



PointMax High-speed Counter/ SSI Encoder Module

5034-ENC, 5034-ENCXT



Allen-Bradley

by ROCKWELL AUTOMATION

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

Encoder Module.....	9
Construct a PointMax I/O System.....	9
Secure Access to the System.....	10
PointMax I/O in a Logix 5000 Control System.....	10
Controller and Software Compatibility.....	11
Module Firmware Updates.....	11
Connections.....	11
Ownership.....	12
Requested Packet Interval.....	12
Connection Optimization.....	12
Listen Only Mode.....	13
Module Inhibiting.....	13
Electronic Keying.....	14
Protected Operations.....	15
Unicast or Multicast Connection.....	16
CIP Sync Time.....	16
Module Data Quality Reporting.....	17
Field Power (SA Power) Loss Detection.....	18
High-speed Counter (HSC) Input.....	18
Ring Counter Type.....	19
Revolution Counter.....	19
Invert Counter Direction.....	19
Counter Modes.....	20
Count/External Direction Mode.....	20
Count/Internal Direction Mode.....	21
Up and Down Pulses Mode.....	21
X1 Quadrature Encoder Mode.....	22
X2 Quadrature Encoder Mode.....	23
X4 Quadrature Encoder Mode.....	24
Configure Counter State When Module is in Program Mode or Inhibited.....	24
Configure Counter State in Communication Fault Mode.....	25
Counter Triggered Function.....	25
Scalar.....	27
Alarms and Alarm Latching.....	27
Input Filtering.....	28
Missing Pulse Detection.....	28

Zero Frequency Detection.....	28
Frequency, Acceleration, and Pulse Width.....	29
Override Inputs.....	30
Synchronous Serial Interface (SSI) Encoder Type.....	30
Communication.....	30
Data Format.....	30
Data Rate.....	31
Word Filter Control.....	31
Direction.....	31
Disable SSI.....	31
SSI Triggered Function.....	31
Example of Using the 5034-ENC Module with a 24-bit SSI Sensor.....	32
Windows Comparator.....	33
Hysteresis Detection and Configuration.....	34
Digital Input Channel.....	36
Software Configurable Input Filters or Delays.....	36
Digital Output Channel.....	36
Controlled by Window.....	36
Pulse Width Modulation.....	37
Triggered Pulse Output.....	40
Override Outputs.....	41
Configure Output State When Module is in Program Mode or Inhibited.....	41
Configure Output State in Communication Fault Mode.....	41
Short Circuit Detection.....	42
Fast Inductive Load Turn-off (Fast Demagnetization).....	43
Events.....	44
Encoder Module Applications.....	47
Add a New Module to a Studio 5000 Logix Designer Application Project.....	47
Discover Modules.....	47
Add a New Module.....	49
I/O Tag Name Conventions.....	50
Access I/O Tags.....	50
Troubleshoot Your Module.....	51
5034-ENC and 5034-ENCXT Details.....	55
Specifications.....	55
Module Configuration.....	62
Overview View.....	62

Device Definition.....	64
Device Information View.....	74
Connection View.....	75
Counter View.....	76
Counter – Input View.....	78
Counter – Alarms View.....	79
Counter – Fault/Program Actions View.....	81
SSI View.....	82
SSI – Input for Store Function View.....	84
Windows View.....	84
Windowxx View.....	85
Input View.....	86
Outputs View.....	87
Outputxx – Controlled by Window View.....	89
Outputxx – PWM View.....	91
Outputxx – Triggered Pulse View.....	93
Events View.....	95
Eventxx View.....	96
Time Synchronization View.....	99
Use the Status Indicators for Troubleshooting.....	100
SA Power Indicator.....	100
Module Status Indicator.....	101
Channel Status Indicator.....	102
Module Tag Definitions.....	103
Configuration Tag Definitions.....	103
Input Tag Definitions.....	117
Output Tag Definitions.....	128
Event Input Tag Definitions.....	133
Event Output Tag Definitions.....	138
Diagnostic Assembly.....	141
Create User-defined Diagnostic Assembly Types.....	141
Diagnostic Assembly.....	142
Diagnostic Channel.....	145
Definitions for Diagnostic Assembly Members.....	149
Create Message Type User Tags.....	152

Preface

About This Publication

This manual describes how to use the PointMax high-speed counter/serial synchronous interface (SSI) encoder module in Logix 5000® control systems.

Make sure that you are familiar with the following:

- Use of a controller in a Logix 5000 control system
- Use of an EtherNet/IP™ network
- Studio 5000 Logix Designer® environment

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. Translated versions are not always available for each revision.

Table 1. Summary of Changes

Topic
Updated the following topics for firmware revision 2.011:
<ul style="list-style-type: none">• Frequency, Acceleration, and Pulse Width on page 29• Windows Comparator on page 33• Hysteresis Detection and Configuration on page 34• Digital Output Channel on page 36• Controlled by Window on page 36• Override Outputs on page 41• Short Circuit Detection on page 42• Module Configuration on page 62• Module Tag Definitions on page 103
Added the following topics for firmware revision 2.011:
<ul style="list-style-type: none">• Pulse Width Modulation on page 37• Triggered Pulse Output on page 40• Events on page 44

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation. You can view or download publications at rok.auto/literature.

Table 2. Additional Resources

Resources	Description
PointMax I/O System Specifications Technical Data, publication 5034-TD001	Provides PointMax I/O system specifications.

Table 2. Additional Resources (continued)

Resources	Description
PointMax I/O System Installation Instructions, publication 5034-IN001	Provides instructions on installing a complete PointMax I/O system.
PointMax EtherNet/IP Adapter User Manual, publication 5034-UM001	Provides information on how to configure and operate PointMax EtherNet/IP adapters.
PointMax Digital I/O Modules User Manual, publication 5034-UM002	Provides information on how to configure and operate PointMax digital I/O modules.
PointMax Analog I/O Modules User Manual, publication 5034-UM003	Provides information on how to configure and operate PointMax analog I/O modules.
PointMax 4-Channel IO-Link Master Module User Manual, publication 5034-UM004	Provides information on how to configure and operate the PointMax IO-Link master module.
PointMax Serial Module User Manual, publication 5034-UM005	Provides information on how to configure and operate the PointMax serial module.
PointMax High-speed Counter/SSI Encoder Module User Manual, publication 5034-UM006	Provides information on how to configure and operate the PointMax high-speed counter/SSI encoder module.
PointMax Analog HART I/O Modules User Manual, publication 5034-UM007	Provides information on how to configure and operate the PointMax analog HART I/O modules.
EtherNet/IP Network Devices User Manual, publication ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
Ethernet Reference Manual, publication ENET-RM002	Describes basic Ethernet concepts, infrastructure components, and infrastructure features.
System Security Design Guidelines Reference Manual, publication SECURE-RM001	Provides guidance on how to conduct security assessments, implement Rockwell Automation® products in a secure system, harden the control system, manage user access, and dispose of equipment.
Safety Guidelines for the Application, Installation, and Maintenance of Solid-state Control, publication SGI-1.1	Designed to harmonize with NEMA Standards Publication No. ICS 1.1-1987 and provides general guidelines for the application, installation, and maintenance of solid-state control in the form of individual devices or packaged assemblies incorporating solid-state components.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Selection and Configuration tools website, rok.auto/systemtools	Helps configure complete, valid catalog numbers and build complete quotes based on detailed product information.
Product Certifications website, rok.auto/certifications	Provides declarations of conformity, certificates, and other certification details.

Download Firmware, AOP, EDS, and Other Files

You can 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](#).

Terminology

This table defines the terms that are used in this publication.

Table 3. Terminology

Acronym	Full Term	Definition
BP	Backplane Power	Power that is generated from module power by the adapter and expansion power module, and supplied to the I/O system through the backplane.
CIP™	Common Industrial Protocol	An industrial communication protocol that is used by Logix 5000® based automation systems on EtherNet/IP™, ControlNet®, and DeviceNet® communication networks.
CIP Sync™	Common Industrial Protocol Synchronization	CIP Sync provides the increased control coordination needed for control applications where absolute time synchronization is vital to achieve real-time synchronization between distributed intelligent devices and systems.
HSC	High-speed Counter	A device that is used to count input signals at a rapid rate. Typically associated with incremental encoders.
MB	Mounting Base	A device that provides data and power connections from the backplane to the installed module.
MP	Module Power	Power that is supplied to the adapter and expansion power module.
MSB	Most Significant Bit	The bit that has the largest value in a multi-bit binary number.
ODVA	Open DeviceNet Vendor Association	A nonprofit association of vendors that are established for the promotion of CIP networks.
PL	Performance Level	ISO 13849-1 safety rating
RIUP	Removal and Insertion Under Power	A feature that enables the device to be connected and disconnected from the system without having to remove power from the system.
RPI	Requested Packet Interval	Time interval (usually in milliseconds) that users are requesting their data be exchanged at
RTB	Removable Terminal Block	A component that is used for wiring field devices to.
SA	Sensor Actuator	A term that is used to describe field-side devices.
SELV	Safety Extra Low Voltage	An electrical system where the voltage level is considered safe under normal or fault conditions, as defined in the EN and IEC standards.
SSI	Serial Synchronous Interface	A serial communication standard for transmitting data between a leader device and a follower device. Typically associated with absolute encoders.
SSV	Sensor Source Voltage	Voltage that is supplied to a sensor.
XT	Harsh Environment	These modules have additional conformal coating and design considerations that add a greater degree of protection when exposed to harsh, corrosive environments.
VCL	Voltage Clamp Low	The low threshold for limiting the voltage level.

Encoder Module

This chapter describes the features of the high-speed counter (HSC) and Synchronous Serial Interface (SSI) encoder module and how to construct a PointMax I/O system.

The module supports removal and insertion under power (RIUP). You must remove the RTB first and then the module.

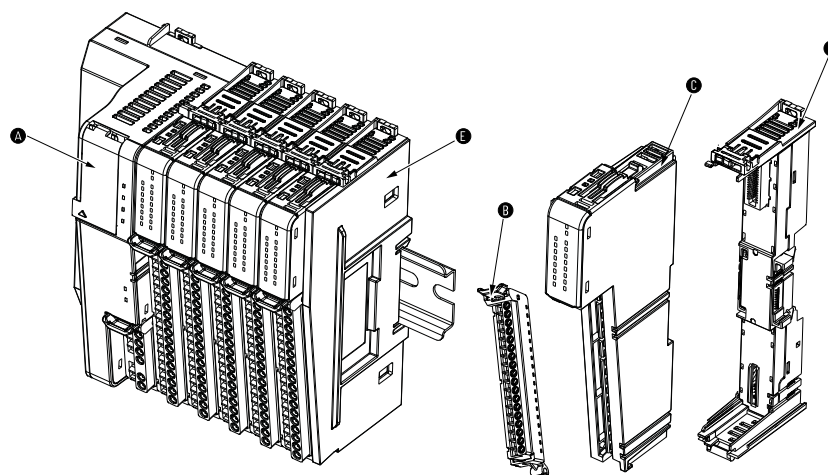
You can configure the module for HSC mode to use with high-speed counters and incremental encoders, or SSI mode to use with SSI encoders.

For technical and environmental specifications details, see PointMax I/O System Specifications Technical Data, publication [5034-TD001](#).

Construct a PointMax I/O System

PointMax I/O system contains the components that are pictured below.

Figure 1. PointMax I/O System



Item	Component Name	Description
A	Adapter	The adapter transfers data between the I/O module and the controller.
B	Removable Terminal Block (RTB)	The RTB contains terminals to terminate wiring for field devices. Also it has interfaces to establish the connection between the RTB and the I/O module.
C	I/O module	The I/O module contains the SA or field power interface and circuitry that is needed to perform specific functions that are related to your application.
D	Mounting Base (MB)	The MB contains mechanical and electrical interfaces to establish the connection between I/O module and the backplane.
E	End cap	It is a dust protection cap for the last module in a rack.

For PointMax I/O system power requirements, see PointMax I/O System Installation Instructions, publication [5034-IN001](#).

Secure Access to the System

To secure access to the device by authorized users only, consider these options:

- Use passwords to help protect the source and execution of the control program.
- Remove the key from the controller.
- Implement physical barriers, such as locked cabinets.

To secure access to the system, consider these options:

- Follow industry best practices to harden your personal computers and servers, including anti-virus/anti-malware and application allow-list solutions.

The recommendations are published at the Rockwell Automation technical support center in Knowledgebase Technote [Rockwell Automation Customer Hardening Guidelines](#).

- Develop and deploy backup and disaster recovery policies and procedures. Test backups on a regular schedule.
- Minimize network exposure for all control system devices and systems, and confirm that they are not accessible from the Internet.
- Locate control system networks and devices behind firewalls and isolate them from the business network.
- Subscribe to the Knowledgebase Technote [Security Advisory Index](#), so you have access to information about security matters that affect Rockwell Automation products.

Follow recommended network practices for products with network interfaces, such as communication ports or web servers. These practices help minimize risk or exposure by unauthorized activity or users. For more information, see:

- Converged Plantwide Ethernet (CPwE) Design and Implementation Guide, publication [ENET-TD001](#)
- EtherNet/IP Network Devices User Manual, publication [ENET-UM006](#)
- Configure System Security Features User Manual, publication [SECURE-UM001](#)

PointMax I/O in a Logix 5000 Control System

IMPORTANT: Throughout this publication, the term Logix 5000 controller refers to the controllers with which you can use PointMax I/O modules in a given capacity.

You can use PointMax I/O modules with Logix 5000 controllers as remote I/O modules only.

You cannot use PointMax I/O modules with all Logix 5000 controllers. For example, you can use PointMax I/O modules with CompactLogix® 5380 and ControlLogix® 5580 controllers but not with CompactLogix 5370 and ControlLogix 5570 controllers.

Logix 5000 controllers use I/O modules to control devices in a control system. The controllers access the modules over an EtherNet/IP network.

See [Connections on page 11](#) for the communication model between controller and I/O modules.

Controller and Software Compatibility

Controller and programming software compatibility requirements apply when you use PointMax I/O modules. A module type and how it is used affect which requirements apply.

You must use Studio 5000 Logix Designer application version 36 or later. For more information, see the Product Compatibility and Download Center at rok.auto/pcdc.

If you use GuardLogix® controllers and need to use up to chassis slot 32, you must use Studio 5000 Logix Designer application version 37 or later.

Module Firmware Updates

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

Firmware information for I/O modules is available at the Rockwell Automation Product Compatibility and Download Center (PCDC). The PCDC is available at: rok.auto/pcdc. At the PCDC, you can use the module catalog number to check for firmware updates. If the catalog number is not available, then no updates exist.

IMPORTANT: Only download firmware and access product release notes from the Rockwell Automation PCDC. Do not download firmware from non-Rockwell Automation sites.

Verify that the firmware revision of the I/O modules that you use is correct before commissioning the system. For more information on how to update your module firmware, see the ControlFLASH Plus Quick Start Guide, publication [CFP-QS001](#).

When the firmware update is in progress, you cannot perform the following operations:

- Another firmware update request
- Module connection request
- Module reset request



WARNING: When you update the PointMax EtherNet/IP adapter, the I/O modules in its chassis may reset and be unable to maintain their Program Mode or Inhibit states. Verify that all equipment controlled by the modules in this chassis are in a stopped state and that all safety critical functions are not affected.

Connections

A connection is a real-time data transfer link between the owner-controller and the module that occupies the slot that the configuration references.

During the module configuration, the connection type determines what data is exchanged between the controller and the module. When you uninhibit a module in online mode, or download the project with modules that are uninhibited to the controller, the controller attempts to establish a connection to each module.

The owner-controller monitors its connection with a module. Any break in the connection, for example, the loss of power to the PointMax I/O system, causes a fault. The Studio 5000 Logix Designer application can monitor the fault status tags to indicate when a fault occurs on a module.

Producer/Consumer Communication

PointMax I/O modules use the Producer/Consumer communication model to produce data without a controller polling them first. The modules produce the data and controllers consume it. That is, the owner-controller and controllers with a Listen Only connection to the module can consume it.

When an input module produces data, the controllers can consume the data simultaneously. Simultaneous data consumption eliminates the need for one controller to send the data to other controllers.

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 PointMax 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.
- Sends the I/O module configuration data to define module behavior and begin operation in the control system.

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

When the owner-controller establishes I/O connection to the module, it sends the configuration data of the module. Two possible results are:

- If the configuration is appropriate to the module, a connection is established and operation begins.
- If the configuration is not appropriate to the module, the data is rejected and the Connection view in the Module Properties indicates that an error occurred. The configuration can be inappropriate for many reasons. For example, a mismatch in electronic keying that helps prevent operation on incompatible module.

Requested Packet Interval

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

Connection Optimization

PointMax I/O system supports automatic optimization for I/O connections.

The following conditions must be met for connections to be included in the optimization:

- I/O modules on the rack have the same RPI
- Unicast
- Sum of data sizes are within the packet limit

This is similar to how I/O systems such as POINT I/O™ support rack optimization or enhanced rack optimization. Functional safety modules, HART modules, and IO-Link master modules are excluded from the optimization.

Listen Only Mode

Any controller in the system can listen to the data from an I/O module. An owner-controller, as described in [Ownership on page 12](#) exchanges data with I/O modules. Other controllers can use a Listen Only connection with the I/O module. In this case, the 'listening' controller can only listen to the data produced by the module. The listening controller does not own the module configuration or exchange other data with the module.

During the I/O configuration process, you can specify a Listen Only connection.

IMPORTANT:

Remember the following:

- The Listen Only mode applies only to standard I/O modules.
- If a controller uses a Listen Only connection, the connection must use the Multicast option. In this case, all other connections to the module, for example, the connection of the owner-controller, must also use the Multicast option.

The Listen Only connection RPI value must be the same as owner connection RPI value.

- If a controller attempts to use a Listen Only connection to a module but the owner-controller connection uses the Unicast option, the attempt at a Listen Only connection fails.
 - The Listen Only controller receives data from the module as long as a connection between an owner-controller and module is maintained.
 - RPI of the Listen Only connection must be the same as owner connection.
 - If the connection between an owner-controller and the module is broken, the module stops sending data and connections to all 'listening controllers' are also broken.
 - Due to Multicast, adding a Listen Only connection does not increase network bandwidth.
 - Only one controller can own the I/O module. The PointMax I/O modules do not support multiple owners of the same module. Other controllers must use a Listen Only connection to the module.
-

Module Inhibiting

Module inhibiting lets you indefinitely suspend a connection between a controller and a module without removing the module from the configuration. This process temporarily stops the connection between the controller and the module.

IMPORTANT: Whenever you inhibit a module with outputs, all outputs change to the state that is configured for Program Mode.

You can use module inhibiting in the following scenarios:

- You want to perform maintenance on the module.
- You want to update a module, for example, update the module firmware revision.
- 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 system.

Use the following procedure to inhibit and uninhibit the module:

1. Inhibit the module – Go to Connection view in the Module Properties dialog, select Inhibit Module and then select Apply or OK.
2. Perform the necessary update.
3. Uninhibit the module – Go to Connection view in the Module Properties dialog, clear Inhibit Module and then select Apply or OK.

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, an 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

Table 4. Electronic Keying Options

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 or compatible 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. • You can use an XT version of the module in place of a non-XT module or non-XT version of the module in place of XT module.

Table 4. Electronic Keying Options (continued)

Keying Option	Description
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> <div><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.</p><p>We strongly suggest that you do not use Disable Keying.</p><p>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></div>
Exact Match	<p>Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.</p> <p>When XT module is configured in Studio 5000 Logix Designer application project, only XT Module residing on XT MB is considered as Exact Match. When non-XT module is configured, only non-XT module (with any MB) is considered as exact match.</p>

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

IMPORTANT: Changing Electronic Keying parameters online interrupts connections to the device and any devices that are connected through the device. Connections from other controllers can also be broken.

If an I/O connection to a device is interrupted, the result can be a loss of data.

For more detailed information on Electronic Keying, see Electronic Keying in Logix 5000 Control Systems Application Technique, publication [LOGIX-AT001](#).

Protected Operations

To maintain the secure operation of your I/O module, operations that can disrupt module operation are restricted based on the module operating mode.

In Protection Mode, the device deactivates services that could disrupt the operation of the device, but the device continues to function. For example, configuration operations or firmware updates are disabled to not impact the operation of the device.

The module enters Implicit Protection Mode as soon as an I/O connection is established to the module. The module exits Implicit Protection Mode as soon as all I/O connections to the module are stopped.

When the device is in Implicit Protection Mode, the mode prevents the following actions:

- Firmware update request
- Module reset request
- Connection or data format change

The following operations can be applied in Studio 5000 Logix Designer application when the modules are in Protection Mode, but as a consequence, the I/O connections are closed and reopened:

- Change Electronic Keying
- Change RPI
- Change Unicast/Multicast

The following operations are allowed when the modules are in Protection Mode:

- Listen-only connection requests from other controllers
- I/O channel configuration change. To apply configuration change without interrupting the I/O connections, there are two methods:
 - Change configurations and apply the changes in the Module Properties dialog.
 - Change configuration tag values and send a Reconfigure Module MSG to the module to apply the changes.

Unicast or Multicast Connection

During module configuration, you must configure the Connection over EtherNet/IP parameter. The configuration choice dictates how input data is broadcast over the network.

The PointMax I/O modules use one of the following methods to broadcast data:

- Multicast - Multicast connections deliver information from one sender to multiple receivers simultaneously. Copies of one transmission are passed to a selected subset of possible destinations.
- Unicast - Unicast connections are point-to-point transmissions between a source node and destination node on the network. A transmission is sent to one destination controller.

IMPORTANT: Unicast is the default setting. Use Unicast because it reduces network bandwidth usage. However, if you are using a ControlLogix 5580 controller in a high availability system, you must use Multicast.

CIP Sync Time

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

These modules are CIP Sync slave-only devices. There must be another module on the network that functions as a master clock.

You can use I/O modules to capture timestamps. The advantage is that CIP Sync is system-wide, so timestamp values are consistent across all modules in the system.

IMPORTANT: *I.CIPSyncValid* = 1 is not sufficient for an application to confirm that the module is ready for application requiring a system time synchronization to start operation. See [Grandmaster Clock Verification on page 17](#) for extra verification needed.

Grandmaster Clock Verification

The application must verify that the module is synchronized with the same Grandmaster Clock as the owner-controller.

- If Status connection is selected in the PointMax EtherNet/IP adapter Device Definition, verify that all of the following are true:
 - I/O module's *I.CipSyncValid* is 1
 - Adapter's *S.CipSyncValid* is 1
 - Adapter's *S.GrandmasterClockID* is the same as the controller's *GrandMasterClockID*
- If Status connection is not selected in the PointMax EtherNet/IP adapter Device Definition, verify that all of the followings are true:
 - I/O module's *I.CipSyncValid* is 1
 - I/O module's *GrandmasterClockID* in *GrandMasterClockInfo* attribute of *TimeSync* object is the same as the controller's *GrandmasterClockID*
- Class: 0x43
- Instance: 0x01
- Attribute: 0x08
- Response: *GrandMasterClockInfo* (see structure below)

GrandMasterClockInfo	STRUCT of
ClockIdentity	USINT[8]
ClockClass	UINT
TimeAccuracy	UINT
OffsetScaledLogVariance	UINT
CurrentUtcOffset	UINT
TimePropertyFlags	WORD
TimeSource	UINT
Priority1	UINT
Priority2	UINT

To get controller's *GrandMasterClockID*, access the *TIMESYNCHRONE* object through the GSV instruction. See [Access the TimeSynchronize object](#).

Module Data Quality Reporting

The module indicates the quality of data that is returned to the owner-controller. Levels of data quality are reported via input tags.

The following inputs tags indicate the level of data quality.

IMPORTANT: Once the condition that causes the Fault or Uncertain tag to change to 1 is removed, the tag automatically resets to 0. The Studio 5000 Logix Designer application controls the tags. You cannot change the status of the tags. Keep in mind that in some system configurations, the tag is not reset immediately after the condition is removed. The tag typically resets after a small delay.

- **Fault** – When the fault bit is set, this tag indicates that the reported data is inaccurate and cannot be trusted for use in your application. Do not use the reported data for control.

If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.

Example causes of inaccurate data for digital outputs include the following:

- Channel fault
 - Field Power Loss condition
 - Short circuit condition
-



You should troubleshoot the module for the typical causes first.

- **Uncertain** – This tag indicates that the reported data can be inaccurate but the degree of inaccuracy is unknown. We recommend that you do not use the reported data for control.

If the tag is set to 1, you know that the data can be inaccurate. You must troubleshoot the module to discover what degree of inaccuracy exists.

An example cause of uncertain data is the counter input signal frequency is outside its designed operating range.



You should monitor the tags in your program to make sure that the application is operating as expected with accurate data.

Field Power (SA Power) Loss Detection

The Field Power Loss Detection feature monitors for the loss of field-side power on the SA power bus.

When the condition is detected, the Fault bit is set at the channel level in the input tag. In the diagnostic assembly, the FieldPowerOff bit is set at the module and channel level structure.

High-speed Counter (HSC) Input

The HSC input interfaces with sensors and incremental encoders.

One signed 32-bit counter and other extended counter functionality are provided. Three high-speed differential inputs (A, B, and Z) are available on the module to use with the counter. You can wire the inputs to use differential line drive output devices, for example, differential or single-ended encoders. You can use sensors, such as inductive or optical sensors, or simple pulse generators.

The counter can operate in the following modes:

- Count/Direction (internal or external)
- Up/Down Pulses
- Quadrature (X1, X2, or X4 counting modes)

The operating mode of a count channel determines how the A and B inputs cause a counter channel to increment or decrement.

Frequency information is calculated and continuously updated with the count data.

You configure the HSC mode in the Device Definition dialog in the Studio 5000 Logix Designer application. For more information, see [Device Definition on page 64](#).

Ring Counter Type

The module counters are ring counters. The current count value changes between configurable Rollover and Rollunder count values.

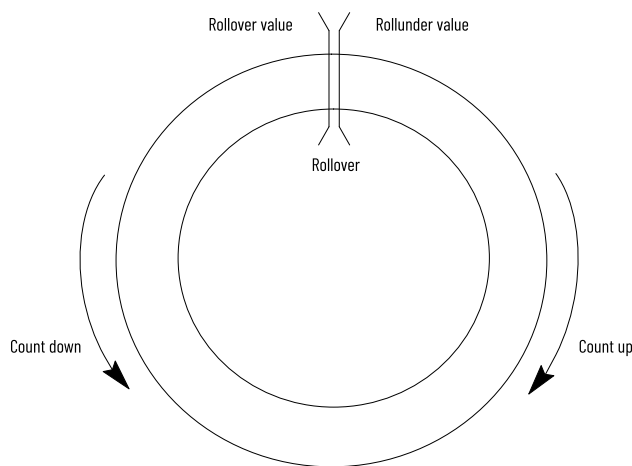
The values are represented in the module tags as follows:

- Current count = *I.Counterxx.Count*
- Rollover count = *O.Counterxx.RolloverValue*
- Rollunder count = *O.Counterxx.RollunderValue*

If a counter is counting up, when the count value reaches the Rollover value - 1, it rolls over to the Rollunder value upon receiving the next count. The rollover tag is set to 1.

