slave station addresses in the Local:x:O1:DF1Master.MaxNormalPollNode and/or Local:x:O1:DF1Master.MaxPriorityPollNode

Figure 15 - Polling list example—Contains two stations: addresses 8 and 9 in Priority Poll. Station 9 is being polled.

Name	Value	Force Mask	Style	Data Type
Local:1:1	{...}	{...}	{...}	AB:5000_DF1_Master:0
Local:1:1.DF1	{...}	{...}	{...}	AB:5000_DF1_Channe...
Local:1:1.DF1Master	{...}	{...}	{...}	AB:5000_DF1_Master_...
Local:1:1.DF1Master.ActiveStation	{...}	{...}	Decimal	SINT[32]
Local:1:1.DF1Master.PriorityPollError	0		Decimal	BOOL
Local:1:1.DF1Master.NormalPollError	0		Decimal	BOOL
Local:1:1.DF1Master.PollGroupSizeError	0		Decimal	BOOL
Local:1:1.DF1Master.LatestPriorityPollNode	9		Decimal	INT
Local:1:1.DF1Master.LatestNormalPollNode	0		Decimal	INT
Local:1:000	{...}	{...}	{...}	AB:5000_ModbusMas...
Local:1:01	{...}	{...}	{...}	AB:5000_DF1_Master_...
Local:1:01.DF1Master	{...}	{...}	{...}	AB:5000_DF1_Master_...
Local:1:01.DF1Master.MaxPriorityPollNode	2		Decimal	INT
Local:1:01.DF1Master.MaxNormalPollNode	0		Decimal	INT
Local:1:01.DF1Master.PollingListUpdateID	0		Decimal	SINT
Local:1:01.DF1Master.PriorityPollNode	{...}	{...}	Decimal	INT[2]
Local:1:01.DF1Master.PriorityPollNode[0]	8		Decimal	INT
Local:1:01.DF1Master.PriorityPollNode[1]	9		Decimal	INT
Local:1:01.DF1Master.NormalPollNode	{...}	{...}	Decimal	INT[2]

Monitor Active Stations

To see what stations are active, view the active station tag. Each bit in the tag represents a station on the link. The stations are numbered in order as a continuous bit-stream file starting with the first bit in the first word. See [Figure 16](#). If the bit is a one, the station is active; if the bit is a zero, the station is inactive.

Figure 16 - Example Active Station File

Name	Value	Force Mask	Style	Data Type
Local:1:1.DF1Master	{...}	{...}	{...}	AB:5000_DF1_Master_...
Local:1:1.DF1Master.ActiveStation	{...}	{...}	Binary	SINT[32]
Local:1:1.DF1Master.ActiveStation[0]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[1]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[2]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[3]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[4]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[5]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[6]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[7]	2#0000_0000		Binary	SINT
Local:1:1.DF1Master.ActiveStation[8]	2#0000_0000		Binary	SINT

Configuring a Message-Based Mode DF1 Half-Duplex Master Station

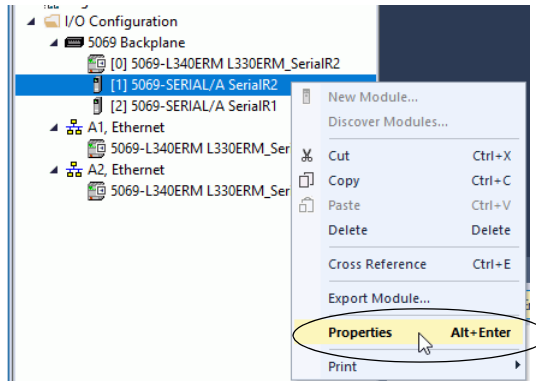
Choose message-based communication mode if you want to use MSG instructions in user programming to communicate with one station at a time. If your application uses satellite transmission or public switched telephone network transmission, consider choosing message-based. Communication to a slave station can be initiated on an as-needed basis.

With message-based mode, you do not have an active station file that you can use to monitor station status. Also, you cannot implement slave station-to-slave station messaging over the telemetry network.

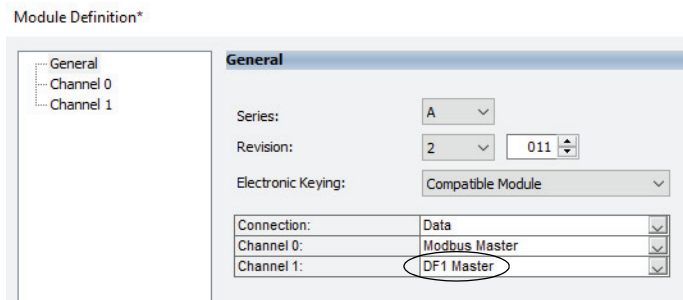
Configuring a Master Station for Message-based Polling Mode

To configure a serial module as a master station using message-based communication, complete the following steps.

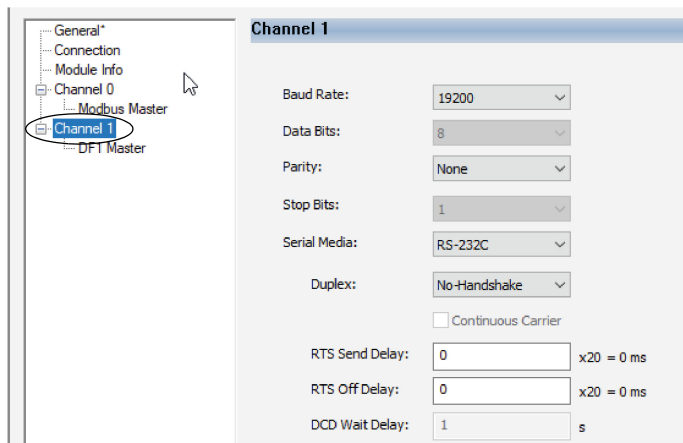
1. Open the Logix Designer application project that includes the serial module.
2. Right-click the module and choose Properties. Configure Channel 1 to DF1 Master if it is not configured to DF1 Master.



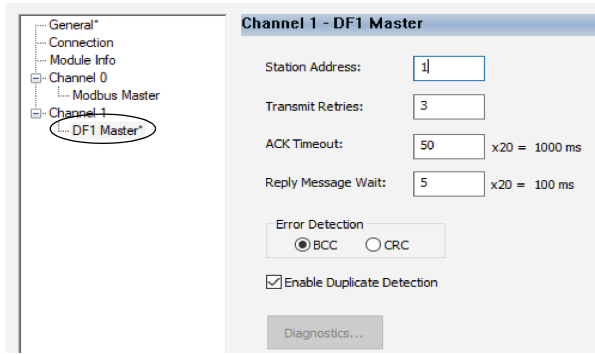
3. Verify that DF1 Master is selected under channel one. Configure Channel 1 to DF1 Master if it is not configured to DF1 Master.



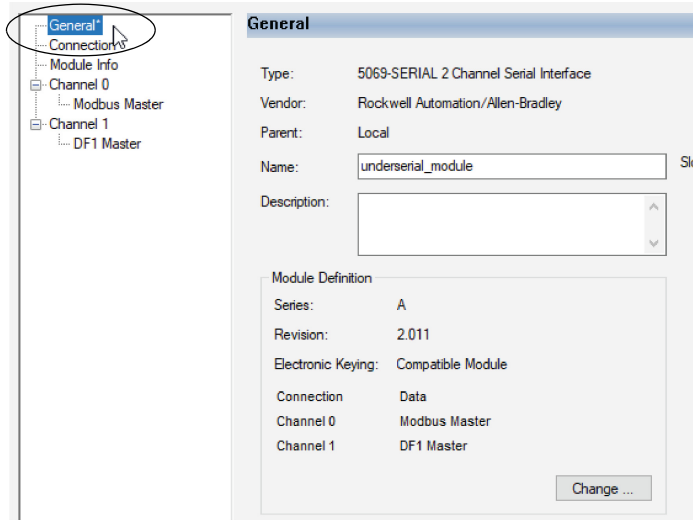
4. Click Channel 1 and configure the parameters as necessary.



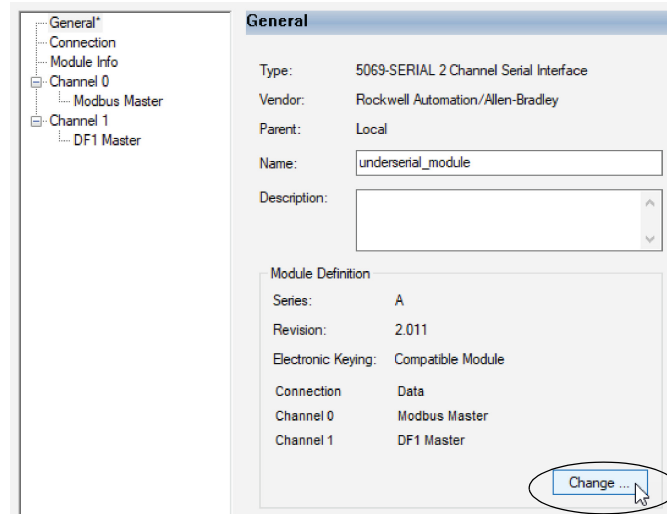
5. Click DF1 Master under the Channel 1 category on the Module Properties dialog box to configure DF1 Master port settings. See [DF1 Master Parameters on page 59](#)



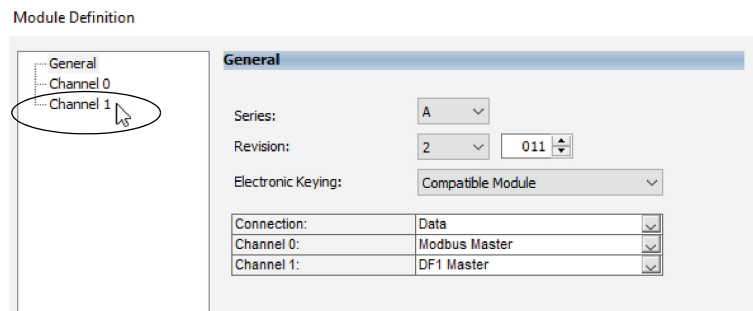
6. Click General in Module properties.



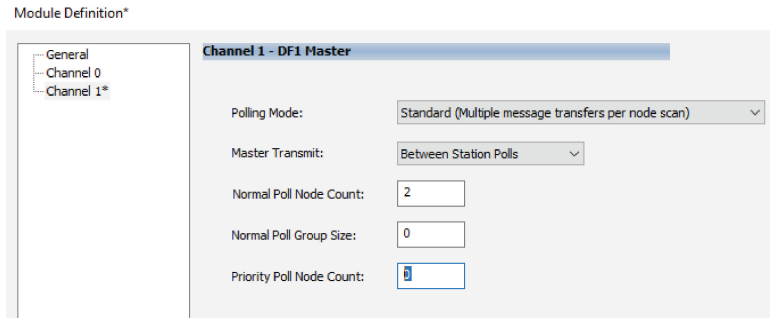
7. Click Change under General Module Properties.



8. Click Channel 1 under Module Definition.



9. Configure the polling settings as necessary.



10. Click OK to accept the changes and click Yes to close the dialog box.

Use the following table to understand the communication parameters needed to configure a serial module as a master station using message-based communication mode parameter.

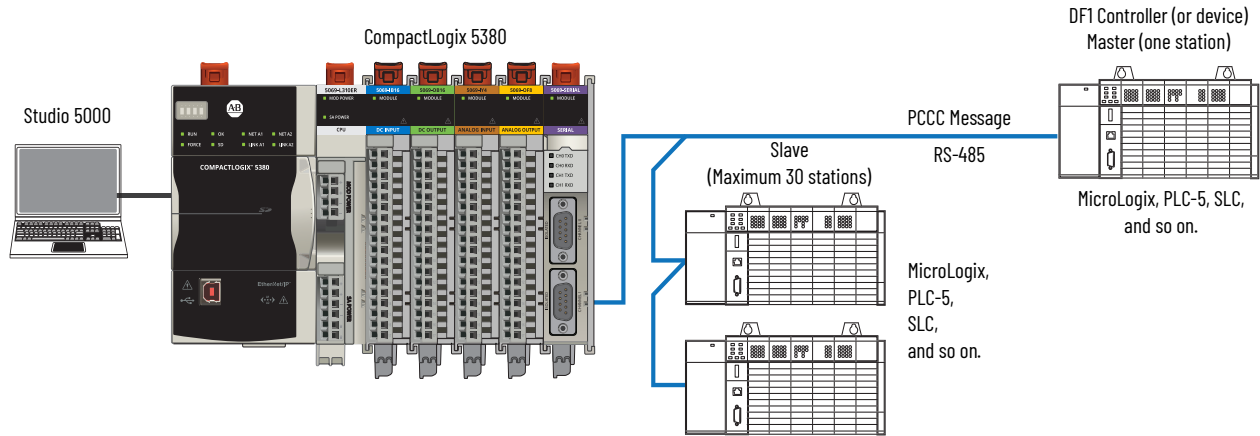
Field	Description	Default
Station Address	The station address for the serial port on the DF1 master. Enter a valid DF1 address (0...254).	0
Transmit Retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value 0...127.	3
ACK Timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0...32767. Limits are defined in 20 ms intervals.	50(1000 ms)
Polling Mode	<p>If you want to:</p> <ul style="list-style-type: none"> Accept unsolicited messages from slave stations, choose MESSAGE BASED (Allow slave to initiate messages.) Slave station-initiated messages are acknowledged and processed after all master station-initiated (solicited) messages. Note: Slave stations can only send messages when they are polled. If the message-based master station never sends a slave station a message, the master station will never send the slave station a poll. Therefore, to regularly obtain a slave station-initiated message from a slave station, you should choose to use standard communications mode instead. Ignore unsolicited messages from slave stations, choose MESSAGE BASED (Do not allow the slave to initiate messages.) Slave station-initiated messages are acknowledged and discarded. The master station acknowledges the slave station-initiated message so that the slave station removes the message from its transmit queue, which allows the next packet that is slated for transmission into the transmit queue. 	Message-Based (slave can initiate messages)
Error Detection	<p>Select BCC or CRC error detection. Configure all stations to use the same type of error checking.</p> <p>BCC—The serial module sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement, but does not detect as many errors as CRC</p> <p>CRC—the serial module sends and accepts messages with a 2-byte CRC for error checking. CRC is a more effective error detection algorithm.</p>	BCC
Enable Duplicate Detection	Select whether the serial module should detect and ignore duplicate messages.	Enabled

DF1 Slave Protocol

The DF1 Slave protocol is used when the serial module is connecting to the DF1 Master device.

In DF1 Slave protocol, RS-485 and Serial Radio Modem can be used.

Figure 17 - DF1 Slave Protocol

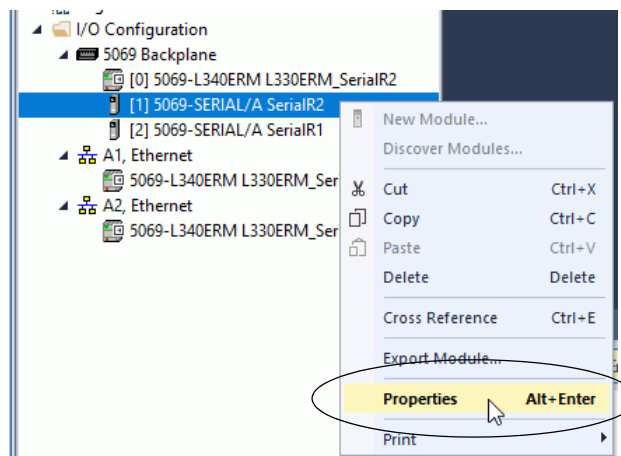


IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

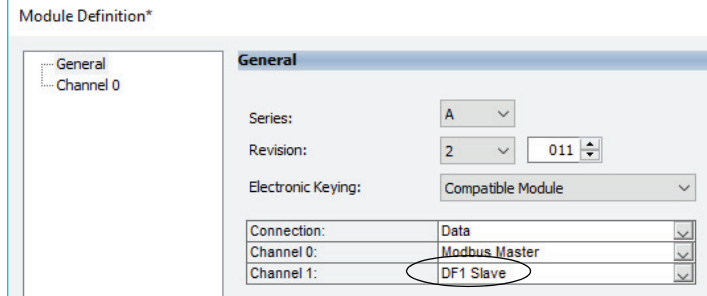
Configuring the Controller as a Slave Station

To configure the controller as a slave station, do the following using your programming software:

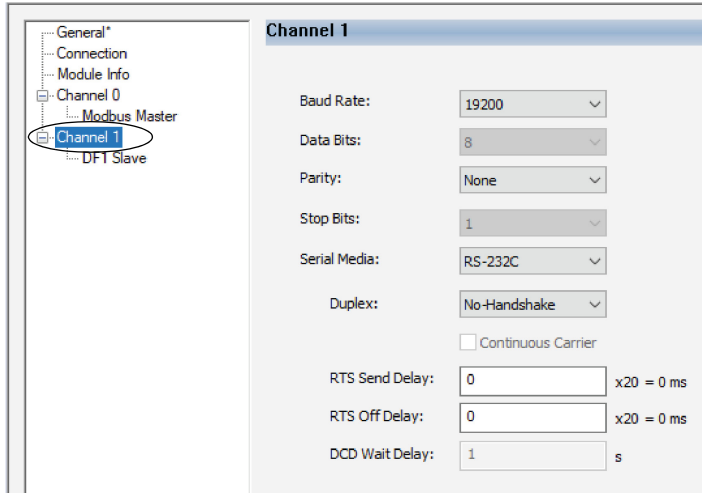
1. Open the Logix Designer application project that includes the serial module.
2. Right-click the module and choose Properties.



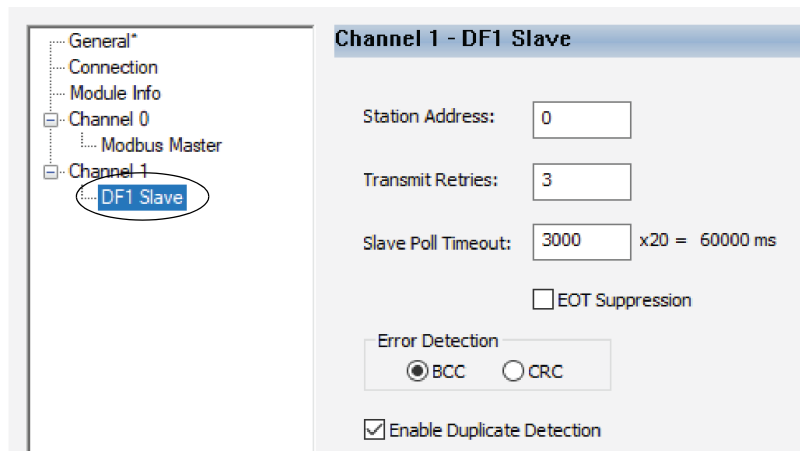
- On the Module Definition dialog box, verify that Channel 1 is configured for DF1 Slave. Configure Channel 1 to DF1 Slave if it is not configured to DF1 Slave.



- Click Channel 1 and configure the parameters as necessary.



- Click DF1 Slave under the Channel 1 category on the Module Properties dialog box to configure DF1 Slave port settings.



- To close the Module Definition dialog box, click OK.

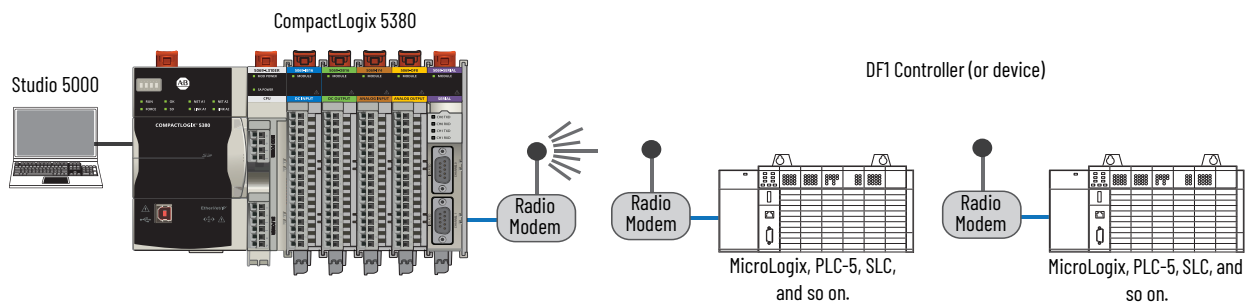
Serial Port Protocol parameters and defaults can be found in the following table. Use the following table to understand the communication parameters you need to specify on the Channel Configuration screen.

Field	Description	Default
Station Address	The station address for the serial port on the DF1 slave. Enter a valid DF1 address (0...254).	0
Transmit Retries	The number of times the slave station retries a message after the first attempt before the station declares the message undeliverable. Enter a value 0...127.	3
Slave Poll Timeout	Specifies the amount of time the slave station waits to be polled by a master before indicating a fault. Enter a value 0...32767. Limits are defined in 20 ms intervals. See 'Configuring Slave Poll Timeout' for further details.	3000 (60,000 ms)
EOT Suppression	Select whether to suppress sending EOT packets in response to a poll. The default is not to suppress sending EOT packets.	Do not suppress
Error Detection	Select BCC or CRC error detection. Configure all stations to use the same type of error checking. BCC —The serial module sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement, but does not detect as many errors as CRC CRC —the serial module sends and accepts messages with a 2-byte CRC for error checking. CRC is a more effective error detection algorithm.	BCC
Enable Duplicate Detection	Select whether the serial module should detect and ignore duplicate messages.	Enabled

DF1 Radio Modem Protocol

The DF1 Radio Modem protocol is used when connecting from the controller to multiple devices over Radio Modems. (This Radio Modem should not analyze/translate protocol by itself)

Figure 18 - DF1 Radio Modem Protocol



IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

Designing Communication for DF1 Radio Modem Protocol

When designing communication using DF1 Radio Modem protocol, you must consider the capabilities of both the controllers and radio modems. The DF1 Radio Modem protocol can only be used with controllers that support and are configured for this protocol.

Determining When to Use DF1 Radio Modem Protocol

If your radio modem can handle full-duplex data port buffering and radio transmission collision avoidance, you can use peer-to-peer message initiation capability in every node (that is, the ladder logic in any node can trigger a MSG instruction to any other node at any time). For messaging between nodes that are outside of radio transmission/reception range of each other, you may use either the Store and Forward capability of the protocol or the repeater capability of the radios.

If your radio modem cannot handle full-duplex data port buffering and radio transmission collision avoidance, you can still use DF1 Radio Modem protocol

in a Master/Slave configuration, with message initiation limited to one master node. If you still require slave node message initiation, then you must use the DF1 Half-Duplex protocol.

The primary advantage of using DF1 Radio Modem protocol for radio modem networks is in the transmission efficiency. Each read/write transaction (command and reply) requires only one transmission by the initiator (to send the command) and one transmission by the responder (to return the reply). The number of transmissions is minimized, radio power is minimized, and throughput is maximized. In contrast, DF1 Half-Duplex protocol requires five transmissions for the DF1 Master to complete a read/write transaction with a DF1 Slave. See [Figure 20 on page 56](#) the DF1 Radio Modem protocol.

An efficiency trade-off exists in that the DF1 Radio Modem protocol does not provide immediate feedback (ACK) to the initiator to indicate that the responder successfully received the communications packet without error.

The Store and Forward capability of the DF1 Radio Modem protocol allows messages between nodes that are outside of radio transmission/reception range of each other to be routed through intermediary nodes that are within range. Each of the intermediary nodes needs a Store and Forward table. The configuration must indicate, based on the source and destination addresses in the message packet, which packets to receive (store) and then rebroadcast (forward). The following section explains how the Store and Forward function is configured.

Figure 19 - Radio Communication

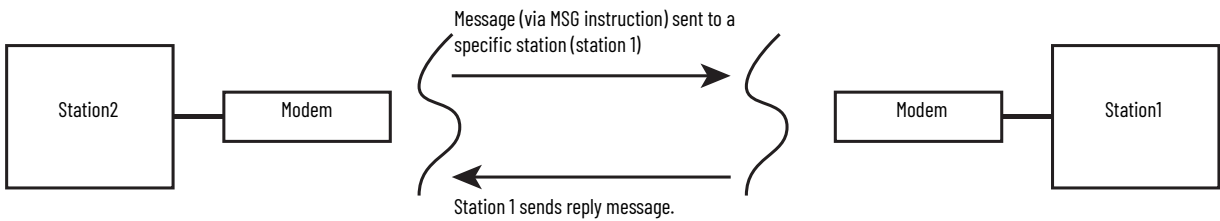
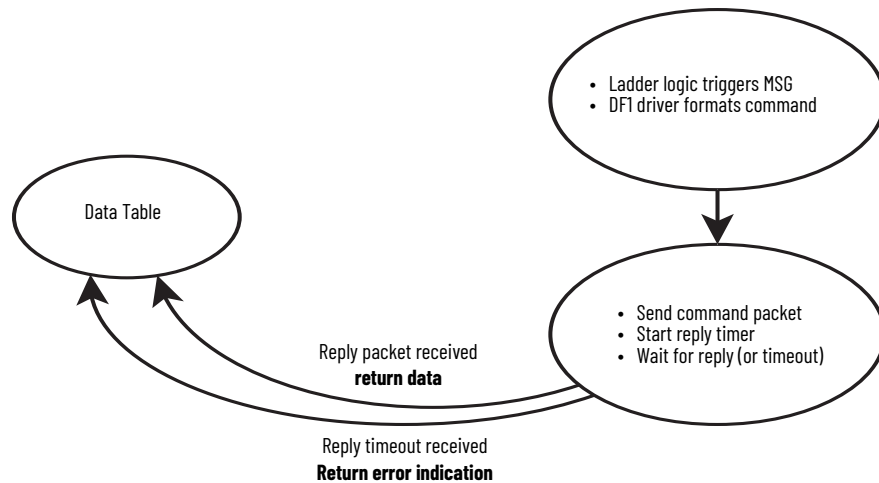


Figure 20 - Read or Write Requests via DF1 Radio Modem

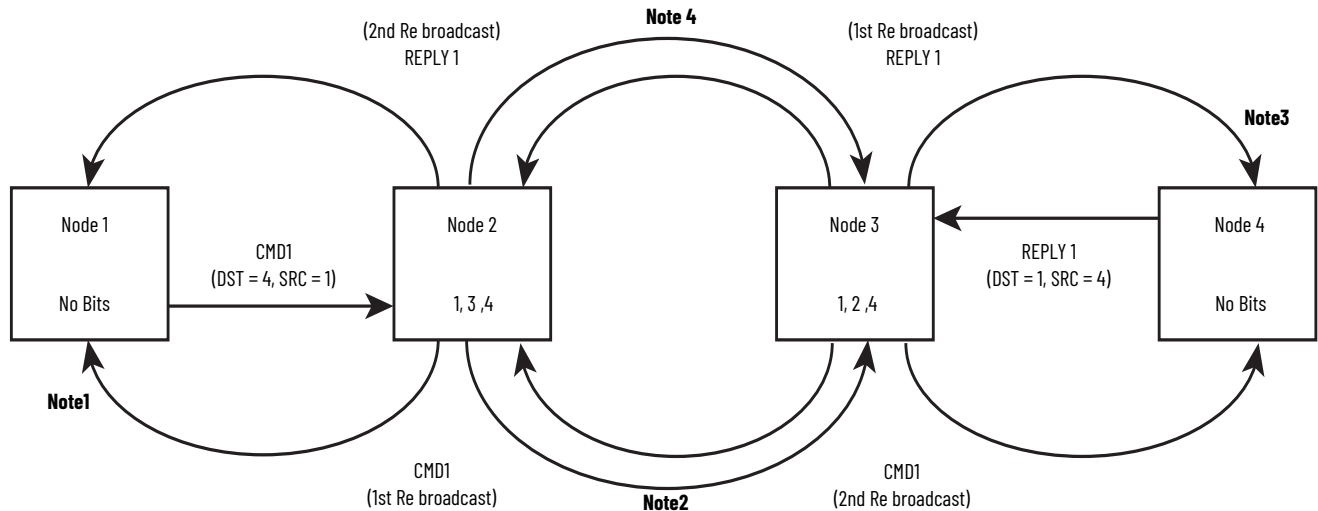


Store and Forward

The Store and Forward Table occupies a 16-word binary data table tag Local:x:O1.DF1RadioModem.StoreAndForward. Each bit in this file corresponds to a DF1 Radio Modem node address. To configure a 5069-SERIAL

to Store and Forward message packets between two other nodes, the bits corresponding to the addresses of those other two nodes must be set. For instance, if node 2 is used to Store and Forward message packets between nodes 1 and 3, then both bits Bx/1 and Bx/3 would have to be set in the Store and Forward Table (Figure 21). You can set bit 255 to enable Store and Forward of broadcast packets, as well.