If a counter is counting down, when the count reaches the Rollunder value, it rolls under to the Rollover value - 1 value upon receiving the next count. The rollunder tag is set to 1.

Figure 2. Ring Counter Diagram



Revolution Counter

Each counter supports a Revolution Counter that counts the Rollover and Rollunder transitions. It increments when a Rollover occurs and decrements when a Rollunder occurs.

Invert Counter Direction

The Invert Counter Direction feature changes the direction of the counter. By default, Invert Counter Direction is disabled.

You disable or enable Invert Counter Direction in Module Properties → Counterxx.

Counter Modes

This section describes the types of counter modes that are available when the module is configured in HSC mode.

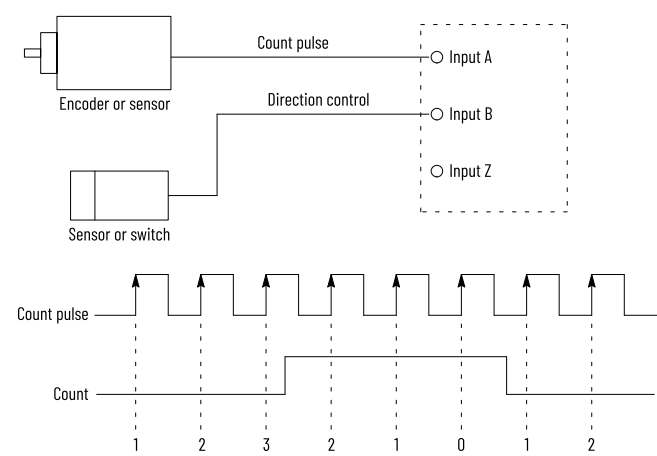
Count/External Direction Mode

In this mode, input B controls the direction of the counter.

When Invert Counter Direction is disabled, the following occurs:

- If input B is set to 0, the counter increments on the rising edges of Input A.
- If the input B is set to 1, the counter decrements on the rising edges of Input A.

Figure 3. Count/External Direction Mode (Invert Counter Direction Disabled)



- Direction Control:
- High = Decrement
 - Low = Increment

When Invert Counter Direction is enabled, the operation of input B reverses. In this case, the following occurs:

- If input B is set to 0, the counter decrements on the rising edge of input A.
- If input B is set to 1, the counter increments on the rising edge of input A.

Table 5. Count/External Direction Counting

Invert Counter Direction Setting	Input A (Count)	Input B (Direction)	Change in Count Value
Disabled	Rising edge	0 or open	1
	Rising edge	1	-1
Enabled	Rising edge	0 or open	-1
	Rising edge	1	1

Count/Internal Direction Mode

In this mode, the *O.Counterxx.Direction* tag determines the direction of the counter.

Table 6. Count/Internal Direction Counting

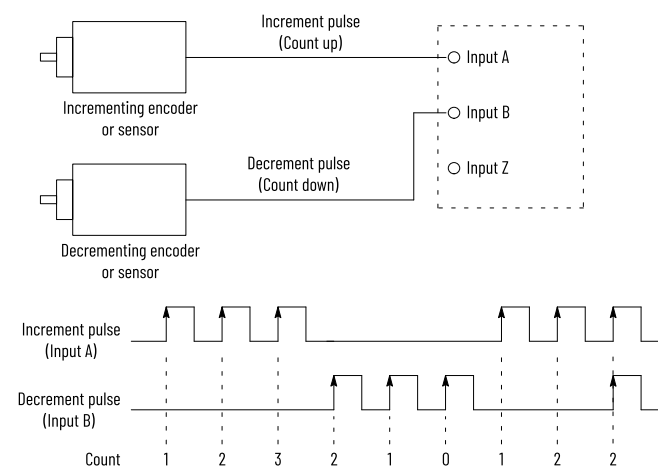
Invert Counter Direction Setting	0.Counterxx.Direct ion Tag	Input A (Count)	Input B (Not Used)	Change in Count Value
Disabled	0	Rising edge	-	1
	1	Rising edge		-1
Enabled	0	Rising edge		-1
	1	Rising edge		1

Up and Down Pulses Mode

In this mode, when Invert Counter Direction is disabled, the following occurs:

- The counter increments on the rising edge of pulses that are applied to input A.
- The counter decrements on the rising edge of pulses that are applied to input B.

Figure 4. Up and Down Pulse Mode (Invert Counter Direction Disabled)



When Invert Counter Direction is enabled, the following occurs:

- The counter decrements on the rising edge of pulses that are applied to input A.
- The counter increments on the rising edge of pulses that are applied to input B.

Table 7. Up and Down Counting

Invert Counter Direction Setting	Input A	Input B	Change in Count Value
Disabled	Rising edge	0, 1	1

Table 7. Up and Down Counting (continued)

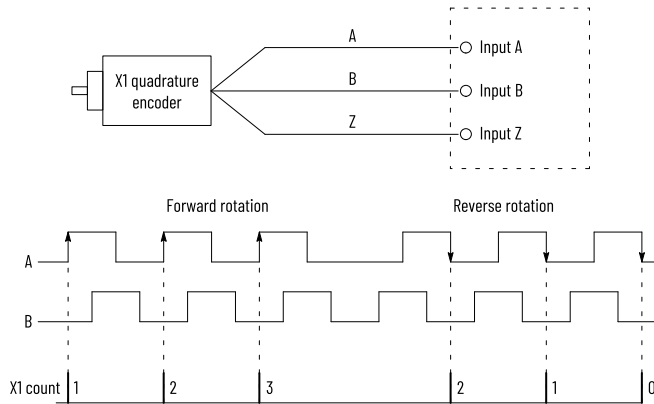
Invert Counter Direction Setting	Input A	Input B	Change in Count Value
Enabled	0, 1	Rising edge	-1
	Rising edge	0, 1	-1
	0, 1	Rising edge	1

X1 Quadrature Encoder Mode

When an X1 quadrature encoder is attached to inputs A and B, the phase relation of inputs A and B determine the count direction.

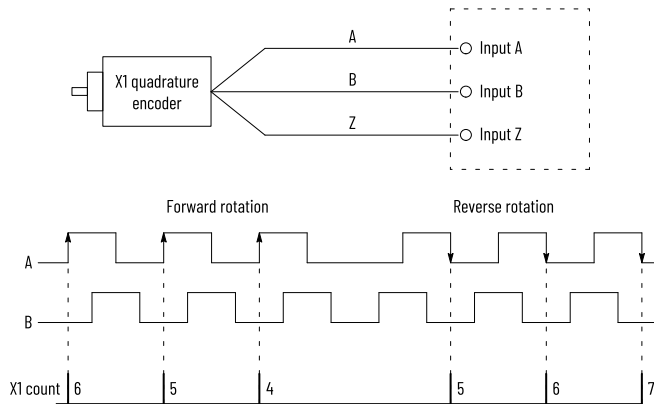
When Invert Counter Direction is disabled, if Input A leads Input B, the counter increments. If Input B leads Input A, the counter decrements.

Figure 5. X1 Quadrature Encoder Mode (Invert Counter Direction Disabled)



When Invert Counter Direction is enabled, if Input A leads Input B, the counter decrements. If Input B leads Input A, the counter increments.

Figure 6. X1 Quadrature Encoder Mode (Invert Counter Direction Enabled)





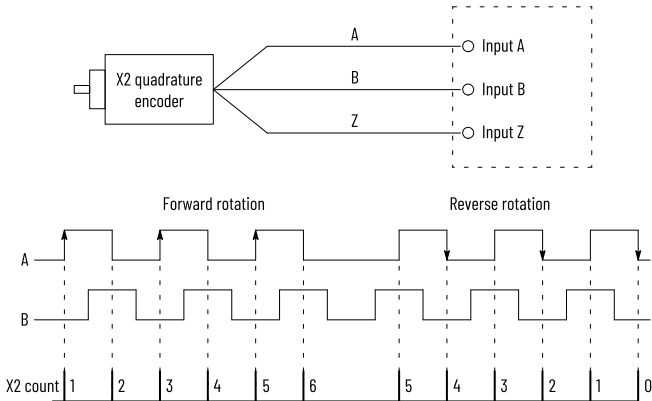
When both Input A and Input B transition simultaneously, instead of in the defined 90° phase separation, the quadrature signal is invalid.
In this case, the *I.Counterxx.QuadratureErrorCount* tag increments.

X2 Quadrature Encoder Mode

The X2 Quadrature Encoder mode operates much like the X1 Quadrature Encoder except that the resolution is doubled. When an X2 quadrature encoder is attached to Inputs A and B, the phase relation of Inputs A and B determine the count direction.

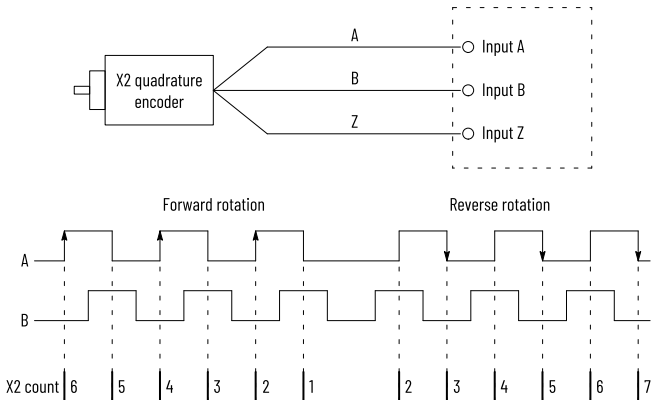
When Invert Counter Direction is disabled, if Input A leads Input B, the counter increments. If Input B leads Input A, the counter decrements.

Figure 7. X2 Quadrature Encoder Mode (Invert Counter Direction Disabled)



When Invert Counter Direction is enabled, if Input A leads Input B, the counter decrements. If Input B leads Input A, the counter increments.

Figure 8. X2 Quadrature Encoder Mode (Invert Counter Direction Enabled)



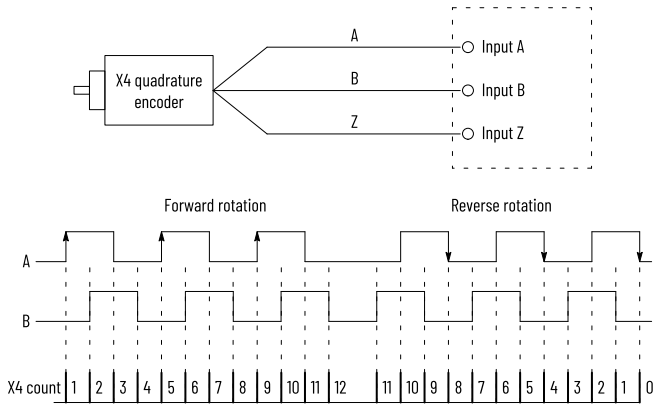
When both Input A and Input B transition simultaneously, instead of in the defined 90° phase separation, the quadrature signal is invalid.
In this case, the *I.Counterxx.QuadratureErrorCount* tag increments.

X4 Quadrature Encoder Mode

The X4 Quadrature Encoder mode operates much like the X1 Quadrature Encoder except that the resolution is quadrupled. When an X4 quadrature encoder is attached to inputs A and B, the phase relation of Inputs A and B determine the count direction.

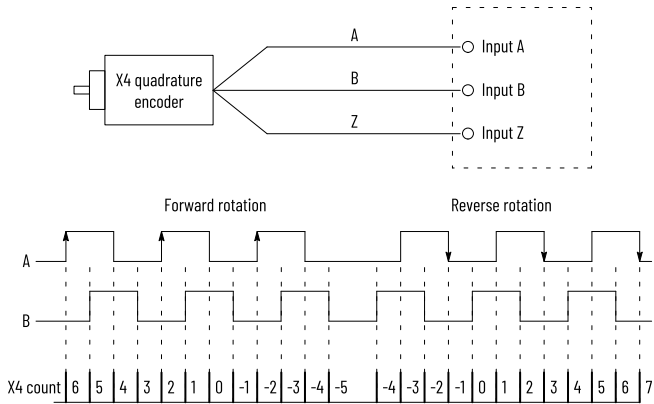
When Invert Counter Direction is disabled, if Input A leads Input B, the counter increments. If Input B leads Input A, the counter decrements.

Figure 9. X4 Quadrature Encoder Mode (Invert Counter Direction Disabled)



When Invert Counter Direction is enabled, if Input A leads Input B, the counter decrements. If Input B leads Input A, the counter increments.

Figure 10. X4 Quadrature Encoder Mode (Invert Counter Direction Enabled)



When both Input A and Input B transition simultaneously, instead of in the defined 90° phase separation, the quadrature signal is invalid.

In this case, the `I.Counterxx.QuadratureErrorCount` tag increments.

Configure Counter State When Module is in Program Mode or Inhibited

You can configure the counter to specific states when the module is in Program Mode or when the module is inhibited.

When the controller switches from Run mode to Program Mode or Inhibited, the counter can behave in the following ways, depending on how you configure the Counter State in Program Mode parameter:

- Continue counting – Default
- Hold its last state
- Transition to a user-defined value

You can choose to reset the stored values when the module is in Program Mode or Inhibited.

When the module is inhibited, the counter behavior follows the Counter State in Program Mode configuration.

Configure Counter State in Communication Fault Mode

You can configure the counter to specific states when a connection fault occurs, that is, the connection between the controller and the module breaks.

When the connection between a controller and module breaks, the counter can behave in the following ways, depending on how you configure the Counter State in Communication Fault Mode parameter:

- Continue counting – Default
- Hold its last state
- Transition to a user-defined value

You can choose to reset the stored values when the module when is in Communication Fault Mode.

When Communication Fails in Program Mode

When communication fails in Program Mode, the counter can behave in the following ways, depending on how you configure the When Communication Fails in Program Mode parameter:

- Leave counter in Program Mode state – The counter behaves as defined by the Counter State in Program Mode selection.
- Change counter to Communication Fault state – The counter behaves as defined by the Counter State in Communication Fault Mode selection.

Counter Triggered Function

The count value is indicated in the *I.Counterxx.Count* tag. You can use an input trigger (Input Z or Input00) to trigger functions for the counter.

You use counter functions to manipulate the count value in the module tags that are associated with each function. You can configure the input trigger for the counter functions in Module Properties → Device Definition → Counter → Input Z Function or Input00 Function.

IMPORTANT: SA power is required to use Input00.

Store

The Store function lets a counter store its current count, direction, instantaneous frequency, acceleration, instantaneous pulse width, and revolution count in the corresponding *I.Counterxx.<StoredTag>* tag.

StoredTag

StoredCount	StoredPulseWidth	ScaledStoredCount
StoredRevolutionCount	StoredAccel	ScaledStoredFrequency
StoredFrequency	StoredDirection	ScaledStoredAccel

You can trigger the Store function using the input trigger or the *O.Counterxx.Store* tag. You can capture the current count value even when the counter is counting at the full 1 MHz speed.

Hold

The Hold function lets a counter hold its current values, regardless of the incoming A or B input data. You can set and trigger the Hold function using the input trigger or the *O.Counterxx.Hold* tag.

The counter enters Hold state upon receiving a rising edge of the input trigger or *O.Counterxx.Hold*. The counter holds its last count value until it exits Hold state upon receiving a falling edge of the input trigger or *O.Counterxx.Hold*.

Load

The Load function lets a counter load new count and revolution count values. You can set and trigger the Load function using the input trigger or the module output tags.

When a Load function is triggered, the value of *O.Counterxx.LoadCountValue* is copied to *I.Counterxx.Count* and the value of *O.Counterxx.LoadRevolutionValue* is copied to *I.Counterxx.RevolutionCount*.

Reset

The Reset function lets the counter reset. When a counter is reset, the *I.Counterxx.Count* tag and *I.Counterxx.RevolutionCount* tag are set to 0.

When the *C.Counterxx.ResetTimeDerivedValues* tag is set to 1, the following values are also reset:

- Frequency, Avg Frequency, Scaled Frequency, and Scaled Freq Avg
- Pulse Width and Avg Pulse Width
- Acceleration, Avg Accel, Scaled Accel, and Scaled Avg Accel

You can trigger the Reset function using the input trigger or the *O.Counterxx.Reset* tag.

Invert Input Trigger

An input trigger can be configured as active high or active low by inverting the input. You can invert the input in Module Properties → Device Definition → Counter → Invert Input Z or Invert Input00.

Inverting the input trigger does not affect the usage of output tag triggers. For example, Input Z is used as the trigger for Hold and it is inverted. In this case, the counter enters the Hold state upon receiving a falling edge of Input Z or rising edge of *O.Counterxx.Hold*. The counter exits the Hold state upon receiving a rising edge of Input Z or falling edge of *O.Counterxx.Hold*.

Example of Using Counter Functions

After power on or power recovery, before using the Count and Revolution tags, these tags must be set to a known value using the Reset or Load function. One method is to perform a reset operation when a certain position is detected.

For example:

1. Input Z is connected to a sensor that detects if the encoder is in the zero position.
2. Input Z is configured to trigger the Reset function.
3. Activate the device that is connected to the encoder to perform one whole loop. Once the encoder is in the zero position, Input Z is activated and it triggers the Reset function. The Count and Revolution is reset to 0.

If the zero position is not position 0, the Load function can be used to set the Count and Revolution to a known value.

Scalar

The Scalar function lets you scale the count, frequency, and acceleration values to user units by multiplying their values in pulses by the Scaling value.

For example, when one encoder revolution is 8192 pulses and you want to read the value in degrees, set the following:

- *O.Scaling* = 0.0439453125 (= $360/8192$)
- *O.RollunderValue* = 0
- *O.RolloverValue* = 8192

Alarms and Alarm Latching

Alarm latching lets you latch a counter alarm once the alarm is triggered. The alarm remains set even if the condition that triggered the alarm disappears, until the alarm is unlatched.

Alarm latching is available on a per counter basis. You can latch the following alarms:

- Missing Pulse Alarm
- Zero Frequency Alarm
- Zero Frequency Average Alarm
- Frequency Alarm
- Frequency Average Alarm
- Pulse Width Alarm
- Pulse Width Average Alarm
- Acceleration Alarm
- Acceleration Average Alarm
- Deceleration Alarm
- Deceleration Average Alarm

For the Missing Pulse and Zero Frequency alarms, you must enable latching using the module tags. Latching is always enabled for other alarms.

Enable Latching

You can enable alarm latching in Module Properties. Go to the Counterxx → Alarms view.

Unlatch Alarms

IMPORTANT: Before you unlatch an alarm, make sure that the condition that triggered the alarm is no longer exists.

Once an alarm is latched, you must use a rising edge of its unlatch tag in the output tags to unlatch the alarm. The alarm type determines which module tag you must change.

Input Filtering

Input Filtering lets you digitally filter out high frequency noise that is inadvertently coupled to the sensor wires. When used, the filter settings directly relate to filter accuracy.

Filters are available for the state transitions from Off to On and On to Off for all module inputs, that is, ABZ inputs and Input00.

For input signal with slow rising and falling time, we recommend using input filtering. You can adjust it to achieve optimum filtering performance and to avoid false measurements.

For more information on available filter timings, see [Specifications on page 55](#).

Missing Pulse Detection

Missing Pulse Detection alerts you when a new pulse is not available to count within a configurable time period. When a missing pulse is detected, the Missing Pulse Alarm is triggered.

To use Missing Pulse Detection, you must complete the following steps:

1. Configure the Missing Pulse Alarm Limit to establish the maximum length of time between pulses before which the Missing Pulse Alarm is triggered.

Go to Module Properties → Counterxx → Alarm to set the Missing Pulse Alarm Limit.

2. Set the *O.Counterxx.MissingPulseAlarmEn* tag to 1.

If this tag is not set to 1, the alarm is not enabled and you are not alerted when a missing pulse occurs.

When the Missing Pulse Alarm is triggered, the *I.Counterxx.MissingPulseAlarm* tag goes to 1.

Missing Pulse Detection is commonly used in high frequency applications. A missing pulse typically indicates the loss of a counting device, for example, an encoder. The loss of a counting device often has a significant impact on the application conditions and immediate system action is used to account for the impact.

Zero Frequency Detection

Zero Frequency Detection alerts you when the frequency that is reported on an input is less than the Zero Frequency limit. When a Zero Frequency condition exists, the Zero Frequency Alarm is triggered.

Configure the *O.Counterxx.ZeroFrequencyAlarmLimit* tag to establish the minimum frequency that is required to be measured at the input before which the Zero Frequency Alarm is triggered.

Zero Frequency Detection is commonly used in lower frequency applications that monitor the continued presence of pulses at a minimum frequency and above. When the Zero Frequency alarm is triggered, the assumption is that the counting device is not lost but instead that the input frequency is low.

Frequency, Acceleration, and Pulse Width

In addition to returning a count value for the counter channel, extended counters measure and return the following for the counter channel:

- Instantaneous and average frequency
- Instantaneous and average pulse width
- Instantaneous and average acceleration

Count/Direction and Quadrature mode counters measure frequency, acceleration, and pulse width based on pulses on Input A.

Up/Down Pulses mode counters measure frequency, acceleration, and pulse width based on pulses on Input A. If there are no pulses on Input A, the counter measures frequency, acceleration, and pulse width based on pulses on Input B.

If the entire period of a pulse is detected on Input B while Input A is off, the measurements are measured on input B on the rising edge of the next pulse detected on Input B.

If the measurements are performed on input B, the measurements start being performed on Input A as soon as the leading edge of a pulse on Input A is detected.

Measured frequency and pulse width are always positive numbers. The direction of the counter, however, is determined by the direction bit.

All frequency, acceleration, and pulse width measurements are measured based on the input signal. The measurements continue even if the following occur when the measurement is taken:

- Rollover
- Rollunder
- Reset
- Load

Average Frequency, Acceleration, and Pulse Width

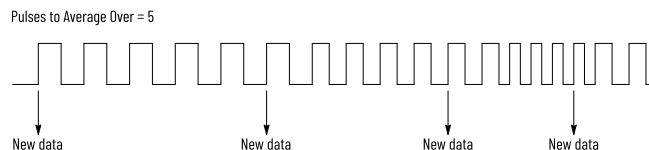
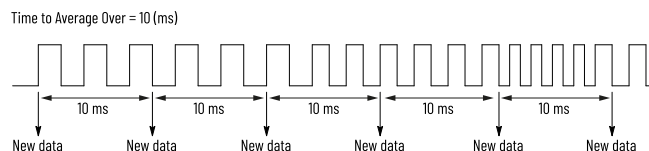
The calculation of average frequency, acceleration, and pulse width is configured in Module Properties → Counter → Average Over.

Table 8. Calculation of Average Over Pulses and Average Over Time

Average Over	Description
Pulses	<p>A new average value is calculated when the specified number of pulses is detected.</p> <p>The number of pulses is configured in Module Properties → Counter → Pulses to Average Over.</p>
Time	<p>A new average value is calculated when the sampling time has elapsed and at least a pulse is detected.</p> <p>The sampling time is configured in Module Properties → Counter → Time to Average Over.</p>



For average acceleration, use Average Over Time to help achieve a less noisy measurement value.

Figure 11. Average Over Pulses**Figure 12. Average Over Time**

Override Inputs

You can override the state of module inputs. This practice is commonly used when troubleshooting an application or testing a control system outside normal operating conditions.

You must change the following tags to override the input state:

- `O.Counterxx.OverrideDataEn` – Set the tag to 1 to enable the `O.Counterxx.OverrideDataValue` to override the input state.
- `O.Counterxx.OverrideDataValue` – Set the tag to the desired value.

Synchronous Serial Interface (SSI) Encoder Type

One signed 32-bit position data is available on the module.

You configure the SSI mode in the Device Definition dialog in the Studio 5000 Logix Designer application. For more information, see [Device Definition on page 64](#).

Communication

To communicate with an SSI sensor the code type and word length (that is, resolution) configuration must match the SSI sensor specification. See the sensor manufacturer's data sheet for the required configuration. These parameters are configured in Module Properties → SSI.

The code type supported are gray code and natural binary.

The word length supported is range of 2...31 bits. This should include position bits (resolution) and trailing bits if standardization is used. For more information, see [Data Format on page 30](#).

Data Format

The SSI sensor data is produced in the `I.SSIxx.Data` tag.

If the type of SSI sensor is Binary, the data format is always Binary. If the type of SSI sensor is Gray code, the displayed data format is always converted from Gray to Binary.

Some SSI sensor has non-position trailing bits on the data returned. This can be truncated by configuring the number of Trailing Bits in Module Properties → SSI – Conversion. This is important to achieve the correct direction and window comparison result.

If the trailing bits has important information such as status information, the *I.SSIxx.RawData* tag can be used to readback this information. The RawData contains the data received from the SSI sensor without any conversion or truncation. The RawData size matches the word length.

Data Rate

The baud rate is the communication rate (bits per seconds) between the SSI sensor and the module's SSI channel. The baud rate selected at the SSI channel equals the frequency of the SSI channel's clock output. You can configure the Baud rate in Module Properties → SSI.

The maximum baud rate that can be used for your application is limited by the transmission line length between the SSI module and sensor. See the SSI sensor data sheet, as well as the PointMax I/O Specifications Technical Data, publication [5034-TD001](#), for communication rate versus transmission line length information.

The time between successive SSI words can be configured in Module Properties → SSI → SSI Word Delay Time.

See the SSI sensor data sheet for the recommended delay time. If the sensor manufacturer specifies a value called monoflop time, use that value for the SSI Word Delay time setting.

The Baud Rate multiplied by Word Length and SSI Word Delay determines the data rate of SSI channel before filter is applied.

Word Filter Control

The Word Filter Control determines the number of successive equal data words that are received from the SSI sensor before updating the produced SSI sensor data (*I.SSIxx.Data* and *I.SSIxx.RawData*).

- Off = Immediately produces data without the previous value comparison.
- Low = 2 successive equal data
- Med = 3 successive equal data
- High = 4 successive equal data
- Max = 5 successive equal data

Direction

The *I.SSIxx.Direction* tag indicates the sensor position direction as per the last SSI sensor movement. Every sampled SSI sensor data (after filtering) is compared with the previous sample to determine the direction. 0 means that the position is moved up. 1 means that the position is moved down.

Disable SSI

You can disable the SSI channel in the Module Properties → SSI view. When disabled, the SSI clock output to the SSI sensor is turned OFF. This may be useful during machine setup or maintenance.

SSI Triggered Function

The SSI sensor data is indicated in the *SSI00.Data* tag. You can use Input00 as a trigger for the Store function for the SSI.

You can configure the input trigger for the SSI functions in Module Properties → Device Definition → SSI → Enable Store via Input00.

IMPORTANT: SA power is required to use Input00.

Store

The Store function lets an encoder store its current data, direction, and raw data in the corresponding *I.SSI00.<StoredTag>* tag.

StoredTag		
StoredData	StoredDirection	StoredRawData

You can trigger the Store function using Input00 or the *O.SSI00.Store* tag. You can capture the current data even when the encoder is moving.

Invert Input Trigger

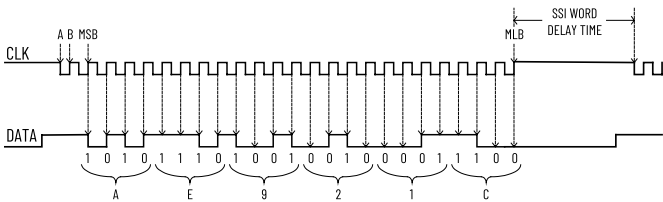
An input trigger can be configured as active high or active low by inverting the input. You can invert the input in Module Properties → Device Definition → SSI → Invert Input00.

Inverting the input trigger does not affect the usage of output tag triggers. For example, Input00 is used as the trigger for Store and it is inverted. In this case, the encoder enters the Store state upon receiving a falling edge of Input00 or rising edge of *O.SSI00.Store*.

Example of Using the 5034-ENC Module with a 24-bit SSI Sensor

The following figure illustrates the SSI clock (CLK) signal that is sent to the SSI sensor by the 5034-ENC module and the SSI data (DATA) coming back to the module from the sensor.

Figure 13. 5034-ENC with 24-bit SSI Sensor Example



The CLK and DATA signals are representative of what you could actually see on an oscilloscope when the module is attached to a SSI sensor in a live system. The SSI module must be configured for a SSI Word length of 24, as well as the proper Data Rate and SSI Word Delay time that are stated in the SSI sensor specifications.

SSI Word Delay time is approximately equal to a value called monoflop time, as stated by some SSI sensor manufacturers. When configuring the 5034-ENC module, enter the monoflop time value for SSI Word Delay time.

Point A of the CLK signal is a falling clock edge. When the SSI sensor sees the first falling clock edge after a delay period, it knows to latch its current position into a parallel-to-serial converter that is located in the sensor.

At point B of the CLK signal is the rising clock edge. The sensor begins to send its serial data to the 5034-ENC module. The module actually starts reading the sensor position data on the next rising clock edge, denoted by MSB in the figure above. MSB is the most significant bit of the data word.

The 5034-ENC module supports only MSB aligned data. This means that the SSI sensor sends the MSB of its data word first, and the least significant bit (LSB) is sent last. Observe that it takes 25 rising clock edges to read a 24-bit data word from a 24-bit SSI sensor. A 25-bit sensor needs 26 rising edges and a 13-bit sensor needs 14 edges. Configure the module for the SSI Word Length as stated in the sensor specifications and the module adds the extra rising edge automatically.

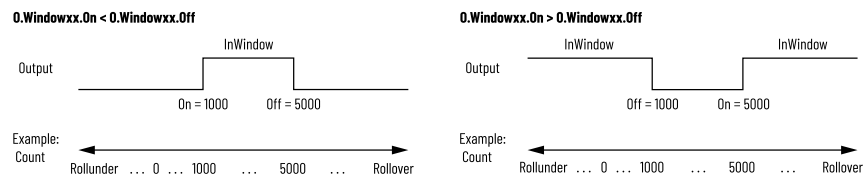
The actual bit values of the current SSI sensor position are shown beneath the Data signal in the figure. A bit value is always the binary (0 or 1) value just before a given rising clock edge. The next binary value is sent by the SSI sensor just after the rising clock edge and is read by the module on the next rising clock edge.

Windows Comparator

Four windows are available on the module. Each window can be used to detect when the input data of the selected mode is within or outside of a configured window. You tie windows to the encoder (HSC or SSI) and to as many as two module outputs.

The following figure shows an example of how a window controls output behavior, using HSC Count as the input data source for the window. For more information, see [Controlled by Window on page 36](#). The windows can operate in different modes.

Figure 14. Windows Behavior Example



You can configure windows to operate in the following modes for HSC:

- Count or scaled count
- Frequency, instantaneous, scaled, or average
- Acceleration, instantaneous, scaled, or average
- Pulse Width, instantaneous or average

You can configure windows to operate in the following modes for SSI:

- Data

The following figures explain how to determine whether the window source is In Window or Not in Window.

Figure 15. $0.Windowxxx.On < 0.Windowxxx.Off$

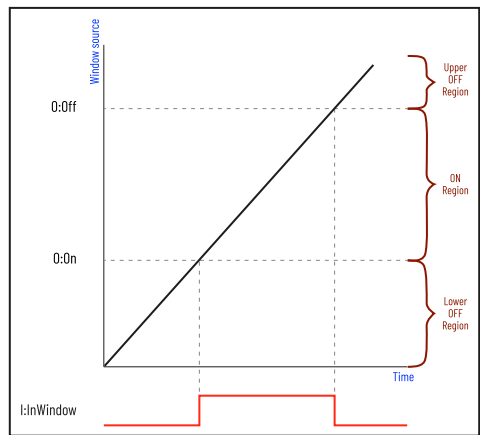
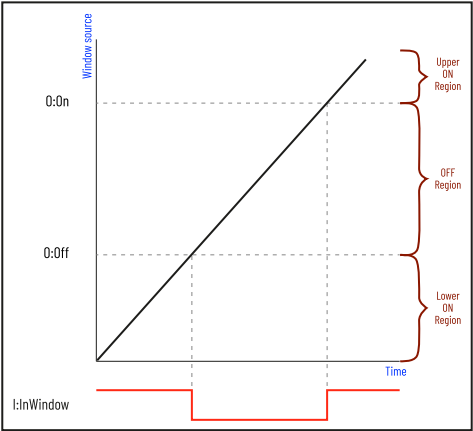


Figure 16. $O.Windowxx.On > O.Windowxx.Off$



Hysteresis Detection and Configuration

Physical vibration can cause an encoder to generate pulses that you do not wish to consider as valid motion. Hysteresis On and Off values are used to eliminate some pulses in either direction as vibration-generated.

You can use the following tags to set the hysteresis values:

- `O.Windowxx.HysteresisOn`
- `O.Windowxx.HysteresisOff`

IMPORTANT:

Remember the following:

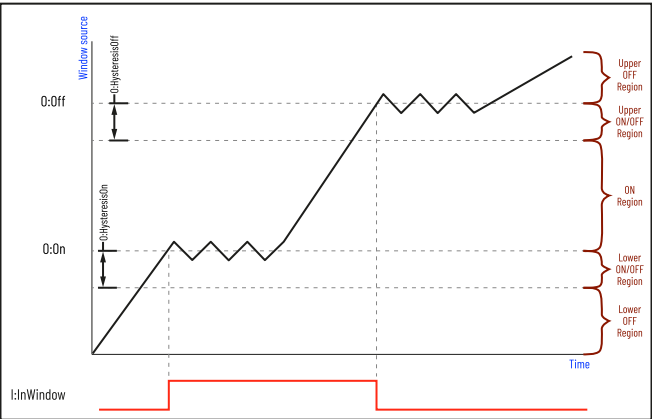
- If a tag value is < 0 , hysteresis is disabled.
- Hysteresis does not apply to Windows configured for Pulse Width mode.

Hysteresis is not used to alter actual count values.

Hysteresis Behavior when $O.Windowxx.On < O.Windowxx.Off$

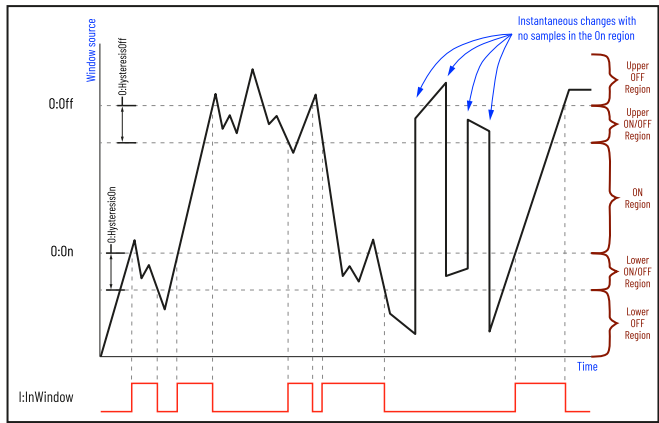
The following figure shows a window with hysteresis as the source of the window varies near the window switching points when $O.Windowxx.On < O.Windowxx.Off$.

Figure 17. Hysteresis Example 1 – Source of Window Varies Near Window Switching Points ($O.Windowxx.On < O.Windowxx.Off$)



The following figure also show a window with hysteresis when $O.Windowxx.On < O.Windowxx.Off$. However, the source of the window, that is, count, frequency, or acceleration varies beyond the hysteresis at the window switching points.

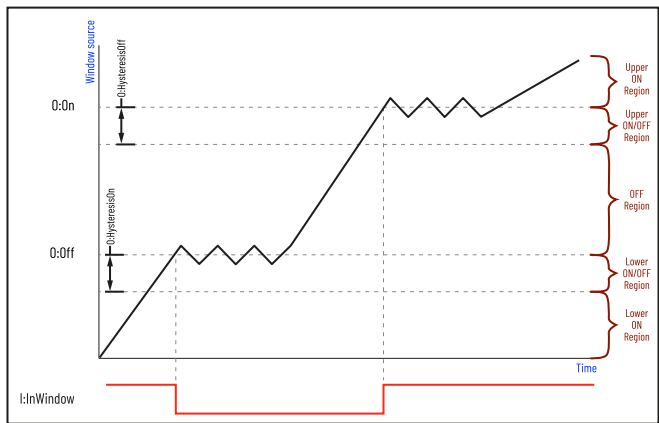
Figure 18. Hysteresis Example 2 – Source of Window Varies Beyond Hysteresis at Window Switching Points ($O.Windowxx.On < O.Windowxx.Off$)



Hysteresis Behavior when $O.Windowxx.On > O.Windowxx.Off$

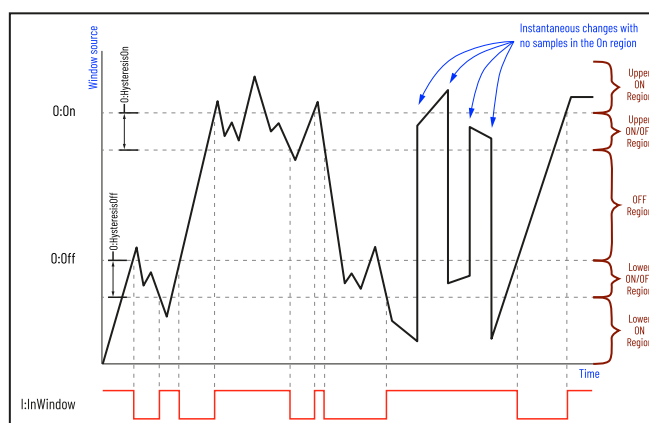
The following example shows a window with hysteresis as the source of the window varies near the window switching points when $O.Windowxx.On > O.Windowxx.Off$.

Figure 19. Hysteresis Example 3 – Source of Window Varies Near Window Switching Points ($O.Windowxx.On > O.Windowxx.Off$)



The following example also show a window with hysteresis when $O.Windowxx.On > O.Windowxx.Off$. However, the source of the window, that is, count, frequency, or acceleration varies beyond the hysteresis at the window switching points.

Figure 20. Hysteresis Example 4 – Source of Window Varies Beyond Hysteresis at Window Switching Points ($0.Windowxx.On > 0.Windowxx.Off$)



Digital Input Channel

One input is available on the module. You can use Input00 as a standard digital input or as a trigger for encoder functions.

Software Configurable Input Filters or Delays

You can adjust On to Off and Off to On filter timing through the Studio 5000 Logix Designer application for the encoder module. These filters improve noise immunity within a signal.

A larger filter value affects the length of delay time for signals from this modules. The filter values are adjustable in the Input view of the Module Properties dialog.

Digital Output Channel

Two outputs are available on the module. You can wire the outputs to use as discrete output devices.

There are different output modes that you can configure for each output:

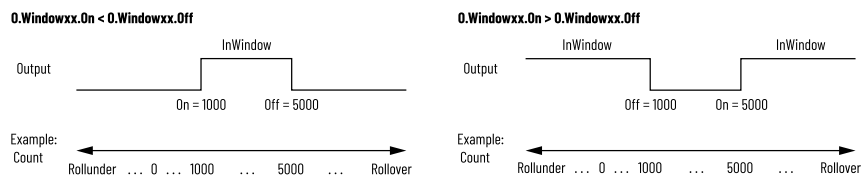
- Use windows comparator to control the outputs. For more information on how to use a module window to turn an output On or Off, see [Controlled by Window on page 36](#).
- Generate pulse width modulation (PWM). For more information on how to configure a PWM signal, see [Pulse Width Modulation on page 37](#).
- Use a local input to trigger a precise timed pulse output. For more information on how to control the pulse output, see [Triggered Pulse Output on page 40](#).

Controlled by Window

You can tie windows to the module outputs. You can tie a window to as many as two outputs.

You can configure a window to turn an output On or Off based on configurable count/frequency/pulse width/acceleration value designation. The state change, that is, from Off to On or On to Off, occurs in less than 10 μ s.

Figure 21. Windows Behavior Example



IMPORTANT: You can override the output using the output tag. See [Override Outputs on page 41](#) for more information.

Pulse Width Modulation

IMPORTANT: This feature is only available with firmware revision 2.011 or later.

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

- Cycle time – The duration of a pulse cycle.
- Duty cycle – The percentage of the cycle time that a pulse is active.

If the cycle time or duty cycle value changes while the output is generating a PWM signal, the changes are not applied until the next cycle of the PWM output. For example, if the duty cycle is 10% and the cycle time is 1.0 second and the cycle time is changed to 0.5 second just after the output turns On, the output stays On for 0.1 second and then turns Off for 0.9 seconds to complete the cycle before the new 0.5 second cycle begins.

For another example, if the cycle time is erroneously set to an hour, a new cycle time does not go into effect until the current cycle of the hour is complete. To trigger the PWM output to restart immediately with a new cycle time or duty cycle, turn Off the output and then it back On.

If the cycle time or pulse width (that is, cycle time * duty cycle or cycle time * [100% - duty cycle]) is outside the valid range for an output, the *I.Outputxx.Error* tag is set and the module responds as described in the following table.

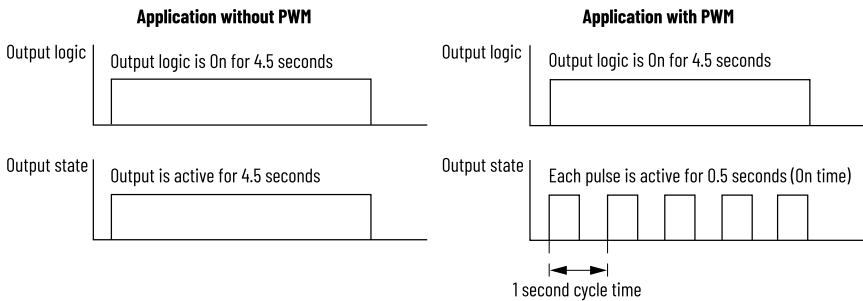
Condition	Result
CycleTime < Minimum cycle time	CycleTime = Minimum cycle time
CycleTime > Maximum cycle time	CycleTime = Maximum cycle time
Duty Cycle > 100%	Output is always in Active state.
Pulse width < Minimum pulse width	Output is always in Idle state.

IMPORTANT: Before PWM functions, you must enable PWM in the module configuration.

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

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

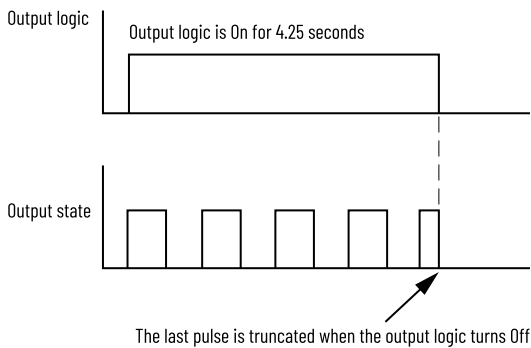
Figure 22. PWM Timing



By default, PWM is configured to continue the output pulse train until the output logic (*O.Outputxx.Data* tag) turns Off. When the output logic turns Off, the output pulse train immediately stops.

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

Figure 23. PWM with Truncated Pulse



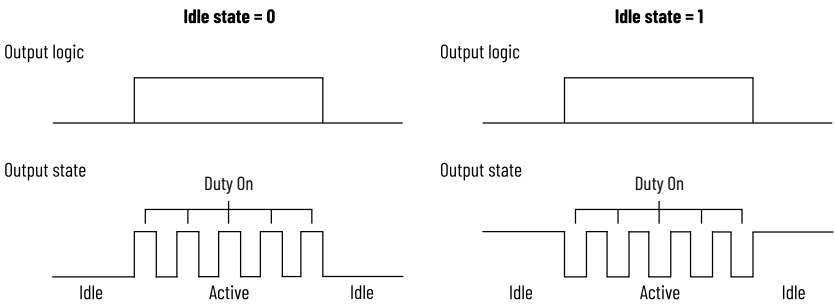
IMPORTANT: When the module goes into Program Mode, Communication Fault mode, or is inhibited, the Program Mode and Communication Fault Mode states that are configured for the module override the PWM output state.

You can modify the default PWM configuration for further control of an output's pulse train.

Idle State

The Idle state indicates the output state when the PWM pulse sequence is not running (PWM is idle).

Figure 24. Output in PWM Idle State



Enforce PTO Duty Cycle

You can limit the duty cycle to be always 50% by enabling the Enforce Duty Cycle at 50.0% for Pulse Train Output (PTO).

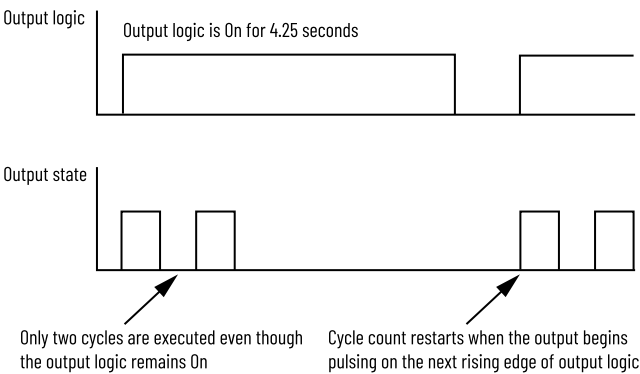
When this configuration is enabled, the *O.Pwmxx.DutyCycle* tag is ignored and generated pulse cycles have 50% duty cycle.

Cycle Count and Execute All Cycles

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

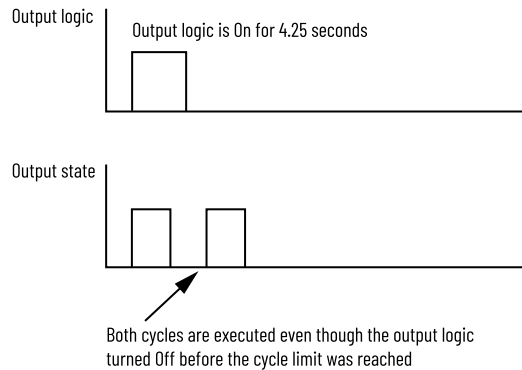
The following figure shows a PWM pulse train that is configured with a cycle count of 2. When the cycle count is reached, the PWM pulse train is stopped and the *I.Pwmxx.Active* tag is cleared.

Figure 25. PWM Cycle Count



If the output logic turns Off before the cycle limit is reached, you can configure the pulse cycles to continue until the cycle limit is reached by enabling the Execute All Cycles option. The following figure shows a cycle limit of 2 with the Execute All Cycles option enabled.

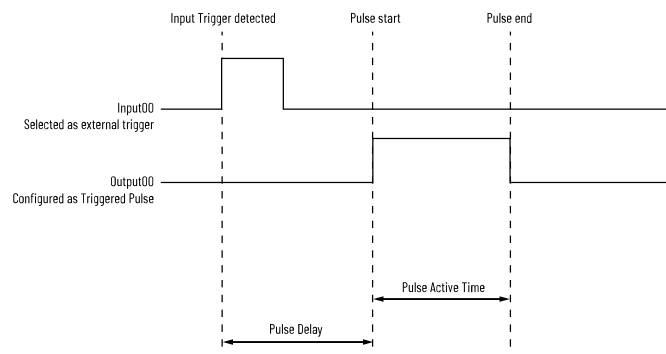
IMPORTANT: When changing cycle count, you should turn off the output first or wait for the Active input tag to turn off. Otherwise, an additional pulse may be generated.

Figure 26. PWM Cycle Count with Execute All Cycles Option

Triggered Pulse Output

IMPORTANT: This feature is only available with firmware revision 2.011 or later.

The Triggered Pulse Output feature allows an output point to generate a precise timed output pulse when triggered by a local input point. When you configure the output, select a local input that acts as a trigger.

Figure 27. Triggered Pulse Output Example

You can control the output pulse timing using the following parameters:

- **Pulse Delay (in seconds):** Defines the delay after the input transition before the output pulse starts.

You can configure this parameter using the output tag *O.Outputxx.PulseDelay*. The valid value range is (1 ms + Input Filter Time)...30 s.

- **Pulse Active Time (in seconds):** Specifies how long the output pulse remains active once triggered after pulse delay.

You can configure this parameter using the output tag *Outputxx.PulseActiveTime*. The valid value range is 10 μ s...30 s.

A rising edge or falling edge of the input signal can initiate the trigger. To initiate the trigger on the falling edge, enable the Invert Trigger option. You can also use the output tag *O.Outputxx.Trigger* to trigger the pulse on its rising edge.

If a triggering transition occurs while the current output pulse is already active or in the delay period (waiting), the new trigger is ignored and *I.Outputxx.TriggerMissed* bit is set to indicate

that a trigger event was missed. You can use the *O.Outputxx.ResetTriggerMissed* bit to reset the *I.Outputxx.TriggerMissed* bit.

IMPORTANT: You can override the output using the output tag. See [Override Outputs on page 41](#) for more information.

Override Outputs

When you configure the output mode as "Controlled by Window" or "Triggered Pulse Output", you can override the state on any module output. This practice is commonly used when troubleshooting an application or testing a control system outside normal operating conditions.

You must change the following tags to override the output state:

- *O.Outputxx.OverrideDataEn* – Set the tag to 1 to enable the *O.Outputxx.OverrideDataValue* to override the output state.
- *O.Outputxx.OverrideDataValue* – Set the tag to the desired value.

Configure Output State When Module is in Program Mode or Inhibited

You can configure individual output points to specific states when the module is in Program Mode or when the module is inhibited.

When the controller switches from Run mode to Program Mode or Inhibited, the output can behave in the following ways, depending on how you configure the Output State in Program Mode parameter:

- Turn off – Default
- Turn on
- Hold its last state
- Local Control – This option is only available when one of the following conditions is true:
 - The encoder type is set to SSI and the SSI channel controls the output.
 - The encoder type is set to HSC, the counter controls this output, and the counter is configured as Countinue Counting in Program Mode.

When the module is inhibited, the output behavior follows the Output State in Program Mode configuration.

Configure Output State in Communication Fault Mode

You can configure individual output points to specific states when a connection fault occurs, that is, the connection between the controller and the module breaks.

Output Behavior Immediately After a Connection Fault

When the connection between a controller and module breaks, the output can behave in the following ways, depending on how the Communication Fault Mode State parameter is configured:

- Off – Output is 0 or Off.
- On – Output is 1 or On.
- Hold – Output maintains the last state.
- Local Control – Output continues to be controlled by the windows.

This option is only available when one of the following conditions is true:

- The encoder type is set to SSI and the SSI channel controls the output via windows.
- The encoder type is set to HSC, the counter controls this output via windows, and the counter is configured as Continue Counting in Program Mode.

The output remains at that state value until the following occurs:

- The connection to the controller is re-established.
- The duration expired, which is based on the value that is defined in the Communication Fault Mode State Duration.

Communication Fault State Duration After Connection Fault

For output behavior after a connection fault, you must define how long the output remains at the specified value before it transitions to a Final Communication Fault Mode State.

You can configure the output to remain at the specific value for the following times:

- Forever
- 1 second
- 2 seconds
- 5 seconds
- 10 seconds

After the Communication Fault Mode State Duration time expires, the output transitions to the user-defined Final Communication Fault Mode State.

Final Communication Fault State

The Final Communication Fault State defines the state to which the output goes after the Communication Fault Mode State Duration time expires.

Once the connection between the controller and module is re-established, the output resumes normal operation.

When Communication Fails in Program Mode

When communication fails in Program Mode, the output can behave in the following ways, depending on how you configure the When Communication Fails in Program Mode parameter:

- Leave outputs in Program Mode state – It causes the output to behave as defined by the Output State in Program Mode selection.
- Change outputs to Communication Fault Mode state – It causes the output to behave as defined by the Output State in Communication Fault Mode selection.

Short Circuit Detection

Short Circuit Detection helps prevent damage to the output that can result when more current is present at the output than it can handle.

A short circuit condition can cause an increase in temperature and trigger a thermal shutoff to prevent damage to the output.

IMPORTANT: Short Circuit Detection is not supported when using PWM/PTO.

When a short circuit condition is detected, the following occurs:

- The output turns off.
- The I/O status indicator for the output flashes red.
- The *I.Outputxx.Fault* tag is set to 1.

When the short-circuit condition is removed, the following occurs:

- The output restarts in its commanded state.
- The I/O status indicator for the output follows the commanded state (off or steady yellow).
- If there are no other faults present, the *I.Outputxx.Fault* tag is reset to 0.

For more information on the maximum current that you can apply to an output, see the PointMax I/O System Specifications Technical Data, publication [5034-TD001](#).

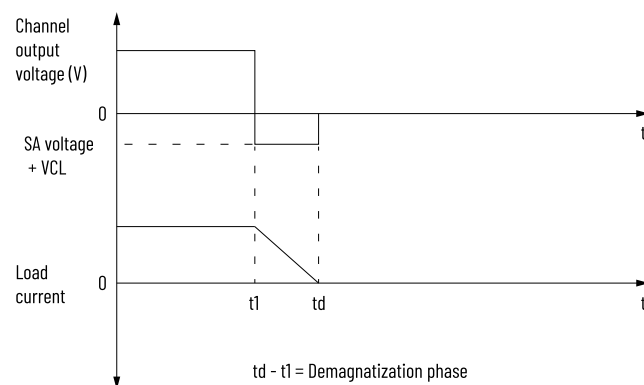
Fast Inductive Load Turn-off (Fast Demagnetization)

Traditionally, inductive loads (such as a solenoid drive) use a free-wheeling diode in parallel with the load to discharge stored energy to reach the Off status (zero current). This process is called Demagnetization.