Figure 21 - Store and Forward in DF1 Radio Modem Protocol



Note 1—The link layer of Node 1 blocks the retransmission of a packet that is received with the SRC byte equal to the receiving node's station add retransmitted.

Note 2—To help prevent Node 2 from retransmitting a duplicate packet, the link layer of Node 2 updates the duplicate packet table with the last 20 packets received.

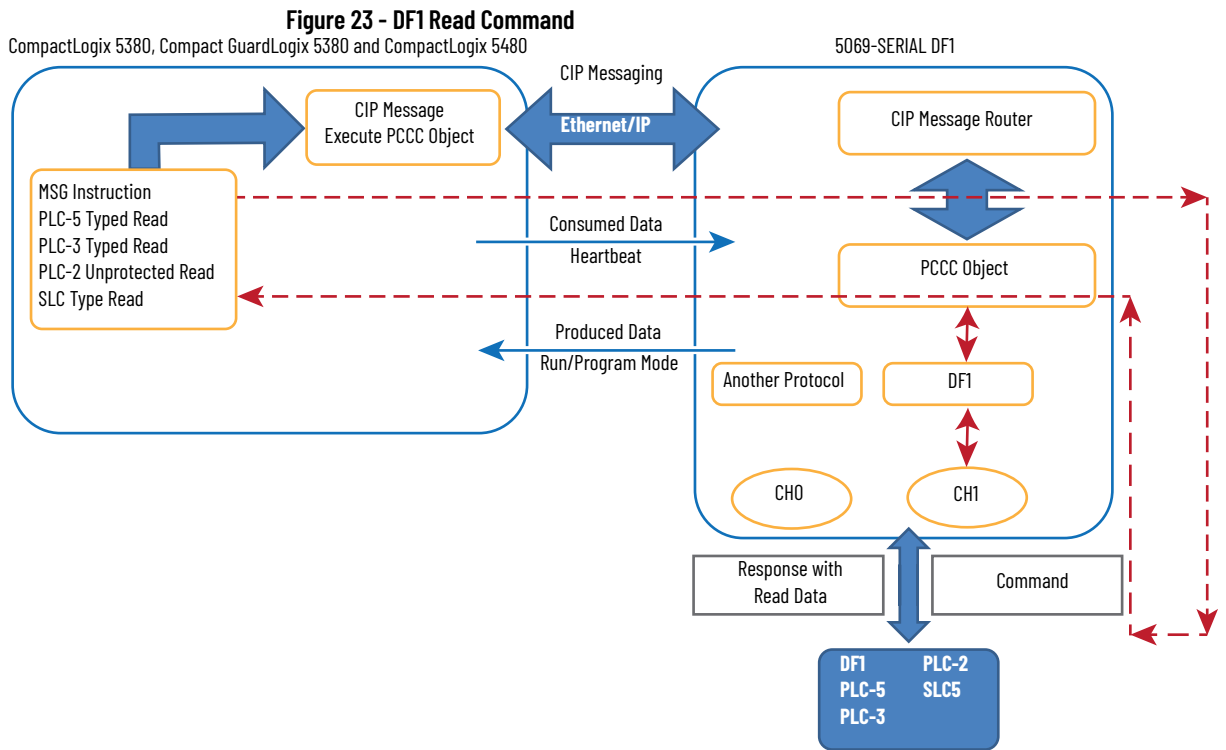
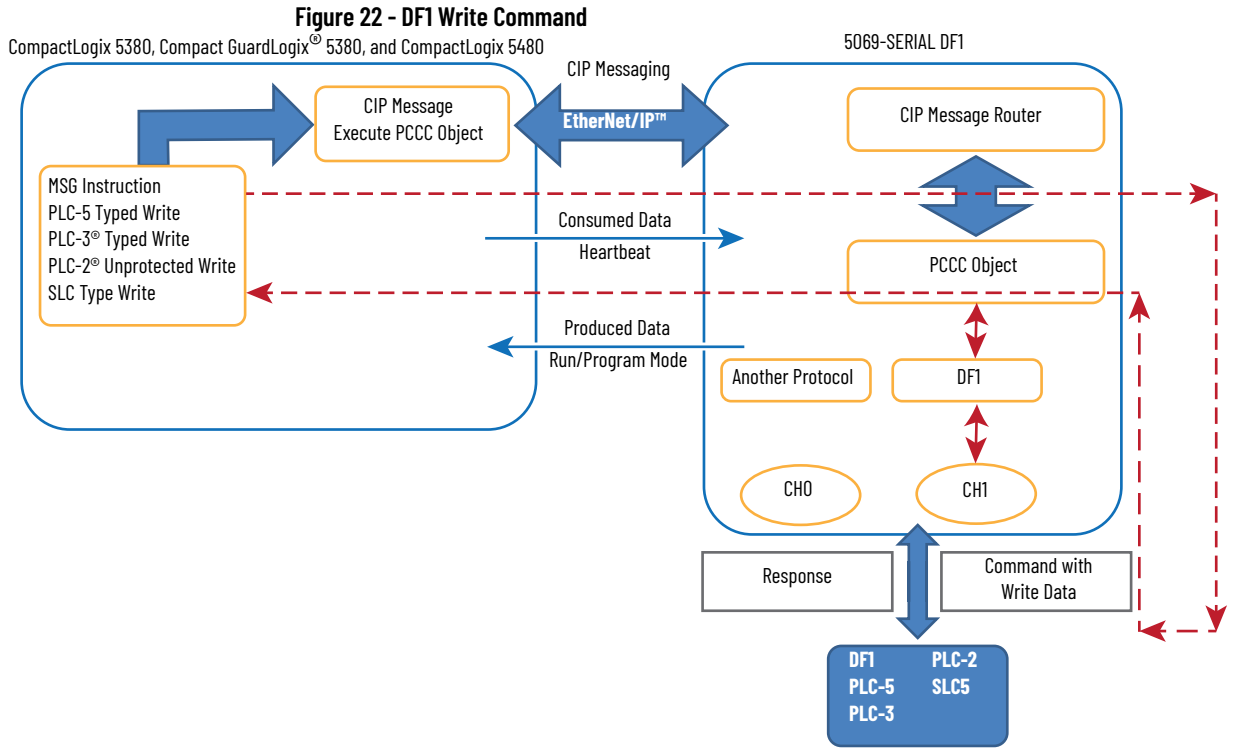
Note 3—The link layer of Node 4 blocks the retransmission of a packet that is received with the SRC byte equal to the receiving node's station add retransmitted.

Note 4—To help prevent Node 3 from retransmitting a duplicate packet, the link layer of Node 3 updates the duplicate packet table with the last 20 packets received.

DF1 Data Exchange

For the DF1 data exchange, the following definitions apply:

- Write—the Controller can write data to DF1.
- Read—the Controller reads data from the DF1.



DF1 Point to Point Parameters

Table 16 - DF1 Point to Point Configuration Parameters

Parameter	Definition	Available Options
Station Address	Station address of the channel.	• 0...254
NAK Receive Limit	Maximum number the module can receive NAK for each message.	• 0...255 (default = 3)
ENQ Transmit Limit	Maximum number the module can send ENQ for each message.	• 0...255 (default = 3)
ACK Timeout	Maximum time the module can wait for ACK for each message.	• 1...32,767 (x20 ms) (default = 50) • 0 = Disabled (Wait Forever)
Embedded Responses	Autodetect Embedded Responses cause the module to use embedded responses only if an embedded response from the serial module is received by the module. Enabled Embedded Responses let the module send embedded responses within packets.	• 0 = Autodetect • 1 = Enabled
Receive Termination Delimiter 1	Configure the channel's receive termination delimiter.	• 8 bit \$00...\$FF (default = \$0D)
Receive Termination Delimiter 2	Configure the channel's receive termination delimiter.	• 8 bit \$00...\$FF (\$FF = disabled)
Error Detection	Detects error protocols.	• 0 = BCC • 1 = CRC (default)
Enable Duplicate Detection	When enabled, duplicate messages are acknowledged and discarded.	• 0 = Disabled • 1 = Enabled (default)

DF1 Master Parameters

Table 17 - DF1 Master Configuration Parameters

Parameter	Definition	Available Options
Station Address	Station address of the channel.	• 0...254
Transmit Retries	Maximum number of times Master can retry sending message to Slave.	• 0...127 (default = 3)
ACK Timeout	Maximum time the module can wait for ACK for each message.	• 1...32,767 (x20 ms) (default = 50) • 0 = Disabled (Wait Forever)
Reply Message Wait	Minimum time between receiving ACK and sending Poll to Slave. Adjust this value to meet Slave response preparation time.	• 1...65,535 (x20 ms) (default = 5) • 0 = Disabled (Wait Forever)
Error Detection	Detects error protocols.	• 0 = BCC • 1 = CRC (default)
Enable Duplicate Detection	When enabled, duplicate messages are acknowledged and discarded.	• 0 = Disabled • 1 = Enabled (default)

DF1 Slave Parameters

Table 18 - DF1 Slave Configuration Parameters

Parameter	Definition	Available Options
Station Address	Station address of the channel.	• 0...254
Transmit Retries	Maximum number of times Slave can retry sending message to Master.	• 0...127 (default = 3)
Slave Poll Timeout	Maximum time Slave can wait for polling from Master when message is sent from Slave.	• 0...32,767 (x20 ms) (default = 3000)
EOT Suppression	Select whether to suppress sending EOT packets in response to poll.	• 0 = Do Not Suppress (default) • 1 = Suppress
Error Detection	Detects error protocols.	• 0 = BCC • 1 = CRC (default)
Enable Duplicate Detection	When enabled, duplicate messages are acknowledged and discarded.	• 0 = Disabled • 1 = Enabled (default)

DF1 Radio Modem Parameter

Table 19 - DF1 Radio Modem Configuration Parameters

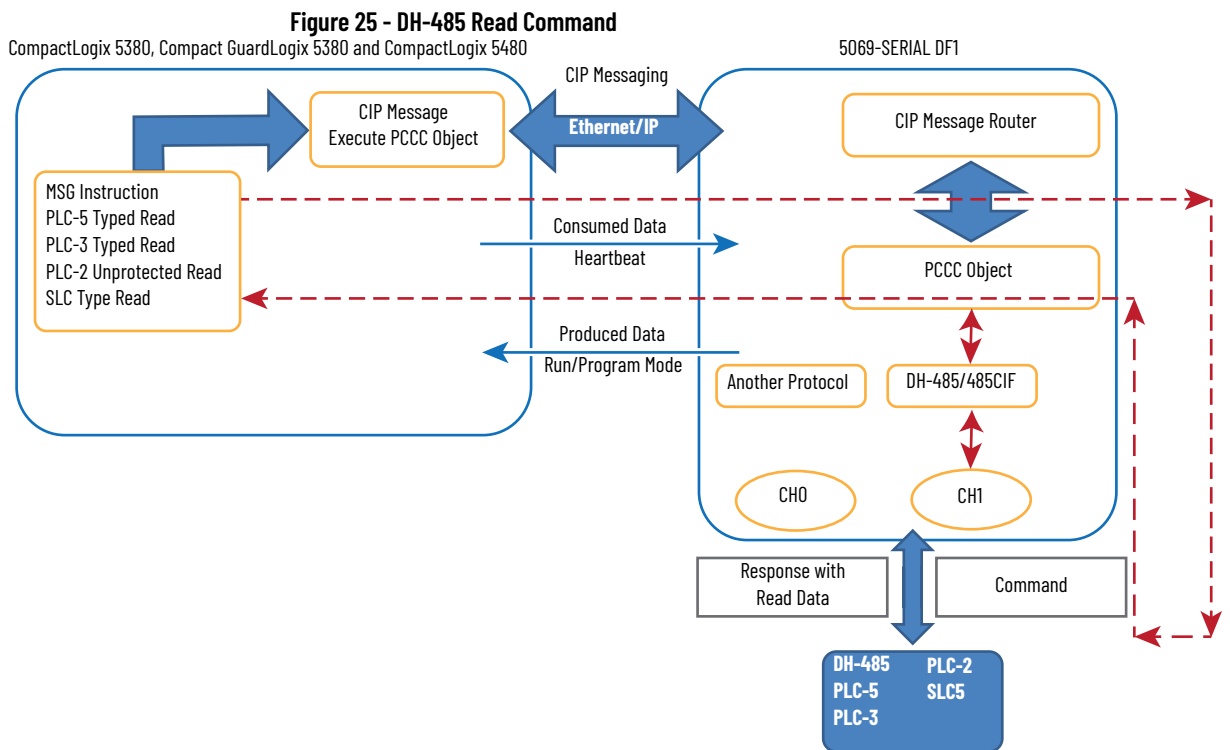
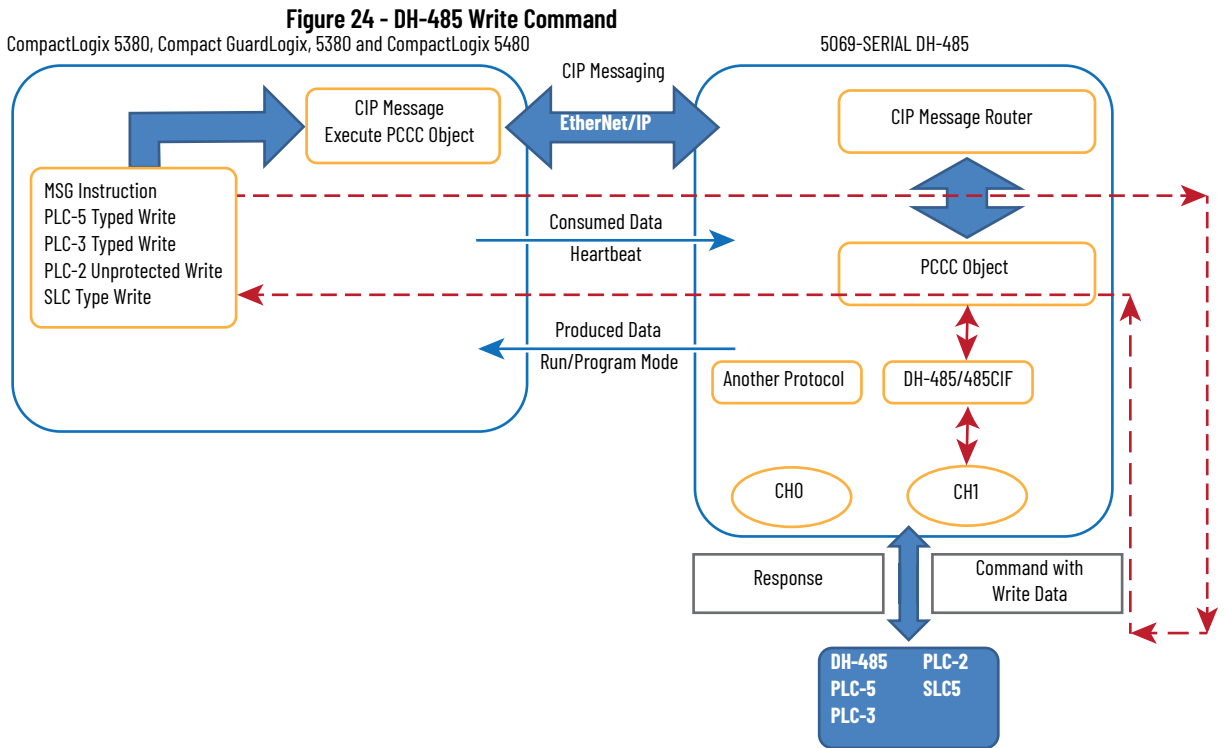
Parameter	Definition	Available Options
Station Address	Station address of the channel.	<ul style="list-style-type: none">• 0...254
Enable Store and Forward	Determine whether Store and Forward (bit) is enabled.	<ul style="list-style-type: none">• 0 = Disabled (default)• 1 = Enabled
Error Detection	Detects error protocols.	<ul style="list-style-type: none">• 0 = BCC• 1 = CRC (default)
Enable Duplicate Detection	When enabled, duplicate messages are acknowledged and discarded.	<ul style="list-style-type: none">• 0 = Disabled (default)• 1 = Enabled

For Example MSG Control Blocks, see [page 127](#).

DH-485 Data Exchange

For the DF1 data exchange, the following definitions apply:

- Write—the Controller can write data to DH-485.
- Read—the Controller reads data from the DH-485.



DH-485 Functions

Table 20 - DH-485 Configuration Parameters

Parameter	Definition	Available Options
Station Address	Station address of the channel.	• 0...31
Max Station Address	Specifies the maximum station address of all devices on the DH-485 network.	• 0...31 (default = 31)
Token Hold Factor	Number of transmissions plus retries that a node holding a token can send onto the datalink each time that it receives the token.	• 1...4 (default = 1)
Error Detection	Detects error protocols.	• 0 = BCC • 1 = CRC (default)
Enable Duplicate Detection	When enabled, duplicate messages are acknowledged and discarded.	• 0 = Disabled (default) • 1 = Enabled

For Example MSG Control Blocks, see [page 127](#).

IMPORTANT DF1 and DH-485 protocols use channel 1 only and must be local.

Compact 5000 I/O Serial Module Configuration

This chapter describes how to configure your Compact 5000™ I/O serial module in a Logix Designer application project.



When a controller establishes a connection to a remote 5069-SERIAL module, it uses a class 3 connection. We recommend that you reserve one class 3 connection on the Compact 5000 EtherNet/IP™ adapter to establish a connection to the module. Otherwise, you can encounter a “Connection Request Error: Module connection limit exceeded” error.

Topic	Page
Add a New Module	63
Edit the Module Configuration	71
View the Module Tags	89

Before You Begin

You must complete the following tasks before you can configure the module:

1. Create a Logix Designer application project.
2. If you use the Compact 5000 I/O serial module as remote I/O module, add a Compact 5000 EtherNet/IP adapter to the project.

For more information on how to add a Compact 5000 EtherNet/IP adapter to a Logix Designer application project, see the EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication [ENET-UM004](#).

IMPORTANT Use Studio 5000 Logix Designer® Version 31 or greater. You must install an Add-On Profile to use the serial module. To find the Add-On Profile, go to the [Product Compatibility and Download Center](#) (PCDC).
Serial firmware V1.011 supports Generic ASCII, Modbus Master and Modbus Slave. Serial firmware revision 2.011 and later with Add-On Profile 2.01 supports Generic ASCII, Modbus Master, Modbus Slave, DF1 and DH-485 protocols.
DF1 and DH-485 are only supported locally on Channel 1.

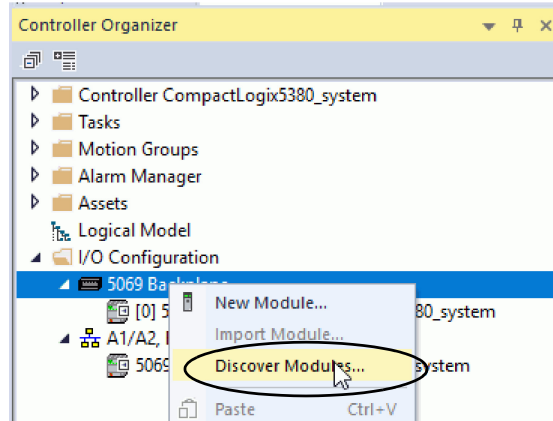
Add a New Module

There are two methods to add modules to your Logix Designer Application project.

- Discover Local and Remote I/O Modules
- Create New Local and Remote I/O Modules

Discover Local I/O Modules

1. Go online with your Logix Designer application.
2. Right-click the 5069 Backplane and choose Discover Modules.

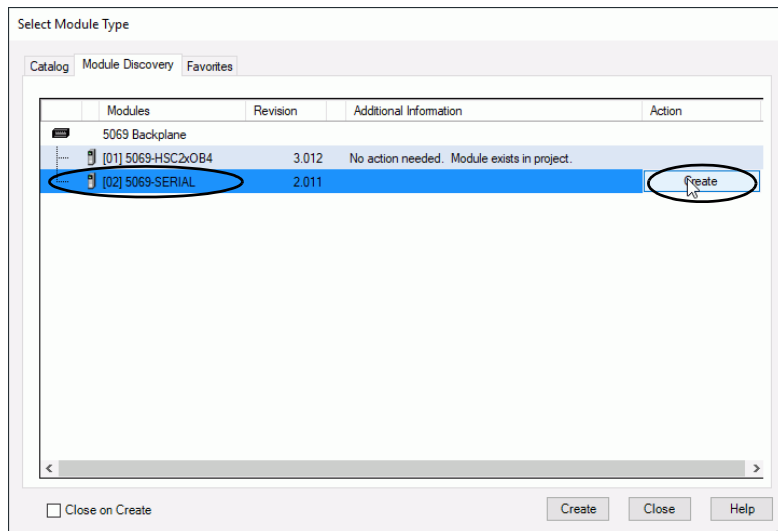


The Logix Designer application automatically detects available modules that are connected to the backplane.

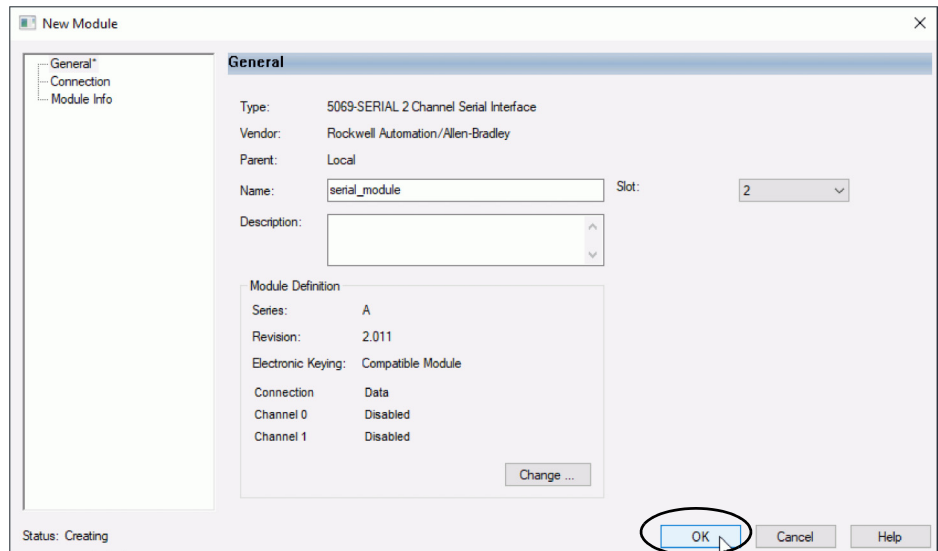
3. Select your module and click Create to add the discovered module to the project.



Selecting close on create will close windows when module creation is complete.



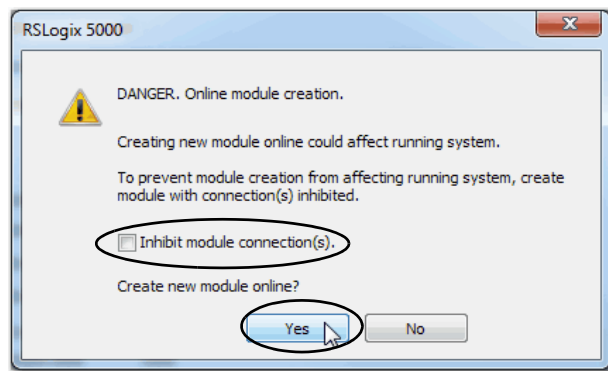
4. Enter the module properties and click OK.



5. At the warning dialog box, click Yes.



If you inhibit the module connection, you must remember to uninhibit the connection later. See [Connection Category on page 77](#)

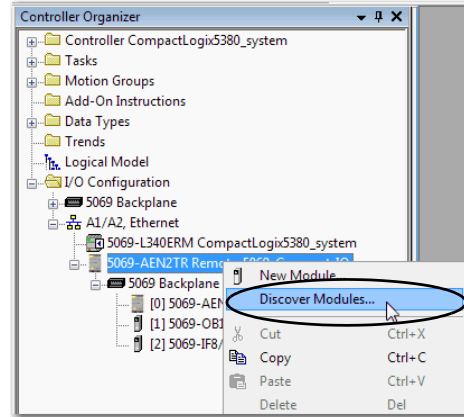


6. Close the Select Module Type dialog box.

Discover Remote I/O Modules

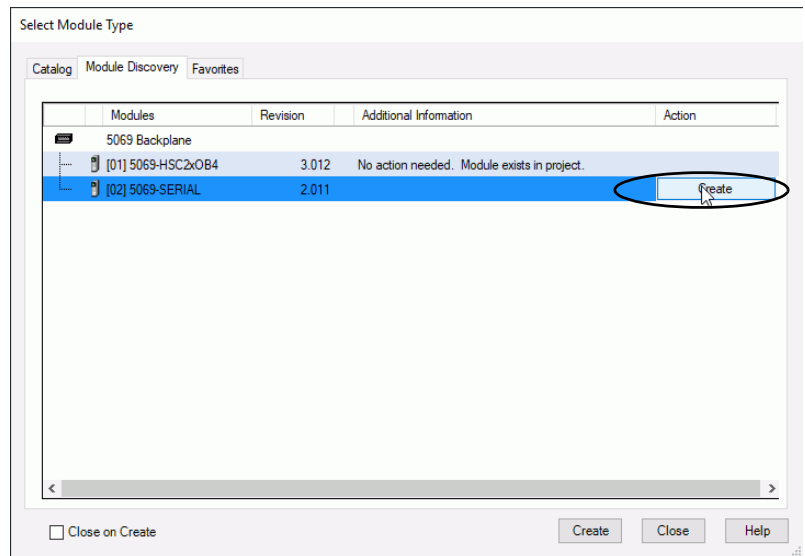
The project must include at Compact 5000 EtherNet/IP adapter.

1. Go online with your Logix Designer application.
2. Right-click the Compact 5000 EtherNet/IP adapter and choose Discover Modules.