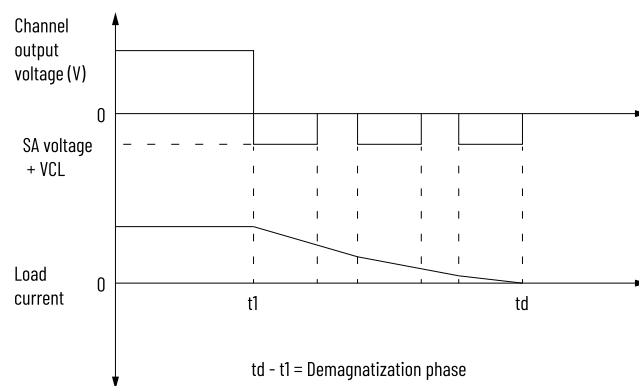
The PointMax high-speed counter/SSI module has the fast inductive load turn-off feature. If the inductive load is within the module's specified inductance and current rating, then no free-wheeling diode is required. Demagnetization is performed faster when compared to using an external free-wheeling diode. For the module inductance and current rating specifications, see the PointMax I/O System Specifications Technical Data, publication [5034-TD001](#).

There are two possible demagnetization scenarios. Typically, inductive load demagnetization happens as illustrated in Demagnetization Example 1. In this scenario, demagnetization happens entirely through the high-side active clamp switch and the output voltage is at a negative voltage of (SA voltage + VCL) during the demagnetization phase.

Figure 28. Demagnetization Example 1



However in some situations, demagnetization happens as illustrated in Demagnetization Example 2. The demagnetization happens through the combination of high-side active clamp switch and low-side switch. The output voltage switches between a negative voltage of (SA voltage + VCL) and approximately 0 volts during the demagnetization phase.

Figure 29. Demagnetization Example 2

Events

IMPORTANT: This feature is only available with firmware revision 2.011 or later.

To use Events, select the “Data with Events” connection type in the Device definition dialog while configuring the module.

Define Event Triggers

When Encoder Type is configured as HSC, the following conditions can trigger events:

- Window00...Window03
- Counter Load
- Counter Store
- Counter Hold
- Counter Reset
- Counter Direction
- Counter Rollunder
- Counter Rollover

When Encoder Type is configured as SSI, the following conditions can trigger events:

- Window00...Window03
- Store
- Direction

You must choose a State Transition for each trigger. The State Transition defines what must occur to trigger the event.

The following State Transition choices are available:

- Not in Window/In Window – Used with the Window event triggers.
- Low/High – Used with the Counter/SSI event triggers.

When the module reports an event to the controller, the module can trigger an event task to execute in the controller. The event task lets you execute a section of logic immediately when an event, or receipt of new data, occurs.

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

IMPORTANT: No event is generated if any of the participating channel is in fault condition.

Independent Point Trigger Definition

This option means that any state change that match can trigger the event.

To use this type of trigger, you must enable the Independent Point Trigger option in the event definition. Set the *EO.Eventxx.IndependentConditionTriggerEn* tag to 1.

Pattern Match Trigger Definition

This option means that a pattern of state changes must match to trigger the event.

To use this type of trigger, you must disable the Independent Point Trigger option in the event definition. Set the *EO.Eventxx.IndependentConditionTriggerEn* tag to 0.

Every state that participates in an event trigger is configured separately. Depending on the event definition, the collective status of all states that matches the pattern triggers the event.

Event Trigger Direction

An event is triggered each time a state change results in one of the following conditions:

- When Rising Edge event trigger is enabled (*EO.Eventxx.EventRisingEn* = 1) – Matching the event trigger definition
- When Falling Edge event trigger is enabled (*EO.Eventxx.EventFallingEn* = 1) – No longer matching the event trigger definition

Do not use the Falling Edge event trigger for the following:

- Counter – Reset, Load, Store, Rollunder, and Rollover
- SSI – Store

Event Latching

To use event latching, you must select the Latch Event checkbox for that input event. Set the *EO.Eventxx.LatchEn* tag to 1.

When event latching is used, the event in the input data must be acknowledged before a new event is sent out.

To acknowledge the current event in the input data, set the *EO.Eventxx.EventNumberAck* tag to the event number (*EI.Eventxx.EventNumber* tag) in the input data.

If the current event is not acknowledged, subsequent events are stored in a queue. The queue can store up to four events. If the queue is full, excess events are dropped.

User-defined Data

You can use as many as three user-defined data values to define an event.

When Encoder Type is configured as HSC, the following data values are available:

- No Additional Data
- Count
- Stored Count
- Scaled Count
- Scaled Stored Count

- Revolution Count
- Stored Revolution Count

When Encoder Type is configured as SSI, the following data values are available:

- No Additional Data
- Data
- Stored Data

You define events when you define a module during module configuration in the Studio 5000 Logix Designer application. For more information, see [Device Definition – Events View on page 73](#).

Additional Event Considerations

When you use the Events feature, also consider the following:

- An Event task only actuates if an event occurs.

IMPORTANT: Make sure that you link the Event task to the Event Input tag, not the Input tag.

Keep in mind that when the Event task executes, the input tag data can have the same data that was sent at the last RPI.

- When you configure the RPI of the input event connection, consider that the minimum interval between two input data packets of event connection is one quarter of the RPI.
- After the event task executes, it does not execute again until the event occurs again.
- When the input event connection is reopened, the *El.Eventxx.<member>* tags are reset to 0.

Encoder Module Applications

This chapter describes how to configure your high-speed counter/SSI encoder module in a Studio 5000 Logix Designer application project. You can use the default module configuration or edit the module configuration.

This chapter does not explain the user-configurable module features that you can edit on different screens in your Logix Designer application project.

Add a New Module to a Studio 5000 Logix Designer Application Project

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

1. Create a Studio 5000 Logix Designer application project.
2. Add a Logix 5000 controller to the project.
3. Add a PointMax EtherNet/IP adapter to the project.

For more information on how to add a PointMax EtherNet/IP adapter to the Studio 5000 Logix Designer application project, see the PointMax EtherNet/IP Adapter User Manual, publication [5034-UM001](#).

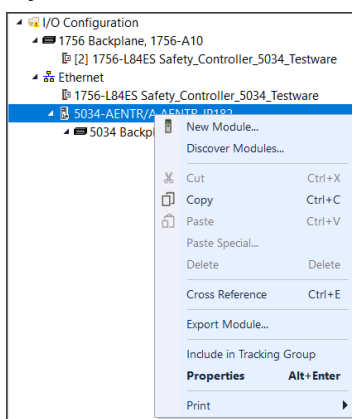
After completing above steps, add your I/O module to the project by following one of the methods:

- Discover Modules
- New Module

Discover Modules

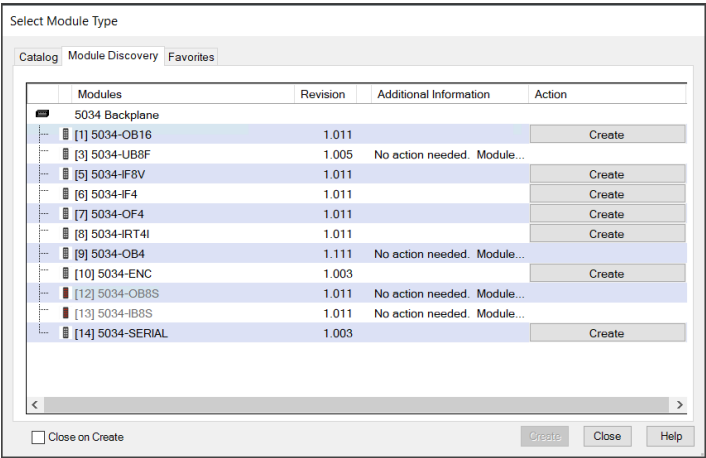
To use the Discover Modules method with PointMax I/O modules, complete these steps:

1. Go online with your Studio 5000 Logix Designer application.
The project must include a PointMax EtherNet/IP adapter.
2. Right-click the PointMax EtherNet/IP adapter and select Discover Modules.

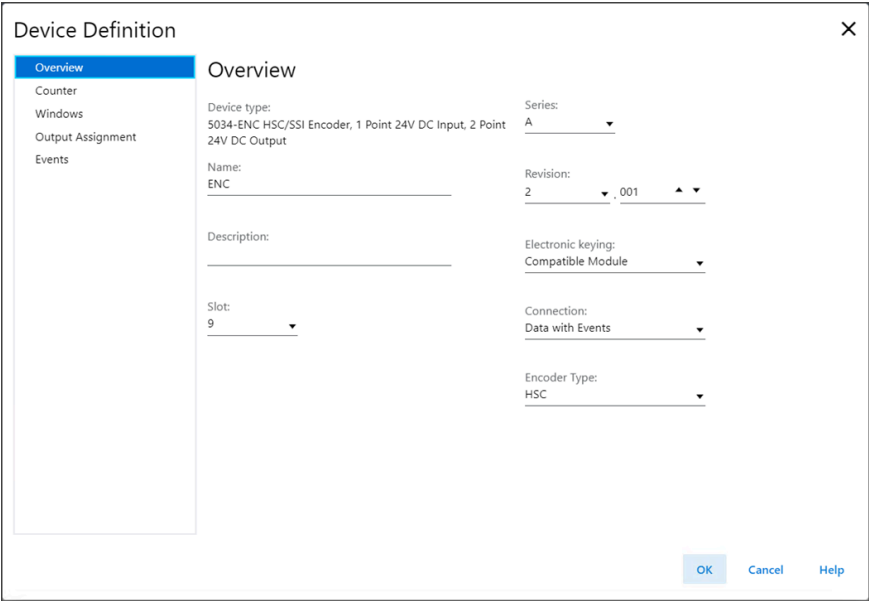


The Studio 5000 Logix Designer application automatically detects available modules that are connected to the backplane. The Select Module Type dialog appears with a list of modules detected.

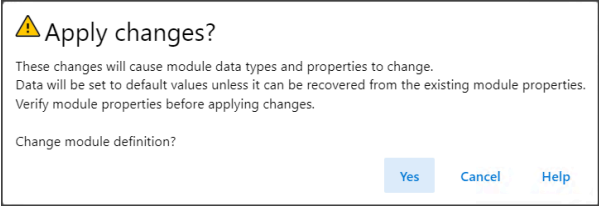
3. In the Select Module Type window, select Create for the module that you want to add.



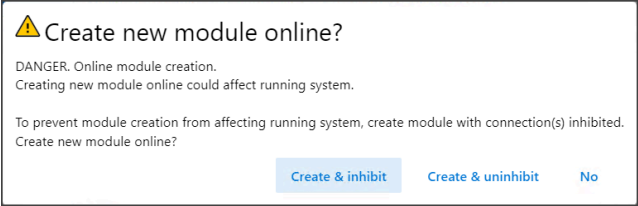
- The New Module window with Device definition dialog appears.
4. In the Device definition dialog, configure the module properties and select OK.



5. If Apply changes warning dialog appears, select Yes.



6. At the Create new module online warning dialog, select Create & inhibit.



7. After adding as many as module you need, close the Select Module Type dialog.

To add additional I/O modules with this method, complete one of the following:

- If you selected the Close on Create checkbox when you created the first I/O module, repeat steps 2...7.
- If you did not select the Close on Create checkbox when you created the first I/O module, repeat steps 3...7.

Add a New Module

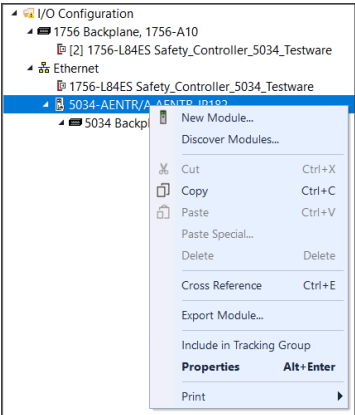
To use the New Module method with PointMax I/O modules, complete these steps:



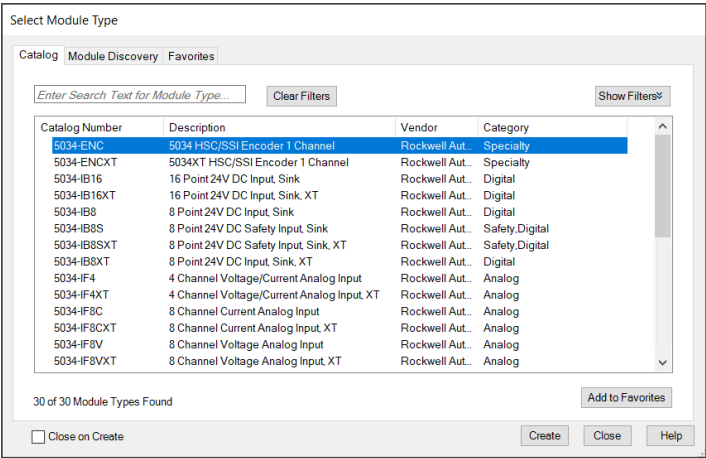
This example shows how to add a module when the Studio 5000 Logix Designer application project is offline.

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 Modules. One exception is that, in step 1, you choose New Module instead of Discover Modules.

1. Right-click the PointMax EtherNet/IP adapter and select New Module.

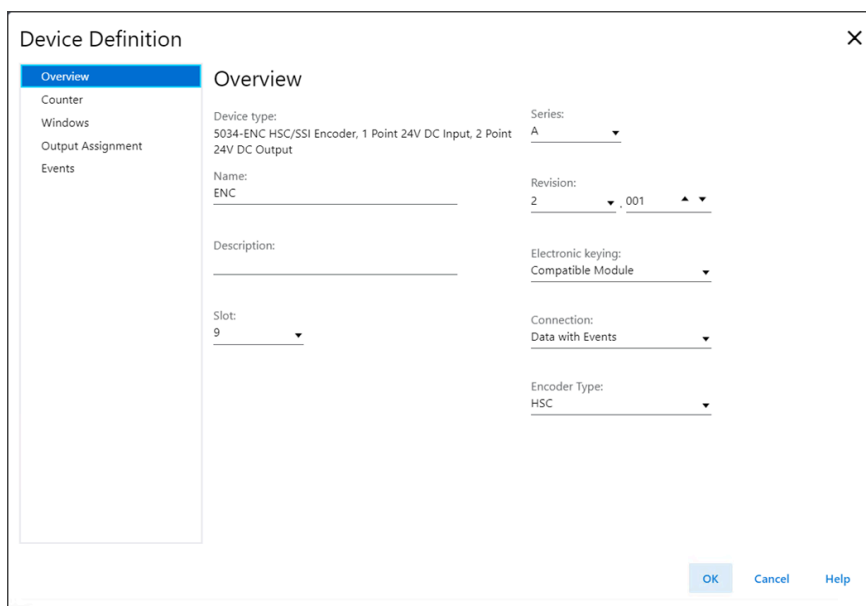


2. In the Select Module Type dialog, choose the module that you want to add and select Create.



The New Module window with Device definition dialog appears.

3. You can select OK to use the default configuration as shown or edit the module configuration.



4. After adding as many as module you need, close the Select Module Type dialog. To add additional I/O modules with this method, complete one of the following:
 - If you selected the Close on Create checkbox when you created the first I/O module, repeat steps 1...3.
 - If you did not select the Close on Create checkbox when you created the first I/O module, repeat steps 2...3.

I/O Tag Name Conventions

The module tag names use defined naming conventions. The conventions are as follows:

Example tag name = *Adapter1:I.Counter00.Count*

- Adapter = Name of the EtherNet/IP adapter in the system
- 1 = Slot number
- I = Tag type

The possible tag types are C (configuration), I (input), O (output), EI (event input), and EO (event output)

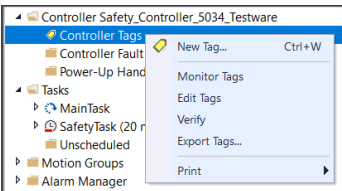
- Counter = Module counter number
- Count = Tag function

In this case, Count represents the count value that is returned to the owner-controller.

Access I/O Tags

To access the I/O tags in the Studio 5000 Logix Designer application, proceed as follows:

1. Open your Studio 5000 Logix Designer application project.
2. Right-click Controller Tags and select Monitor Tags.



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

A screenshot of the 'Controller Tags - Safety_Controller_5034_Testware(controller)' window. The 'Scope' is set to 'Safety_Controller_5' and 'Show' is set to 'All Tags'. The table below lists various tags with their names, values, force masks, styles, and data types.

Name	Value	Force Mask	Style	Data Type
Adapter:1.C	[...]	[...]		AB:5000_HSC1_C_DI1_DO2_0661:C:0
Adapter:1.I	[...]	[...]		AB:5000_HSC1_DI1_DO2_0661:I:0
Adapter:1.I.RunMode	0		Decimal	BOOL
Adapter:1.I.ConnectionFaulted	0		Decimal	BOOL
Adapter:1.I.DiagnosticActive	0		Decimal	BOOL
Adapter:1.I.DiagnosticSequenceCount	0		Decimal	SINT
Adapter:1.I.Counter00	[...]	[...]		CHANNEL_HSC_CTRL2:I:0
Adapter:1.I.Window00	[...]	[...]		AB:5000_Window_Struct:I:0
Adapter:1.I.Window01	[...]	[...]		AB:5000_Window_Struct:I:0
Adapter:1.I.Window02	[...]	[...]		AB:5000_Window_Struct:I:0
Adapter:1.I.Window03	[...]	[...]		AB:5000_Window_Struct:I:0
Adapter:1.I.Input00	[...]	[...]		CHANNEL_DI:I:0
Adapter:1.I.Output00	[...]	[...]		CHANNEL_DO:I:0
Adapter:1.I.Output01	[...]	[...]		CHANNEL_DO:I:0
Adapter:1.O	[...]	[...]		AB:5000_HSC1_DO2_0661:O:0
Adapter:S	[...]	[...]		AB:5000_EN2_DLR_Base32:S:0

Troubleshoot Your Module

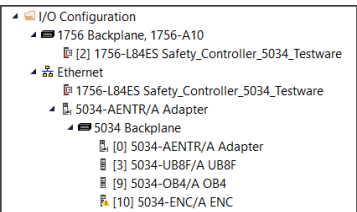
You can use one of the following methods to troubleshoot your module:

- Use Module Status Indicators
See the respective module-specific sections for information on status indicators.
- Use Studio 5000 Logix Designer application.

Use Studio 5000 Logix Designer Application

Check the Controller Organizer to see if there is a warning icon on the module.

Figure 30. Warning Icon in Controller Organizer



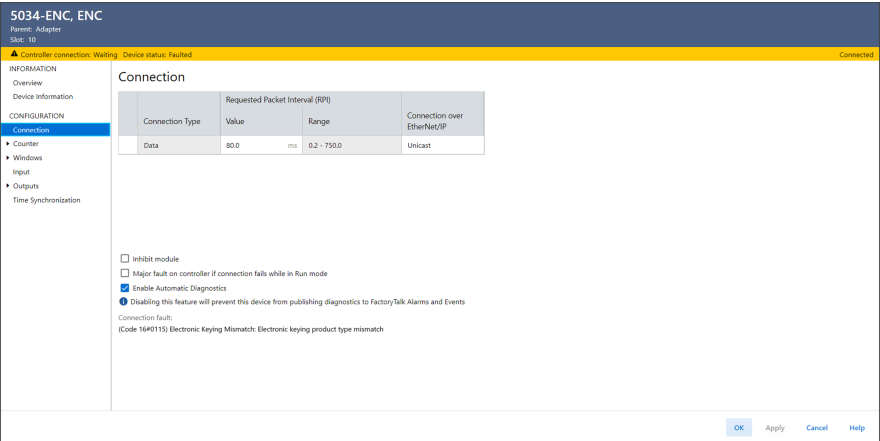
If a warning icon is present, open the module properties and check the status bar to identify the type of fault.

Figure 31. Fault Message in Status Bar



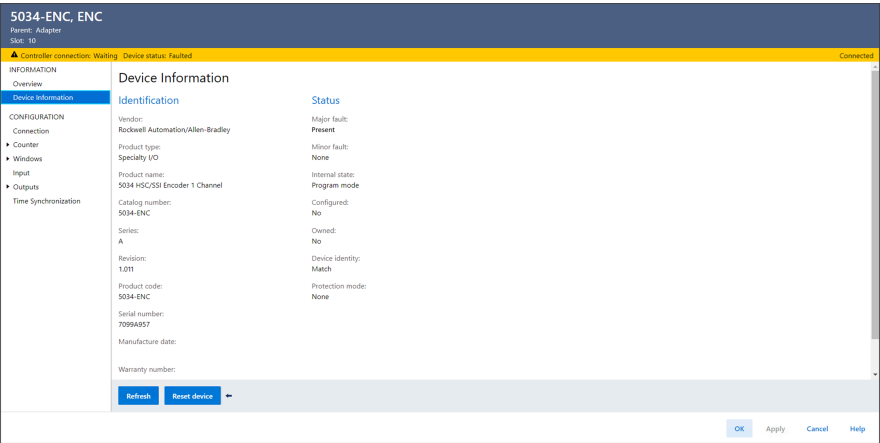
If connection is faulted, go to the Connection view and check the error code in the Connection Fault area.

Figure 32. Connection Fault Description with Error Code



Error Code	Action
0x204	<ol style="list-style-type: none"> 1. Open the adapter module properties. 2. Go to Chassis Information view and check the base status. If there is a fault, check the base details to identify the fault.
Keying Error	<p>Go to Device Information view to confirm that the physical module is compliant or matches the configured identity.</p> <p>For more information, see Electronic Keying in Logix 5000 Control Systems Application Technique, publication LOGIX-AT001.</p>
0x10	<p>Go to Device Information view and check the firmware revision and major fault status.</p> <ul style="list-style-type: none"> • If firmware update is required, update the firmware. • If the major fault is recoverable, reset the module. • If the major fault is unrecoverable, replace the module.

Figure 33. Example for Device Information View with Major Fault Present



If a counter, input, or output is faulted, go to the respective view and check the Diagnostics to see the diagnostic condition.

Figure 34. Example for Counter Diagnostics

Counter Diagnostics

Fault Exist:

No

Data Uncertain:

No

Field Power:

Present

Field Power On Timestamp:

None

Field Power Off Timestamp:

None

Frequency Overrange Fault:

Not Present

Quadrature Error Count:

0

OK

Help

Figure 35. Example for Input00 Diagnostics

Input00 Diagnostics

Fault Exist:

No

Data Uncertain:

No

Field Power:

Present

Field Power On Timestamp:

None

Field Power Off Timestamp:

None

OK

Help

Figure 36. Example for Outputxx Diagnostics

Output00 Diagnostics

Fault Exist:

No

Data Uncertain:

No

Field Power:

Present

Field Power On Timestamp:

None

Field Power Off Timestamp:

None

Short Circuit Fault:

Not Present

Short Circuit Fault Timestamp:

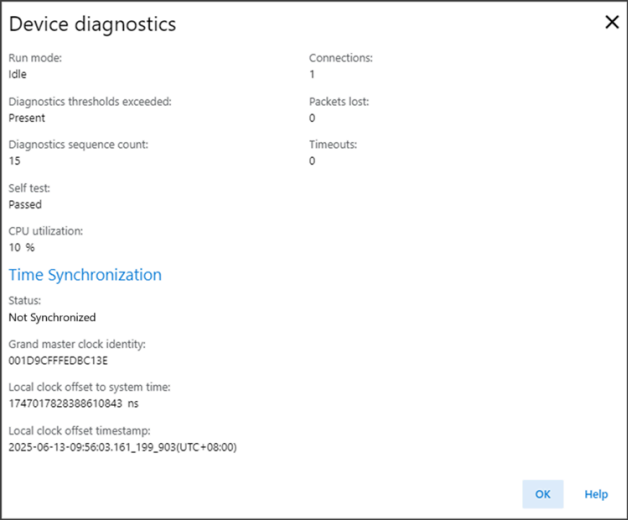
None

OK

Help

If a Time Sync fault is present, go to the Device Diagnostics dialog to check the Time Synchronization status and Grandmaster clock identity.

Figure 37. Example for Device Diagnostics

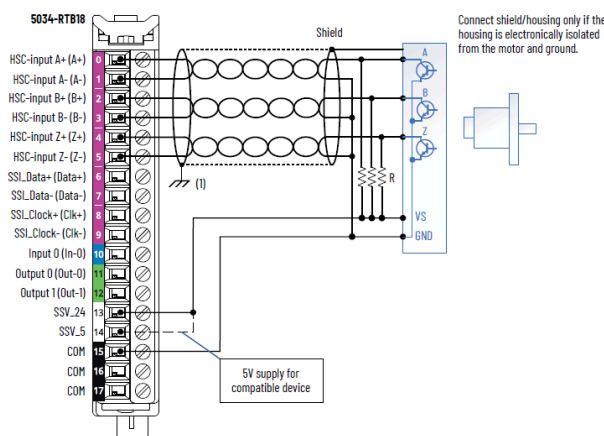


The Grandmaster clock identity must match with the network Grandmaster clock identity. Also, check the Time Synchronization status of the adapter in the adapter Device Diagnostics dialog.

5034-ENC and 5034-ENCXT Details

5034-ENC and 5034-ENCXT HSC/SSI Encoder 1 Channel Module

Figure 38. 5034-ENC and 5034-ENCXT Wiring Diagram – HSC/Incremental Encoder



(1) If the module is installed on a DIN rail that is grounded, you can connect the cable shield to the terminal labeled Chassis instead of Earth Ground.

IMPORTANT: When using the module in HSC mode, do not connect wiring to the SSI terminals (Data, Clk).

IMPORTANT: External resistors, as indicated by the “R” in the diagram, are required if they are not internal to the encoder. The pull-up resistor (R) value depends on the power supply value.

To calculate the maximum resistor value, use the following formula:

$$R = VDC - V_{min} / I_{min}$$

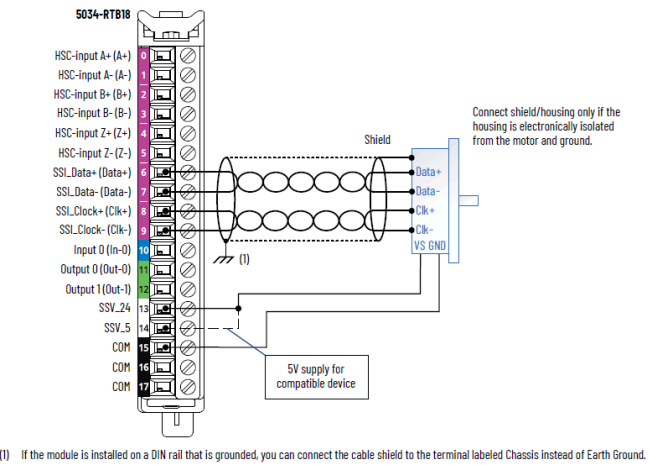
Where:

- R = Maximum pull-up resistor value
- VDC = Power supply voltage (either SSV_24 or SSV_5)
- V_{min} = 3.0V DC
- I_{min} = 2.5 mA

Supply By	Power Supply Voltage (V DC)	Pull-up Resistor Value (R), Max (Ω)(1)
SSV_24	10	2800
	24	8400
	30	10,800
SSV_5	5	800

(1) Resistance values can change, depending on your application. The minimum resistor (R) value depends on the current-sinking capability of the encoder.

Figure 39. 5034-ENC and 5034-ENCXT Wiring Diagram – SSI Encoder



IMPORTANT: When using the module in SSI mode, do not connect wiring to the HSC terminals (A, B, Z).

Figure 40. 5034-ENC and 5034-ENCXT Wiring Diagram – Input/Output

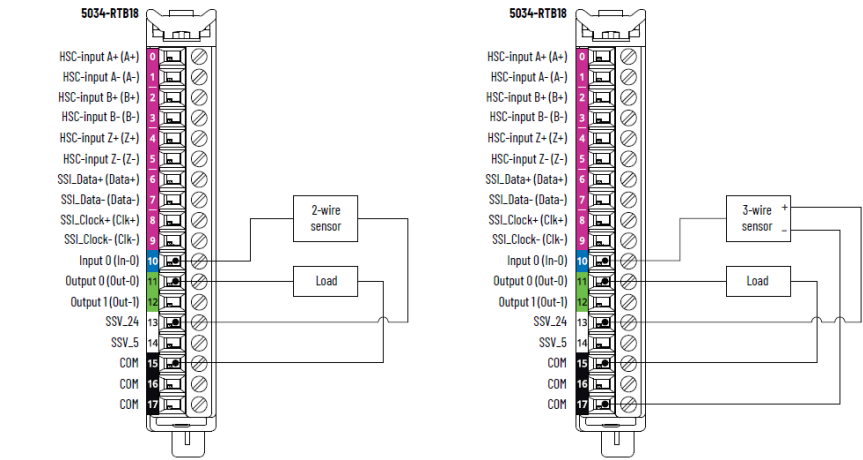


Figure 41. 5034-ENC and 5034-ENCXT Functional Block Diagram

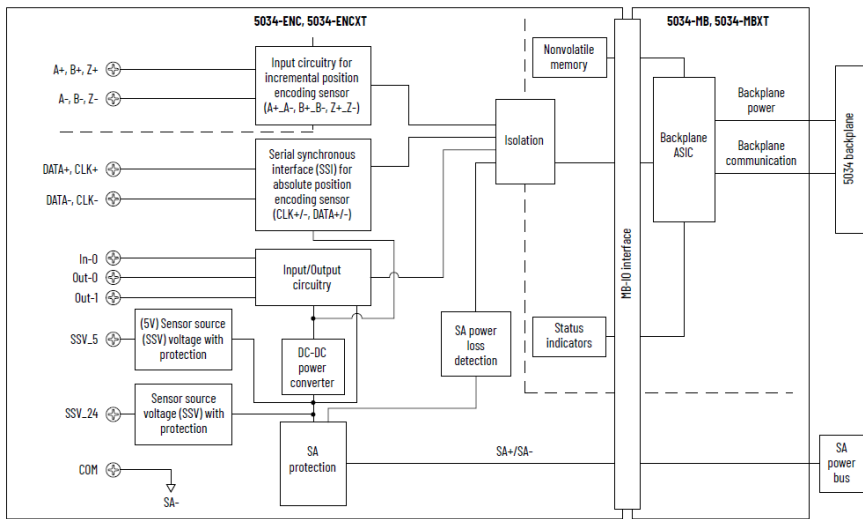


Table 9. Technical Specifications - 5034-ENC, 5034-ENCXT - HSC

Attribute	5034-ENC, 5034-ENCXT
On-state voltage range Channel A, B, Z	3...32V DC
On-state current, min Channel A, B, Z	2.5 mA @ 3V DC
On-state current, nom Channel A, B, Z	4.0 mA @ 24V DC
On-state current, max Channel A, B, Z	8.0 mA @ 32V DC
Off-state voltage, max Channel A, B, Z	1.5V DC
Off-state current, max Channel A, B, Z	0.5 mA
Input pulse width, min Off-to-On On-to-Off	125 ns
Input filter time Off-to-On On-to-Off	0 ns, 100 ns, 200 ns, 500 ns, 1 μ s, 2 μ s, 5 μ s (Default for Input A and Input B), 10 μ s, 20 μ s, 50 μ s, 100 μ s, 200 μ s, 500 μ s, 1 ms (Default for Input Z), 2 ms, 5 ms, 10 ms, 20 ms, 50 ms
Phase separation, min	100 ns
Input pulse frequency, max	1.0 MHz
Input pulse frequency, min	0.1 Hz

Table 10. Technical Specifications - 5034-ENC, 5034-ENCXT - SSI

Attribute	5034-ENC, 5034-ENCXT
SSI mode encoder type	Any absolute encoder supporting standard SSI protocol including linear, rotary, and optical distance measuring devices in MSB-aligned data format The physical interface for clock and data signals is RS-422.
SSI data rate supported	125 kHz (Default), 250 kHz, 500 kHz, 1 MHz, 2 MHz
SSI bits per word supported	2...31 bits 13 bits (Default)
SSI word delay time	16...65,535 μ s

Table 10. Technical Specifications - 5034-ENC, 5034-ENCXT - SSI (continued)

Attribute	5034-ENC, 5034-ENCXT
	64 μ s (Default)
SSI code type supported	Binary, Gray (Gray to binary conversion supported)
SSI direction	Increasing or decreasing SSI count indication
SSI data comparator	Up to four window comparators
SSI data latching	Supported using In-0 input
SSI cable type	UL CM/AWM 2464/CSA Type CMG FT4 or similar cable utilizing shielded twisted pairs for D+/- and C+/- connections. See sensor manufacturer for actual cable required for the SSI sensor under use. In-0 input can be separate from SSI cable.
SSI cable length, max	Depending on the desired SSI data rate: 125 kHz - 320 m (1050 ft) 250 kHz - 160 m (525 ft) 500 kHz - 60 m (195 ft) 1 MHz - 20 m (65 ft) 2 MHz - 8 m (25 ft)

Table 11. Technical Specifications - 5034-ENC, 5034-ENCXT - Digital Input

Attribute	5034-ENC, 5034-ENCXT
On-state voltage range	10...30V DC
On-state current, min	2.0 mA
On-state current, nom	4.5 mA
On-state current, max	5.0 mA
Off-state voltage, max	5V DC
Off-state current, max	1.5 mA
Input impedance, min	2 k Ω @ 10V DC
Input impedance, nom	5.3 k Ω @ 24V DC
Input impedance, max	15 k Ω @ 30V DC
Input delay time (screw to backplane), max	150 μ s
Off-to-On	

Table 11. Technical Specifications - 5034-ENC, 5034-ENCXT - Digital Input (continued)

Attribute	5034-ENC, 5034-ENCXT
On-to-Off	
Input pulse width, min	10 μ s
Off-to-On	
On-to-Off	
Input filter time	0 ns, 10 μ s, 20 μ s, 50 μ s, 100 μ s, 200 μ s, 500 μ s, 1 ms (Default), 2 ms, 5 ms, 10 ms, 20 ms, 50 ms
Off-to-On	
On-to-Off	
Timestamp of inputs (sequence of events)	Not supported

Table 12. Technical Specifications - 5034-ENC, 5034-ENCXT - Digital Output

Attribute	5034-ENC, 5034-ENCXT
On-state voltage range	10...30V DC
On-state voltage drop, max	0.25V DC
On-state current per point, min	1.0 mA
Off-state voltage, max	5V DC with 1 mA min load
Off-state leakage current per point, max ⁽¹⁾	0.5 mA
Output current rating per channel, max	0.5 A
Output current rating per module, max	1 A
Surge current per point, max	1.2 A for 10 ms, repeatable every 3 s
Inductive load allowed, max	1.2 H
Output clamping voltage for inductive load when turn off	(SA voltage + VCL)...0 VCL value: Minimum is -63V, Typical is -55V, Maximum is -49V
Output delay time (backplane to screw), max	150 μ s
Off-to-On	
On-to-Off	
PWM/PTO support	Yes Maximum switching frequency: 100 kHz (10 μ s period)

Table 12. Technical Specifications - 5034-ENC, 5034-ENCXT - Digital Output (continued)

Attribute	5034-ENC, 5034-ENCXT
	Minimum switching frequency: 0.033 Hz (30 s period) Minimum pulse width: 5 μ s Maximum load current (resistive type) per point: <ul style="list-style-type: none"> • 200 mA for $F_{SW} \leq 10$ kHz • 100 mA for $10 \text{ kHz} < F_{SW} \leq 25$ kHz • 40 mA for $25 \text{ kHz} < F_{SW} \leq 100$ kHz Where F_{SW} is the switching frequency
Triggered pulse output support	Yes
Open load detection diagnostics	Not supported
Output short circuit/overload detection	Yes
Output short circuit/overload protection	Yes
Pilot duty rating	1.2 A inrush current, 0.5 A rated current, DC-14
Output states in program mode per point	Hold Last State On Off (Default) Local Control
Output states in communication fault mode per point	Hold Last State On Off (Default) Local Control
Duration of fault mode per point	1 s 2 s 5 s 10 s Forever (Default)
Output final state after fault mode per point	On Off (Default)

(1) Recommended loading resistor - To limit the effects of leakage current through solid-state outputs, you can connect a loading resistor in parallel with your load. For 24V DC operation, use a 5.6 k Ω , 0.5 W resistor for transistor operation.

Table 13. General Specifications - 5034-ENC, 5034-ENCXT

Attribute	5034-ENC, 5034-ENCXT
Number of HSC	1 group of A+/A-, B+/B-, and Z+/Z-
Number of SSI	1 group of Data+/Data- and Clock+/Clock-
Number of inputs	1 channel, sinking
Number of outputs	2 channels, sourcing
SA power voltage, nom	24V DC
SA power voltage range	10...30V DC
SA power current, nom	1.6 A
SA power current, max	1.7 A
SA power current at no load	12 mA
SA reverse polarity protection	Yes
SSV voltage range, SSV_24	Follow SA power supply
SSV voltage range, SSV_5	5V ($\pm 5\%$)
SSV current, SSV_24, max	500 mA
SSV current, SSV_5, max	300 mA
SSV short-circuit protection, SSV_24	Yes
SSV short-circuit protection, SSV_5	Yes
Power dissipation, max ⁽¹⁾	1.22 W
Thermal dissipation, max ⁽¹⁾	4.16 BTU/hr
Isolation voltage	250V (continuous), Basic Insulation Type, System to Field 250V (continuous), Basic Insulation Type, HSC input to other Field side 250V (continuous), Basic Insulation Type, HSC input to System No isolation between SA power and Digital input/output points No isolation between SA power and SSI ports No isolation between SA power and SSV_24/SSV_5 No isolation between individual Digital input/output points
RIUP support	Yes
CIP Sync	Slave only ordinary clock
Electronic keying	Electronic keying via programming software

Table 13. General Specifications - 5034-ENC, 5034-ENCXT (continued)

Attribute	5034-ENC, 5034-ENCXT
RTB key positions (slot)	3, 7, 12
RTB supported	5034-RTB18, 5034-RTB18S
Wiring category ⁽²⁾	2 - Signal ports 2 - Power ports
Wiring specification	See Wiring Specifications for Removable Terminal Blocks in the PointMax I/O System Specifications Technical Data, publication 5034-TD001
Indicators	1 green/red module status indicator 1 green/red SA power status indicator 5 yellow/red I/O status indicators 1 yellow I/O status indicator
Dimensions (HxWxD), approx	123.6 x 15.0 x 66.4 mm (4.87 x 0.59 x 2.61 in.)
Weight, approx	45.0 g (1.59 oz.) - 5034-ENC 47.0 g (1.66 oz.) - 5034-ENCXT
Enclosure type	None (Open-style)
North American temp code	T4
UKEX/ATEX temp code	T4
IECEx temp code	T4

(1) Value is measured at 60 °C (140 °F). Power dissipation varies with temperature.

(2) Use this Conductor Category information for planning conductor routing. See the Industrial Automation Wiring and Grounding Guidelines, publication [1770-4.1](#). Use this Conductor Category information for planning conductor routing as described in the appropriate System Level Installation Manual.

Module Configuration

Before you start configuring your I/O module, you must create a Studio 5000 Logix Designer application project for the Logix 5000 controller that owns the I/O module. The project includes configuration data for the module.

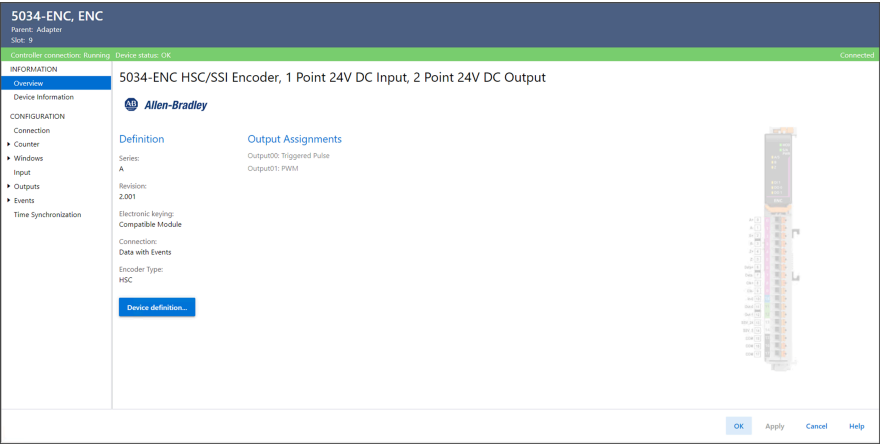
The Studio 5000 Logix Designer application transfers the project to the owner-controller during the program download. Data is then transferred to the I/O modules over the EtherNet/IP network.

The I/O modules can operate immediately after receiving and applying the configuration data.

Overview View

Use Overview to view the definition of a device, including the device type, revision, electronic keying, connection, and other device-specific values.

Figure 42. Overview View Example



The Overview view includes these parameters:

Parameter	Description
Series	Displays the series of the device.
Revision	Displays the revision of the device.
Electronic Keying	<p>Displays the electronic keying that is used for the device. Electronic keying compares the device that is defined in the project to the installed device. If keying fails, a fault occurs.</p> <p>Available options:</p> <ul style="list-style-type: none">Exact MatchCompatible ModuleDisable Keying <div><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></div> <p>For detailed information on Electronic keying, see Electronic Keying in Logix 5000 Control Systems Application Technique, publication LOGIX-AT001.</p>
Connection	Displays the connection type and the communication format.

Parameter	Description
Encoder Type	Displays whether the Encoder Type that is selected is HSC or SSL. Some assignment options are not available depending on the Encoder Type that is selected.
Output Assignments	Displays the operation mode that is assigned for each output.
Device Definition	Opens Device definition dialog to change the revision, electronic keying, channel mode, and other device-specific values.

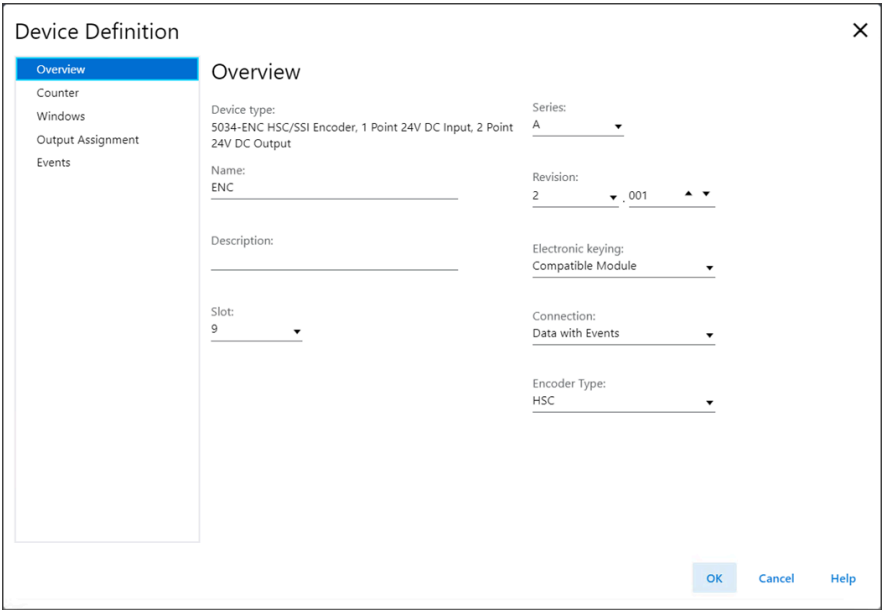
Device Definition

This section shows definitions for specific configurations chosen in the Device Definition dialog. To change the definition of a device, select Device definition in the Overview view.

Device Definition – Overview View

Use Overview view in the Device Definition dialog to define a device or change the device definition.

Figure 43. Device Definition – Overview View Example



Overview includes these parameters:

Table 14. Overview Parameters

Parameter	Definition	Available Choices
Device Type	Displays the device catalog number and type.	Device-specific

Table 14. Overview Parameters (continued)

Parameter	Definition	Available Choices
Name	Enter an IEC 61131 compliant device name. If an invalid character is entered in this field, or if the name exceeds 40 characters, the software ignores the character.	All valid values
Description	Enter the description of the device.	All valid values
Slot	Specify the slot number where the device resides. Only slots between 1 and the maximum number of I/O devices are valid depending on the platform. When the device is created, the slot number defaults to the first available slot position. When the controller is changed to one supporting a smaller maximum I/O count, the current slot value may no longer be valid.	1...32
Series	Specifies the series of the device.	Device-specific
Revision	Specifies the major and minor revisions of the device. The valid range for minor revision is from 1...255.	Device-specific
Electronic Keying	<p>Defines the electronic keying used for the device. Electronic keying compares the device defined in the project to the installed device. If keying fails, a fault occurs.</p> <hr/> <div>  <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> </div> <hr/> <p>For detailed information on Electronic keying, see Electronic Keying in Logix 5000 Control Systems Application Technique, publication LOGIX-AT001.</p>	<ul style="list-style-type: none"> Exact Match Compatible Module Disable Keying
Connection	Specify the type of data transferred between the device and controller. For the Listen Only choice, the controller and device establish communication without the controller sending any configuration or output data to the device. A full input data connection is established but depends on	<ul style="list-style-type: none"> Data Listen Only Data Data with Events Listen Only Data with Events

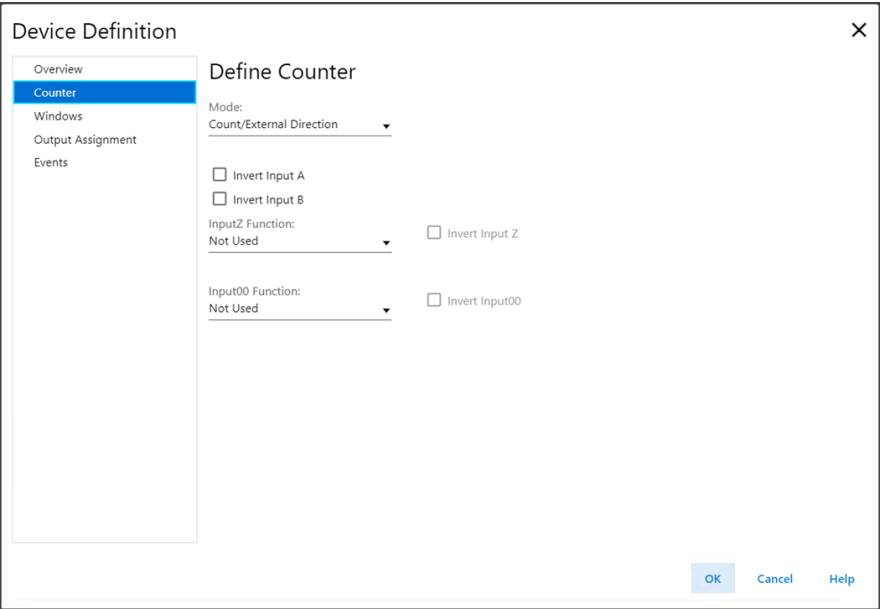
Table 14. Overview Parameters (continued)

Parameter	Definition	Available Choices
	the connection between the owner-controller and the device.	
Encoder Type	Specify the type of encoder in which the module operates.	<ul style="list-style-type: none">• HSC• SSI

Device Definition – Define Counter View

Use Define Counter view to define the counter parameters when the Encoder Type is configured as HSC.

Figure 44. Device Definition – Define Counter View Example



Define Counter includes these parameters:

Table 15. Define Counter – Parameters

Parameter	Definition	Available Choices
Mode	Defines the mode in which the counter operates.	<ul style="list-style-type: none">• Count/External Direction• Count/Internal Direction• Up/Down Pulses• Quadrature X1

Table 15. Define Counter – Parameters (continued)

Parameter	Definition	Available Choices
		<ul style="list-style-type: none"> • Quadrature X2 • Quadrature X4
Invert Input A	<p>Determines whether to invert Input A.</p> <p>This parameter is only available when the counter mode is configured as Count/External Direction, Count/Internal Direction, or Up/Down Pulses.</p>	<ul style="list-style-type: none"> • Cleared – Do not invert the input. • Selected – Invert the input.
Invert Input B	<p>Determines whether to invert Input B.</p> <p>This parameter is only available when the counter mode is configured as Count/External Direction, Count/Internal Direction, or Up/Down Pulses.</p>	<ul style="list-style-type: none"> • Cleared – Do not invert the input. • Selected – Invert the input.
Invert Input A, B	<p>Determines whether to invert Input AB.</p> <p>This parameter is only available when the counter mode is configured as Quadrature X1, Quadrature X2, or Quadrature X4.</p>	<ul style="list-style-type: none"> • Cleared – Do not invert the input. • Selected – Invert the input.
Input Z Function	<p>Defines the function for Input Z.</p> <p>Each counter function (Reset, Store, Load, Hold) can only be triggered by one input.</p> <p>For example, if you select the Input Z Function as "Reset, Hold", then the options that contain "Reset" and "Hold" are not available to select for the Input00 Function.</p> <p>If you select any option for Input Z Function, then Input00 Function is automatically set to Not Used.</p>	<ul style="list-style-type: none"> • Not Used • Hold • Load • Load, Hold • Reset • Reset, Hold • Store • Store, Hold • Store, Load • Store, Load, Hold • Store, Reset • Store, Reset, Hold
Invert Input Z	Determines whether to invert Input Z.	<ul style="list-style-type: none"> • Cleared – A rising edge on Input Z triggers the counter functions. • Selected – A falling edge on Input Z triggers the counter functions.
Input00 Function	<p>Defines the function for Input00.</p> <p>Each counter function (Reset, Store, Load, Hold) can only be triggered by one input.</p>	<ul style="list-style-type: none"> • Not Used • Hold • Load

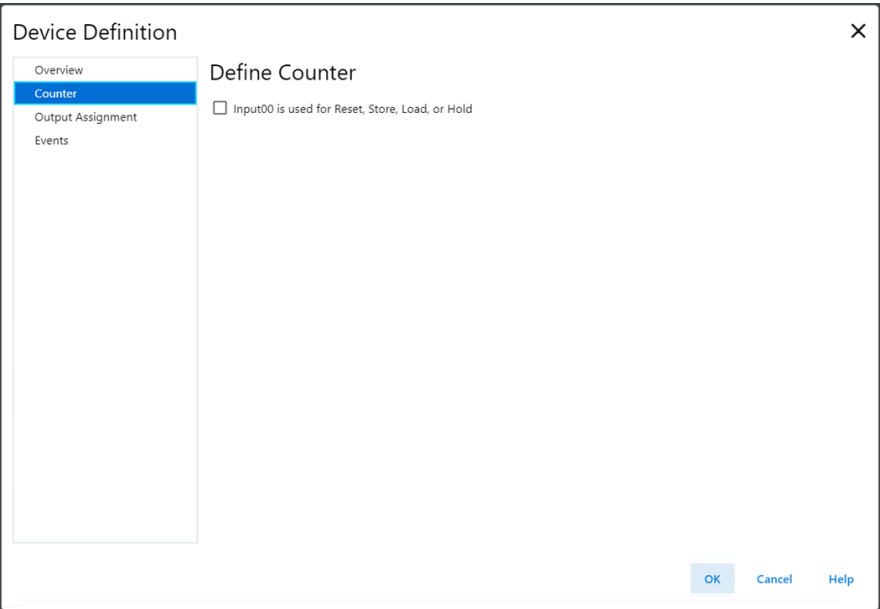
Table 15. Define Counter – Parameters (continued)

Parameter	Definition	Available Choices
	For example, if you select the Input Z Function as "Reset, Hold", then the options that contain "Reset" and "Hold" are not available to select for the Input00 Function. If you select any option for Input Z Function, then Input00 Function is automatically set to Not Used.	<ul style="list-style-type: none">• Load, Hold• Reset• Reset, Hold• Store• Store, Hold• Store, Load• Store, Load, Hold• Store, Reset• Store, Reset, Hold
Invert Input00	Determines whether to invert Input00.	<ul style="list-style-type: none">• Cleared – A rising edge on Input00 triggers the counter functions.• Selected – A falling edge on Input00 triggers the counter functions.

Device Definition – Define Counter View (Listen Only)

Use Define Counter to define the counter parameters when the Encoder Type is configured as HSC and Connection is configured as Listen Only Data.