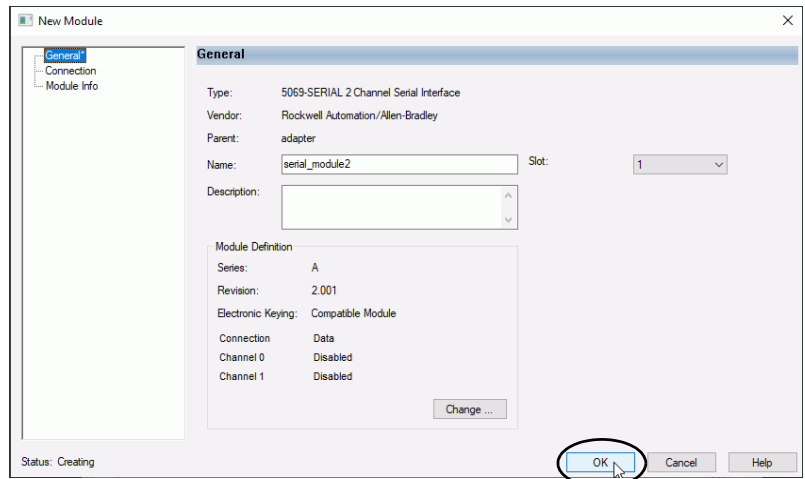


The Logix Designer application automatically detects available modules that are connected to the backplane.

3. Click Create to add the discovered module to your project.



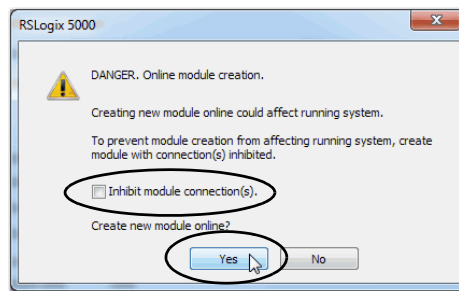
- Enter the module properties and click OK.



- At the warning dialog box, make sure that Inhibit module connection is selected and click Yes.



If you inhibit the module connection, you must remember to uninhibit the connection later. See [Connection Category on page 77](#)



- Close the Select Module Type dialog box.

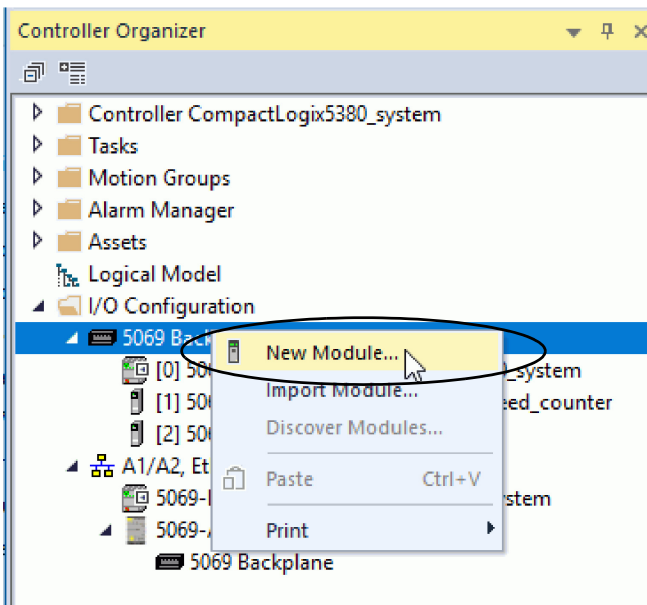
New Local I/O Modules



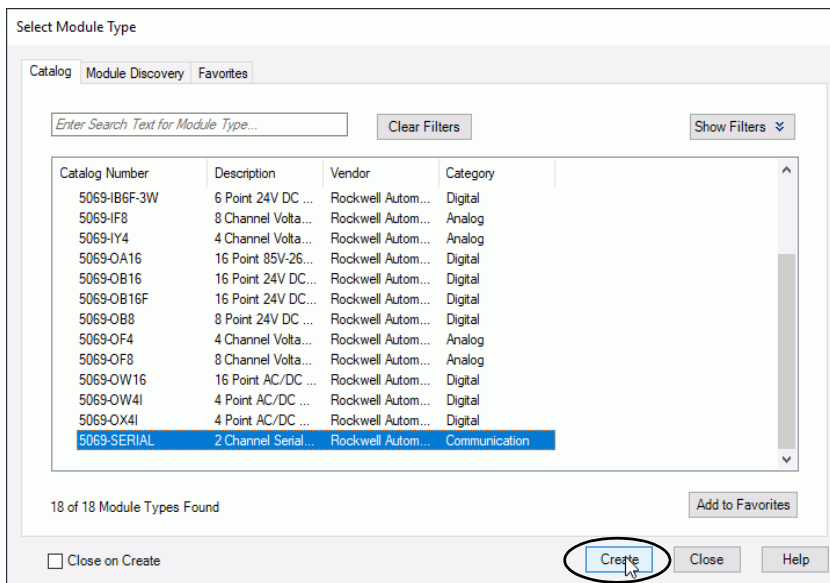
This example shows how to add a local I/O module when the Logix Designer application project is offline.

You can add new modules when the project is online, if desired. In this case, the steps are similar to the steps described in [Discover Local I/O Modules on page 64](#). One exception is that, in step 1, you choose New Module instead of Discover Modules.

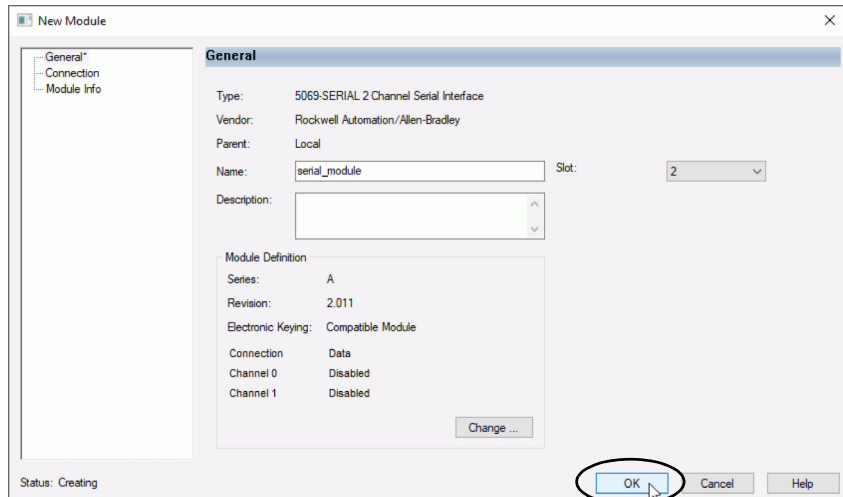
1. Right-click the 5069 Backplane and choose New Module.



2. Select the module and click create.



3. Enter the module properties and click OK.



New Remote I/O Module

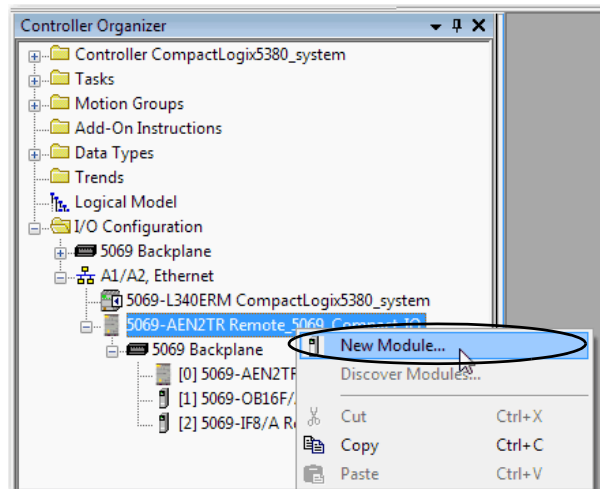
The project must include at Compact 5000 EtherNet/IP adapter.



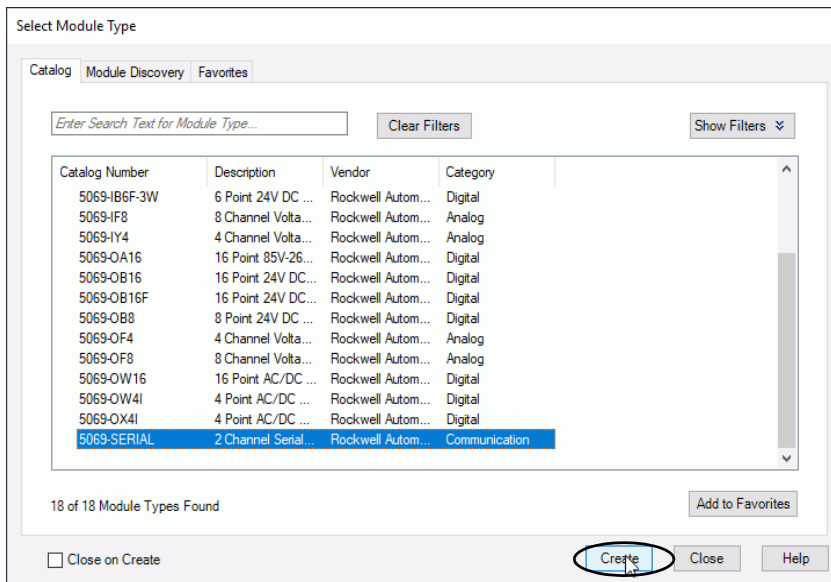
This example shows how to add a local I/O module when the Logix Designer application project is offline.

You can add new modules when the project is online, if desired. In this case, the steps are similar to the steps described in [Discover Local I/O Modules on page 64](#). One exception is that, in step 1, you choose New Module instead of Discover Modules.

1. Right-click the 5069 Compact EtherNet/IP adapter and choose New Module.

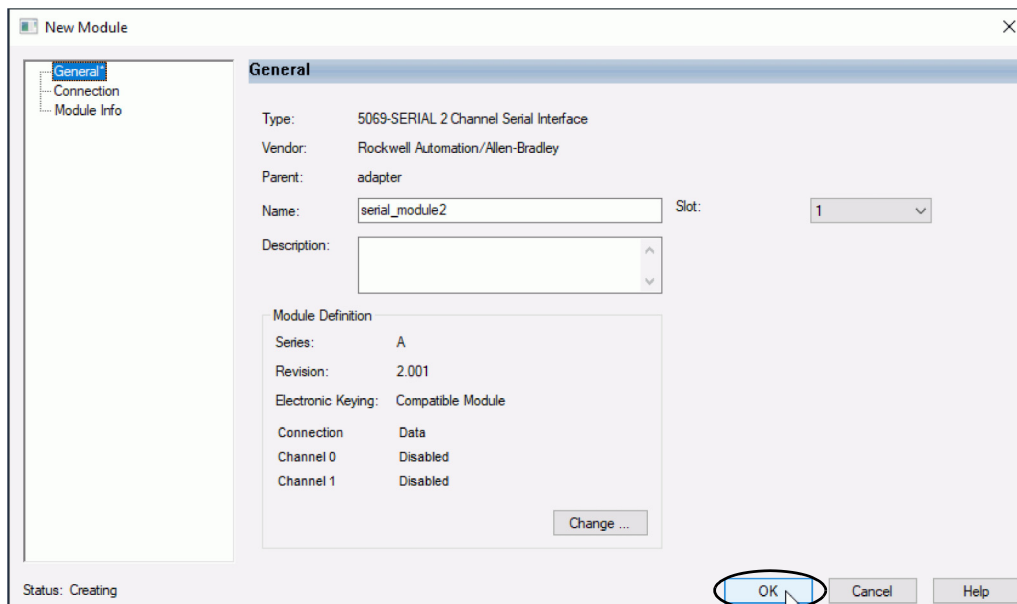


2. Select the module and click Create.



The New Module dialog box appears with a list of categories on the left side. The number and type of categories varies by module type.

3. You can click OK to use the default configuration as shown or edit the module configuration. The rest of this chapter describes how to edit module configuration categories.



Edit the Module Configuration

Click the category names in the New Module dialog box to view and change the configuration parameters.

IMPORTANT This chapter shows how to edit configuration when you add the module to the Logix Designer application project. If you access the module configuration after it is added to the project, the dialog box is named Module Properties. The same categories are displayed as the categories displayed on the New Module dialog box.

Some Compact 5000 I/O modules categories are specific to the module type. The following categories apply to all Compact 5000 I/O modules and are described in this section.

- [General Category](#)
- [Connection Category](#)
- [Module Info Category](#)
- [View the Module Tags](#)

General Category

The General category appears first when you create a module. The parameters in this category are the same for all Compact 5000 I/O modules.

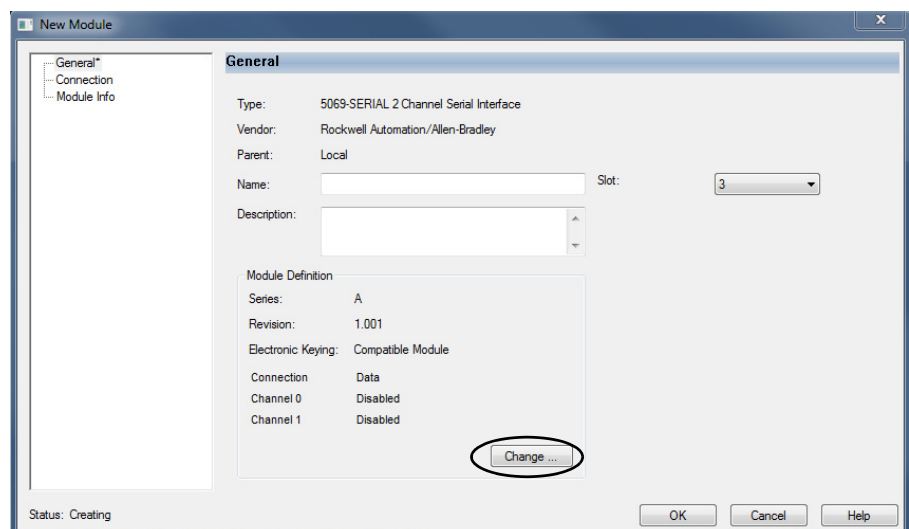
You use this category to complete the following optional tasks:

- Name the module.
- Assign a slot number. (required)
- Describe the module.
- Access the Module Definition.

Module Definition Parameters

Module Definition parameters are available on the General tab of the Module Properties dialog box in the Logix Designer application project. The module definition can only be edited during offline mode.

To access the module definition parameters, click change on the general screen.



The Module Definition dialog box opens.

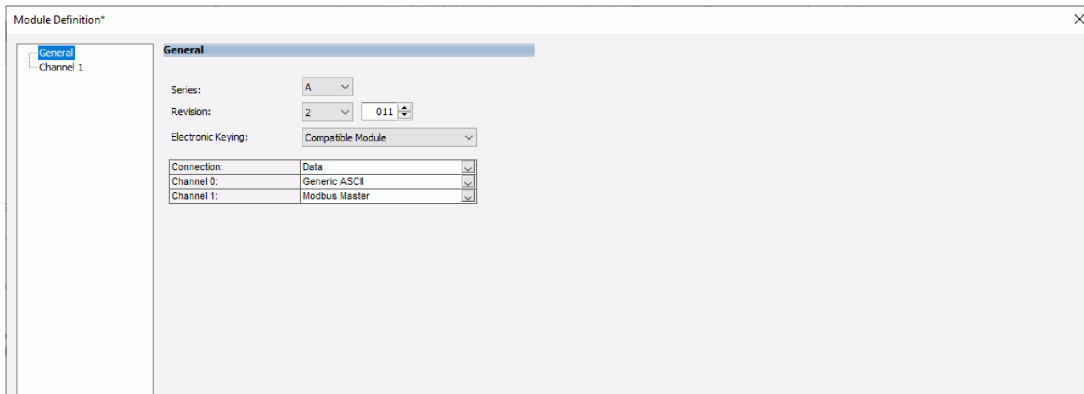


Table 21 describes the parameters in the Module Definition dialog box.

Table 21 - Module Definition Parameters

Parameter	Definition	Available Choices ⁽¹⁾
Series	Module hardware series	Module-specific
Revision	Module firmware revision, including major and minor revision levels	Module-specific
Electronic Keying	Software method by which you reduce the possibility of using the wrong device in a control system. For more information, see the following: <ul style="list-style-type: none"> • View the Module Tags on page 89 • Electronic Keying in Logix 5000™ Control Systems Application Technique, publication LOGIX-AT001 	Exact Match Compatible Module Disable Keying
Connection	<ul style="list-style-type: none"> • Definition establishes a connection between the controller and the module. 	Data
Channel 0/1	Determines the following for the module type you configure: <ul style="list-style-type: none"> • Determines the protocol that is used on Channel 0 or 1 	Disabled Generic ASCII Modbus Master Modbus Slave For more information, see Table 22 .
Channel 1	Determines the following for the module type you configure: <ul style="list-style-type: none"> • Determines the protocol that is used on Channel 1 • Used locally only • Supported in firmware revision 2.011 and later with Add-On Profile 2.01 	DF1 Point to Point DF1 Master DF1 Slave DF1 Radio Modem DH-485 For more information, see Table 22 .

(1) The choices that are available vary by module type and catalog number.

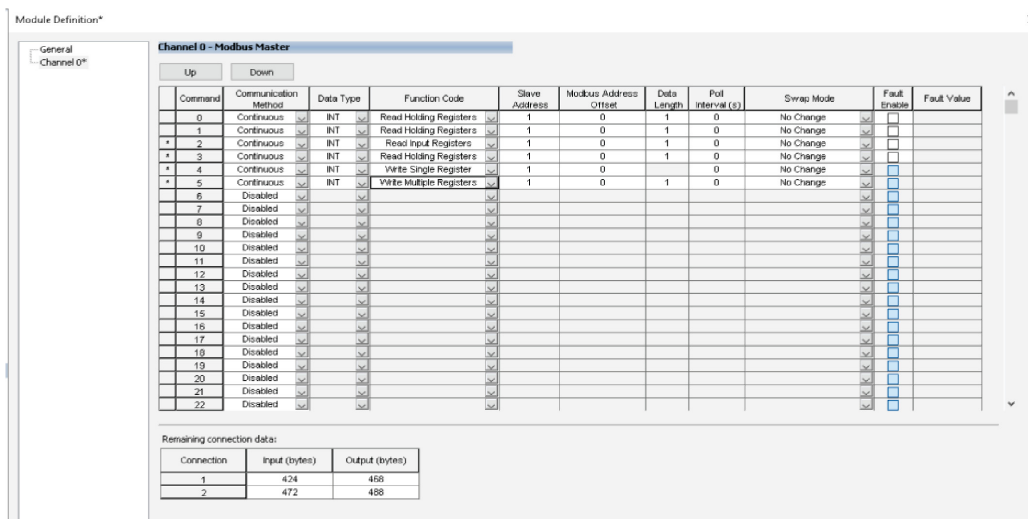
Table 22 - Communication Mode Definitions

Communication Mode	Definition
Disabled	The channel is unused and no physical connection is enabled between the controller and the serial module.
Generic ASCII	A general mode of serial communication where you can define any user data to be transmitted or received in the communication.
Modbus Master	The device sends Modbus queries or write commands to the slaves devices connected to it.
Modbus Slave	The device operates as a slave to an external master and waits for commands from the master.
DF1 Point to Point	The device sends DF1 queries or write commands to the devices connected to it.
DF1 Master	The device sends DF1 master queries or write commands to the slaves devices connected to it.
DF1 Slave	The device operates as a slave to an external master and waits for commands from the master.
DF1 Radio Modem	The device sends DF1 queries or write commands to the devices connected to it.
DH-485	The device sends DH-485 queries or write commands to the devices connected to it.

You can have additional configurable parameters depending on your parameter choice for a channel. What you choose can have additional parameters.

See [Figure 26](#) for an example of Modbus Master options and [Figure 27](#) for an example of Modbus Slave options

Figure 26 - Modbus Master Module Definition Parameters



IMPORTANT On the module definition screen, you will see two options at the top for moving the commands up and down. If either of these options are used, make sure that the user program is adjusted to reflect the new location of the command or the program will show an error.

Table 23 - Master Command List Modbus Master Parameters

Parameter	Definition
Communication Method	Communication Method: <ul style="list-style-type: none"> Disabled Continuous - sending the command based on the Poll interval value. Conditional - only for Write command and triggered when the write value has changed. <ul style="list-style-type: none"> After initialization, the base value for conditional is 0. So if the first consumed data is not 0, the conditional command is transmitted. If you want to use value 0, you must set another value and reset the value to 0 again.
Data Type	<ul style="list-style-type: none"> BOOL INT REAL
Function Code	<ul style="list-style-type: none"> Read Coil Status- This code reads Modbus addresses 000000...065535. These bit values indicate coil status. Read Input Status- This code reads Modbus addresses 100000...165535. These read-only bit values indicate discrete input status. Read Holding Registers - This code reads Modbus addresses 400000...465535. This is a 16-bit word value. Read Input Registers - This code reads Modbus addresses 300000...365535. They are also 16-bit word values, but are read-only data. The Modbus Master cannot write to these registers. Force Single Coil - This code writes to Modbus addresses 000000...065535. This command writes to only one coil. Preset Single Register - This code writes to Modbus addresses 400000...465535. This command writes to only one coil. Force Multiple Coil - This code writes to multiple coil values to the slave addresses 000000...065535. Preset Multiple Registers - This code writes to multiple register values to the slave device at addresses 400000...465535 For more information about Master Command List Function Codes, see Appendix B .
Slave Address	Node Address of the Modbus Slave device (1...247 and 0 for broadcast).
Modbus Address Offset (0-based)	Offset to the actual Modbus Address. Holding Register address of 400003 is equal to 00003 in the Modbus Address offset. (0...65535).
Data Length	Number of data points being read (1...125 for registers, 1...2000 for Coil and discrete inputs) or written (1...123 for registers, 1...1968 for Coil).
Poll Interval	<ul style="list-style-type: none"> 0 - keeps repeating as fast as possible. 1...32,767- the time, in seconds, for the command to be sent periodically in Continuous mode.
Swap Mode	<ul style="list-style-type: none"> No Change - no swapping of data Word Swap - Words are swapped before sending out, for example, 11112222 = 22221111 Byte Swap - each byte is swapped. For example, 11223344 = 22114433 Word and Byte swap - both word and byte is swapped.
Fault Enable	<ul style="list-style-type: none"> Check to write the Fault Value into the received data if this Read Command fails. This does not apply to write commands.
Fault Value	<ul style="list-style-type: none"> User-defined value to replace received data.

Master Command List Limitations

- A maximum of 50 commands can be created. The commands are subject to available connection memory.
- Each Modbus Master supports up to two data connections.
- The First Data Connection supports a max of 464 bytes of read data and 476 bytes of write data. The Second Data Connection supports a max of 472 bytes of read data and 488 bytes of write data.
- Each command can have a max of either 125 words read or 123 words write or 2000 coil/discrete input read or 1968 coils written.
- Each Read command uses:
 - Four bytes of input data for command status
 - Two bytes of input data per holding register or input register read.
 - One byte of input data per every 1...8 coils or discrete inputs read.
 - Padding to align command to 4 bytes.
- Each Write command uses:
 - Four bytes of input data for command status
 - Two bytes of output data per holding register written.
 - One byte of output data per every 1...8 coils written.
 - Padding to align command to 4 bytes.
- Remaining available data is displayed.
- An error message appears when connection memory is exceeded.

Master Command Memory Usage

Input and output data size is available in the Add-On Profile.

- View data size to verify data has not exceeded data available per channel:

Remaining connection data:		
Connection	Input (bytes)	Output (bytes)
1	424	468
2	472	488

Current Manual Calculation Check

The following is an example of master command memory usage.

- Maximum single connection configuration for registers:

Command	Communication Method	Data Type	Function Code	Slave Address	Modbus Address Offset	Data Length	Poll Interval (s)	Swap Mode	Fault Enable	Fault Value
0	Continuous	INT	Read Holding Registers	1	0	125	0	No Change	<input type="checkbox"/>	
1	Continuous	INT	Read Holding Registers	1	125	98	0	No Change	<input type="checkbox"/>	
2	Conditional	INT	Write Multiple Registers	1	0	123		No Change	<input type="checkbox"/>	
3	Conditional	INT	Write Multiple Registers	1	123	118		No Change	<input type="checkbox"/>	

- 223 words * 2 bytes/word read = 446 input bytes
- 241 words * 2 bytes/word written = 482 output bytes

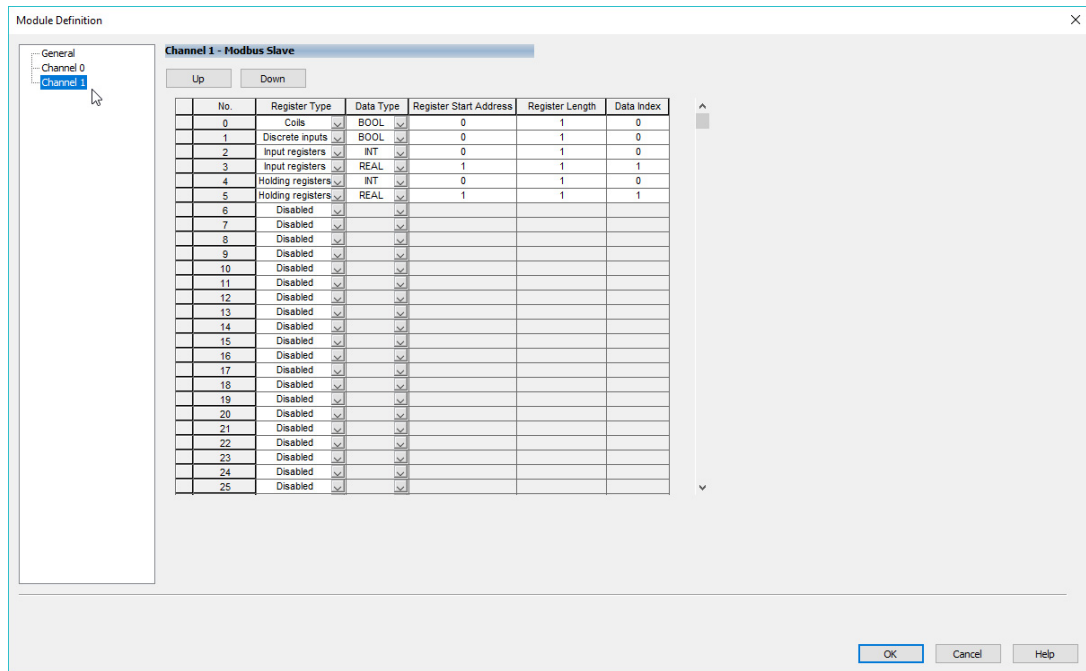
- Maximum single connection configuration for coils:

Command	Communication Method	Data Type	Function Code	Slave Address	Modbus Address Offset	Data Length	Poll Interval (s)	Swap Mode	Fault Enable	Fault Value
0	Continuous	BOOL	Read Coils	1	0	2000	0	No Change	<input type="checkbox"/>	
1	Continuous	BOOL	Read Coils	1	2000	1576	0	No Change	<input type="checkbox"/>	
2	Conditional	BOOL	Write Multiple Coils	1	0	1968		No Change	<input type="checkbox"/>	
3	Conditional	BOOL	Write Multiple Coils	1	1968	1888		No Change	<input type="checkbox"/>	

- 3576 bits / 8 bits/byte read = 447 input bytes
- 3856 bits / 8 bits/byte = 482 output bytes

Modbus Slave Example

Figure 27 - Modbus Slave Module Definition Parameters



IMPORTANT On the module definition screen, you will see two options at the top for moving the commands up and down. If either of these options are used, make sure that the user program is adjusted to reflect the new location of the command or the program will show an error.

Table 24 - Modbus Slave Module Definition Parameters

Parameter	Definition
Register Type	<ul style="list-style-type: none"> Disabled (default) Coils Discrete Inputs Input Registers Holding Registers
Data Type	<ul style="list-style-type: none"> BOOL INT REAL
Register Start Address	<ul style="list-style-type: none"> 0...65535 (default = 0)
Register Length	Dependent on Data Type <ul style="list-style-type: none"> BOOL: 1...128 (default = 1) INT: 1...100 (default = 1) REAL: 1...50 (default = 1)
Data Index	Location of data in the output or input tags depending on the register type defined. For more information, see the following section on Data Index .

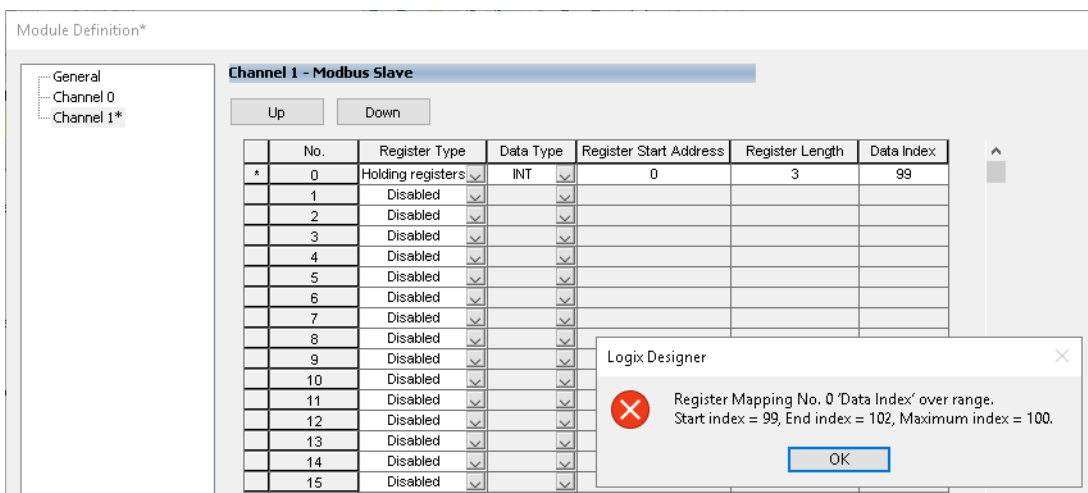
Data Index

This value indicates the offset of the register type address in controller tags that allow the serial module to read or write required data to or from the controller.

Each register type has a predefined array size that can be used for the Modbus Slave as indicated in the [Modbus Slave Address Table Limits](#). For example, a Holding register has the limit of 100 INT. The array size is 0...99 and the Data index is 0...99 when the Holding register is used depending on the size and the location where you want to store the information.

If you exceed the data limit, an error can occur. This error would appear as in [Figure 28](#).

Figure 28 - Data Index Error



Modbus Slave Address Table Limits

- Up to 30 data point ranges can be created, subject to available memory:
 - 200 byte maximum of Holding registers (up to 100 INTs or 50 REALs)
 - 200 byte maximum of Input registers (up to 100 INTs or 50 REALs)
 - Up to 128 Coils (Data Indexes 0...15 at 8-bit boundaries)
 - Up to 128 Discrete inputs (Data Indexes 0...15 at 8-bit boundaries)

Modbus Slave Data Mapping Example

Channel 1 - Modbus Slave						
		Up		Down		
No.	Register Type	Data Type	Register Start Address	Register Length	Data Index	
0	Holding registers	INT	3	3	0	
1	Holding registers	INT	32000	7	3	
2	Holding registers	INT	999	90	10	
3	Coils	BOOL	0	8	0	
4	Coils	BOOL	10	1	1	
5	Coils	BOOL	32000	17	2	
6	Coils	BOOL	999	88	5	

- Local:1:O1.Slave.HoldingRegister[0...2] = 400003...400005
- Local:1:O1.Slave.HoldingRegister[3...9] = 432000...432006
- Local:1:O1.Slave.HoldingRegister[10...99] = 400999...410088
- Local:1:O1.Slave.Coil[0].o...[0].7 = 000000...000007
- Local:1:O1.Slave.Coil[1].o = 000010
- Local:1:O1.Slave.Coil[2].o...[4].o = 0320000...032016
- Local:1:O1.Slave.Coil[5].o...[15].7 = 000999...001086

Local:2:O1.Slave	{...}	{...}		AB:5000_Mod
Local:2:O1.Slave.Run	0	Decimal	BOOL	
Local:2:O1.Slave.SequenceNumber	0	Decimal	INT	
Local:2:O1.Slave.HoldingRegister	{...}	{...}	Decimal	INT[100]
Local:2:O1.Slave.Coil	{...}	{...}	Decimal	SINT[16]
Local:2:O1.Slave.InputRegister	{...}	{...}	Decimal	INT[100]
Local:2:O1.Slave.DiscreteInput	{...}	{...}	Decimal	SINT[16]

Connection Category

The Connection category lets you complete the following tasks:

- Set the RPI rate. For more information on the RPI, see [page 27](#).
- Inhibit the module. For more information on inhibit the module, see [page 28](#).
- The connection over EtherNet/IP will be Unicast only.
- Configure whether a connection failure while the controller is in Run module causes a major or minor fault.



The Module Fault area of the Connection category is useful during module troubleshooting. For more information on the Module Fault area, see [page 92](#).

Generic ASCII

Figure 29 – Generic ASCII Connection

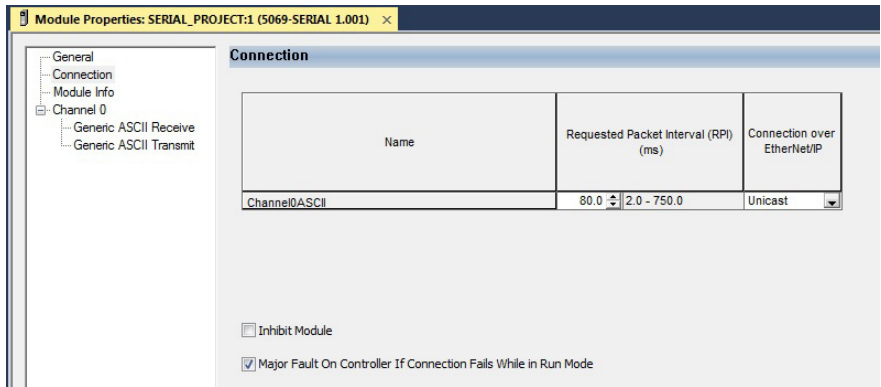


Figure 30 – Generic ASCII Module Info

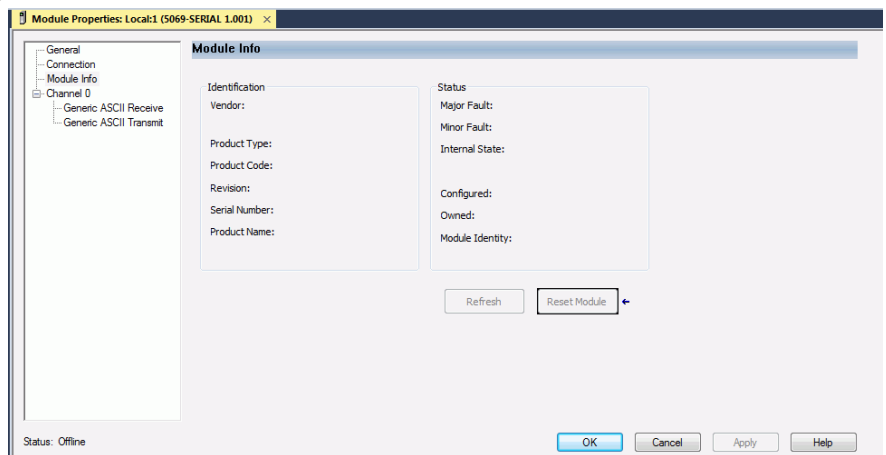
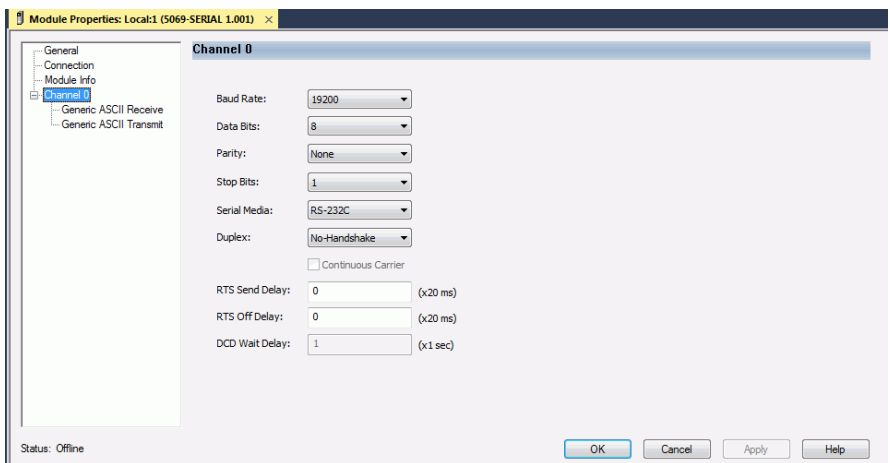


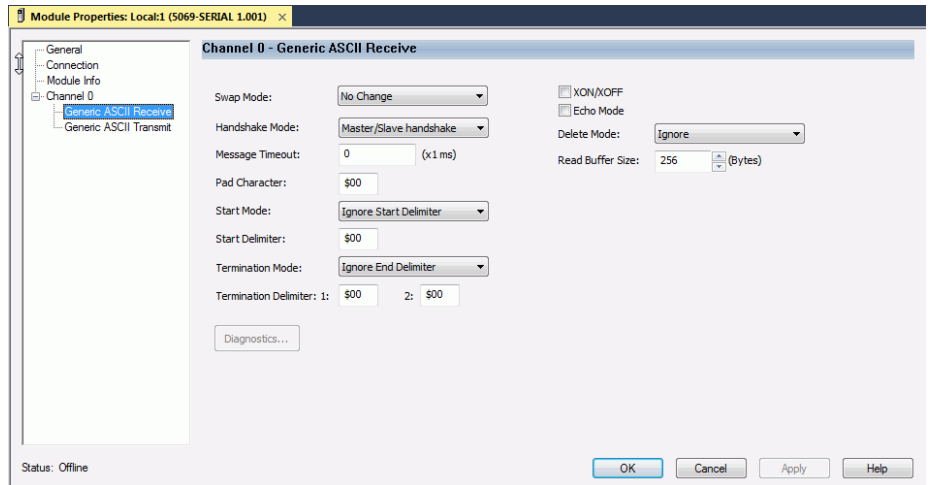
Figure 31...Figure 33 show the communication port that defines the communication rate, serial media setting, and receive and transmit settings.

Figure 31 – Generic ASCII Channel Parameters



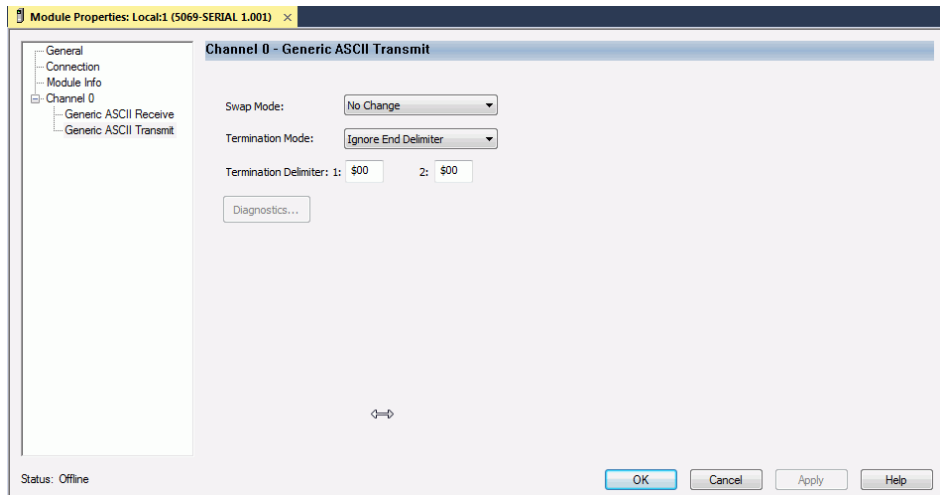
For module function definitions, see [Common Module Functions on page 31](#)

Figure 32 - Generic ASCII Receive



For more information, see [Generic ASCII Receive Functions on page 37](#)

Figure 33 - Generic ASCII Transmit



For more information, see [Generic ASCII Transmit Functions on page 34](#)

Modbus Master and Slave

Figure 34 - Modbus Master and Slave Connection

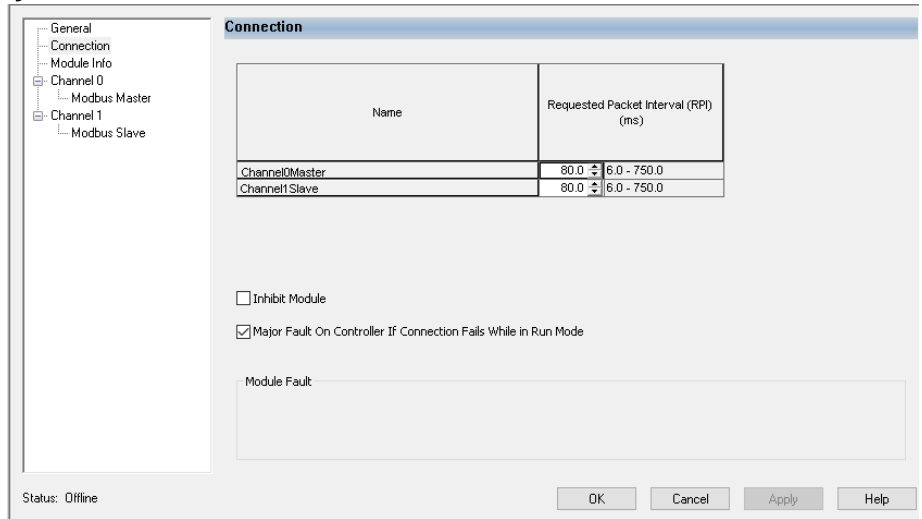
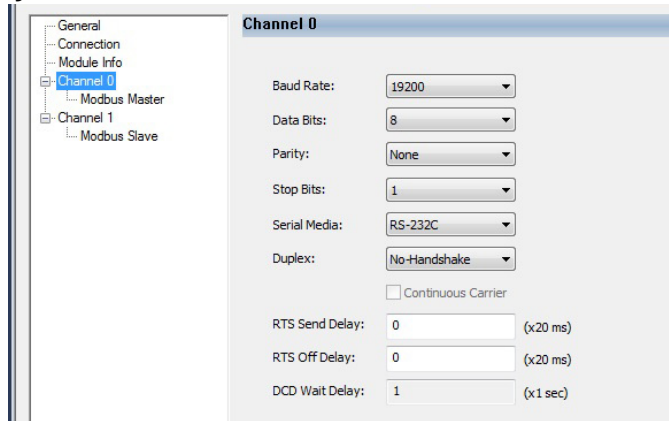


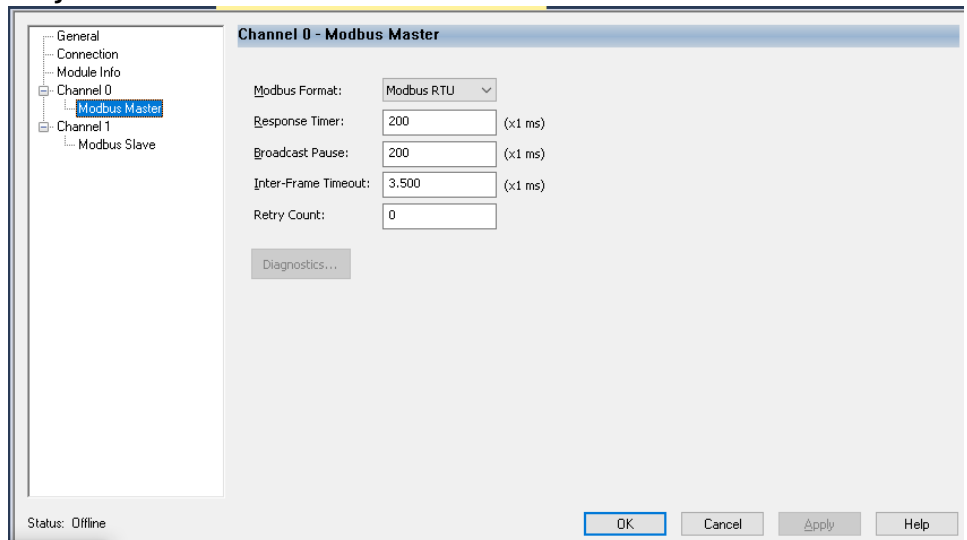
Figure 35 - Channel 0 and Channel 1 Parameters



For module function definitions, see [Common Module Functions on page 31](#).

[Figure 36](#) and [Figure 37](#) show the detailed communication settings for Modbus Master and Modbus Slave.

Figure 36 - Modbus Master Channel Parameters



For Modbus Master function definitions, see [Modbus Master Functions on page 40](#).

Figure 37 - Modbus Slave Channel Parameters



For Modbus Slave function definitions, see [Modbus Slave Functions on page 43](#).

DF1 Point to Point

Figure 38 - Connection

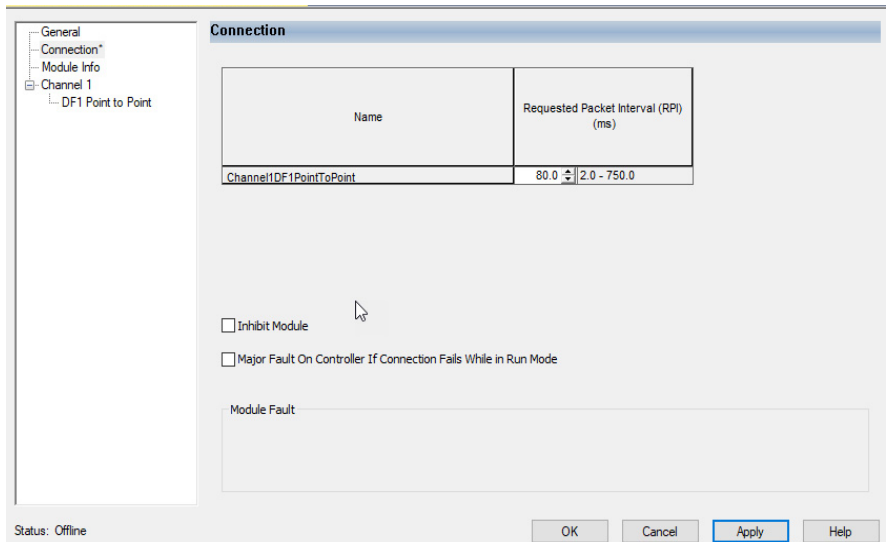


Figure 39 - Module Info

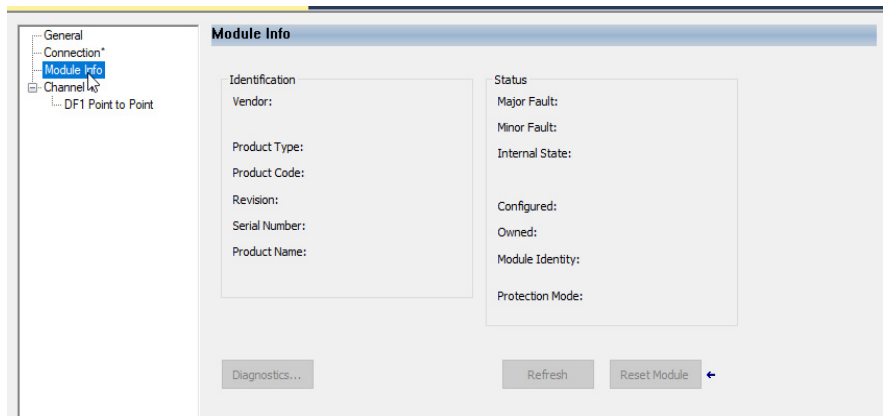
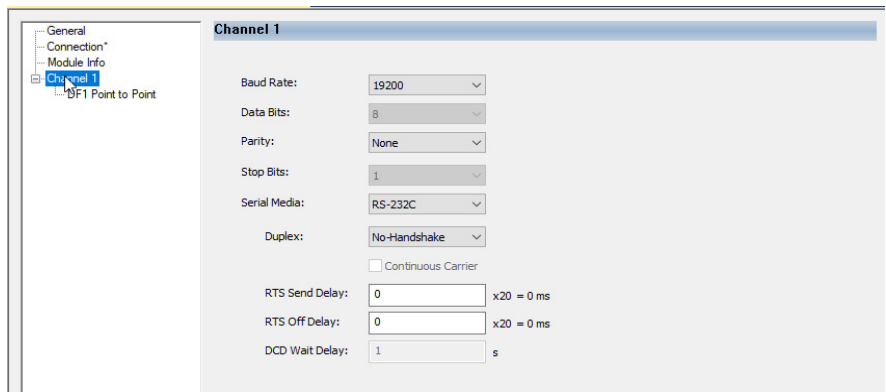


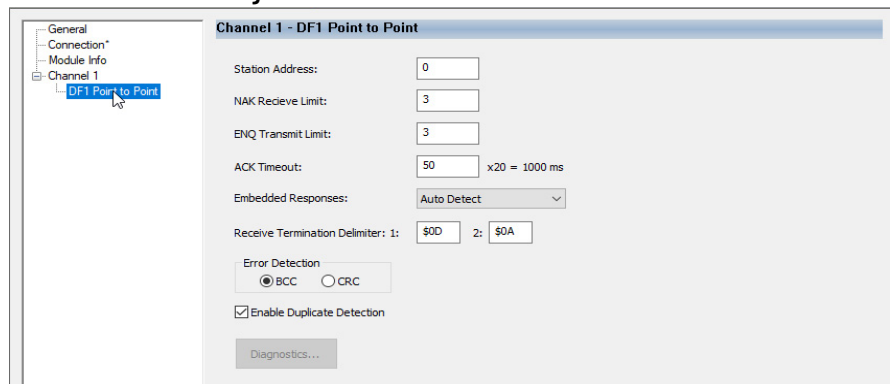
Figure 40 and Figure 41 show the communication port that defines the communication rate and transmission settings.

Figure 40 - Channel Parameters



For module function definitions, see [Common Module Functions on page 31](#)

Figure 41 - Transmission Settings



For DF1 Point to Point function definitions, see [DF1 Point to Point Parameters on page 69](#)

DF1 Master

Figure 42 - Connection

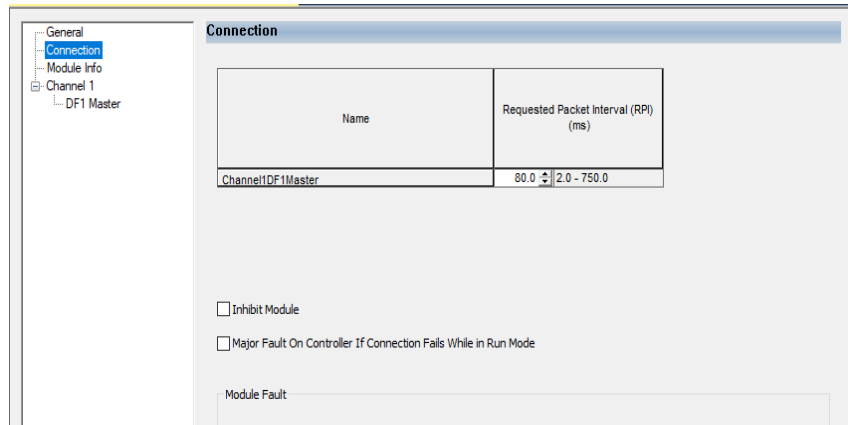


Figure 43 - Module Info

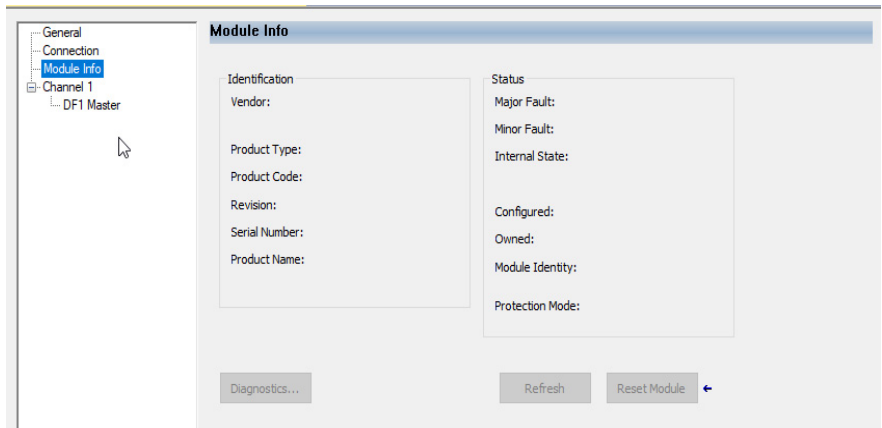
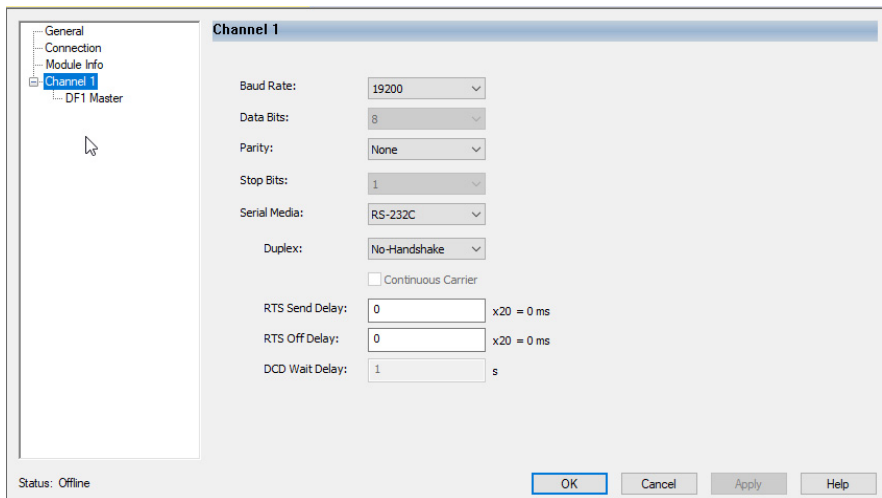


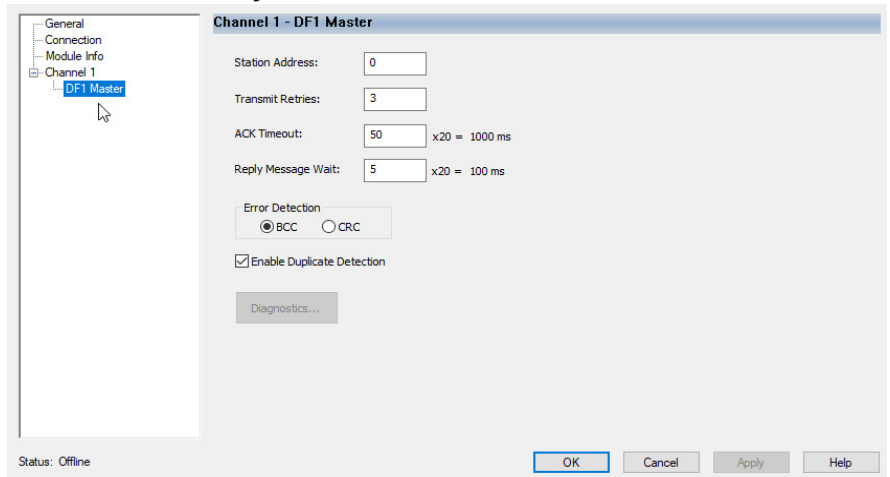
Figure 44 and Figure 45 show the communication port that defines the communication rate and transmission settings.