Figure 45. Device Definition – Define Counter View (Listen Only) Example



Define Counter (Listen Only) includes these parameters:

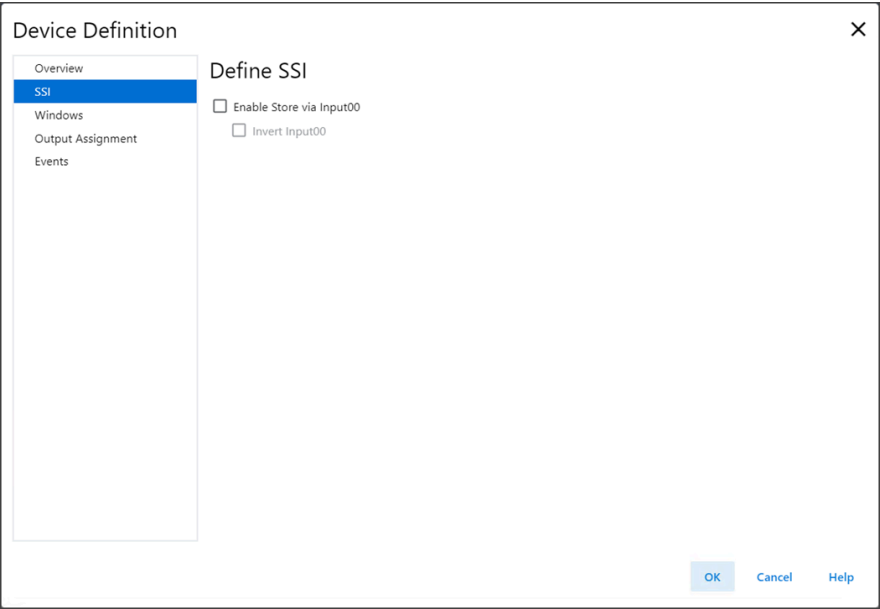
Table 16. Define Counter (Listen Only) - Parameters

Parameter	Definition	Available Choices
Input00 is used for Reset, Store, Load, or Hold	Determines whether to use Input00 to trigger counter functions.	<ul style="list-style-type: none">• Cleared – Do not use Input00 for Reset, Store, Load, or Hold• Selected – Use Input00 for Reset, Store, Load, or Hold

Device Definition – Define SSI View

Use Define SSI view to define the SSI parameters when the Encoder Type is configured as SSI.

Figure 46. Device Definition – Define SSI View Example



Define SSI includes these parameters:

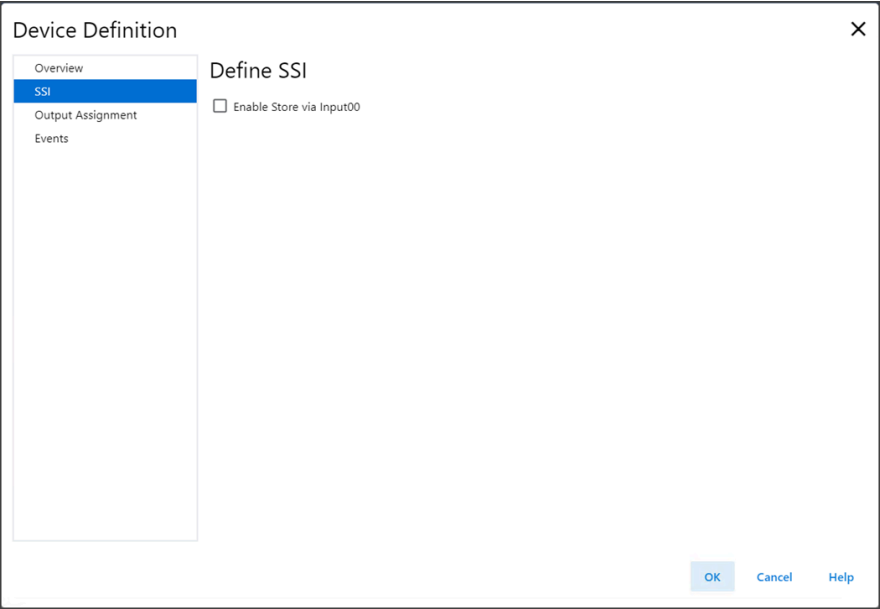
Table 17. Define SSI - Parameters

Parameter	Definition	Available Choices
Enable Store via Input00	Defines whether the Store function is triggered using Input00.	<ul style="list-style-type: none">• Cleared – Store function is not triggered using Input00.• Selected – Store function is triggered using Input00.
Invert Input00	Defines whether to invert the indicated input.	<ul style="list-style-type: none">• Cleared – A rising edge on Input00 triggers the Store function.• Selected – A falling edge on Input00 triggers the Store function.

Device Definition – Define SSI View (Listen Only)

Use Define SSI view to define the SSI parameters when the Encoder Type is configured as SSI and Connection is configured as Listen Only Data.

Figure 47. Device Definition – Define SSI view (Listen Only) Example



Define SSI (Listen Only) includes these parameters:

Table 18. Define SSI (Listen Only) – Parameters

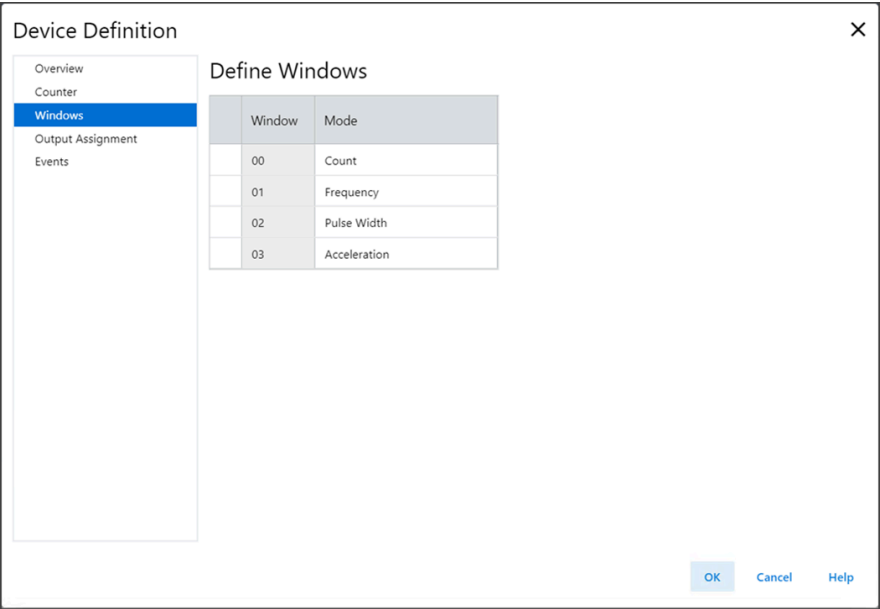
Parameter	Definition	Available Choices
Enable Store via Input00	Defines whether the Store function is triggered using Input00.	<ul style="list-style-type: none">Cleared – Store function is not triggered using Input00.Selected – Store function is triggered using Input00.

Device Definition – Define Windows View

Use Define Windows view to define the Windows parameters.

This view is not available when the Connection is configured as Listen Only Data.

Figure 48. Device Definition – Define Windows Example



Define Windows includes these parameters:

Table 19. Define Windows – Parameters

Parameter	Definition	Available Choices
Window	The window to which this configuration applies	0...3
Mode	Defines the counter or SSI variable for the window to operate on.	Options for HSC mode: <ul style="list-style-type: none">CountScaled CountFrequencyScaled Frequency

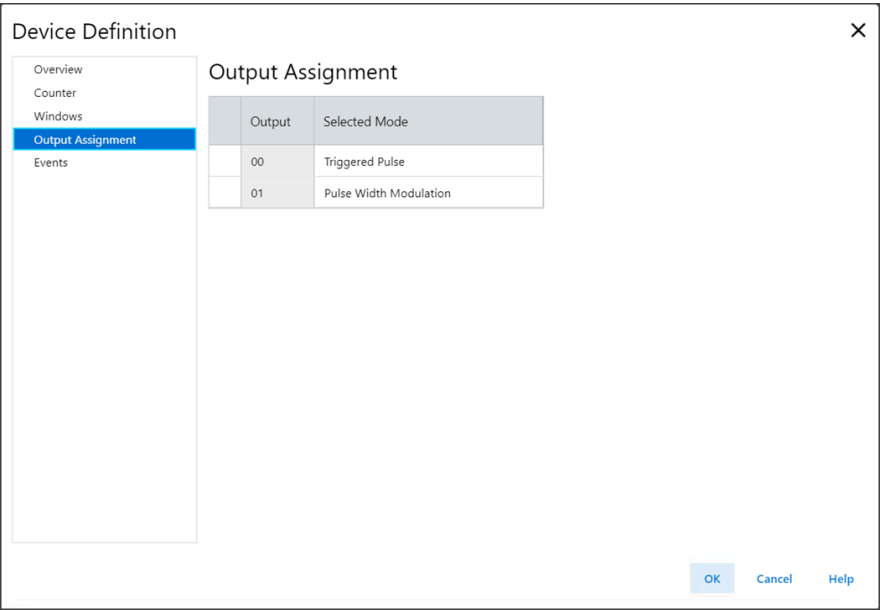
Table 19. Define Windows – Parameters (continued)

Parameter	Definition	Available Choices
		<ul style="list-style-type: none">Average FrequencyScaled Average FrequencyPulse WidthAverage Pulse WidthAccelerationAverage AccelarationScaled AccelerationScaled Average Acceleration Options for SSI mode: <ul style="list-style-type: none">Data

Device Definition – Output Assignments View

Use Output Assignments view to define the operating mode of the outputs.

Figure 49. Device Definition – Output Assignments View Example



Output Assignments includes these parameters:

Table 20. Output Assignments – Parameters

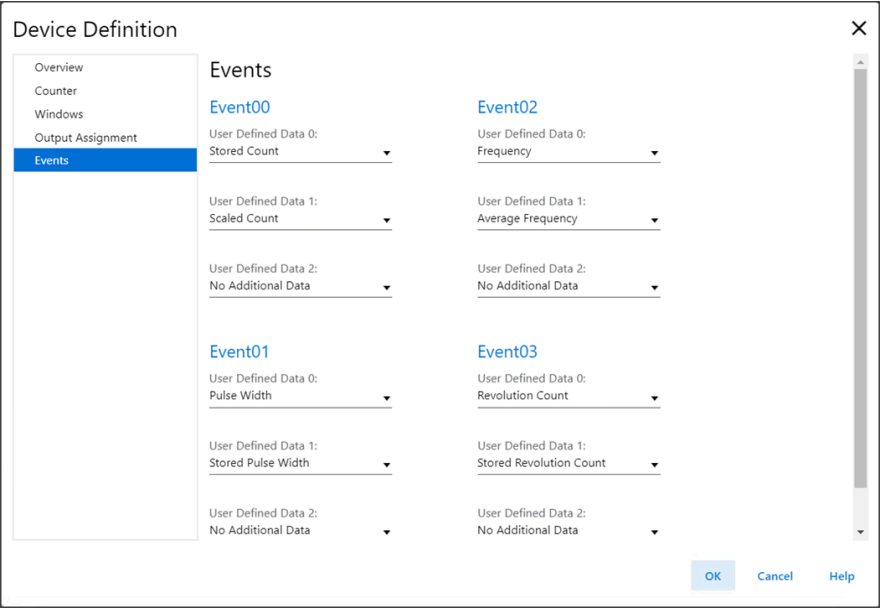
Parameter	Definition	Available Choices
Output	The output to which this configuration applies.	0...1
Selected Mode	Defines the mode in which the output operates.	<ul style="list-style-type: none">DisabledControlled by WindowPulse Width ModulationTriggered Pulse

Device Definition – Events View

Use Events view to define the Events parameters.

This view is not available when the Connection is configured as Data or Listen Only Data.

Figure 50. Device Definition – Events View Example



Events includes these parameters:

Table 21. Events – Parameters

Parameter	Definition	Available Choices
Eventxx – User Defined Data x	Associates user-defined data instances with Events on the module. You can choose as many as three instances for each event.	<p>When Encoder Type is configured as HSC:</p> <ul style="list-style-type: none">No Additional DataStored CountScaled CountScaled Stored CountFrequency

Table 21. Events - Parameters (continued)

Parameter	Definition	Available Choices
		<ul style="list-style-type: none">• Average Frequency• Stored Frequency• Scaled Frequency• Scaled Average Frequency• Scaled Stored Frequency• Pulse Width• Average Pulse Width• Stored Pulse Width• Acceleration• Average Acceleration• Count• Revolution Count• Stored Revolution Count• Stored Acceleration• Scaled Acceleration• Scaled Stored Acceleration• Scaled Average Acceleration <p>When Encoder Type is configured as SSI:</p> <ul style="list-style-type: none">• No Additional Data• Data• Stored Data• Raw Data• Stored Raw Data

Device Information View

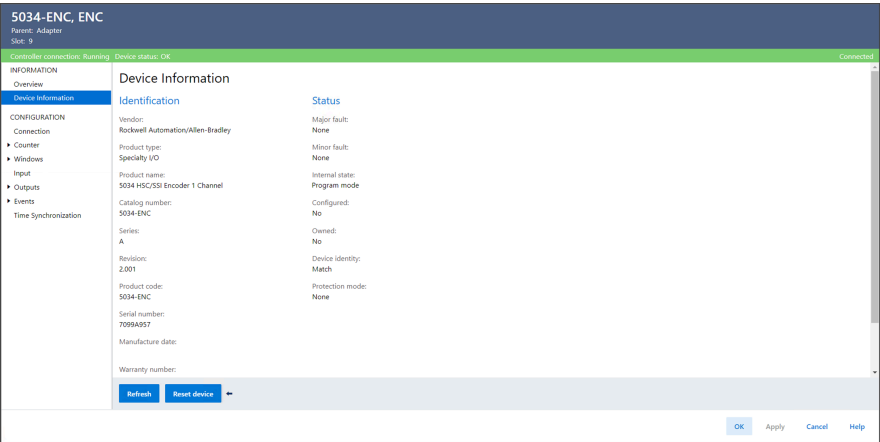
Use Device Information to view device and status information when the device is online.

Use Device Information to:

- Determine the identity of the device.
- View the device's current operational state.
- View whether the device was configured by the owner controller.

- View whether an owner controller is currently connected to the device.
- Retrieve the latest information from the device.
- Reset a device to its power-up state.
- If supported, view the protection mode of the device.

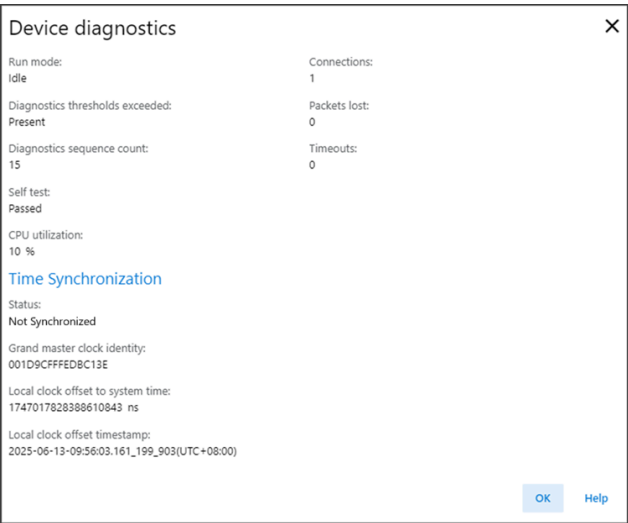
Figure 51. Device Information View Example



Device Diagnostics

To view the device diagnostics, select Diagnostics in the Device Information view.

Figure 52. Device Diagnostics Example



Connection View

Use Connection to define controller-to-device behavior.

- Set the RPI rate. For more information, see [Requested Packet Interval on page 12](#).
- Select Unicast or Multicast connection to use on the EtherNet/IP network. For more information, see [Unicast or Multicast Connection on page 16](#).
- View the reason for a connection fault.



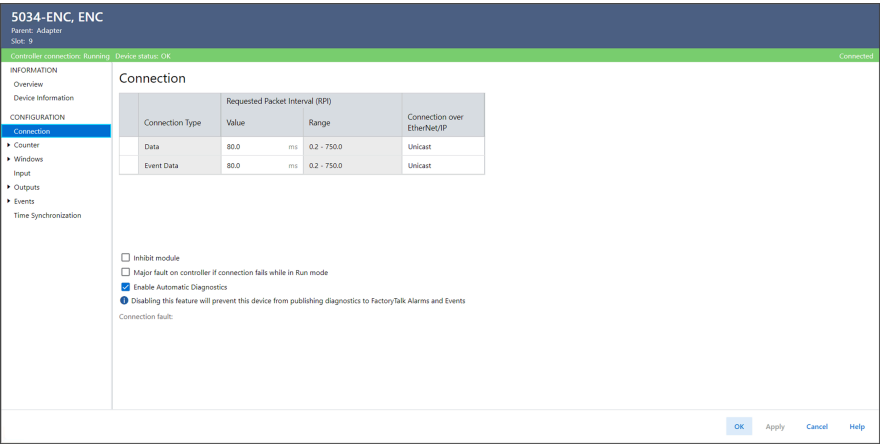
If there is a connection fault, the Connection Fault area displays the error code with a description that helps you to troubleshoot the module. For more information, see [Troubleshoot Your Module on page 51](#).

- Inhibit the module. For more information, see [Module Inhibiting on page 13](#).
- Configure whether a connection failure while the controller is in Run mode causes a major or minor fault.
- Enable or disable Automatic Diagnostics.



ATTENTION: If you do not enable Automatic Diagnostics, the device does not publish diagnostics to FactoryTalk Alarms and Events.

Figure 53. Connection View Example

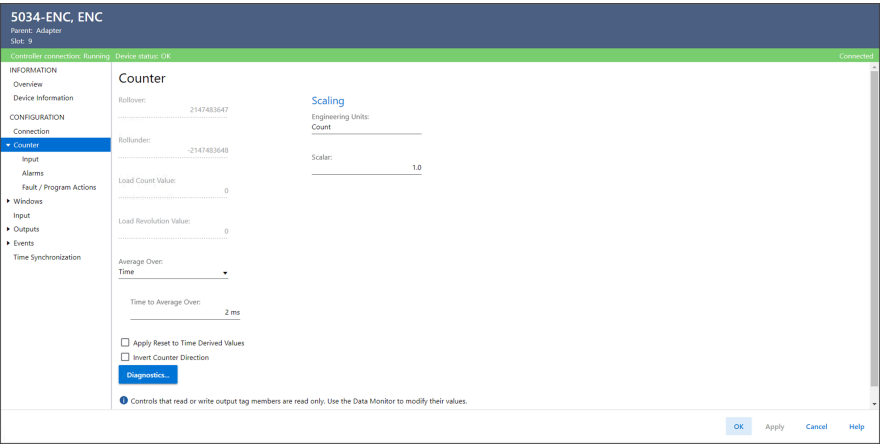


Counter View

Counter is available only if you choose HSC for the Encoder Type parameter in the Device Definition dialog.

Counter shows a summary of the module counter configuration

Figure 54. Counter View Example



The Counter view includes these parameters:

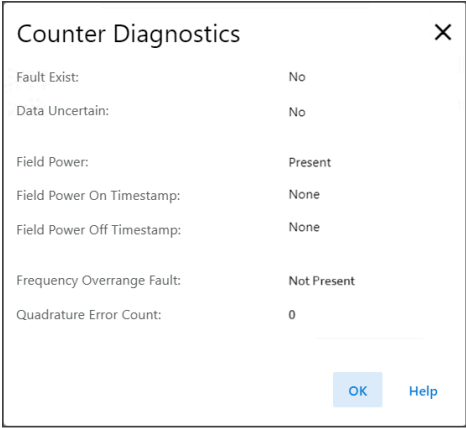
Parameter	Description
Rollover	Displays the value of the RolloverValue output tag. If a counter is counting up, when the counter value reaches the Rollover value - 1, it rolls over to the Rollunder value upon receiving the next count. The rollover tag is set to 1.
Rollunder	Displays the value of the RollunderValue output tag. If a counter in counting down, when the count reaches the Rollunder value, it rolls under to the Rollover value - 1 value upon receiving the next count. The rollunder tag is set to 1.
Load Count Value	Displays the preset number of counts to load into the counter when the Load function is triggered.
Load Revolution Value	Displays the preset number of revolutions to load into the counter when the Load function is triggered.
Average Over	Defines whether to average based on the number of pulses or over a period of time when calculating average frequency, pulse width, and acceleration. <ul style="list-style-type: none"> Pulses – Average based on the number of pulses Time – Average based over a period of time
Pulses to Average Over	Defines the number of pulses to average over. This setting is only available when Counter Average Over is set to Pulses.
Time to Average Over	Defines the duration in milliseconds to average over. This setting is only available when Counter Average Over is set to Time.
Apply Reset to Time Derived Values	Specifies whether to reset the time derived values when the counter is reset. <ul style="list-style-type: none"> Cleared – Do not reset the time derived values Selected – Reset the time derived values
Invert Counter Direction	Specifies whether to invert the direction of the counter. <ul style="list-style-type: none"> Cleared – Do not invert the counter direction Selected – Invert the counter direction
Engineering Units	Defines the engineering unit that the Count, Frequency, and Acceleration values in pulses are scaled to.

Parameter	Description
	For example, if the engineering unit is "abc", the units of the scaled input tags are as follows: <ul style="list-style-type: none">ScaledCount: abcScaledFrequency: abc/sScaledAccel: abc/s²
Scalar	Defines the value that is used to multiply Count, Frequency, and Acceleration values in pulses to achieve the scaled values in engineering units.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Counter Diagnostics

To view the Counter diagnostics, select Diagnostics in the Counter view.

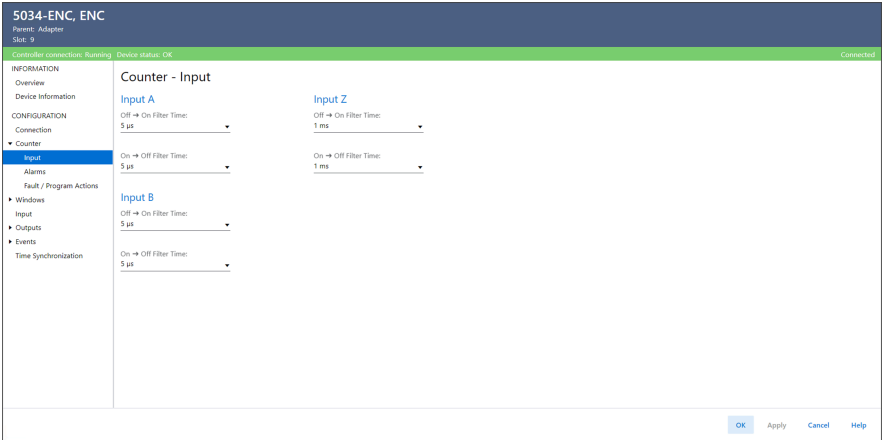
Figure 55. Counter Diagnostics Example



Counter – Input View

Use Counter – Input to configure the Off → On and On → Off filter times for the different inputs.

Figure 56. Counter – Input View Example



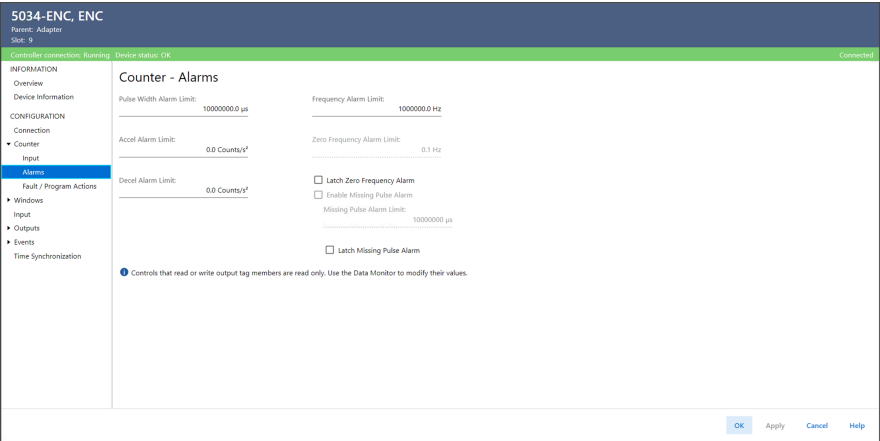
The Counter – Input view includes these parameters:

Parameter	Description
Input A Off → On Filter Time	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input A On → Off Filter Time	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.
Input B Off → On Filter Time	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input B On → Off Filter Time	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.
Input A, B Off → On Filter Time	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input A, B On → Off Filter Time	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.
Input Z Off → On Filter Time	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input Z On → Off Filter Time	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.
Input00 Off → On Filter Time	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid. This parameter is not visible if Input00 Function is set to Not Used in the Device Definition dialog.
Input00 On → Off Filter Time	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid. This parameter is not visible if Input00 Function is set to Not Used in the Device Definition dialog.

Counter – Alarms View

Use Counter – Alarms to configure the alarms for the counter.

Figure 57. Counter – Alarms View Example



The Counter – Alarms view includes these parameters:

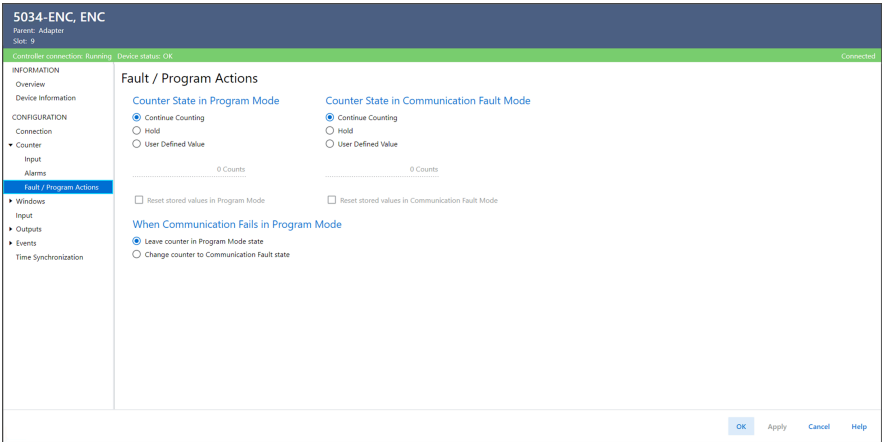
Parameter	Description
Pulse Width Alarm Limit	Defines the maximum pulse width that is allowed before the pulse width alarm is triggered. If the instantaneous or average pulse width exceeds the maximum pulse width, the pulse width alarm is triggered.
Accel Alarm Limit	Defines the maximum acceleration rate that is allowed before the acceleration alarm is triggered. If the instantaneous or average acceleration rate exceeds the maximum acceleration rate, the acceleration alarm is triggered. A value of 0 disables the Alarm.
Decel Alarm Limit	Defines the maximum deceleration rate that is allowed before the deceleration alarm is triggered. If the instantaneous or average acceleration rate is less than the maximum deceleration rate, the deceleration alarm is triggered. A value of 0 disables the Alarm.
Frequency Alarm Limit	Defines the frequency limit that is allowed before the frequency alarm is triggered. If the instantaneous or average frequency exceeds this limit, the frequency alarm is triggered.
Zero Frequency Alarm Limit	Displays the value that the frequency or average frequency must be less than to trigger the zero frequency alarm.
Latch Zero Frequency Alarm	Determines whether the zero frequency alarm latches so that it does not clear until explicitly unlatched.
Enable Missing Pulse Alarm	Determines whether the missing pulse alarm is enabled.