Figure 44 - Channel Parameters



For module function definitions, see [Common Module Functions on page 31](#)

Figure 45 - Transmission Settings



For DF1 Master function definitions, see [DF1 Master Parameters on page 69](#)

DF1 Slave

Figure 46 - Connection

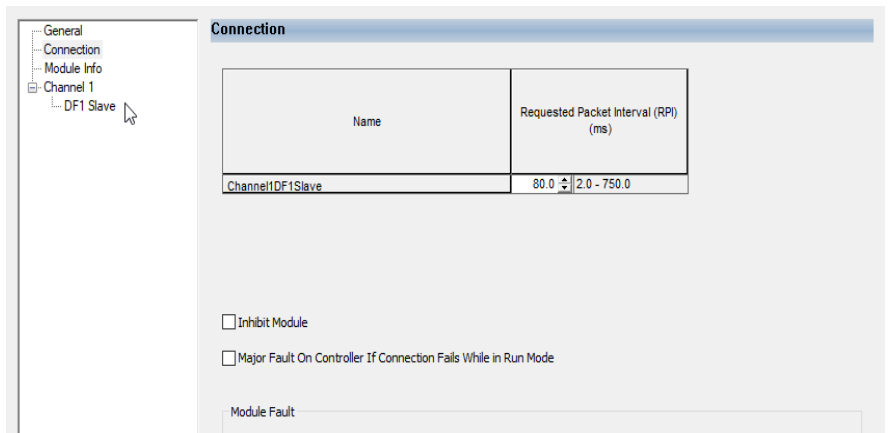
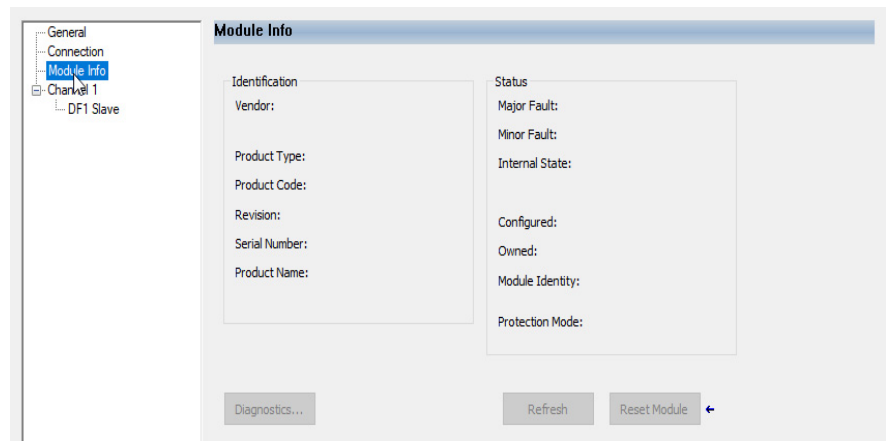
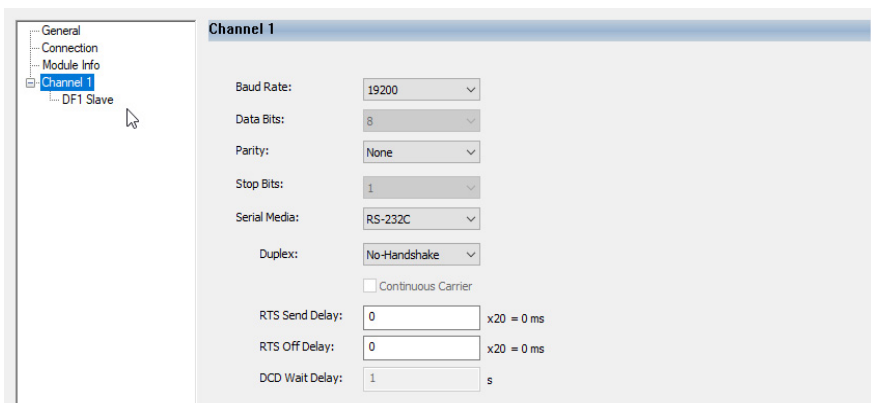


Figure 47 - Module Info



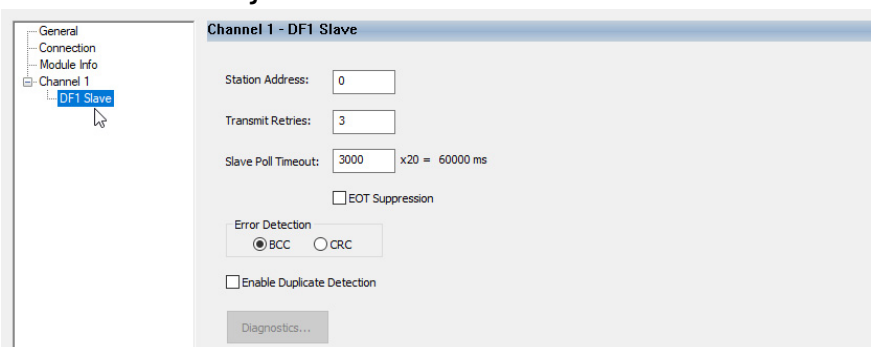
[Figure 48](#) and [Figure 49](#) show the communication port that defines the communication rate and transmission settings.

Figure 48 - Channel Parameters



For module function definitions, see [Common Module Functions on page 31](#)

Figure 49 - Transmission Settings



For DF1 Master function definitions, see [DF1 Slave Parameters on page 70](#)

DF1 Radio Modem

Figure 50 - Connection

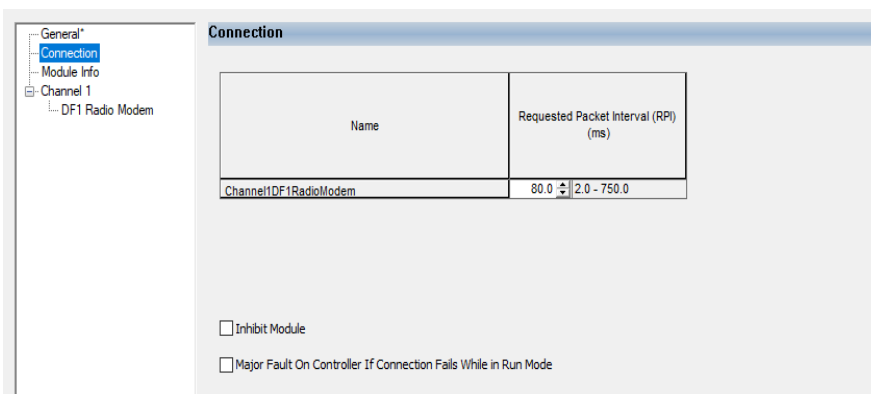


Figure 51 - Module Info

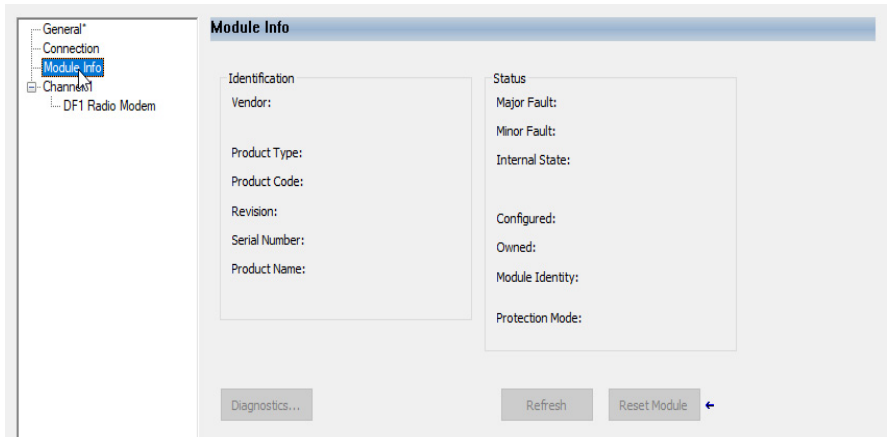
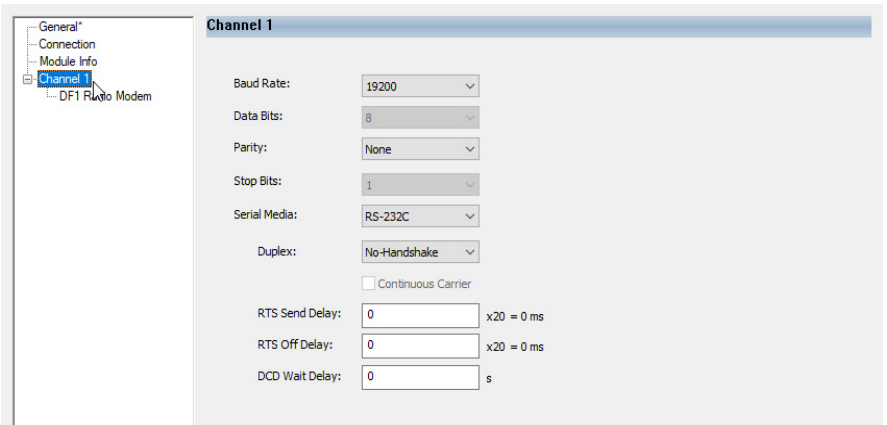


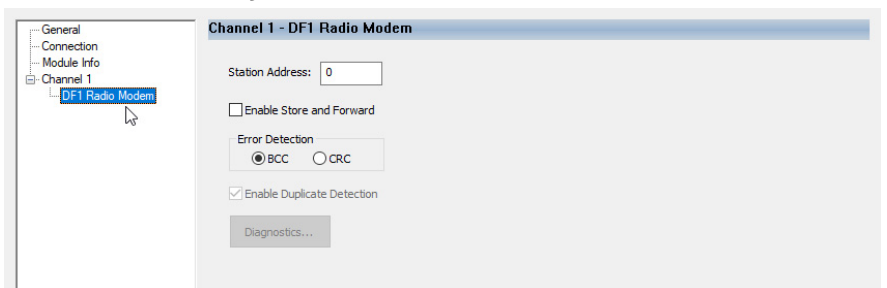
Figure 52 and Figure 53 show the communication port that defines the communication rate and transmission settings.

Figure 52 - Channel Parameters



For module function definitions, see [Common Module Functions on page 31](#)

Figure 53 - Transmission Settings



For DF1 Master function definitions, see [DF1 Radio Modem Parameters on page 70](#)

DH-485

Figure 54 - Connection

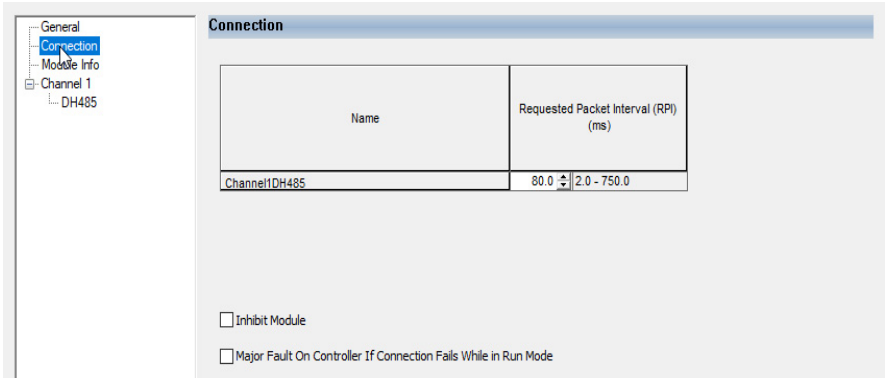


Figure 55 - Module Info

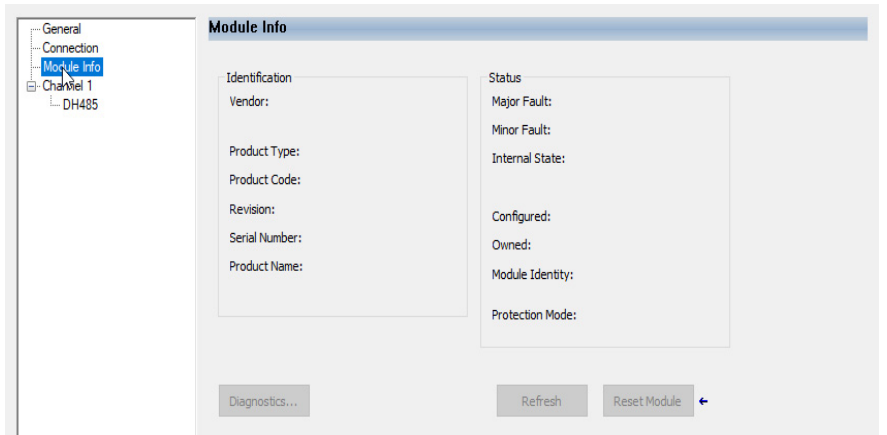
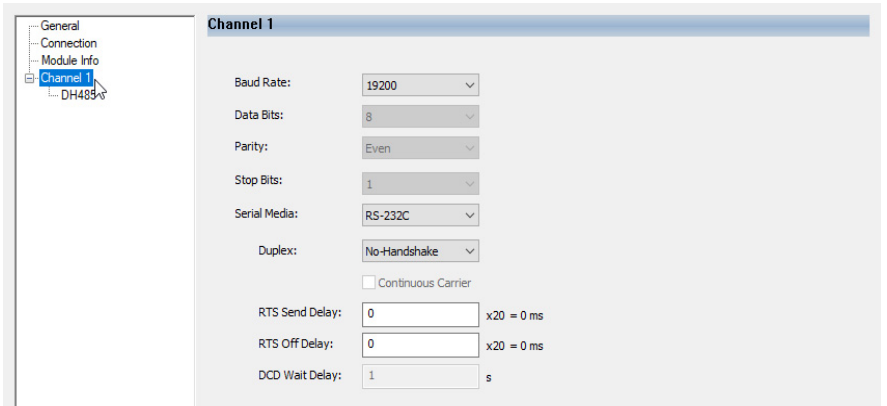


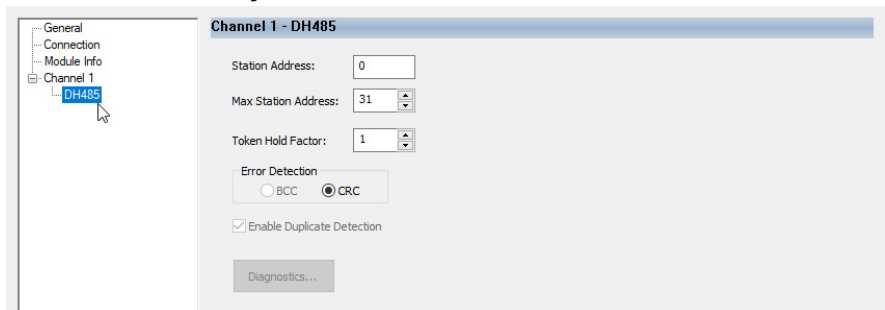
Figure 56 and Figure 57 show the communication port that defines the communication rate and transmission settings.

Figure 56 - Channel Parameters



For module function definitions, see [Common Module Functions on page 31](#)

Figure 57 - Transmission Settings

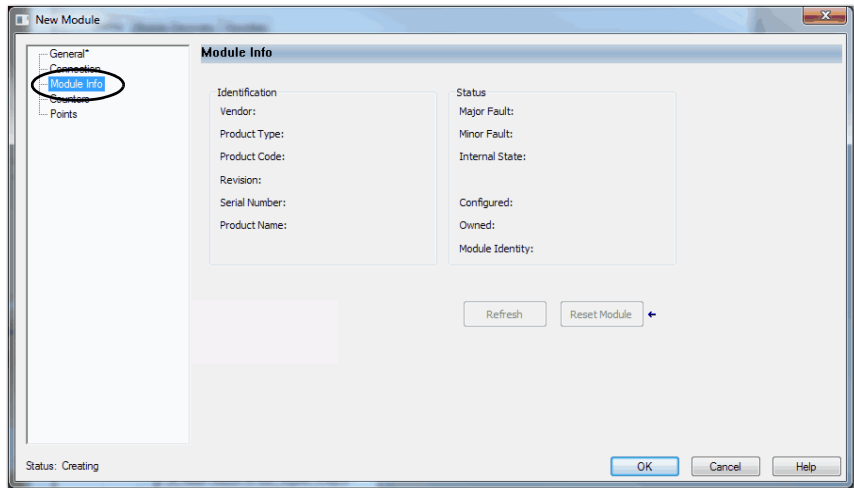


For DH-485 function definitions, see [DH-485 Functions on page 72](#)

Module Info Category

The Module Info category displays module and status information about the module when the project is online. You can use this category to complete the following:

- Determine the identity of the module.
- Refresh the data on the screen.
- Reset the module.

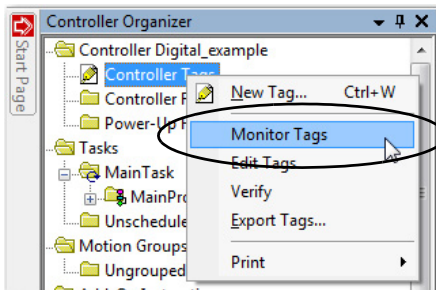


View the Module Tags

When you create a module, the Logix Designer application creates a set of tags that you can view in the Tag Editor. Each configured feature on your module has a distinct tag that is available for use in the controller program logic.

Complete the following steps to access the tags for a module.

1. In the Controller Organizer, right-click Controller Tags and choose Monitor Tags.



The Controller Tags dialog box appears with data.

2. To view the tags, click the + symbols as shown.

The screenshot shows the 'Controller Tags' dialog box. At the top, it says 'Scope: Digital_example' and 'Show: All Tags'. Below is a table with columns: Name, Value, Force Mask, Style, and Data Type. The table lists various tags, many of which are collapsed with a minus sign (-) in the 'Name' column. A red circle highlights the plus sign (+) next to the tag 'remote_ethernet_adapter:1.C.Pt01.InputOnFilter', indicating that clicking it will expand the tag to show its sub-entries.

Name	Value	Force Mask	Style	Data Type
- remote_ethernet_adapter:1.C	{...}	{...}	{...}	AB:5000_DI16_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt00	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
- remote_ethernet_adapter:1.C.Pt01	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter	13		Decimal	SINT
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.0	1		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.1	0		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.2	1		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.3	1		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.4	0		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.5	0		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.6	0		Decimal	BOOL
- remote_ethernet_adapter:1.C.Pt01.InputOnFilter.7	0		Decimal	BOOL
+ remote_ethernet_adapter:1.C.Pt01.InputOnFilter	13		Decimal	SINT
+ remote_ethernet_adapter:1.C.Pt02	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt03	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt04	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt05	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt06	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt07	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt08	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt09	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt10	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt11	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt12	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt13	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt14	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0
+ remote_ethernet_adapter:1.C.Pt15	{...}	{...}	{...}	AB:5000_DI_Channel_IB16.C:0

For more information on module tags, see [Module Tags on page 93](#).

Notes:

Troubleshoot Your Module

Compact 5000 I/O Serial Module Status Indicators

Figure 58 shows the Compact 5000 I/O serial Module Status indicators.

Figure 58 - Compact 5000 I/O Serial Status Indicators

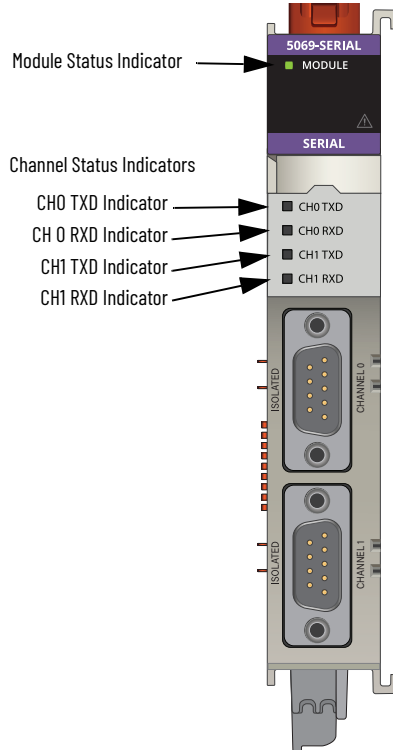


Table 25 - Channel Status Indicator Descriptions

Indicator State	Description	Recommended Action
Off	Channel is disabled or there is no communication	None
Steady Yellow	Communicating	None
Flashing Yellow	Communicating	None
Flashing Red	Serial Port Communication Error	Cycle module power. Check serial port configuration and setup.

Table 26 describes the Module (MOD) Status indicator on Compact 5000™ I/O Serial modules.

Table 26 - Module Status Indicator - Compact 5000 I/O Module

Indicator State	Description	Recommended Action
Off	The module is not powered.	None if your application does not use the module If your application uses the module and it is expected to be operating, complete the following: <ul style="list-style-type: none"> • Confirm that the system is powered. • Confirm that the module is installed properly.
Steady green	The module has a connection to the owner-controller and is operating normally.	None
Flashing green	The following conditions exist: <ul style="list-style-type: none"> • Both channels are disabled. • The module is powering up. • The module has powered up successfully. • One of the following: The module does not have a connection to the controller. A connection can result from missing, incomplete, or incorrect module configuration. 	Complete the following: <ul style="list-style-type: none"> • Troubleshoot your Logix Designer application to determine what is helping prevent a connection from the module to the controller and correct the issue. Confirm that the system conditions require the controller to be in Remote Run mode or Run mode, transition the controller to one of those modes.
Steady red	The module experienced a nonrecoverable fault.	Complete the following actions: <ol style="list-style-type: none"> 1. Cycle power to the module. If the status indicator remains in the steady red state, replace the module.
Flashing red	One of the following conditions exists: <ul style="list-style-type: none"> • A module firmware update is in progress. • A module firmware update attempt failed. • The device has experienced a recoverable fault. • A connection to the module has timed out. Note: Continuously executing IOT instruction can cause loss of connection. An interval of at least 1 ms between IOT instruction execution is necessary.	Complete one of the following: <ul style="list-style-type: none"> • Let the firmware update progress complete. • Reattempt a firmware update after one fails. • Use the Logix Designer application to determine the cause of the module fault. • The Connection and Module Info categories of the modules configuration indicate the fault type. • To clear a recoverable fault, complete one of the following: <ul style="list-style-type: none"> – Cycle module power. – Click Reset Module in the Logix Designer application project via the Module Info category of the Module Properties dialog box. If the fault does not clear after cycling power and clicking Reset Module, contact Rockwell Automation® Technical Support. • Use the Logix Designer application to determine if a connection has timed out. The Connection category in the Module Properties for the module indicates the module state, including if a connection has timed out. If a connection has timed out, determine the cause and correct it. For example, a cable failure can cause a connection timeout.

Module Tags

Topic	Page
Access the Tags	95
Channel Configured Generic ASCII Tags	96
Channel Configured for Generic ASCII	97
Channel Configured for Modbus Master	100
Channel Configured for Modbus Slave	103
Channel Configured for DF1—Point to Point	105
Channel Configured for DF1—Master	107
Channel Configured for DF1—Slave	109
Channel Configured for DF1—Radio Modem	110
Channel Configured for DH-485	111

Name Conventions

The module tags use defined naming conventions. The conventions are as follows:

- Module name
- Slot number
- Tag type and channel number—If Generic ASCII or Modbus Slave is used in the Module Definition for the channel.
- Tag type, channel number, and number of connections—If Modbus Master is used in the Module Definition for the channel.
- Parameter

Generic ASCII and Modbus Slave Name Conventions

The following is an example for a Generic ASCII or Modbus Slave tag name. The conventions for the example, 'SERIAL_PROJECT:1:Io.RunMode', would be the following:

- SERIAL_PROJECT= name of the module
- 1 = slot number
- Io = tag type (input) and channel number (o)
 - The possible tag types are I (input) and O (output)
 - The possible channel numbers are 0 or 1
- RunMode = Parameter

Modbus Master Name Conventions

The following is an example for a Modbus Slave tag name. The conventions for the example, 'SERIAL_PROJECT:1:I00.RunMode', would be the following:

- SERIAL_PROJECT= name of the module
- 1 = slot number
- I00 = tag type (input) and channel number (0), connection number(0)
 - The possible tag types are I (input) and O (output)
 - The possible channel numbers are 0 or 1
 - The possible connection numbers are 0 or 1
- RunMode = Parameter

DF1 Name Conventions

The following is an example for a DF1 tag name. The conventions for the example, 'Local:1:I1.RunMode', would be the following:

- Local= name of the module
- 1 = slot number
- I1 = tag type (input), channel number (1)
 - The possible tag types are I (input) and O (output)
 - The channel number is 1
- RunMode = Parameter

DH-485 Name Conventions

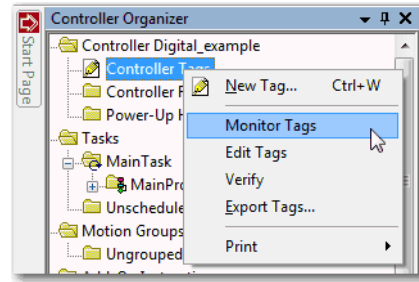
The following is an example for a DH-485 Master tag name. The conventions for the example, 'Local:1:I1.RunMode', would be the following:

- Local= name of the module
- 1 = slot number
- I1= tag type (input), channel number (1)
 - The possible tag types are I (input) and O (output)
 - The channel number is 1
- RunMode = Parameter

Access the Tags

You can view tags from the Tag Editor.

1. Open your Logix Designer application project.
2. Right-click Controller Tags and choose Monitor Tags.



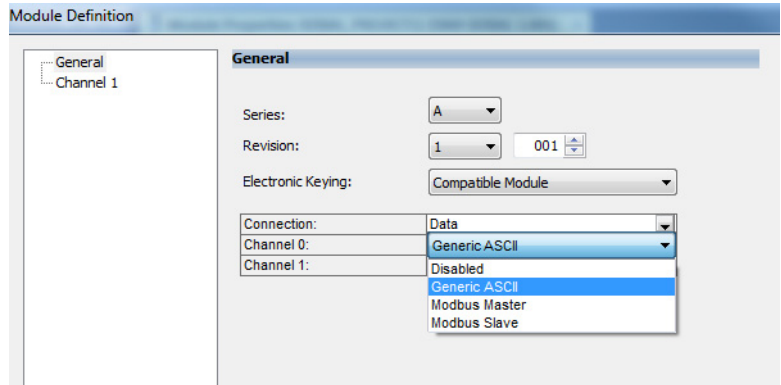
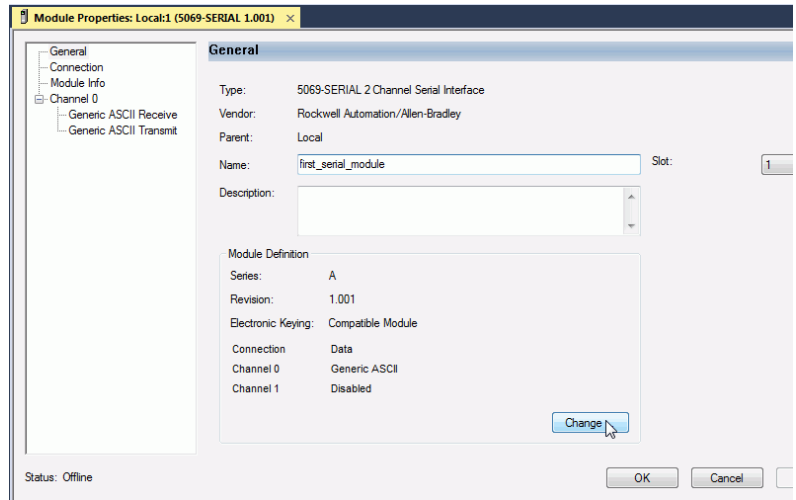
3. Open the tags as necessary to view specific tags.

 A screenshot of the 'Controller Tags - Serial_application(controller)' window. The window title is 'Controller Tags - Serial_application(controller)'. The 'Scope' is set to 'Serial_applicator' and 'Show' is set to 'All Tags'. There is a search filter 'Enter Name Filter...'. The table below lists various tags with their names, values, force masks, styles, and data types.

Name	Value	Force Mask	Style	Data Type
Local:1:ID		{...}	{...}	AB:5000_ASCII:0
Local:1:O0		{...}	{...}	AB:5000_ASCII:O:0
Local:1:O0.ASCII		{...}	{...}	AB:5000_ASCII_Channel:O:0
Local:1:O0.ASCII.TxD	0		Decimal	SINT
Local:1:O0.ASCII.RxD	0		Decimal	SINT
Local:1:O0.ASCII.TxD.Length			Decimal	INT
Local:1:O0.ASCII.ClearBuffer	0		Decimal	BOOL
Local:1:O0.ASCII.DTR	0		Decimal	BOOL
Local:1:O0.ASCII.RTS	0		Decimal	BOOL
Local:1:O0.ASCII.EXEC	0		Decimal	BOOL
Local:1:O0.ASCII.TxD.Data	{...}	{...}	Decimal	SINT[256]

Channel Configured Generic ASCII Tags

This section describes the tags that are created when you choose the Generic ASCII option for a channel in the module definition dialog box as shown in the following graphics.



Channel Configured for Generic ASCII

Input Tags

The following image shows the tags that are described in the following table.

[Table 27](#) describes the input tags of channel 1 configured to the Generic ASCII.

Name	Value	Force Mask	Style	Data Type
Local:1:IO	{...}	{...}	{...}	AB:5000_ASCII:IO
Local:1:IO.RunMode	0		Decimal	BOOL
Local:1:IO.ConnectionFaulted	0		Decimal	BOOL
Local:1:IO.DiagnosticActive	0		Decimal	BOOL
Local:1:IO.DiagnosticSequenceCount	0		Decimal	SINT
Local:1:IO.ASCII	{...}	{...}	{...}	AB:5000_ASCII_Chan...
Local:1:IO.ASCII.Fault	0		Decimal	BOOL
Local:1:IO.ASCII.Uncertain	0		Decimal	BOOL
Local:1:IO.ASCII.TxDataLost	0		Decimal	BOOL
Local:1:IO.ASCII.RxDataLost	0		Decimal	BOOL
Local:1:IO.ASCII.ParityError	0		Decimal	BOOL
Local:1:IO.ASCII.TxFIFOEmpty	0		Decimal	BOOL
Local:1:IO.ASCII.RxFIFOEmpty	0		Decimal	BOOL
Local:1:IO.ASCII.NonDelimitedRecord	0		Decimal	BOOL
Local:1:IO.ASCII.HandshakeError	0		Decimal	BOOL
Local:1:IO.ASCII.NewData	0		Decimal	BOOL
Local:1:IO.ASCII.TxDataSent	0		Decimal	BOOL
Local:1:IO.ASCII.TxDataLengthInvalid	0		Decimal	BOOL
Local:1:IO.ASCII.RxDataLengthInvalid	0		Decimal	BOOL
Local:1:IO.ASCII.FramingError	0		Decimal	BOOL
Local:1:IO.ASCII.BufferOverRun	0		Decimal	BOOL
Local:1:IO.ASCII.CTS	0		Decimal	BOOL
Local:1:IO.ASCII.RTS	0		Decimal	BOOL
Local:1:IO.ASCII.DSR	0		Decimal	BOOL
Local:1:IO.ASCII.DCD	0		Decimal	BOOL
Local:1:IO.ASCII.DTR	0		Decimal	BOOL
Local:1:IO.ASCII.XOFF	0		Decimal	BOOL
Local:1:IO.ASCII.BREAK	0		Decimal	BOOL
Local:1:IO.ASCII.TxAck	0		Decimal	SINT
Local:1:IO.ASCII.RxID	0		Decimal	SINT
Local:1:IO.ASCII.RxDataLength	0		Decimal	INT
Local:1:IO.ASCII.RxData	{...}	{...}	Decimal	SINT[256]
Local:1:IO0	{...}	{...}	{...}	AB:5000_ASCII:O:0

Table 27 - Generic ASCII Input Module Tags

Name	Data Type	Definition	Valid Values
Ix.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.ASCII.Fault	BOOL	Indicates that channel data is inaccurate and cannot be trusted for use in the application.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>

Table 27 - Generic ASCII Input Module Tags

Name	Data Type	Definition	Valid Values
Ix.ASCII.Uncertain	BOOL	Indicates that the channel data can be inaccurate but the degree of inaccuracy is not known.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.
Ix.ASCII.TxDataLost	BOOL	The transmitted data was lost. Until Clear Buffer, this bit continues to set.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.RxDataLost	BOOL	The received data was lost. Until Clear Buffer, this bit continues to set.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.ParityError	BOOL	Status that shows whether a parity error has occurred or not.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.TxFIFOEmpty	BOOL	Data in transmit FIFO. The FIFO is not empty. The output FIFO has not sent all of its data to the ASCII device.	<ul style="list-style-type: none"> 0 = Not Empty 1 = Empty
Ix.ASCII.RxFIFOEmpty	BOOL	Data in the receive FIFO. The FIFO is not empty. The input FIFO has not sent all of its data to the interface.	<ul style="list-style-type: none"> 0 = Not Empty 1 = Empty
Ix.ASCII.NonDelimitedRecord	BOOL	An input record is received and sent to the interface that was not triggered by receiving a delimiter character. This event occurs when either the buffer is filled to its maximum receive size or a Message Timeout has occurred.	<ul style="list-style-type: none"> 0 = Not Produced 1 = Produced
Ix.ASCII.HandshakeError	BOOL	Handshake error. Used for Handshake mode only.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.NewData	BOOL	New data. Used for Handshake mode only.	<ul style="list-style-type: none"> 0 = None 1 = New Data
Ix.ASCII.TxDataSent	BOOL	Indicates that the module has sent the data indicated by the Tx Transaction ID and can accept more transmit data.	<ul style="list-style-type: none"> 0 = Not Complete 1 = Complete
Ix.ASCII.TxDataLengthInvalid	BOOL	Indicates whether the Ox.ASCII.TxDataLength is valid.	<ul style="list-style-type: none"> 0 = Correct 1 = Length of TxDataLength is illegal
Ix.ASCII.RxDataLengthInvalid	BOOL	Indicates whether the Ox.ASCII.RxDataLength is valid	<ul style="list-style-type: none"> 0 = Correct 1 = Length of RxDataLength is illegal⁽¹⁾
Ix.ASCII.FramingError	BOOL	Indicates whether an error in framing has occurred.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.BufferOverrun	BOOL	Indicates whether a buffer overrun has occurred.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.ASCII.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.ASCII.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.ASCII.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.ASCII.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.ASCII.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.ASCII.XOFF	BOOL	Flag for detection Xoff (Flow control)	<ul style="list-style-type: none"> 0 = On 1 = XOff is detected
Ix.ASCII.BREAK	BOOL	Status shows whether a Break Interrupt occurred or not. If a Break Interrupt is detected, the module keeps receiving binary data from the Serial Port. The Receive data includes null Data (0x00).	<ul style="list-style-type: none"> 0 = Not Detected 1 = Detected
Ix.ASCII.TxAck	SINT	Feedback from the latest Transmit Transaction ID.	• -128...+127
Ix.ASCII.RxID	SINT	Notification of receiving by updating number.	• -128...+127
Ix.ASCII.RxDataLength	INT	Length of Produce Data of each channel.	• 1...256
Ix.ASCII.RxData[x]	SINT (256)	Received data from serial port.	• -128...+127

(1) Length greater than the Max buffer length defined in the module definition page.

Output Tags

This screen capture shows the tags that are described in the following table.

[Table 28](#) describes the output tags of channel 1 configured to the Generic ASCII

Name	Value	Force Mask	Style	Data Type
Local:1:ID		{...}	{...}	AB:5000_ASCII:I:0
Local:1:II		{...}	{...}	AB:5000_ASCII:I:0
Local:1:O0		{...}	{...}	AB:5000_ASCII:O:0
Local:1:O0.ASCII		{...}	{...}	AB:5000_ASCII_Channel:O:0
Local:1:O0.ASCII.TxID	0		Decimal	SINT
Local:1:O0.ASCII.RxID	0		Decimal	SINT
Local:1:O0.ASCII.TxDataLength	0		Decimal	INT
Local:1:O0.ASCII.ClearBuffer	0		Decimal	BOOL
Local:1:O0.ASCII.DTR	0		Decimal	BOOL
Local:1:O0.ASCII.RTS	0		Decimal	BOOL
Local:1:O0.ASCII.EXEC	0		Decimal	BOOL
Local:1:O0.ASCII.RxDataLength	0		Decimal	INT
Local:1:O0.ASCII.TxData	{...}	{...}	Decimal	SINT[256]
Local:1:O0.ASCII.TxData[0]	0		Decimal	SINT

Table 28 - Generic ASCII Output Module Tags

Name	Data Type	Definition	Valid Values
Ox.ASCII.TxID	SINT	This ID is incremented when you want to transmit data from the serial port.	• -128...+127 ⁽¹⁾
Ox.ASCII.RxID	SINT	This ID is incremented when you receive data from the serial port. It is only used in Master/Slave Handshake Mode.	• -128...+127 ⁽¹⁾
Ox.ASCII.TxDataLength	INT	Length of Transmitted Data of each channel.	• 1...256
Ox.ASCII.RxDataLength	INT	Length of Received Data of each channel.	• 1...256
Ox.ASCII.ClearBuffer	BOOL	If Clear Buffer the bit changes from 0 to 1, the Receive and Transaction buffer is cleared. In Half Duplex, the RTS signal level is cleared (set to Inactive) When it is at 0: Clearbuffer is triggered.	• 0 = No Change • 1 = Buffer Cleared
Ox.ASCII.DTR	BOOL	Signal level of DTR line sent out when rising edge of EXEC bit is detected.	• 0 = Inactive • 1 = Active
Ox.ASCII.RTS	BOOL	Signal level of the RTS line sent out when rising edge of EXEC bit is detected.	• 0 = Inactive • 1 = Active
Ox.ASCII.EXEC	BOOL	If EXEC bit changes 0 to 1, FW will output a signal level that is set in the DTR/RTS tag.	• 0 = Inactive • 1 = Active
Ox.ASCII.TxData[x]	SINT (256)	Output data from module.	• -128...+127

(1) The value of 0 must be skipped except during module power-up.

Channel Configured for Modbus Master

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type
Local:1:I0		{...}	{...}	AB:5000_ASCII:I0
Local:1:I1		{...}	{...}	AB:5000_ASCII:I0
Local:1:O0		{...}	{...}	AB:5000_ASCII:O:0
Local:1:O1		{...}	{...}	AB:5000_ASCII:O:0
Local:2:I00		{...}	{...}	AB:5000_ModbusMaster_Status_E33DF780:I:0
Local:2:I00.RunMode	0		Decimal	BOOL
Local:2:I00.ConnectionFaulted	0		Decimal	BOOL
Local:2:I00.DiagnosticActive	0		Decimal	BOOL
Local:2:I00.DiagnosticSequenceCount	0		Decimal	SINT
Local:2:I00.Fault	0		Decimal	BOOL
Local:2:I00.Uncertain	0		Decimal	BOOL
Local:2:I00.LastExecutedCommand	0		Decimal	SINT
Local:2:I00.Command00	{...}	{...}		AB:5000_Modbus_W_Response_Struct:I:0
Local:2:I00.Command00.Fault	0		Decimal	BOOL
Local:2:I00.Command00.Uncertain	0		Decimal	BOOL
Local:2:I00.Command00.ParityError	0		Decimal	BOOL
Local:2:I00.Command00.ResponseTimeout	0		Decimal	BOOL
Local:2:I00.Command00.FramingError	0		Decimal	BOOL
Local:2:I00.Command00.BufferOverRun	0		Decimal	BOOL
Local:2:I00.Command00.SlaveErrorCode	0		Decimal	SINT
Local:2:I00.Command00.ExecutionCount	0		Decimal	SINT
Local:2:O00	{...}	{...}		AB:5000_ModbusMaster_Run_B429E292:O:0
Local:2:O00.Run	0		Decimal	BOOL
Local:2:O00.Command00	{...}	{...}		AB:5000_Modbus_W_I400000_001_Struct:O:0
Local:2:O00.Command00.Data400000	0		Decimal	INT

Input Tags

In the following table, the xx in the tag names represents the channel number because the module has two channels, and both channels support the use of Modbus Master.



The yy in the tag names represents the Modbus Master command number and the zzzzzz represents the Modbus data address.

Table 29 - Modbus Master Input Tags

Name	Data Type	Definition	Valid Values
Ixx.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ixx.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ixx.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ixx.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ixx.Fault	BOOL	Indicates that channel data is inaccurate and cannot be trusted for use in the application.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ixx.Uncertain	BOOL	Indicates that the channel data can be inaccurate but the degree of inaccuracy is not known.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ixx.LastExecutedCommand	SINT	Indicates the number of the latest executed command, and the Update Counter indicates the status for command level. If this value is updated, either a new command is executed, not changed, command is not executed, or have not finished yet.	<ul style="list-style-type: none"> -(value shown before any command) 0...49
Ixx.Commandyy.Fault	BOOL	Indicates that channel data is inaccurate and cannot be trusted for use in the application. The number of Commandyy tags (where yy can be 0...49) are dynamically created depending on the command list you create.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ixx.Commandyy.Uncertain	BOOL	Indicates that the channel data can be inaccurate but the degree of inaccuracy is not known.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ixx.Commandyy.Disabled	BOOL	Status whether the Modbus Command is disabled and not executed on every scan of the module.	<ul style="list-style-type: none"> 0 = Enabled 1 = Disabled
Ixx.Commandyy.ParityError	BOOL	Status that shows whether a parity error has occurred or not.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ixx.Commandyy.ResponseTimeout	BOOL	Shows if there was a response timeout.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ixx.Commandyy.FramingError	BOOL	Shows if there was a framing error.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ixx.Commandyy.BufferOverrun	BOOL	Shows if a buffer overrun has occurred.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ixx.Commandyy.SlaveErrorCode	SINT	Indicates if a slave error code has occurred. <ul style="list-style-type: none"> IMPORTANT: A slave error code is not equal to an exception code. 	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ixx.Commandyy.ExecutionCount	SINT	The number of times the command is executed.	<ul style="list-style-type: none"> -128...127
Ixx.Commandxx.Datazzzzzz	SINT, INT, or REAL	Command read response data. The data for this tag is dynamic according to how the command list is created.	<ul style="list-style-type: none"> -128...+127 -32768...+32767

Output Tags

In the following table, the *xx* in the tag name represents the channel number because the module has two channels, and both channels support the use of Modbus Master.

Name	Value	Force Mask	Style	Data Type
Local:1:000		{...}	{...}	AB:5000_ModbusV
Local:1:000.Run		0	Decimal	BOOL
Local:1:000.Command00Disable		0	Decimal	BOOL
Local:1:000.Command01Disable		0	Decimal	BOOL
Local:1:000.Command03Disable		0	Decimal	BOOL

Table 30 - Modbus Master Output Tags

Name	Data Type	Definition	Valid Values
Oxx.Run	BOOL	Channel's operating state.	<ul style="list-style-type: none"> 0 = Idle 1 = Run⁽¹⁾
Oxx.CommandyyDisable	BOOL	Disable Modbus Master command runtime.	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Oxx.Commandyy.Datazzzzzz	SINT, INT, or REAL	Command write data. The number of Commandyy tags (where yy can be 0...49) are dynamically created depending on the command list you create.	<ul style="list-style-type: none"> -128...+127 -32768...+32767

(1) User logic must set the Run bit in order for the Modbus Master commands to execute.



The *yy* in the tag names represents the Modbus Master command number and the *zzzzzz* represents the Modbus data address.

Channel Configured for Modbus Slave

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type
Local:3:10	{...}	{...}	{...}	AB:5000_ModbusSlave:I0
Local:3:10.RunMode	0		Decimal	BOOL
Local:3:10.ConnectionFaulted	0		Decimal	BOOL
Local:3:10.DiagnosticActive	0		Decimal	BOOL
Local:3:10.DiagnosticSequenceCount	0		Decimal	SINT
Local:3:10.Slave	{...}	{...}	{...}	AB:5000_ModbusSlave_Channel:I0
Local:3:10.Slave.Fault	0		Decimal	BOOL
Local:3:10.Slave.Uncertain	0		Decimal	BOOL
Local:3:10.Slave.CRC_LRCErr	0		Decimal	BOOL
Local:3:10.Slave.ParityError	0		Decimal	BOOL
Local:3:10.Slave.IllegalDataAddress	0		Decimal	BOOL
Local:3:10.Slave.FramingError	0		Decimal	BOOL
Local:3:10.Slave.BufferOverRun	0		Decimal	BOOL
Local:3:10.Slave.SequenceNumberAck	0		Decimal	INT
Local:3:10.Slave.MasterUpdateCount	0		Decimal	INT
Local:3:10.Slave.HoldingRegister	{...}	{...}	Decimal	INT[100]
Local:3:10.Slave.Coil	{...}	{...}	Decimal	SINT[16]
Local:3:00	{...}	{...}	{...}	AB:5000_ModbusSlave:O:0
Local:3:00.Slave	{...}	{...}	{...}	AB:5000_ModbusSlave_Channel:O:0
Local:3:00.Slave.Run	0		Decimal	BOOL
Local:3:00.Slave.SequenceNumber	0		Decimal	INT
Local:3:00.Slave.HoldingRegister	{...}	{...}	Decimal	INT[100]
Local:3:00.Slave.Coil	{...}	{...}	Decimal	SINT[16]
Local:3:00.Slave.InputRegister	{...}	{...}	Decimal	INT[100]
Local:3:00.Slave.DiscreteInput	{...}	{...}	Decimal	SINT[16]

Input Tags

Table 31 - Modbus Slave Input Tags

Name	Data Type	Definition	Valid Values
Ix.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.Slave.Fault	BOOL	Indicates that channel data is inaccurate and cannot be trusted for use in the application.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.</p> <p>IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ix.Slave.Uncertain	BOOL	Indicates that the channel data can be inaccurate but the degree of inaccuracy is not known.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data <p>If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.</p> <p>IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>
Ix.Slave.CRC_LRCError	BOOL	Status shows that CRC (LRC) Error is occurred or not	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.Slave.ParityError	BOOL	Status that shows whether a parity error has occurred or not.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.Slave.IllegalDataAddress	BOOL	Status shows that user requests out of Modbus Register Address	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.Slave.BufferOverrun	BOOL	Status shows that whether Over Run is occurred or not in ASIC.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.Slave.FramingError	BOOL	Shows if there was a framing error.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.Slave.SequenceNumberAck	INT	Acknowledges the sequence number.	<ul style="list-style-type: none"> -32768...+32767
Ix.Slave.MasterUpdateCount	INT	Any Modbus change will update this counter.	<ul style="list-style-type: none"> -32768...+32767
Ix.Slave.HoldingRegister[x] ⁽¹⁾	INT	Produce Data that are written by Modbus Master as Data in Produce Tag.	<ul style="list-style-type: none"> -32768...+32767
Ix.Slave.Coil[x] ⁽²⁾	SINT	Produce Data that are written by Modbus Master as Data in Produce Tag.	<ul style="list-style-type: none"> -128...+127

(1) X represents any possible value 0...99

(2) X represents any possible value 0...15

Output Tags

Table 32 - Modbus Slave Output Tags

Name	Data Type	Definition	Valid Values
Ox.Slave.Run	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run⁽¹⁾
Ox.Slave.SequenceNumber	INT	Sequence number for updating slave data from controller	• -32768...+32767
Ox.Slave.HoldingRegister[x] ⁽²⁾	INT	Location of holding register values defined by user for the serial module	• -32768...+32767
Ox.Slave.Coil[x] ⁽³⁾	SINT	Location of slave coil values defined by user for the serial module	• -128...+127
Ox.Slave.InputRegister[x] ⁽⁴⁾	INT	Location of input register values defined by user for the serial module	• -32768...+32767
Ox.Slave.DiscreteInput[x] ⁽⁵⁾	SINT	Location of discrete input values defined by user for the serial module	• -128...+127

(1) The Run bit is to start the update of the output (O) tags values into the Serial module. The serial module will always respond to the external Modbus master, but they will be using the old values if the RUN bit is not enabled but new data is on the output O tag.

(2) X represents any possible value 0...99

(3) X represents any possible value 0...15

(4) X represents any possible value 0...99

(5) X represents any possible value 0...15

Channel Configured for DF1—Point to Point

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mas	Style	Data Type	Description	Constant
Local:3:I1	{...}	{...}		AB:5000_DF1_PointToPoint:I:0		<input type="checkbox"/>
Local:3:I1.DF1	{...}	{...}		AB:5000_DF1_Channel:I:0		
Local:3:I1.DF1.RunMode	0		Decimal	BOOL		
Local:3:I1.DF1.ConnectionFaulted	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticActive	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticSequenceCount	0		Decimal	SINT		
Local:3:I1.DF1.CTS	0		Decimal	BOOL		
Local:3:I1.DF1.RTS	0		Decimal	BOOL		
Local:3:I1.DF1.DSR	0		Decimal	BOOL		
Local:3:I1.DF1.DCD	0		Decimal	BOOL		
Local:3:I1.DF1.DTR	0		Decimal	BOOL		
Local:3:I1.DF1.MsgProcessing	0		Decimal	BOOL		
Local:3:I1.DF1Modem	{...}	{...}		AB:5000_DF1_Modem_Channel:I:0		
Local:3:I1.DF1Modem.TxAck	0		Decimal	SINT		
Local:3:I1.DF1Modem.RxId	0		Decimal	SINT		
Local:3:I1.DF1Modem.RxDataLength	0		Decimal	INT		
Local:3:I1.DF1Modem.RxData	{...}	{...}	Decimal	SINT[128]		
Local:3:O1	{...}	{...}		AB:5000_DF1_PointToPoint:O:0		<input type="checkbox"/>
Local:3:O1.DF1Modem	{...}	{...}		AB:5000_DF1_Modem_Channel:O:0		
Local:3:O1.DF1Modem.TxiD	0		Decimal	SINT		
Local:3:O1.DF1Modem.DTR	0		Decimal	BOOL		
Local:3:O1.DF1Modem.RTS	0		Decimal	BOOL		
Local:3:O1.DF1Modem.EXEC	0		Decimal	BOOL		
Local:3:O1.DF1Modem.TxDataLength	0		Decimal	INT		
Local:3:O1.DF1Modem.TxData	{...}	{...}	Decimal	SINT[128]		

Input Tags

Table 33 - DF1 Point to Point Input Tags

Name	Data Type	Definition	Valid Values
Ix.DF1.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.DF1.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DF1.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DF1.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.DF1.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.MsgProcessing	BOOL	Determine if instruction message is processing	<ul style="list-style-type: none"> 0 = Not Processing 1 = Processing
Ix.DF1Modem.TxACK	SINT	Feedback from the latest Transmit Transaction ID.	<ul style="list-style-type: none"> -128...+127
Ix.DF1Modem.RxD	SINT	Notification of receiving by updating number.	<ul style="list-style-type: none"> -128...+127
Ix.DF1Modem.RxDataLength	INT	Length of Produce Data of each channel.	<ul style="list-style-type: none"> 1...128
Ix.DF1Modem.RxData[x]	SINT (128)	Received data from serial port.	<ul style="list-style-type: none"> -128...+127

Output Tags

Table 34 - DF1 Point to Point Output Tags

Name	Data Type	Definition	Valid Values
Ox.DF1Modem.TxID	SINT	This ID is incremented when you want to transmit data from the serial port.	<ul style="list-style-type: none"> -128...+127⁽¹⁾
Ox.DF1Modem.DTR	BOOL	Signal level of DTR line sent out when rising edge of EXEC bit is detected.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ox.DF1Modem.RTS	BOOL	Signal level of the RTS line sent out when rising edge of EXEC bit is detected.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ox.DF1Modem.EXEC	BOOL	If EXEC bit changes 0 to 1, FW will output a signal level that is set in the DTR/RTS tag.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ox.DF1Modem.TxDataLength	INT	Length of Transmitted Data of each channel.	<ul style="list-style-type: none"> 1...128
Ox.DF1Modem.TxData[x]	SINT (128)	Output data from module.	<ul style="list-style-type: none"> -128...+127

(1) The value of 0 must be skipped except during module power-up.

Channel Configured for DF1–Master

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type	Description
Local:3:I1		{...}	{...}	AB:5000_DF1_Master:I:0	
Local:3:I1.DF1		{...}	{...}	AB:5000_DF1_Channel:I:0	
Local:3:I1.DF1.RunMode	0		Decimal	BOOL	
Local:3:I1.DF1.ConnectionFaulted	0		Decimal	BOOL	
Local:3:I1.DF1.DiagnosticActive	0		Decimal	BOOL	
Local:3:I1.DF1.DiagnosticSequenceCount	0		Decimal	SINT	
Local:3:I1.DF1.CTS	0		Decimal	BOOL	
Local:3:I1.DF1.RTS	0		Decimal	BOOL	
Local:3:I1.DF1.DSR	0		Decimal	BOOL	
Local:3:I1.DF1.DCD	0		Decimal	BOOL	
Local:3:I1.DF1.DTR	0		Decimal	BOOL	
Local:3:I1.DF1.MsgProcessing	0		Decimal	BOOL	
Local:3:I1.DF1Master	{...}	{...}		AB:5000_DF1_Master_Channel:I:0	
Local:3:I1.DF1Master.ActiveStation	{...}	{...}	Decimal	SINT[32]	
Local:3:I1.DF1Master.PriorityPollError	0		Decimal	BOOL	
Local:3:I1.DF1Master.NormalPollError	0		Decimal	BOOL	
Local:3:I1.DF1Master.PollGroupSizeError	0		Decimal	BOOL	
Local:3:I1.DF1Master.LatestPriorityPollNode	0		Decimal	INT	
Local:3:I1.DF1Master.LatestNormalPollNode	0		Decimal	INT	
Local:3:O1	{...}	{...}		AB:5000_DF1_Master_P0N0:O:0	
Local:3:O1.DF1Master	{...}	{...}		AB:5000_DF1_Master_P0N0_Channel:O:0	
Local:3:O1.DF1Master.MaxPriorityPollNode	0		Decimal	INT	
Local:3:O1.DF1Master.MaxNormalPollNode	0		Decimal	INT	
Local:3:O1.DF1Master.PollingListUpdateID	0		Decimal	SINT	

Input Tags

Table 35 - DF1 Master Input Tags

Name	Data Type	Definition	Valid Values
Ix.DF1.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.DF1.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DF1.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DF1.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.DF1.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.MsgProcessing	BOOL	Determine if instruction message is processing.	<ul style="list-style-type: none"> 0 = Not Processing 1 = Processing
Ix.DF1Master.ActiveStation	SINT (32)	Determine if station is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1Master.PriorityPollError	BOOL	Shows if there was a priority polling error.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.DF1Master.NormalPollError	BOOL	Shows if there was a normal polling error.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.DF1Master.PollGroupSizeError	BOOL	Shows if the Normal Poll Group size is greater than the Max Normal Poll Node.	<ul style="list-style-type: none"> 0 = Non-occurrence 1 = Occurrence
Ix.DF1Master.LatestPriorityPollNode	INT	Address of Priority Poll Node currently being polled.	<ul style="list-style-type: none"> -32768...+32767
Ix.DF1Master.LatestNormalPollNode	INT	Address of Normal Poll Node currently being polled.	<ul style="list-style-type: none"> -32768...+32767

Output Tags

Table 36 - DF1 Master Output Tags

Name	Data Type	Definition	Valid Values
Ox.DF1Master.MaxPriorityPollNode	INT	Defines the maximum number of nodes in the Priority Poll Node.	<ul style="list-style-type: none"> -32768...+32767
Ox.DF1Master.MaxNormalPollNode	INT	Defines the maximum number of nodes in the Normal Poll Node.	<ul style="list-style-type: none"> -32768...+32767
Ox.DF1Master.PollingListUpdateID	SINT	This ID is incremented when you want to update the polling list in the serial module.	<ul style="list-style-type: none"> -128...+127
Ox.DF1Master.PriorityPollNode ⁽¹⁾	INT[N]	Defines the poll sequence the DF1 Master polls for priority node.	-
Ox.DF1Master.NormalPollNode ⁽¹⁾	INT[M]	Defines the poll sequence the DF1 Master polls for normal node.	-

(1) If you configure the polling in the module definition, there will be two more tag array generated. The array is dynamically generated depending on what you configure in the definition.