Parameter	Description
Missing Pulse Alarm Limit	Displays the length of time that must elapse between detecting the leading edge of consecutive pulses before the missing pulse alarm is triggered.
Latch Missing Pulse Alarm	Determines whether the missing pulse alarm latches so that it does not clear until explicitly unlatched.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Counter – Fault/Program Actions View

Use Counter– Fault/Program Actions to configure the behavior of the counter when the module is in Program mode or Fault mode, or when a communication fault occurs.

Figure 58. Counter – Fault/Program Actions View Example



The Counter – Fault/Program Actions view includes these parameters:

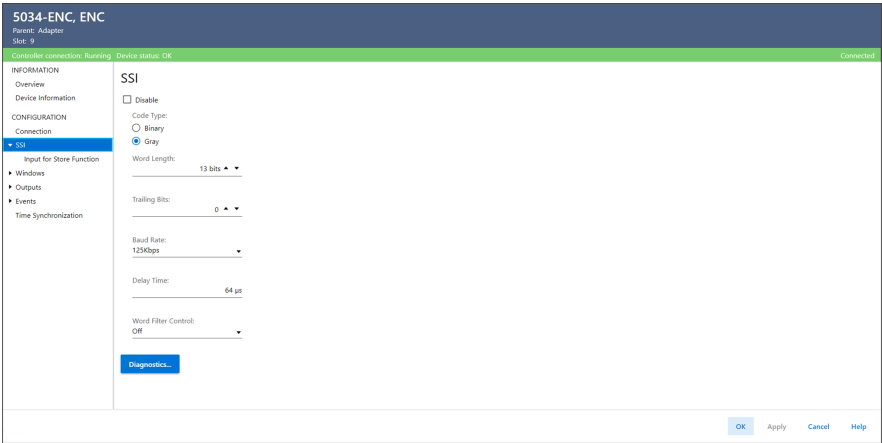
Parameter	Description
Counter State in Program Mode	<p>Determines the behavior of the counter when the controller transitions to Program Mode or Inhibit mode. While in Program Mode or Inhibit mode, counters from the controller are ignored and the counter behaves as follows:</p> <ul style="list-style-type: none"> Continue counting Hold – Retain the last count and resume counting from this count when the controller transitions to Run Mode. User-defined value – Set the count to the user-defined value and resume counting from this value when the controller transitions to Run Mode.
Counter State in Program Mode User-defined Value	Defines the value that is set to count when Counter State in Program Mode is selected as User-defined value.

Parameter	Description
Reset stored values in Program Mode	Determines whether to reset the stored values when the controller transitions to Program Mode or Inhibit mode.
Counter State in Communication Fault Mode	Determines what happens to the counter after a communication fault occurs and the module enters Communication Fault Mode. <ul style="list-style-type: none">Countinue countingHold – Retain the last count and resume counting from this count when the controller transitions to Run Mode.User-defined value – Set the count to the user-defined value and resume counting from this value when the controller transitions to Run Mode.
Counter State in Communication Fault Mode User-defined Value	Defines the value that is set to count when Counter State in Communication Fault Mode is selected as User-defined value.
Reset stored values in Communication Fault Mode	Determines whether to reset the stored values after a communication fault occurs and the module enters Communication Fault Mode.
When Communication Fails in Program Mode	Determines the counter action if communication fails in Program Mode. <ul style="list-style-type: none">Leave counter in Program Mode state – Follow the behavior for Program Mode.Change counter to Communication Fault Mode state – Follow the behavior for Communication Fault Mode.

SSI View

SSI is available only if you choose SSI for the Encoder Type parameter in the Device Definition dialog.

Figure 59. SSI View Example



The SSI view includes these parameters:

Parameter	Description
Disable	Enables or disables the SSI input. When the SSI input is disabled, the module stops the clock signal output.
Code Type	<p>Determines the code type of the SSI sensor.</p> <ul style="list-style-type: none"> Binary Gray - When this option is selected, Gray to Binary conversion is always enabled. <p><i>I.SSI00.Data</i> contains the value after Gray to Binary conversion and <i>I.SSI00.RawData</i> contains the original data received.</p>
Word Length	Determines the length of the received serial data word in bits.
Trailing Bits	Determines the number of bits to shift right in SSI raw data to get <i>I.SSIxx.Data</i> .
Baud Rate	<p>Determines the speed of the SSI data clock.</p> <ul style="list-style-type: none"> 125K 250K 500K 1M 2M
Delay Time	Determines the delay in microseconds between successive SSI data words.
Word Filter Control	<p>Determines the corresponding number of successive equal SSI data words that the module receives from the SSI sensor to update the real time present SSI data word.</p> <ul style="list-style-type: none"> Off Low (2 successive values) Med (3 successive values) High (4 successive values) Max (5 successive values)
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

SSI Diagnostics

To view the SSI diagnostics, select Diagnostics in the SSI view.

Figure 60. SSI Diagnostics Example

SSI Diagnostics

Fault Exist:

No

Data Uncertain:

No

Field Power:

Present

Field Power On Timestamp:

None

Field Power Off Timestamp:

None

OK

Help

SSI – Input for Store Function View

Use SSI – Input for Store Function to configure the Off → On and On → Off filter times for the input.



This view is only available when you use Input00 as a trigger for the "Store" counter function.

Figure 61. SSI – Input for Store Function View Example

5034-ENC, ENC

Parent: Adapter

ID# 9

Configure | Operation | Running | Device status: OK

Connected

INFORMATION

Overview

Device Information

CONFIGURATION

Connection

SSI

Input for Store Function

Windows

Outputs

Events

Time Synchronization

SSI - Input for Store Function

Input00

Filter

Off → On:

1 ms

On → Off:

1 ms

OK

Apply

Cancel

Help

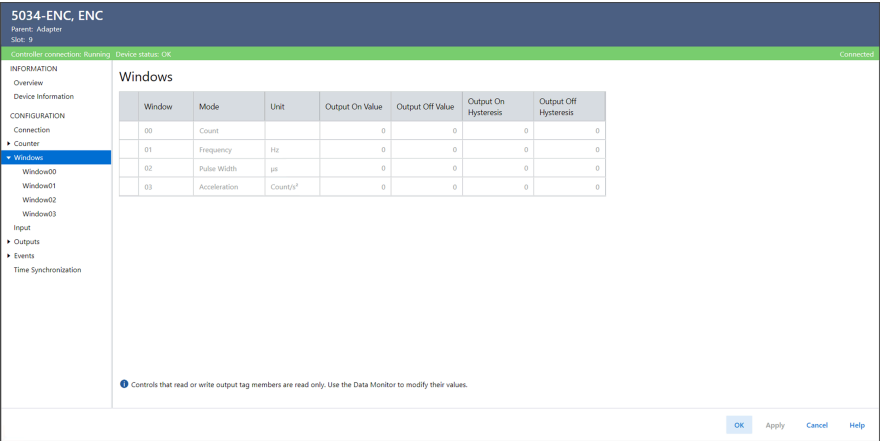
The SSI – Input for Store Function view includes these parameters:

Parameter	Description
Input00 Off → On	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input00 On → Off	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.

Windows View

Windows show a summary of the module windows configuration.

Figure 62. Windows View Example



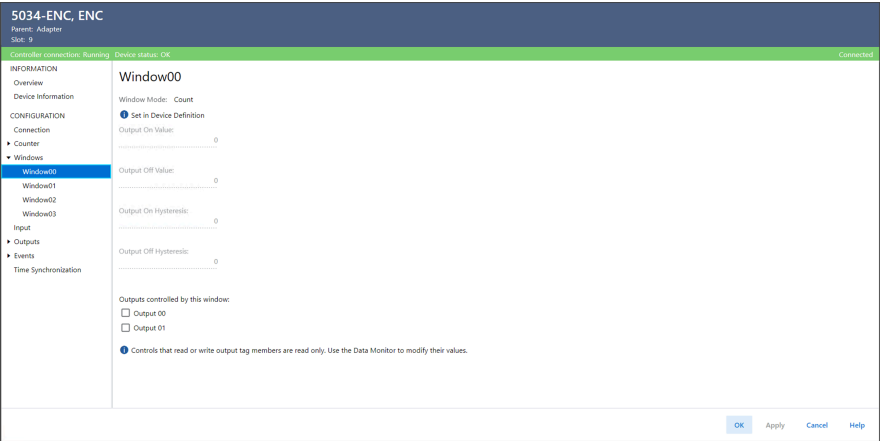
The Windows view includes these parameters:

Parameter	Description
Window	Displays the window number to which this configuration applies.
Mode	Displays which input tag value the window operates on.
Unit	Displays the unit that is associated with the configured Mode for the window.
Output On Value	Displays the value that defines one side of the window thresholds.
Output Off Value	The outputs (if tied to the window) and <i>I.Windowxx.InWindow</i> tag switch state at this threshold according to the diagram Figure 14: Windows Behavior Example on page 33 .
Output On Hysteresis	Displays the negative offset from the Output On Value to apply hysteresis to.
Output Off Hysteresis	Displays the negative offset from the Output Off Value to apply hysteresis to.

Windowxx View

Windowxx shows a specific module windows configuration.

Figure 63. Windowxx View Example



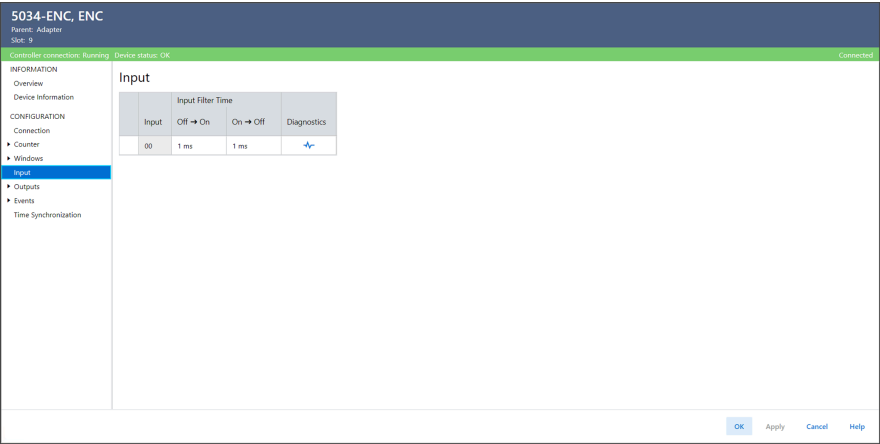
The Windowxx view includes these parameters:

Parameter	Description
Window Mode	Displays the mode that the window is configured for.
Output On Value	Displays the value that defines one side of the window thresholds.
Output Off Value	The outputs (if tied to the window) and <i>I.Windowxx.InWindow</i> tag switch state at this threshold according to the diagram Figure 14: Windows Behavior Example on page 33 .
Output On Hysteresis	Displays the negative offset from the Output On Value to apply hysteresis to.
Output Off Hysteresis	Displays the negative offset from the Output Off Value to apply hysteresis to.
Outputs Controlled by this Window Output00	Determines whether this window controls the indicated output.
Outputs Controlled by this Window Output01	Determines whether this window controls the indicated output.

Input View

Use Input to view or configure the input for the device.

Figure 64. Input View Example



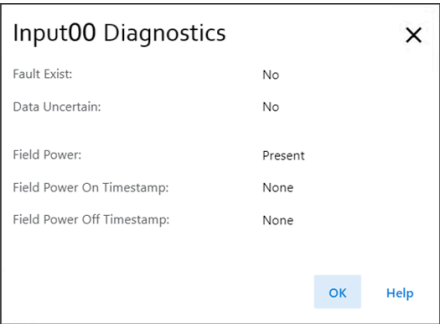
The Input view includes these parameters:

Parameter	Description
Input	Displays the Input number to which this configuration applies.
Input Filter Time Off → On	Defines how long an Off to On input transition must remain in the On state before the device considers the transition valid.
Input Filter Time On → Off	Defines how long an On to Off input transition must remain in the Off state before the device considers the transition valid.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Input00 Diagnostics

To view the Input00 diagnostics, select Diagnostics in the Input view.

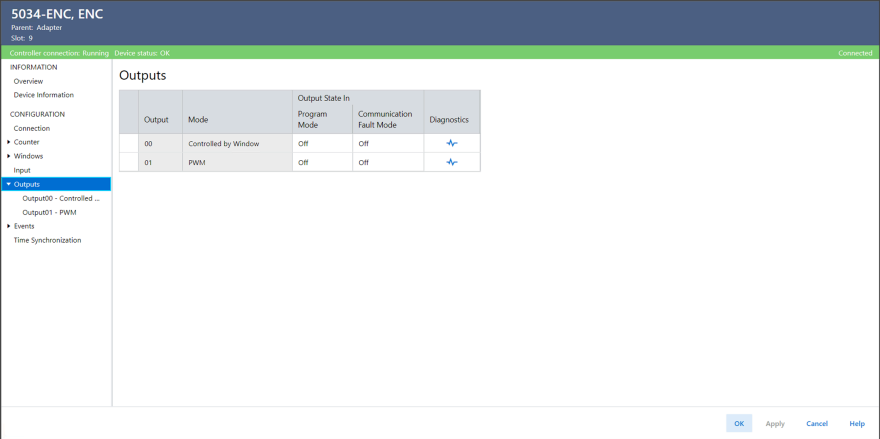
Figure 65. Input00 Diagnostics Example



Outputs View

Use Outputs to view or configure the outputs for the device.

Figure 66. Outputs View Example



The Outputs view includes these parameters:

Parameter	Description
Output	Displays the output number to which this configuration applies.
Mode	Displays the mode for the output.
Output State in Program Mode	Determines the behavior of each output when the controller transitions to Program Mode or Inhibit mode. While in Program Mode or Inhibit mode, outputs from the controller are ignored and the output behaves as follows: <ul style="list-style-type: none">Off – Output is 0 or Off.On – Output is 1 or On.Hold – Output maintains the last state.Local Control – Output continues to be controlled by the window.
Output State in Communication Fault Mode	Determines what happens to the output of each point after a communication fault occurs and the module enters Communication Fault Mode. <ul style="list-style-type: none">Off – Output is 0 or Off.On – Output is 1 or On.Hold – Output maintains the last state.Local Control – Output continues to be controlled by the window.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Outputxx Diagnostics

To view the Outputxx diagnostics, select Diagnostics Outputs view.

Figure 67. Outputxx Diagnostics Example

Output00 Diagnostics

Fault Exist:

No

Data Uncertain:

No

Field Power:

Present

Field Power On Timestamp:

None

Field Power Off Timestamp:

None

Short Circuit Fault:

Not Present

Short Circuit Fault Timestamp:

None

OK

Help

Outputxx – Controlled by Window View

Use Outputxx – Controlled by Window to view or configure a specific output for the device. This view is available when the output if configured as Digital Output in the Device Definition dialog.

Figure 68. Outputxx – Controlled by Window View Example

5034-ENC, ENC

Parent: Adapter

Slot: 0

Communication connection summary Device status: OK Connected

INFORMATION

Overview

Device Information

CONFIGURATION

Connection

Counter

Windows

Input

Outputs

Output00 - Controlled by Window

Output01 - PWM

Events

Time Synchronization

Output00 - Controlled by Window

Output State in Program Mode

Program state

Off

When Communication Fails in Program Mode

☒ Leave outputs in Program Mode state

☐ Change outputs to Communication Fault Mode state

Output State in Communication Fault Mode

Communication fault state

Off

Communication fault state duration

Forever

Final communication fault state

Off

Diagnostics...

OK

Apply

Cancel

Help

The Outputxx – Controlled by Window view includes these parameters:

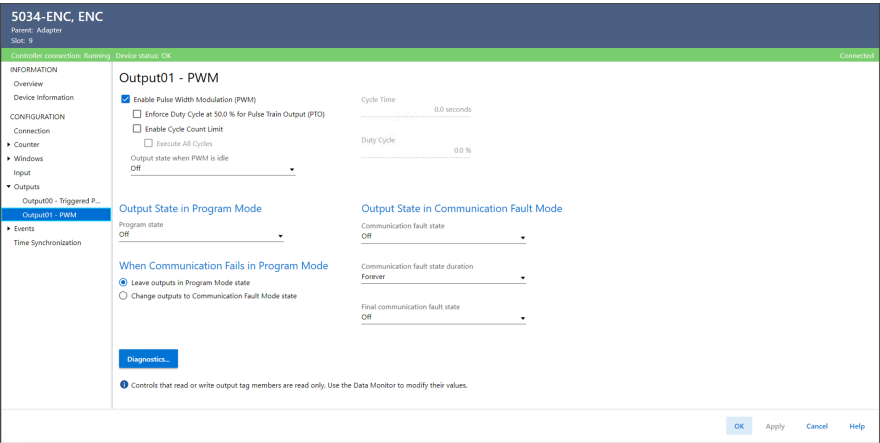
Parameter	Description
Output State In Program Mode Program State	Determines the behavior of each output when the controller transitions to Program Mode or Inhibit mode. While in Program Mode or Inhibit mode, outputs from the controller are ignored and the output behaves as follows: <ul style="list-style-type: none">Off – Output is 0 or Off.On – Output is 1 or On.Hold – Output maintains the last state.Local Control – Output continues to be controlled by the windows.

Parameter	Description
	<p>This option is only available when one of the following conditions is true:</p> <ul style="list-style-type: none"> - The encoder type is set to SSI and the SSI channel controls the output via windows. - The encoder type is set to HSC, the counter controls this output via windows, and the counter is configured as Continue Counting in Program Mode.
Output State in Communication Fault Mode Communication Fault State	<p>Determines what happens to the output of each point after a communication fault occurs and the module enters Communication Fault Mode.</p> <ul style="list-style-type: none"> • Off – Output is 0 or Off. • On – Output is 1 or On. • Hold – Output maintains the last state. • Local Control – Output continues to be controlled by the windows. <p>This option is only available when one of the following conditions is true:</p> <ul style="list-style-type: none"> - The encoder type is set to SSI and the SSI channel controls the output via windows. - The encoder type is set to HSC, the counter controls this output via windows, and the counter is configured as Continue Counting in Communication Fault Mode.
Output State in Communication Fault Mode Communication Fault State Duration	Determines the length of time the output state in the Communication Fault Mode is held before the Final Fault State.
Output State in Communication Fault Mode Final Communication Fault State	<p>Determines whether the module transitions to On or Off after the Communication Fault Mode Output State Duration elapses. The default final state is Off.</p> <p>If you choose Forever for Duration, you cannot choose a final state. The module retains its current Communication Fault Mode state.</p>
When Communication Fails in Program Mode	<p>Determines the output action if communication fails in Program Mode.</p> <ul style="list-style-type: none"> • Leave outputs in Program Mode state – Follow the behavior for Program Mode. • Change outputs to Communication Fault Mode state – Follow the behavior for Communication Fault Mode.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Outputxx – PWM View

Use Outputxx – PWM to view or configure the points for the device.

Figure 69. Outputxx – PWM View Example



The Outputxx – PWM view includes these parameters:

Parameter	Description
Enable Pulse Width Modulation (PWM)	Determines whether the output is Pulse Width Modulation or Digital Output.
Enforce Duty Cycle at 50.0% for Pulse Train Output (PTO)	Determines whether to enforce the PTO specific duty cycle, which is fixed at 50%. If enabled, <i>O.Pwmxx.DutyCycle</i> is ignored.
Enable Cycle Count Limit	Determines whether to generate a defined number of pulse cycles when there is a rising edge in the data or continuous pulse cycles when data is On. <ul style="list-style-type: none"> 0 = Pulse cycles continue to occur until <i>O.Outputxx.Data</i> is set to 0. 1 = Allow only a fixed number of pulse cycles to occur when there is a rising edge in <i>O.Outputxx.Data</i>.
Execute All Cycles	Only applicable when Enable Cycle Count Limit is enabled. <p>When <i>O.Outputxx.Data</i> transitions to 0:</p> <ul style="list-style-type: none"> 0 = Stop generating pulses immediately 1 = Continue generating pules until the Pulse Cycle Limit is reached

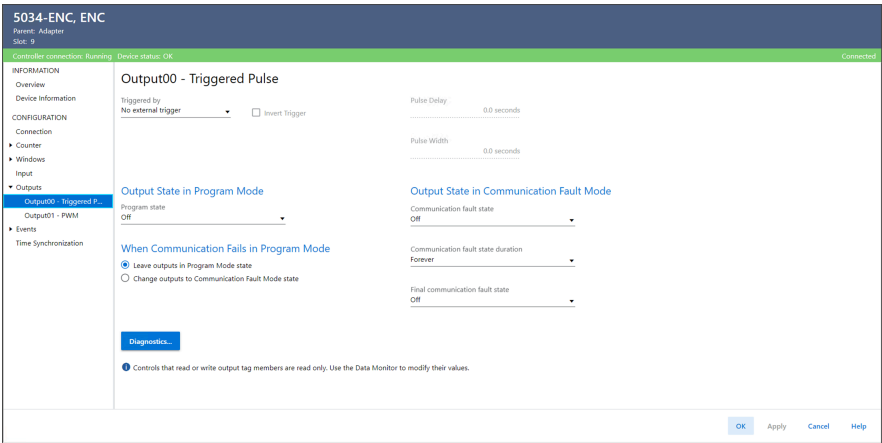
Parameter	Description
Output State when PWM is Idle	<p>Determines the state of the output when PWM is idle.</p> <ul style="list-style-type: none"> 0 = When PWM is idle, the output is Off. It also means that when PWM is running, the active state per <i>O.Pwmxx.DutyCycle</i> is the output in On state. 1 = When PWM is idle, the output is On. It also means that when PWM is running, the active state per <i>O.Pwmxx.DutyCycle</i> is the output in Off state.
Cycle Time	Displays the period in seconds of one pulse cycle.
Duty Cycle	Displays the percentage of the pulse cycle time when the output is in an active state.
Cycle Count Limit	<p>Displays the number of pulse cycles to generate when there is a rising edge in the data.</p> <p>Only applicable when Enable Cycle Count Limit is enabled.</p>
Output State in Program Mode Program State	<p>Determines the behavior of each output when the controller transitions to Program Mode or Inhibit mode. While in Program Mode or Inhibit mode, outputs from the controller are ignored and the output behaves as follows:</p> <ul style="list-style-type: none"> Off – Output is 0 or Off. On – Output is 1 or On.
Output State in Communication Fault Mode Communication Fault State	<p>Determines what happens to the output of each point after a communication fault occurs and the module enters Communication Fault Mode.</p> <ul style="list-style-type: none"> Off – Output is 0 or Off. On – Output is 1 or On.
Output State in Communication Fault Mode Communication Fault State Duration	Determines the length of time the output state in the Communication Fault Mode is held before the Final Fault State.
Output State in Communication Fault Mode Final Communication Fault State	<p>Determines whether the module transitions to On or Off after the Communication Fault Mode Output State Duration elapses. The default final state is Off.</p> <p>If you choose Forever for Duration, you cannot choose a final state. The module retains its current Communication Fault Mode state.</p>

Parameter	Description
Output State When Communication Fails in Program Mode	Determines the output action if communication fails in Program Mode. <ul style="list-style-type: none">Program Mode – Indicates leaving outputs in Program Mode state.Communication Fault Mode – Indicates changing outputs to Communication Fault Mode state.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Outputxx – Triggered Pulse View

Use Outputxx – Triggered Pulse to view or configure a specific output for the device. This view is available when the output is configured as Digital Output in the Device Definition dialog.

Figure 70. Outputxx – Triggered Pulse View Example



The Outputxx – Triggered Pulse view includes these parameters:

Parameter	Description
Triggered by	Specifies the input that triggers the pulse function. <ul style="list-style-type: none">No external trigger – No external trigger is selected. However, you can use 0.Outputxx.Trigger tag to trigger the pulse.Input00 – Input00 activates the output pulse. Input00 is only available when it is not used as a HSC counter function trigger or SSI store trigger.

Parameter	Description
Invert Trigger	<p>This option inverts the trigger edge for the input point that is selected in the 'Trigger by' parameter.</p> <ul style="list-style-type: none"> 0 = A rising edge of the input triggers the output pulse. 1 = A falling edge of the input triggers the output pulse.
Pulse Delay	Specifies the delay time in seconds between the trigger event and the start of the output pulse.
Pulse Active Time	Specifies the active duration in seconds for the generated output pulse.
Output State In Program Mode Program State	<p>Determines the behavior of each output when the controller transitions to Program Mode or Inhibit mode. While in Program Mode or Inhibit mode, outputs from the controller are ignored and the output behaves as follows:</p> <ul style="list-style-type: none"> Off – Output is 0 or Off. On – Output is 1 or On. Local Control – Output continues to operate as configured. <p>This option is only available when Input00 is not used as an encoder trigger.</p>
Output State in Communication Fault Mode Communication Fault State	<p>Determines what happens to the output of each point after a communication fault occurs and the module enters Communication Fault Mode.</p> <ul style="list-style-type: none"> Off – Output is 0 or Off. On – Output is 1 or On. Local Control – Output continues to operate as configured. <p>This option is only available when Input00 is not used as an encoder trigger.</p>
Output State in Communication Fault Mode Communication Fault State Duration	Determines the length of time the output state in the Communication Fault Mode is held before the Final Fault State.
Output State in Communication Fault Mode Final Communication Fault State	<p>Determines whether the module transitions to On or Off after the Communication Fault Mode Output State Duration elapses. The default final state is Off.</p> <p>If you choose Forever for Duration, you cannot choose a final state. The module retains its current Communication Fault Mode state.</p>

Parameter	Description
Output State When Communication Fails in Program Mode	Determines the output action if communication fails in Program Mode. <ul style="list-style-type: none"> Program Mode – Leave outputs in Program Mode state. Communication Fault Mode – Change outputs to Communication Fault Mode state.
Diagnostics	Opens the Diagnostics dialog, which contains diagnostic information.

Events View

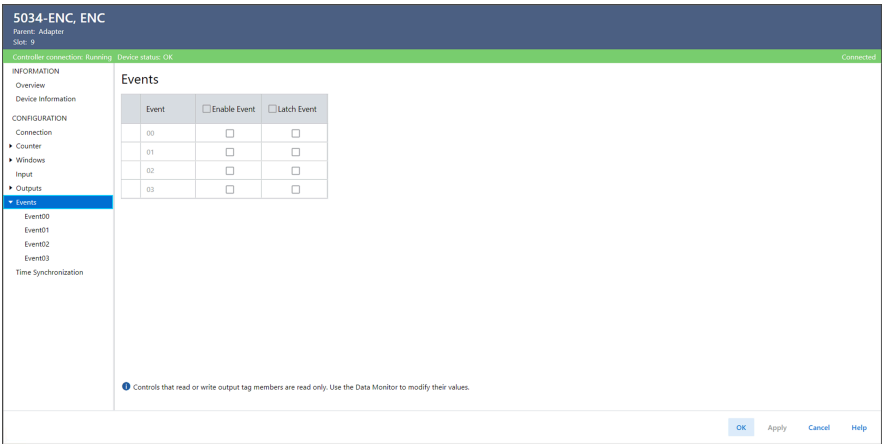
The Events view shows an overview of the configuration values for all events. You can perform certain tasks only when the project is online.