Channel Configured for DF1–Slave

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type	Description	Constant
Local:3:I1	{...}	{...}		AB:5000_DF1:I:0		<input type="checkbox"/>
Local:3:I1.DF1	{...}	{...}		AB:5000_DF1_Channel:I:0		
Local:3:I1.DF1.RunMode	0		Decimal	BOOL		
Local:3:I1.DF1.ConnectionFaulted	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticActive	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticSequenceCount	0		Decimal	SINT		
Local:3:I1.DF1.CTS	0		Decimal	BOOL		
Local:3:I1.DF1.RTS	0		Decimal	BOOL		
Local:3:I1.DF1.DSR	0		Decimal	BOOL		
Local:3:I1.DF1.DCD	0		Decimal	BOOL		
Local:3:I1.DF1.DTR	0		Decimal	BOOL		
Local:3:I1.DF1.MsgProcessing	0		Decimal	BOOL		

Input Tags

Table 37 - DF1 Slave Input Tags

Name	Data Type	Definition	Valid Values
Ix.DF1.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.DF1.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DF1.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DF1.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.DF1.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.MsgProcessing	BOOL	Determine if instruction message is processing	<ul style="list-style-type: none"> 0 = Not Processing 1 = Processing

DF1 Slave mode does not have output tags.

Channel Configured for DF1–Radio Modem

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type	Description	Constant
Local:3:I1	{...}		{...}	AB:5000_DF1:I:0		<input type="checkbox"/>
Local:3:I1.DF1	{...}		{...}	AB:5000_DF1_Channel:I:0		
Local:3:I1.DF1.RunMode	0		Decimal	BOOL		
Local:3:I1.DF1.ConnectionFaulted	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticActive	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticSequenceCount	0		Decimal	SINT		
Local:3:I1.DF1.CTS	0		Decimal	BOOL		
Local:3:I1.DF1.RTS	0		Decimal	BOOL		
Local:3:I1.DF1.DSR	0		Decimal	BOOL		
Local:3:I1.DF1.DCD	0		Decimal	BOOL		
Local:3:I1.DF1.DTR	0		Decimal	BOOL		
Local:3:I1.DF1.MsgProcessing	0		Decimal	BOOL		
Local:3:O1	{...}		{...}	AB:5000_DF1_RadioMode...		<input type="checkbox"/>
Local:3:O1.DF1RadioModem	{...}		{...}	AB:5000_DF1_RadioMode...		
Local:3:O1.DF1RadioModem.StoreAndForward	{...}		Decimal	INT[16]		

Input Tags

Table 38 - DF1 Radio Modem Input Tags

Name	Data Type	Definition	Valid Values
Ix.DF1.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.DF1.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DF1.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DF1.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.DF1.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.MsgProcessing	BOOL	Determine if instruction message is processing.	<ul style="list-style-type: none"> 0 = Not Processing 1 = Processing

Output Tags

Table 39 - DF1 Radio Modem Output Tags

Name	Data Type	Definition	Valid Values
Ox.DF1RadioModem.StoreAndForward	INT (16)	Defines the stations that need to be involved in the Store and Forward function.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active

Channel Configured for DH-485

This screen capture shows the tags that are described in the following tables.

Name	Value	Force Mask	Style	Data Type	Description	Constant
Local:3:I1	{...}		{...}	AB:5000_DF1:I:0		<input type="checkbox"/>
Local:3:I1.DF1	{...}		{...}	AB:5000_DF1_Channel:I:0		
Local:3:I1.DF1.RunMode	0		Decimal	BOOL		
Local:3:I1.DF1.ConnectionFaulted	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticActive	0		Decimal	BOOL		
Local:3:I1.DF1.DiagnosticSequenceCount	0		Decimal	SINT		
Local:3:I1.DF1.CTS	0		Decimal	BOOL		
Local:3:I1.DF1.RTS	0		Decimal	BOOL		
Local:3:I1.DF1.DSR	0		Decimal	BOOL		
Local:3:I1.DF1.DCD	0		Decimal	BOOL		
Local:3:I1.DF1.DTR	0		Decimal	BOOL		
Local:3:I1.DF1.MsgProcessing	0		Decimal	BOOL		

Input Tags

Table 40 - DH-485 Input Tags

Name	Data Type	Definition	Valid Values
Ix.DF1.RunMode	BOOL	Channel's operating state	<ul style="list-style-type: none"> 0 = Idle 1 = Run
Ix.DF1.ConnectionFaulted	BOOL	Indicates if a connection is running. The module sets this tag to 0 when connected. If the module is not connected, it changes the tag to 1.	<ul style="list-style-type: none"> 0 = Connection running 1 = Connection not running
Ix.DF1.DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics active 1 = One or more diagnostics are active or the prognostics threshold is reached
Ix.DF1.DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	<ul style="list-style-type: none"> -128...+127 The value of 0 is skipped except during module power-up.
Ix.DF1.CTS	BOOL	Indicates whether the CTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.RTS	BOOL	Indicates whether the RTS is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DSR	BOOL	Indicates whether the DSR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DCD	BOOL	Indicates whether the DCD is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.DTR	BOOL	Indicates whether the DTR is active.	<ul style="list-style-type: none"> 0 = Inactive 1 = Active
Ix.DF1.MsgProcessing	BOOL	Determine if instruction message is processing	<ul style="list-style-type: none"> 0 = Not Processing 1 = Processing

DH-485 mode does not have output tags.

Notes:

Modbus Command Codes

Topic	Page
Read Coil Status (Function Code 01)	113
Read Input Status (Function Code 02)	114
Read Holding Registers (Function Code 03)	115
Read Input Registers (Function Code 04)	116
Force Single Coil (Function Code 05)	117
Preset Single Register (Function Code 06)	118
Force Multiple Coils (Function Code 15)	119
Preset Multiple Registers (Function Code 16)	120

Master Command List Function Codes

Read Coil Status (Function Code 01)

Query

This function allows you to obtain the ON/OFF status of logic coils (Modbus ox range) used to control discrete outputs from the addressed slave only. Broadcast mode is not supported with this function code. In addition to the slave address and function fields, the message requires that the information field contain the initial coil address to be read (Starting Address) and the number of locations that are interrogated to obtain status data.

The addressing allows up to 2000 coils to be obtained at each request. However, the specific slave device can have restrictions that lower the maximum quantity. The coils are numbered from zero; (coil number 1 = zero, coil number 2 = one, coil number 3 = two, and so on).

The following table is a sample read output status request to read coils 0020 to 0056 (37 coils) from slave device number 11.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Point High	Data Start Point Low	Number of Points High	Number of Points Low	Error Check Field (2 bytes)
0B	01	00	13	00	25	CRC

Response

An example response to Read Coil Status is as shown in the following table. The data is packed one bit for each coil. The response includes the slave address, function code, quantity of data characters, the data characters, and error checking. Data is packed with one bit for each coil (1 = ON, 0 = OFF). The low-order bit of the first character contains the addressed coil, and the remainder follows. For coil quantities that are not even multiples of eight, the last characters are completed with zeros at high-order end. The quantity of data characters is always specified as quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the slave interface device is serviced at the end of a controller's scan, data reflects coil status at the end of the scan. Some slaves limit the quantity of coils provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status from sequential scans.

Node Address	Function Code	Byte Count	Data Coil Status 20...27	Data Coil Status 28...35	Data Coil Status 36...43	Data Coil Status 44...51	Data Coil Status 52...56	Error Check Field (2 bytes)
0B	01	05	CD	6B	B2	0E	1B	CRC

The status of coils 20...27 is shown as CD (HEX) = 1100 1101 (Binary). Reading from left to right, this status shows that coils 27, 26, 23, 22, and 20 are all on. The other Data Coil Status bytes are decoded similarly. Due to the quantity of coil statuses that are requested, the last data field, which is shown 1B (HEX) = 0001 1011 (Binary), contains the status of only five coils (52...56) instead of eight coils. The three leftmost bits are provided as zeros to fill the 8-bit format.

Read Input Status (Function Code 02)

Query

This function allows you to obtain the ON/OFF status of discrete inputs (Modbus IX range) in the addressed slave. PC Broadcast mode is not supported with this function code. In addition to the slave address and function fields, the message requires that the information field contain the initial input address to be read (Starting Address) and the number of locations that are interrogated to obtain status data.

The addressing allows up to 2000 inputs to be obtained at each request; however, the specific slave device can have restrictions that lower the maximum quantity. The inputs are numbered from zero; (input 10001 = zero, input 10002 = one, input 10003 = two, and so on, for a 584).

The following table is a sample read input status request to read inputs 10197 to 10218 (22 coils) from slave number 11.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Point High	Data Start Point Low	Number of Points High	Number of Points Low	Error Check Field (2 bytes)
0B	02	00	C4	00	16	CRC

Response

An example response to Read Input Status is as shown in the following table. The data is packed one bit for each input. The response includes the slave address, function code, quantity of data characters, the data characters, and error checking. Data is packed with one bit for each input (1=ON, 0=OFF). The lower-order bit of the first character contains the addressed input, and the remainder follows. For input quantities that are not even multiples of eight, the last characters are completed with zeros at high-order end. The quantity of data characters is always specified as a quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the slave interface device is serviced at the end of a controller's scan, the data reflect input status at the end of the scan. Some slaves limit the quantity of inputs provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status for sequential scans.

Node Address	Function Code	Byte Count	Data Discrete Input 10197...10204	Data Discrete Input 10205...10212	Data Discrete Input 10213...10218	Error Check Field (2 bytes)
0B	02	03	AC	DB	35	CRC

The status of inputs 10197...10204 is shown as AC (HEX) = 10101 1100 (binary). Reading left to right, this show that inputs 10204, 10202, and 10199 are all on. The other input data bytes are decoded similar.

Due to the quantity of input statuses that are requested, the last data field that is shown as 35 HEX = 0011 0101 (binary) contains the status of only 6 inputs (10213...10218) instead of 8 inputs. The two leftmost bits are provided as zeros to fill the 8-bit format.

Read Holding Registers (Function Code 03)

Query

This function allows you to retrieve the contents of holding registers 4xxxx (Modbus 4x range) in the addressed slave. The registers can store the numerical values of associated timers and counters that can be driven to external devices. The addressing allows retrieving up to 125 registers at each request; however, the specific slave device can have restrictions that lower this maximum quantity. The registers are numbered form zero (40001 = zero, 40002 = one, and so on). The broadcast mode is not allowed.

The following example reads registers 40108...40110 (three registers) from slave number 11.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Registers High	Data Start Registers Low	Data Number of Registers High	Data Number of Registers Low	Error Check Field (2 bytes)
0B	02	00	6B	00	03	CRC

Response

The addressed slave responds with its address and the function code, followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high-order bits and the second, the low-order bits.

Because the slave interface device is normally serviced at the end of the controller's scan, the data reflect the register content at the end of the scan. Some slaves limit the quantity of register content provided each scan; thus for large register quantities, multiple transmissions are made using register content from sequential scans.

In the example below, the registers 40108...40110 have the decimal contents 555, 0, and 100 respectively.

Node Address	Function Code	Byte Count	High Data	Low Data	High Data	Low Data	High Data	Low Data	Error Check Field (2 bytes)
0B	03	06	02	2B	00	00	00	64	CRC

Read Input Registers (Function Code 04)

Query

This function retrieves the contents of the controller's input registers from the Modbus 3x range. These locations receive their values from devices that are connected to the I/O structure and can only be referenced, not altered from within the controller. The addressing allows retrieving up to 125 registers at each request; however, the specific slave device can have restrictions that lower this maximum quantity. The registers are numbered for zero (30001 = zero, 30002 = one, and so on). Broadcast mode is not allowed.

The following example requests the contents of register 30009 in slave number 11.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Point High	Data Start Point Low	Data Number of Points High	Data Number of Points Low	Error Check Field (2 bytes)
0B	04	00	08	00	01	CRC

Response

The addressed slave responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high-order bits and the second, the low-order bits.

Because the slave interface is normally serviced at the end of the controller's scan, the data reflect the register content at the end of the scan. Each PC limits the quantity of register contents provided each scan; thus for large register quantities, multiple PC scans are required, and the data that is provided is from sequential scans.

In the following example, the register 30009 contains the decimal value 0.

Node Address	Function Code	Byte Count	Data Input Register High	Data Input Register Low	Error Check Field (2 bytes)
0B	04	02	00	00	CRC

Force Single Coil (Function Code 05)

Query

This Function Code forces one coil (Modbus 0x range) either ON or OFF. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coil is disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 0001 = zero, coil 0002 = one, and so on). The data value 65,280 (FF00 HEX) sets the coil ON and the value zero turns it OFF; all other values are illegal and do not affect that coil.

The use of slave address 00 (Broadcast Mode) forces all attached slaves to modify the desired coil.



Functions 5, 6, 15, and 16 are the only messages that are recognized as valid for broadcast.

The following example is a request to slave number 11 to turn ON coil 0173.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Bit High	Data Start Bit Low	Number of Bits High	Number of Bits Low	Error Check Field (2 bytes)
0B	05	00	AC	FF	00	CRC

Response

The normal response to the Command Request is to retransmit the message as received after the coil state has been altered.

Node Address	Function Code	Data Coil Bit High	Data Coil Bit Low	Data On/Off	Data	Error Check Field (2 bytes)
0B	05	00	AC	FF	00	CRC

The forcing of a coil via Modbus function 5 happens regardless of whether the addressed coil is disabled or not (In ProSoft products, the coil is only affected if you implement the necessary Ladder Logic).

IMPORTANT The Modbus protocol excludes standard functions for testing or changing the DISABLE state of discrete inputs or outputs. Where applicable, this can be accomplished via device-specific Program commands (In ProSoft products, this is only accomplished through Ladder Logic programming).

Coils that are reprogrammed in the controller logic program are not automatically cleared upon power-up. Thus, if such a coil is set ON by function Code 5 and (even months later), an output is connected to that coil, the output is 'hot'.

Preset Single Register (Function Code 06)

Query

This Function Code allows you to modify the contents of a Modbus 4x range in the slave. This code writes to one register only. Any holding register that exists within the controller can have its contents changed by this message. However, because the controller is actively scanning, it can also alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller. Unused high-order bits must be set to zero.

When used with slave address zero (Broadcast mode), all slave controllers load the specified register with the contents specified.



- Functions 5, 6, 15, and 16 are the only messages that are recognized as valid for broadcast.
- This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

The following example is a request to write the value '3' to register 40002 in slave 11.

Node Address	Function Code	Data Start Bit High	Data Start Bit Low	Preset Data Register High	Preset Data Register Low	Error Check Field (2 bytes)
0B	06	00	01	00	03	CRC

Response

The response to a preset single register request is to retransmit the query message after the register has been altered.

Node Address	Function Code	Data Register High	Data Register Low	Preset Data Register High	Preset Data Register Low	Error Check Field (2 bytes)
0B	06	00	01	00	03	CRC

Force Multiple Coils (Function Code 15)

Query

This function forces each coil (Modbus 0x range) in a consecutive block of coils to a desired ON or OFF state. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coils are disabled, the controller can also alter the state of the coil.

Coils are numbered from zero (coil 00001 = zero, coil 00002 = one, and so on). The desired status of each coil is packed in the data field, one bit for each coil (1= ON, 0= OFF). The use of slave address 0 (Broadcast Mode) forces all attached slaves to modify the desired coils.



Functions 5, 6, 15, and 16 are the only messages that are recognized as valid for broadcast.

The following example forces 10 coils starting at address 20 (13 HEX). The two data fields, CD =1100 and 00 = 0000 000, indicate that coils 27, 26, 23, 22, and 20 are to be forced on.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Coil Address High	Coil Address Low	Number of Coils High	Number of Coils Low	Byte Count	Force Data High 20...27	Force Data Low 28...29	Error Check Field (2 bytes)
0B	0F	00	13	00	0A	02	CD	01	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Node Address	Function Code	Coil Address High	Coil Address Low	Number of Coils High	Number of Coils Low	Error Check Field (2 bytes)
0B	0F	00	13	00	0A	CRC

Preset Multiple Registers (Function Code 16)

Query

The Function Code allows you to modify the contents of a Modbus 4x range in the slave. This writes up to 125 registers at a time. Since the controller is actively scanning, it can alter the content of any holding register at any time.



Functions 5, 6, 15, and 16 are the only messages that are recognized as valid for broadcast.

The following example is a request to write two registers starting at register 4002 in slave 11.



This is the structure of the message being sent out to the Modbus network. The following byte values are in hexadecimal display.

Node Address	Function Code	Data Start Address High	Data Start Address Low	Number of Points High	Number of Points Low	Byte Count	High Data	Low Data	High Data	Low Data	Error Check Field (2 bytes)
0B	10	00	01	00	02	04	00	0A	01	02	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Node Address	Function Code	Data Start Address High	Data Start Address Low	Number of Points High	Number of Points Low	Error Check Field (2 bytes)
0B	10	00	01	00	02	CRC

Programming Example

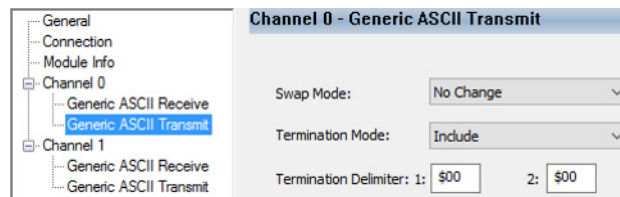
Topic	Page
Generic ASCII Sample Code	121
Modbus Master Command List	124
Modbus Slave Address Mapping Table	124
Modbus Sample Code Configuration Example	124

Generic ASCII Sample Code

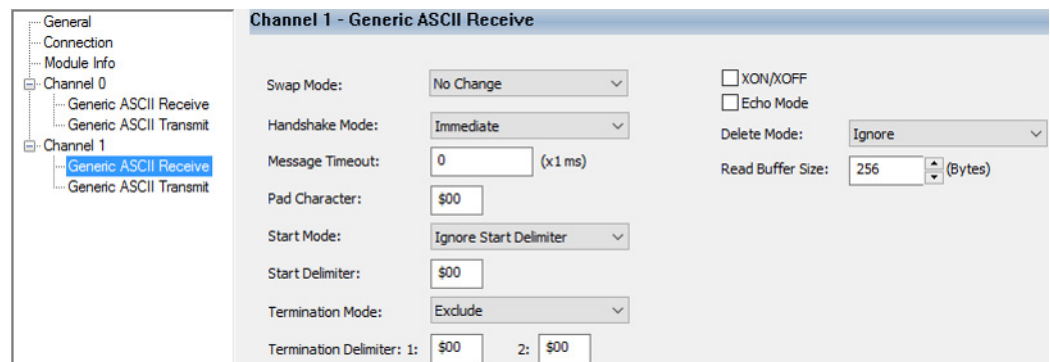
The following images show sample code for Generic ASCII Transmit data and Generic ASCII Receive data.

Transmit and Receive Channel Configurations

The following image shows the Generic ASCII Transmit Configuration.

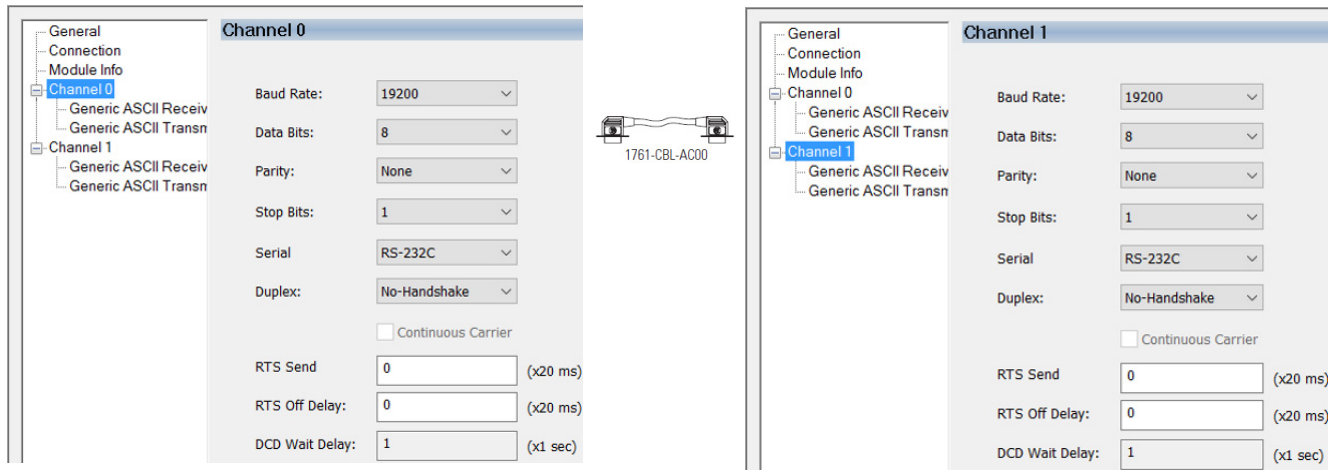


The following image shows the Generic ASCII Receive Configuration.

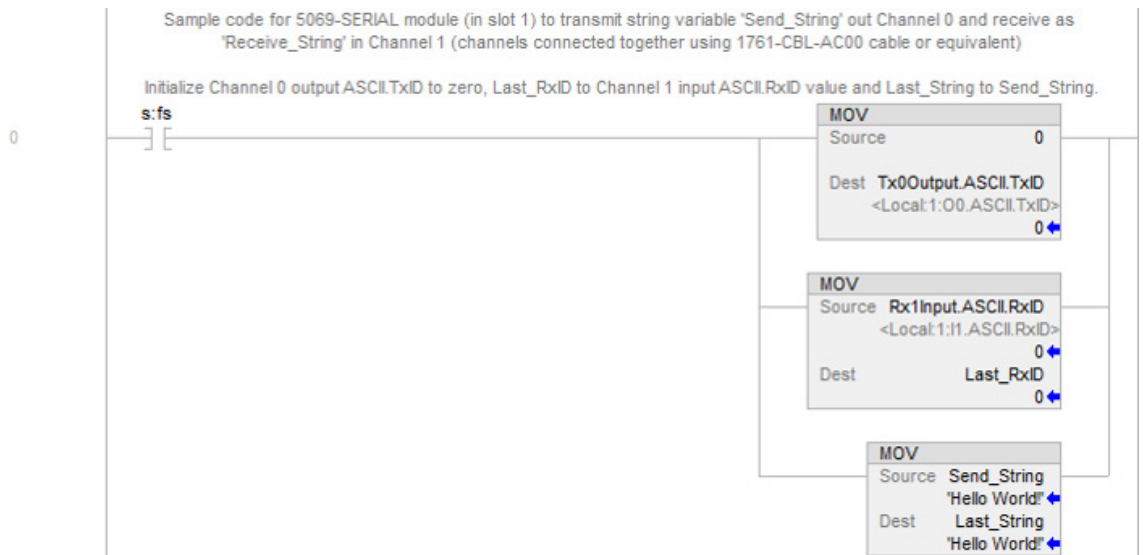


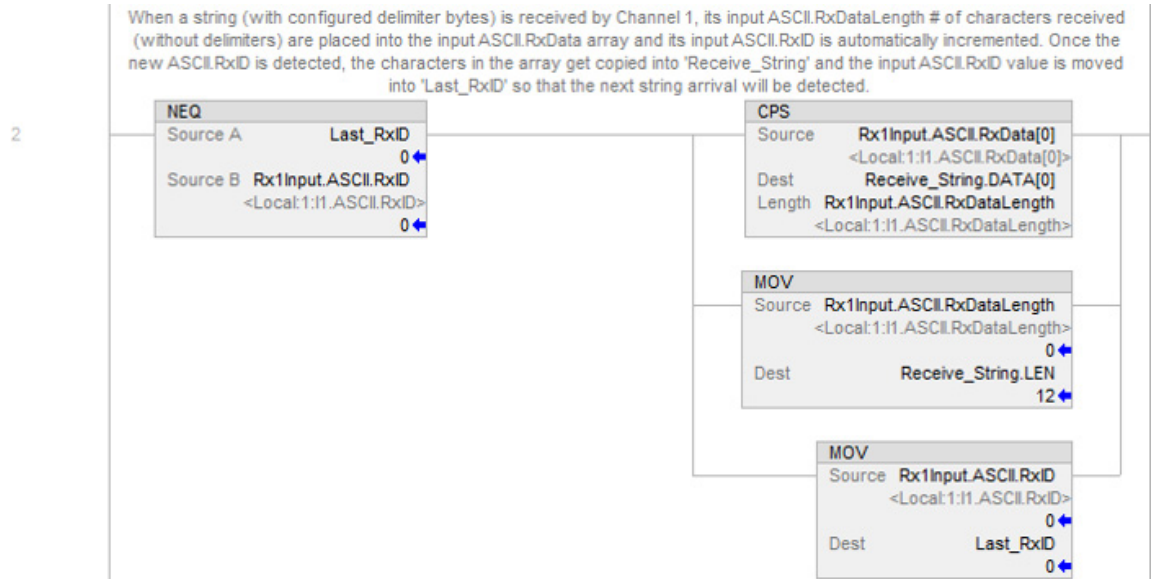
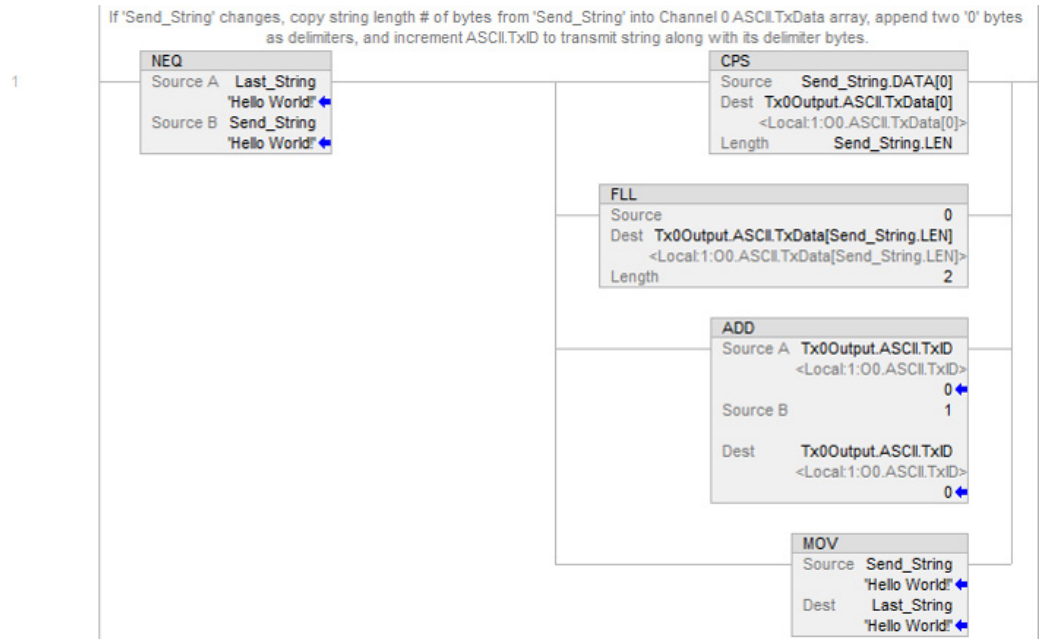
To test the ASCII capabilities by transmitting out Channel 0 and receive in Channel 1, connect the two serial ports together with a 1761-CBL-AC00 cable. On the Module Properties Channel screens, verify that both channels are configured identically for RS-232C. See the following images.

Figure 59 - Channel 0 to Channel 1 Communication



Sample Code





Modbus Sample Code Configuration

The following images show sample code configuration for Modbus Master and Modbus Slave communication.

Modbus Master Command List

Channel 0 - Modbus Master										
Up Down										
Command	Communication Method	Data Type	Function Code	Slave Address	Modbus Address Offset	Data Length	Poll Interval (s)	Swap Mode	Fault Enable	Fault Value
0	Continuous	INT	Read Holding Registers	1	0	100	0	No Change	<input checked="" type="checkbox"/>	65535
1	Continuous	BOOL	Read Coils	1	0	128	0	No Change	<input type="checkbox"/>	
2	Continuous	INT	Read Input Registers	1	0	100	0	No Change	<input type="checkbox"/>	
3	Continuous	BOOL	Read Discrete Inputs	1	0	128	0	No Change	<input type="checkbox"/>	
4	Conditional	INT	Write Single Register	1	0			No Change	<input type="checkbox"/>	
5	Conditional	BOOL	Write Single Coil	1	0			No Change	<input type="checkbox"/>	

Modbus Slave Address Mapping Table

Channel 1 - Modbus Slave						
Up Down						
No.	Register Type	Data Type	Register Start Address	Register Length	Data Index	
0	Holding registers	INT	0	100	0	
1	Input registers	INT	0	100	0	
2	Coils	BOOL	0	128	0	
3	Discrete inputs	BOOL	0	128	0	

Modbus Sample Code Configuration Example

To test the Modbus capabilities by configuring Channel 0 for Master and Channel 1 for Slave, connect the two serial ports together with a 1761-CBL-AC00 cable. On the Module Properties Channel screen, verify that both channels are configured identically for RS-232C.

Channel 0

Baud Rate: 19200

Data Bits: 8

Parity: None

Stop Bits: 1

Serial: RS-232C

Duplex: No-Handshake

Continuous Carrier

RTS Send: 0 (x20 ms)

RTS Off Delay: 0 (x20 ms)

DCD Wait Delay: 1 (x1 sec)

1761-CBL-AC00

Channel 1

Baud Rate: 19200

Data Bits: 8

Parity: None

Stop Bits: 1

Serial: RS-232C

Duplex: No-Handshake

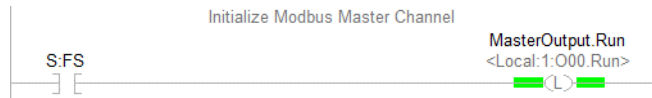
Continuous Carrier

RTS Send: 0 (x20 ms)

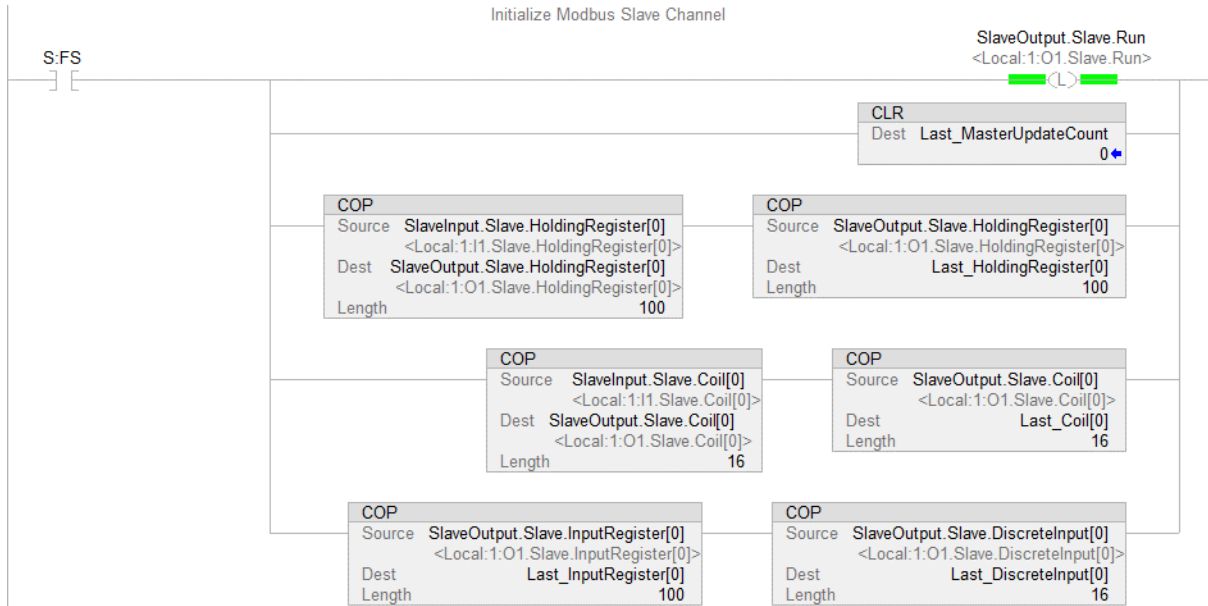
RTS Off Delay: 0 (x20 ms)

DCD Wait Delay: 1 (x1 sec)

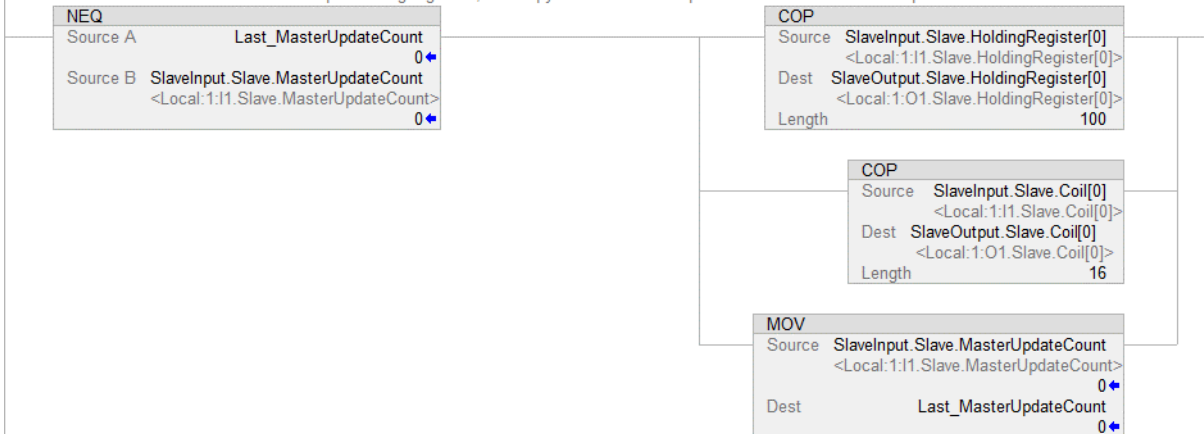
Modbus Master Sample Code

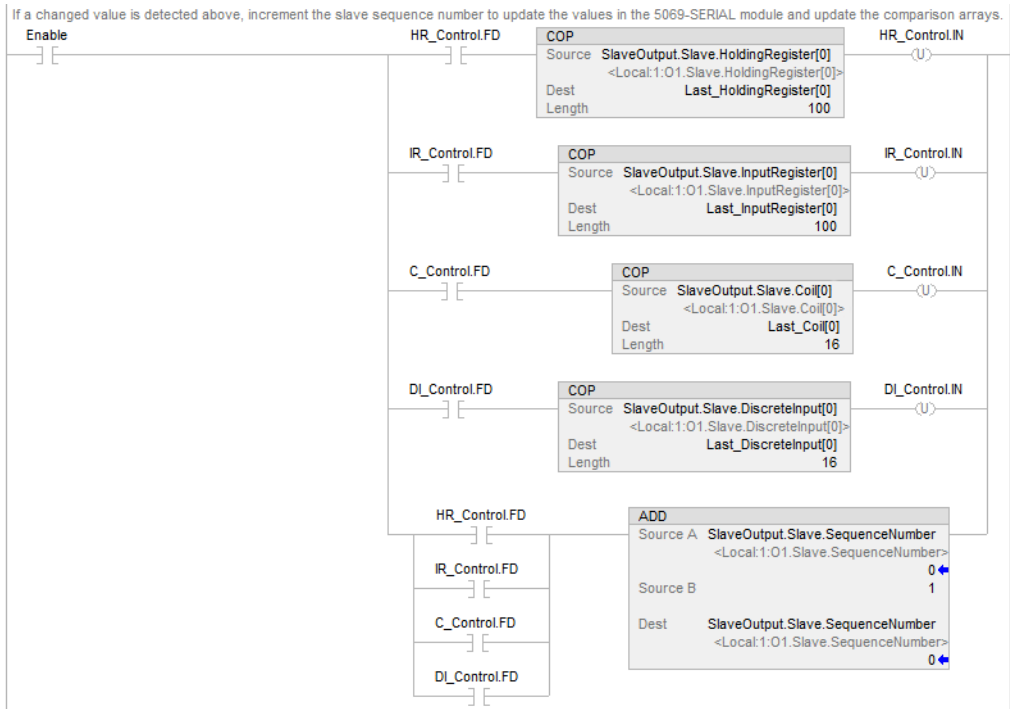
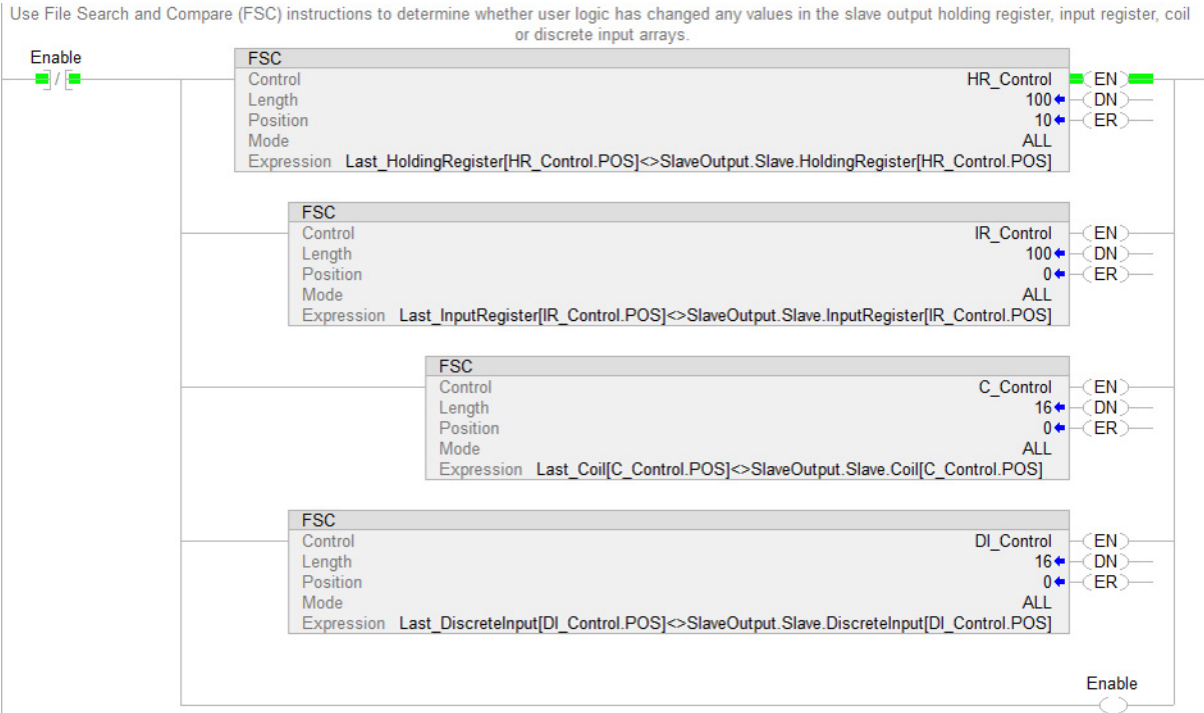


Modbus Slave Sample Code



Whenever MasterUpdateCount increments, indicating that a write command has been received by the slave, copy the latest slave input holding register values to the slave output holding registers, and copy the latest slave input coil values to the slave output coil.





Example MSG Control Blocks

The following is coding examples to create MSG Read and Write control blocks:

- [Logix write message to another Logix controller example.](#)
- [Read MSG from a Logix to a Logix Controller example.](#)
- [Write MSG from a Logix to a PLC-5® or SLC™ 500 controller example.](#)
- [Read MSG from a Logix to a PLC-5 or SLC 500 controller example.](#)
- [Write MSG instruction from a Logix to an SLC or MicroLogix™ controller example.](#)
- [Read MSG instruction from a Logix to an SLC or MicroLogix Controller](#)

Figure 60 - [1] 5069-SERIAL/A SerialR2

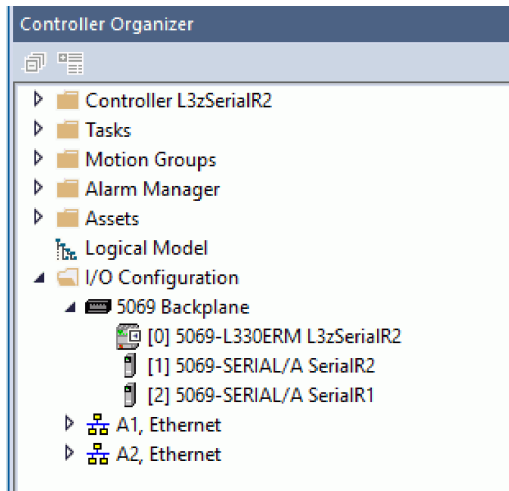


Figure 61 - Logix write message to another Logix controller example.

Local controller-scoped tag containing data to be written.

Remote controller-scoped tag into which data will be written.

Path (out serial port) to station (254).

Figure 62 - Read MSG from a Logix to a Logix Controller example.

The diagram shows a 'Read Message from Logix' block with 'Message Control message2'. Below it are two 'Message Configuration - message2' windows. The left window shows 'Message Type: PLC5 Typed Read', 'Source Element: @remotedata', and 'Destination Element: read'. The right window shows 'Path: SerialR2, 3, 254' and 'Communication Method: CIP'. Arrows point from the text labels to the corresponding fields in the configuration windows.

Local controller-scoped tag containing data to be read.

Remote controller-scoped tag into which data will be copied.

Path (out serial port) to station (254).

Figure 63 - Write MSG from a Logix to a PLC-5® or SLC™ 500 controller example.

The diagram shows a 'Write Message to PLC-5' block with 'Message Control message2'. Below it are two 'Message Configuration - message2' windows. The left window shows 'Message Type: PLC5 Typed Write', 'Source Element: write', and 'Destination Element: S:23'. The right window shows 'Path: SerialR2, 3, 254' and 'Communication Method: CIP'. Arrows point from the text labels to the corresponding fields in the configuration windows.

Local controller-scoped tag containing data to be written.

Remote PLC-5 data table address into which data will be written.

Path (out serial port) to station (254).

Figure 64 - Read MSG from a Logix to a PLC-5 or SLC 500 controller example.

The diagram shows a 'Read Message from PLC-5' block with 'Message Control message4'. Below it are two 'Message Configuration - message4' windows. The left window shows 'Message Type: PLC5 Typed Read', 'Source Element: S:23', and 'Destination Element: read'. The right window shows 'Path: SerialR2, 3, 199' and 'Communication Method: CIP'. Arrows point from the text labels to the corresponding fields in the configuration windows.

Remote PLC-5 data table where data will be read from.

Local controller-scoped tag into which data will be copied.

Path (out serial port) to station (199).

Figure 65 - Write MSG instruction from a Logix to an SLC or MicroLogix™ controller example.

The diagram shows a 'Write Message to SLC/MicroLogix' instruction (MSG) with 'Message Control message5'. It is connected to a ladder logic network. Below are two configuration windows for 'message5':

- Message Configuration - message5 (Left):**
 - Message Type: SLC Typed Write
 - Source Element: write
 - Number of Elements: 1
 - Destination Element: N7:0
- Message Configuration - message5 (Right):**
 - Path: SerialR2_3_222
 - Communication Method: CP
 - Channel: K
 - Destination Link: 0
 - Source Link: 0
 - Destination Node: 0 (Octal)

Annotations with arrows point to the 'write' source element, 'N7:0' destination element, and the 'SerialR2_3_222' path field.

Figure 66 - Read MSG instruction from a Logix to an SLC or MicroLogix Controller

The diagram shows a 'Read Message from SLC/MicroLogix' instruction (MSG) with 'Message Control message6'. It is connected to a ladder logic network. Below are two configuration windows for 'message6':

- Message Configuration - message6 (Left):**
 - Message Type: SLC Typed Read
 - Source Element: N7:0
 - Number of Elements: 1
 - Destination Element: read
- Message Configuration - message6 (Right):**
 - Path: SerialR2_3_222
 - Communication Method: CP
 - Channel: K
 - Destination Link: 0
 - Source Link: 0
 - Destination Node: 0 (Octal)

Annotations with arrows point to the 'N7:0' source element, 'read' destination element, and the 'SerialR2_3_222' path field.

Notes:

ASCII Conversion Tables

ASCII Conversions

[Table 41](#) defines the conversions between decimal, octal, hex, and binary values and the ASCII character or control associated with that value.

Table 41 - ASCII Conversions

Decimal	Octal	Hex	Binary	ASCII Character or Control	Decimal	Octal	Hex	Binary	ASCII Character or Control
0	0	00	0000000	Control Shift P, NUL	25	31	19	0011001	Control Y, EM
1	1	01	0000001	Control A, SOH	26	32	1A	0011010	Control Z, SUB
2	2	02	0000010	Control B, STX	27	33	1B	0011011	Control Shift K, ESC
3	3	03	0000011	Control C, ETX	28	34	1C	0011100	Control Shift L, FS
4	4	04	0000100	Control D, EOT	29	35	1D	0014101	Control Shift M, GS
5	5	05	0000101	Control E, ENQ	30	36	1E	0011110	Control Shift N, RS
6	6	06	0000110	Control F, ACK	31	37	1F	0011111	Control Shift O, US
7	7	07	0000111	Control G, Rings bell	32	40	20	0100000	Space, SP
8	10	08	0001000	Control H, Backspace on some terminals	33	41	21	0100001	!
9	11	09	0001001	Control I, Horizontal tab on some terminals	34	42	22	0100010	"
10	12	0A	0001010	Control J, Line feed	35	43	23	0100011	#
11	13	0B	0001011	Control K, VT	36	44	24	0100100	\$
12	14	0C	0001000	Control L, Form feed on some terminals	37	45	25	0100101	%
13	15	0D	0001101	Control M, Carriage return	38	46	26	0100110	&
14	16	0E	0001110	Control N, SO	39	47	27	0100111	'
15	17	0F	0001111	Control O, SI	40	50	28	0101000	(
16	20	10	0010000	Control P, DLE	41	51	29	0101001)
17	21	11	0010001	Control Q, DC1	42	52	2A	0101010	*
18	22	12	0010010	Control R, DC2	43	53	2B	0101011	+
19	23	13	0010011	Control S, DC3	44	54	2C	0101100	,
20	24	14	0010100	Control T, DC4	45	55	2D	0101101	-
21	25	15	0010101	Control U, NAK	46	56	2E	0101110	.
22	26	16	0010110	Control V, SYN	47	57	2F	0101111	/
23	27	17	0010111	Control W, EB	48	60	30	0110000	0
24	30	18	0011000	Control X, CAN	49	61	31	0110001	1
50	62	32	0110010	2	85	125	55	1010101	U
51	63	33	0110011	3	86	126	56	1010110	V
52	64	34	0110100	4	87	127	57	1010111	W
53	65	35	0110101	5	88	130	58	1011000	X
54	66	36	0110110	6	89	131	59	1011001	Y
55	67	37	0110111	7	90	132	5A	1011010	Z
56	70	38	0111000	8	91	133	5B	1011011	[
57	71	39	0111001	9	92	134	5C	1011100	\
58	72	3A	0111010	:	93	135	5D	1011101]
59	73	3B	0111011	;	94	136	5E	1011110	^
60	74	3C	0111100	<	95	137	5F	1011111	_

Table 41 - ASCII Conversions

Decimal	Octal	Hex	Binary	ASCII Character or Control	Decimal	Octal	Hex	Binary	ASCII Character or Control
61	75	3D	0111101	=	96	140	60	1100000	\
62	76	3E	0111110	>	97	141	61	1100001	a
63	77	3F	0111111	?	98	142	62	1100010	b
64	100	40	1000000	@	99	143	63	1100011	c
65	101	41	1000001	A	100	144	64	1100100	d
66	102	42	1000010	B	101	145	65	1100101	e
67	103	43	1000011	C	102	146	66	1100110	f
68	104	44	1000100	D	103	147	67	1100111	g
69	105	45	1000101	E	104	150	68	1101000	h
70	106	46	1000110	F	105	151	69	1101001	i
71	107	47	1000111	G	106	152	6A	1101010	j
72	110	48	1001000	H	107	153	6B	1101011	k
73	111	49	1001001	I	108	154	6C	1101100	l
74	112	4A	1001010	J	109	155	6D	1101101	m
75	113	4B	1001011	K	110	156	6E	1101110	n
76	114	4C	1001100	L	111	157	6F	1101111	o
77	115	4D	1001101	M	112	160	70	1110000	p
78	116	4E	1001110	N	113	161	71	1110001	q
79	117	4F	1001111	O	114	162	72	1110010	r
80	120	50	1010000	P	115	163	73	1110011	s
81	121	51	1010001	Q	116	164	74	1110100	t
82	122	52	1010010	R	117	165	75	1110101	u
83	123	53	1010011	S	118	166	76	1110110	v
84	124	54	1010100	T	119	167	77	1110111	w
120	170	78	1111000	X	124	174	7C	1111100	
121	171	79	1111001	y	125	175	7D	1111101	}
122	172	7A	1111010	z	126	176	7E	1111110	~
123	173	7B	1111011	{	127	177	7F	1111111	DEL

Numerics

5069-FPD field potential distributor 19

A

access module tags 95

ASCII conversion tables 131

C

compatibility

Logix 5000 controllers with Compact 5000 I/O digital modules 10

configuration parameters 28

configure the module 20

connections 20
 multiple connections 20
 create a new module
 discover local I/O modules 64
 discover remote I/O modules 71
 new local I/O modules 76
 new remote I/O module 76

connections 20

multiple connections 20

construct the system 15

controller organizer

monitor tag 89
 view module tag 89

D

data exchange

received from serial port 30
 sent to serial port 29

disable keying 26

E

edit the module configuration 76

connection category 77
 general category 71
 module definition 71
 module info category 88

electronic keying 22

compatible module 26
 disable keying 26
 exact match 26

F

fault reporting 25

function codes 113

force multiple coils 119
 force single coil 117
 preset single register 118
 read coil status 113
 read holding registers 115
 read input registers 116
 read input status 114

G

generic ASCII

input tags 97
 output tags 99

L

local I/O modules 12

Logix 5000 controllers

compatibility with Compact 5000 I/O digital modules 10

Logix Designer application

tag editor 95
 view module tag 89

M

modbus master

data exchange
 read command 35, 37, 58, 61
 write command 35
 input tags 100
 output tag 102

modbus master sample code 125

modbus slave

data exchange 37
 read command 36
 write command 36
 input tags 104, 106, 108, 109, 111
 output tags 105, 106, 108, 110

modbus slave sample code 125

module definition

dialog box 71

module location

local I/O modules 12

module status indicator

description, digital I/O module 91 - 92

module tag

viewing 89

module tags

tag editor 95

N

new local I/O modules 76

O

ownership 20

P

Power Compact 5000 I/O Serial Module

system-side power 18

programming example 121

protected mode 22

protected operations 22

exit and enter 22

R

remote I/O module 13

S

serial module

 diagram 11

 parts 11

serial module status indicators 91

software compatibility 10

software configurable 23

status reporting 25

V

view the module tags 89

Additional Resources

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

Resource	Description
Compact 5000 I/O EtherNet/IP Adapters Installation Instructions, publication 5069-IN003	Describes how to install and wire the Compact 5000™ I/O EtherNet/IP adapters.
Compact 5000 I/O Serial Module Install Instructions, publication 5069-IN022	Describes how to install the Compact 5000 I/O Serial Module.
Compact 5000 I/O Modules Specifications Technical Data, publication 5069-TD001	Provides specifications for Compact 5000 I/O EtherNet/IP adapters and Compact 5000 I/O modules.
CompactLogix 5380 and Compact GuardLogix 5380 Controllers User Manual, publication 5069-UM001	Describes how to use CompactLogix™ 5380 and Compact GuardLogix® 5380 controllers.
Compact 5000 I/O Digital Modules User Manual, publication 5069-UM004	Describes how to use Compact 5000 I/O digital modules.
Compact 5000 I/O Modules Analog Modules User Manual, publication 5069-UM005	Describes how to use Compact 5000 I/O analog modules.
5000 Series High-speed Counter Modules in Logix 5000 Control Systems User Manual, publication 5000-UM006	Describes how to use the Compact 5000 I/O.
Compact 5000 EtherNet/IP Adapters User Manual, publication 5069-UM007	Describes how to use the 5069-AENTR and 5069-AEN2TR EtherNet/IP adapters.
ControlLogix DH-485 Communication Module, publication 1756-UM532	Describes how to use the ControlLogix® DH-485 Communication Module.
ControlLogix 5580 and GuardLogix 5580 Controllers User Manual, publication 1756-UM543	Describes how to use ControlLogix 5580 and GuardLogix 5580 controllers.
SCADA System Application Guide, publication AG-UM008	Application guide for SCADA systems.
Electronic Keying in Logix 5000 Control Systems Application Technique, publication LOGIX-AT001	Describes how to use electronic keying in Logix 5000™ control system applications.
DF1 Protocol and Command Set Reference Manual, publication 1770-RM516	Describes how to use electronic keying in Logix 5000 control system applications.
DF1 Protocol and Command Set Reference Manual, publication 1770-RM516	Describes how to use electronic keying in Logix 5000 control system applications.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, rok.auto/certifications .	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at rok.auto/literature.

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	rok.auto/support
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	rok.auto/techdocs
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

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.

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.

Allen-Bradley, Compact 5000, Compact I/O, CompactLogix, ControlLogix, expanding human possibility, GuardLogix, Logix 5000, MicroLogix, PLC-2, PLC-3, PLC-5, Rockwell Automation, SLC, Stratix, Studio 5000, and Studio 5000 Logix Designer are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark 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, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

UNITED KINGDOM: Rockwell Automation Ltd. Pitfield, Kiln Farm Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800, Fax: (44)(1908) 261-917

Publication 5069-UM003D-EN-P - October 2022

Supersedes Publication 5069-UM003C-EN-P - January 2022

Copyright © 2022 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.