This view is only available when you select the “Data with Events” connection type in the Device definition. See [Device Definition on page 64](#).

IMPORTANT: You cannot configure events on the Events view. The parameters that are displayed in this view are read-only.

You must use the Event Output tags to configure an event. For more information, see [Event Output Tag Definitions on page 138](#).

Figure 71. Events View Example



The Events view include these parameters:

Parameter	Description	Member Name	Value
Event	The event number to which this configuration applies.	—	—
Enable Event	Displays whether the corresponding event trigger is active. Events are triggered when conditions match the event definition.	EO.Eventxx.En	<ul style="list-style-type: none"> 0 = Event trigger definition is not active and events are not triggered when conditions match the definition 1 = Event trigger definition is active and events are triggered when conditions match the definition
Latch Event	Displays whether an event is latched until it is acknowledged.	EO.Eventxx.LatchEvent	<ul style="list-style-type: none"> 0 = Event is not latched. New events overwrite old events. 1 = Event is latched. Events are latched until acknowledged. New events are not reported until the current event is acknowledged. Up to four new events are stored in the memory buffer. When the memory buffer is full, additional events are lost.

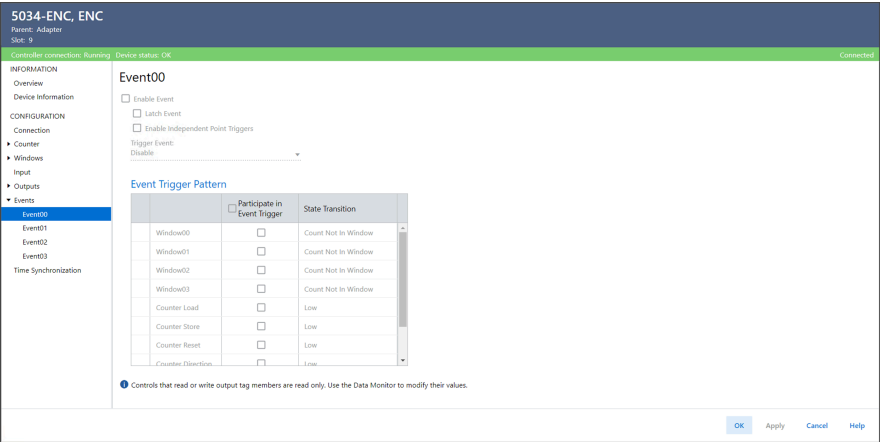
Eventxx View

The Eventxx view, where 'xx' represents the index of an event definition, shows the configuration options available for that event. You cannot configure the parameters from this view, you must use the configuration tags.

Use this view to review the following tasks:

- Whether the event is enabled.
- Whether the event is latched.
- Whether independent point triggers for the event is enabled.
- Check the trigger condition for the event.
- Check the participants and their State Transition behavior for the event.

Figure 72. Eventxx View Example



The Eventxx view includes these parameters.

Parameter	Description	Member Name	Value
Enable Event	Displays whether the corresponding event trigger is active. Events are triggered when conditions match the event definition.	EO.Eventxx.En	<ul style="list-style-type: none"> 0 = Event trigger definition is not active and events are not triggered when conditions match the definition 1 = Event trigger definition is active and events are triggered when conditions match the definition
Latch Event	Displays whether an event is latched until it is acknowledged.	EO.Eventxx.LatchEvent	<ul style="list-style-type: none"> 0 = Event is not latched. New events overwrite old events. 1 = Event is latched. Events are latched until acknowledged. New events are not reported until the current event is acknowledged. Up to four new events are stored in the memory buffer. When the memory buffer is full, additional events are lost.
Enable Independent Point Triggers	Displays whether each condition indicated in the Event trigger definition can initiate an event independently.	EO.Eventxx.IndependentConditionTriggerEn	<ul style="list-style-type: none"> 0 = Event is triggered when all participating conditions in the event trigger achieve their configured values 1 = Event is triggered when any participating condition in the event trigger achieves their configured value
Trigger Event	Triggers an event each time a state change results in one of the following conditions: <ul style="list-style-type: none"> Matching the event trigger definition No longer matching the event trigger definition 	<ul style="list-style-type: none"> EO.Eventxx.EventRisingEn EO.Eventxx.EventFallingEn 	<p>Values (EventRisingEn, EventFallingEn) represent:</p> <ul style="list-style-type: none"> (0, 0): Event trigger is disabled. (1, 0): Event triggers on transitions to match the configured pattern. (0, 1): Event triggers on transitions to not match the configured pattern. (1, 1): Event triggers on transitions to either match or not match the configured pattern.
Event Trigger Pattern – Participate in Event Trigger	Displays whether the specified condition participates in the Event trigger definition for the event.	<p>When Encoder Type is configured as HSC:</p> <ul style="list-style-type: none"> EO.Eventxx.InWindowxxSelect EO.Eventxx.ResetSelect EO.Eventxx.HoldSelect EO.Eventxx.LoadSelect EO.Eventxx.StoreSelect EO.Eventxx.RolloverSelect 	<ul style="list-style-type: none"> 0 = Condition does not participate in the Event trigger definition. 1 = Condition participates in the Event trigger definition.

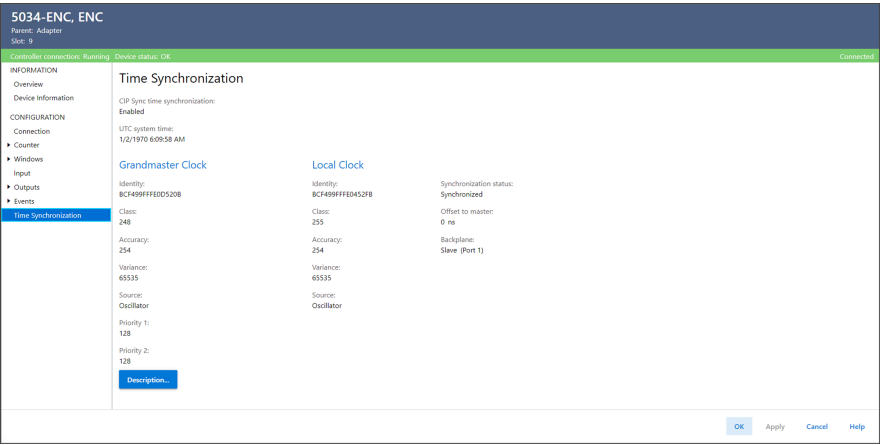
Parameter	Description	Member Name	Value
		<ul style="list-style-type: none"> EO.Eventxx.Rollunder Select EO.Eventxx.DirectionSelect <p>When Encoder Type is configured as SSI:</p> <ul style="list-style-type: none"> EO.Eventxx.InWindowxxSelect EO.Eventxx.StoreSelect EO.Eventxx.DirectionSelect 	
Event Trigger Pattern – State Transition	Displays the state of the condition to match after the transition.	<p>When Encoder Type is configured as HSC:</p> <ul style="list-style-type: none"> EO.Eventxx.InWindowxxValue EO.Eventxx.ResetValue EO.Eventxx.HoldValue EO.Eventxx.LoadValue EO.Eventxx.StoreValue EO.Eventxx.RolloverValue EO.Eventxx.RollunderValue EO.Eventxx.DirectionValue <p>When Encoder Type is configured as SSI:</p> <ul style="list-style-type: none"> EO.Eventxx.InWindowxxValue EO.Eventxx.StoreValue EO.Eventxx.DirectionValue 	<p>For InWindowxxValue:</p> <ul style="list-style-type: none"> 0 = Not in Window matches the event pattern 1 = In Window matches the event pattern <p>For other event tags:</p> <ul style="list-style-type: none"> 0 = Value of 0 matches the event pattern 1 = Value of 1 matches the event pattern

Time Synchronization View

The Time Synchronization view is read-only and displays the CIP Sync status information about the module when the project is online. Time Synchronization displays the following information:

- CIP Sync time synchronization status
- UTC system time
- Grandmaster Clock information
- Grandmaster Clock description
- Local Clock information

Figure 73. Time Synchronization View Example



Grandmaster Clock Description

The Grandmaster clock description dialog provides detailed information about the grandmaster clock.

To view the Grandmaster clock description, select Description in the Time Synchronization view.

Figure 74. Grandmaster Clock Description Example

Grandmaster clock description

User name:

test

User location:

Protocol address:

00.00.00.00

Physical address:

00-00-00-00

Clock type:

Boundary Clock

Manufacturer name:

Rockwell Automation

Model:

1756-L84ES/B_4584_BuildVerificat

Serial number:

01087894

Hardware revision:

3.1

Firmware revision:

36.11.4584

Software revision:

Profile identity:

00-21-6C-00-01-00

Physical protocol:

ControlBus

Network protocol:

Local

Port number:

1

Close

Help

Use the Status Indicators for Troubleshooting

- PointMax I/O modules use the following status indicators:
- SA Power Indicator – This indicator operates the same for all PointMax I/O modules.
 - Module Status Indicator – This indicator operates the same for all PointMax I/O modules.
 - Channel Status Indicator – This indicator operates differently based on the module type you are using.

SA Power Indicator

Table 22. Interpret SA Power Indicator

Indicator State	Description	Recommended Action
Off	The module is not powered or outside its designed operating range.	Complete the following actions: 1. Confirm that the system is powered. 2. Confirm that the module is installed properly.
Steady green	There is SA power to the module.	None
Steady red	There is no SA power to the module or SA power is not in the valid range.	Complete the following actions: 1. Confirm that SA power wiring is properly connected to the adapter, expansion power module, or 5034-MBSA mounting base. 2. Check the following: <ul style="list-style-type: none">Check that the SA voltage is in the correct range.If an external power supply is used, confirm that the power supply is turned on.Confirm that there is sufficient voltage supplied to the module.

Table 22. Interpret SA Power Indicator (continued)

Indicator State	Description	Recommended Action
		<ul style="list-style-type: none"> - Make sure that the mounting base to mounting base connection is properly secured. <p>Go to Chassis Information view in the Module Properties of the adapter to check the Field Power status of mounting bases installed in all slots.</p>

Module Status Indicator

Table 23. Interpret Module Status Indicator

Indicator State	Description	Recommended Action
Off	The module is not powered.	<p>Complete the following actions:</p> <ol style="list-style-type: none"> 1. Confirm that the system is powered. 2. 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	<p>One of the following conditions exist:</p> <ul style="list-style-type: none"> • The module has powered up successfully. • The module is OK, but it does not have a connection. <p>No connection can result from missing, incomplete, or incorrect module configuration.</p>	<p>Complete the following actions:</p> <ul style="list-style-type: none"> • Troubleshoot your Studio 5000 Logix Designer application to determine what is preventing 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.	<p>Complete the following actions:</p> <ol style="list-style-type: none"> 1. Cycle power to the module. 2. If the status indicator remains in the steady red state, replace the module.
Flashing red	<p>One of the following conditions exist:</p> <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. 	<p>Complete one of the following:</p> <ul style="list-style-type: none"> • Let the firmware update progress complete. • Reattempt a firmware update after one fails. • Use the Studio 5000 Logix Designer application to determine whether a module recoverable fault is occurred. It is shown in Device Information view. <p>To clear a recoverable fault, complete one of the following:</p> <ul style="list-style-type: none"> - Cycle module power. - Select Reset device in the Studio 5000 Logix Designer project via the Device Information view of the Module Properties dialog. <p>If the fault does not clear after cycling power and module reset, contact Rockwell Automation Technical Support.</p>

Table 23. Interpret Module Status Indicator (continued)

Indicator State	Description	Recommended Action
		<ul style="list-style-type: none"> Use the Studio 5000 Logix Designer application to determine if a connection has timed out. The Connection view in the Module Properties indicates whether a connection request 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.
Flashing red/green	The module is in a self-test mode.	None

Channel Status Indicator

Table 24. HSC-A/SSI Channel Status Indicator

Indicator State	Description	Recommended Action
Off	In HSC mode: <ul style="list-style-type: none"> The input signal is Off. In SSI mode: <ul style="list-style-type: none"> The module is commanded to stop retrieving SSI data. 	One of the following: <ul style="list-style-type: none"> Confirm that the input is configured properly. Confirm that there is backplane power supplied through the EtherNet/IP adapter.
Steady yellow	In HSC mode: <ul style="list-style-type: none"> The input signal is On. In SSI mode: <ul style="list-style-type: none"> The module is commanded to retrieve SSI data. 	None
Flashing red	A Field Power Loss condition exists.	Locate and correct the cause of field power loss condition.

Table 25. HSC-B and HSC-Z Status Indicators

Indicator State	Description	Recommended Action
Off	One of the following: <ul style="list-style-type: none"> The input signal is Off. The module is in SSI mode. 	One of the following: <ul style="list-style-type: none"> Confirm that the input is configured properly. Confirm that the correct mode is selected.
Steady yellow	The input signal is On.	None
Flashing red	A Field Power Loss condition exists.	Locate and correct the cause of field power loss condition.

Table 26. Digital Output Channel Status Indicators

Indicator State	Description	Recommended Action
Off	The output is Off.	One of the following: <ul style="list-style-type: none"> Confirm that the output is configured properly. Confirm that there is backplane power supplied through the EtherNet/IP adapter.
Steady yellow	The output is On.	None
Flashing red	One of the following: <ul style="list-style-type: none"> A Field Power Loss condition exists. A Short Circuit condition is detected. 	One of the following: <ul style="list-style-type: none"> Locate and correct the cause of field power loss condition. Locate and correct the cause of Short Circuit condition.

Table 27. Digital Input Channel Status Indicator

Indicator State	Description	Recommended Action
Off	One of the following: <ul style="list-style-type: none"> The input is Off. A Field Power Loss condition exists. 	One of the following: <ul style="list-style-type: none"> Confirm that the input is configured properly. Confirm that there is backplane power supplied through the EtherNet/IP adapter. Locate and correct the cause of field power loss condition.
Steady yellow	The input is On.	None

Module Tag Definitions

Module tags are created when you add a module to the Studio 5000 Logix Designer application project.

The set of module tags associated with a module depends on the module type and Module Definition choices made during module configuration.

The tables contained in this section list all of the tags available with a module. Not all tags in the list are used when that module type is added to a project. Tag use varies by module configuration.

Configuration Tag Definitions

Table 28. Configuration Tags

Name	Data Type	Definition	Valid Values
Windowxx.CounterSelect	SINT	Counter with which the window is associated.	0 = Counter 0
Windowxx.OutputxxSelect	BOOL	Window controls the indicated output, that is, any of Output00...Output01 If a Window controls the output, when the selected HSC/SSI data is in the window, the output is On.	<ul style="list-style-type: none"> 0 = Window does not control output 1 = Window controls output

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
		For the definition of "in the window", see Control Outputs with Windows on page .	
Counter00.InputOffOnFilterA	SINT	The amount of time that a signal must be in the On state before the Input A data indicates the On state	<ul style="list-style-type: none"> • 0 = 0 ns • 1 = 100 ns • 2 = 200 ns • 3 = 500 ns • 4 = 1 μs • 5 = 2 μs • 6 = 5 μs • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms • 17 = 20 ms • 18 = 50 ms
Counter00.InputOnOffFilterA	SINT	The amount of time that a signal must be in the Off state before the Input A data indicates the Off state	<ul style="list-style-type: none"> • 0 = 0 ns • 1 = 100 ns • 2 = 200 ns • 3 = 500 ns • 4 = 1 μs • 5 = 2 μs • 6 = 5 μs • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.InputOffOnFilterAB (Quadrature)	SINT	The amount of time that a signal must be in the On state before the Input AB data indicates the On state	<ul style="list-style-type: none"> 0 = 0 ns 1 = 100 ns 2 = 200 ns 3 = 500 ns 4 = 1 μs 5 = 2 μs 6 = 5 μs 7 = 10 μs 8 = 20 μs 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.InputOnOffFilterAB (Quadrature)	SINT	The amount of time that a signal must be in the Off state before the Input AB data indicates the Off state	<ul style="list-style-type: none"> 0 = 0 ns 1 = 100 ns 2 = 200 ns 3 = 500 ns 4 = 1 μs 5 = 2 μs 6 = 5 μs 7 = 10 μs 8 = 20 μs

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.InputOffOnFilterB	SINT	The amount of time that a signal must be in the On state before the Input B data indicates the On state	<ul style="list-style-type: none"> 0 = 0 ns 1 = 100 ns 2 = 200 ns 3 = 500 ns 4 = 1 μs 5 = 2 μs 6 = 5 μs 7 = 10 μs 8 = 20 μs 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.InputOnOffFilterB	SINT	The amount of time that a signal must be in the Off state before the Input B data indicates the Off state	<ul style="list-style-type: none"> 0 = 0 ns 1 = 100 ns 2 = 200 ns 3 = 500 ns 4 = 1 μs

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> • 5 = 2 μs • 6 = 5 μs • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms • 17 = 20 ms • 18 = 50 ms
Counter00.InputOffOnFilterZ	SINT	The amount of time that a signal must be in the On state before the Input Z data indicates the On state	<ul style="list-style-type: none"> • 0 = 0 ns • 1 = 100 ns • 2 = 200 ns • 3 = 500 ns • 4 = 1 μs • 5 = 2 μs • 6 = 5 μs • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 17 = 20 ms 18 = 50 ms
Counter00.InputOnOffFilterZ	SINT	The amount of time that a signal must be in the Off state before the Input Z data indicates the Off state	<ul style="list-style-type: none"> 0 = 0 ns 1 = 100 ns 2 = 200 ns 3 = 500 ns 4 = 1 μs 5 = 2 μs 6 = 5 μs 7 = 10 μs 8 = 20 μs 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.InputOffOnFilterTrigger	SINT	<p>The amount of time that a signal must be in the On state before the trigger input data indicates the On state</p> <p>Only applicable when the standard digital input is selected as the trigger of one or more counter functions (that is, Store, Reset, Load, Hold).</p>	<ul style="list-style-type: none"> 0 = 0 ns 7 = 10 μs 8 = 20 μs 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 17 = 20 ms 18 = 50 ms
Counter00.InputOnOffFilterTrigger	SINT	<p>The amount of time that a signal must be in the Off state before the trigger input data indicates the Off state</p> <p>Only applicable when the standard digital input is selected as the trigger of one or more counter functions (that is, Store, Reset, Load, Hold).</p>	<ul style="list-style-type: none"> 0 = 0 ns 7 = 10 μs 8 = 20 μs 9 = 50 μs 10 = 100 μs 11 = 200 μs 12 = 500 μs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Counter00.AvgOverPulses	INT	<p>The number of pulses to average when calculating average frequency, pulse width, and acceleration</p> <p>Only applicable when AvgOverPulsesEn is true.</p>	1...32,767
Counter00.AvgOverTime	INT	<p>The duration in milliseconds to average when calculating average frequency, pulse width, and acceleration</p> <p>Only applicable when AvgOverPulsesEn is false.</p>	2...1000
Counter00.InvertInputA	BOOL	Invert Input A	<ul style="list-style-type: none"> 0 = Do not invert Input A 1 = Invert Input A
Counter00.InvertInputAB (Quadrature)	BOOL	Invert Input AB	<ul style="list-style-type: none"> 0 = Do not invert Input AB 1 = Invert Input AB
Counter00.InvertInputB	BOOL	Invert Input B	<ul style="list-style-type: none"> 0 = Do not invert Input B 1 = Invert Input B
Counter00.InvertInputZ	BOOL	When Input Z is selected as the trigger for a counter function (Store, Load, Hold, Reset), this tag determines	<ul style="list-style-type: none"> 0 = A rising edge triggers the function 1 = A falling edge triggers the function

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
		whether a rising edge or falling edge triggers the function.	
Counter00.InvertDirectionInput	BOOL	Determines whether to invert the direction of the counter.	<ul style="list-style-type: none"> 0 = Do not invert the counter direction 1 = Invert the counter direction
Counter00.InvertInputTrigger	BOOL	When Input00 is selected as the trigger for a counter function (Store, Load, Hold, Reset), this tag determines whether a rising edge or falling edge triggers the function.	<ul style="list-style-type: none"> 0 = A rising edge triggers the function 1 = A falling edge triggers the function
Counter00.ZeroFrequencyAlarmLatchEn	BOOL	Latches the Zero Frequency Alarm when set so that it does not clear until explicitly unlatched.	<ul style="list-style-type: none"> 0 = Latching is disabled 1 = Latching is enabled
Counter00.ResetTimeDerivedValues	BOOL	Indicates that a Reset sets frequency and average frequency, pulse width, and acceleration to zero.	<ul style="list-style-type: none"> 0 = Reset does not reset Time-derived values 1 = Reset resets Time-derived values
Counter00.MissingPulseAlarmLatchEn	BOOL	Latches the Missing Pulse Alarm when set so that it does not clear until explicitly unlatched.	<ul style="list-style-type: none"> 0 = Latching is disabled 1 = Latching is enabled
Counter00.AvgOverPulsesEn	BOOL	Determines whether to average over the number of pulses or time when calculating average frequency, pulse width, and acceleration.	<ul style="list-style-type: none"> 0 = Average over time 1 = Average over pulses
Counter00.ProgramToFaultEn	BOOL	<p>Determines counter action if a connection faults while the module is in a Program state.</p> <p>The counter can remain in Program Mode or transition to a safe state for Communication Fault Mode.</p>	<ul style="list-style-type: none"> 0 = Stay in Program Mode 1 = Go to Communication Fault Mode
Counter00.ProgModeStoredValuesReset	BOOL	Determines whether to reset values for Stored tags when the controller transitions to Program Mode or Inhibit mode.	<ul style="list-style-type: none"> 0 = Do not reset the values for Stored tags 1 = Reset the values for Stored tags
Counter00.FaultModeStoredValuesReset	BOOL	Determines whether to reset values for Stored tags for the counter when the controller transitions to Communication Fault Mode.	<ul style="list-style-type: none"> 0 = Do not reset the values for Stored tags 1 = Reset the values for Stored tags
Counter00.Scaling	REAL	Scaled Count, Scaled Frequency, and Scaled Acceleration values in engineering units are calculated by multiplying the values in pulses by Scaling.	Any positive value
Counter00.FrequencyAlarmLimit	REAL	<p>Maximum frequency allowed before the Frequency alarm is triggered.</p> <p>If the instantaneous or average frequency is greater than this value, the Frequency Alarm is triggered.</p>	0.1...1,000,000 Hz
Counter00.PulseWidthAlarmLimit	REAL	Maximum pulse width allowed before the Pulse Width alarm is triggered.	0.125...10,000,000 μ s

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
		If the instantaneous or average pulse width exceeds the maximum pulse width, the Pulse Width alarm is triggered	
Counter00.AccelAlarmLimit	REAL	Maximum acceleration rate allowed before the Acceleration alarm is triggered. If the instantaneous or average acceleration rate exceeds the maximum acceleration rate, the Acceleration alarm is triggered.	Any positive value
Counter00.DecelAlarmLimit	REAL	Maximum deceleration rate allowed before the Deceleration alarm is triggered. If the instantaneous or average acceleration rate is less than the maximum deceleration rate, the Deceleration alarm is triggered.	Any positive value
Counter00.ProgMode	SINT	Determines counter action when the controller transitions to Program Mode or the connection to the module is inhibited.	<ul style="list-style-type: none"> 0 = Continue Counting Continue counting when the controller transitions to Program Mode or the connection to the module is inhibited. 1 = Hold Retain the last count and resume counting from this count when the controller transitions to Run Mode. 2 = User-defined Value Set the count to the user-defined value in ProgValue and resume counting from this value when the controller transitions to Run.
Counter00.FaultMode	SINT	Determines counter action when a connection fault occurs.	<ul style="list-style-type: none"> 0 = Continue Counting Continue counting when a connection fault occurs. 1 = Hold Retain the last count and resume counting from this count when the controller transitions to Run Mode. 2 = User-defined Value Set the count to the user-defined value in FaultValue and resume counting from this value when the controller transitions to Run Mode.
Counter00.ProgValue	SINT	The user-defined value to use when ProgMode is set to User-defined Value.	Any value

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
Counter00.FaultValue	SINT	The user-defined value to use when FaultMode is set to User-defined Value.	Any value
SSI00.Disable	BOOL	Enables or disables the SSI input.	<ul style="list-style-type: none"> 0 = SSI is enabled 1 = SSI is disabled
SSI00.CodeType	BOOL	Determines the code type of the SSI data.	<ul style="list-style-type: none"> 0 = SSI data is in Binary format 1 = SSI data is in Gray format <p>I.SSI00.Data contains the value after Gray to Binary conversion and I.SSI00.RawData contains the original data received.</p>
SSI00.InvertStoreInput	BOOL	Inverts the active state of the digital input that is used as the Store function trigger.	<ul style="list-style-type: none"> 0 = A rising edge triggers the function 1 = A falling edge triggers the function
SSI00.WordBitLength	SINT	Determines the length of the received serial data word in bits.	2...31 data bits
SSI00.BaudRate	SINT	Determines the speed of the SSI data clock.	<ul style="list-style-type: none"> 7 = 125 Kbps 8 = 250 Kbps 9 = 500 Kbps 10 = 1 Mbps 11 = 2 Mbps
SSI00.WordFilterControl	SINT	Determines the corresponding number of successive equal SSI data words that the module receives from the SSI sensor to update the realtime present SSI data word.	<ul style="list-style-type: none"> 1 = Off 2 = Low (2 successive values) 3 = Med (3 successive values) 4 = High (4 successive values) 5 = Max (5 successive values)
SSI00.TrailingBitLength	SINT	Determines the number of bits to shift right in SSI data.	0...16 bits
SSI00.DelayTime	DINT	Determines the delay time in microseconds between successive SSI data words.	16...65,535 µs
SSI00.StoreInputOffOnFilter	SINT	<p>Determines the amount of time that a signal must be in the On state before the Store function trigger input data indicates the On state.</p> <p>Only applicable when the standard digital input is selected as the Store function trigger.</p>	<ul style="list-style-type: none"> 0 = 0 ns 7 = 10 µs 8 = 20 µs 9 = 50 µs 10 = 100 µs 11 = 200 µs

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms • 17 = 20 ms • 18 = 50 ms
SSI00.StoreInputOnOffFilter	SINT	<p>Determines the amount of time that a signal must be in the Off state before the Store function trigger input data indicates the Off state.</p> <p>Only applicable when the standard digital input is selected as the Store function trigger.</p>	<ul style="list-style-type: none"> • 0 = 0 ns • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms • 17 = 20 ms • 18 = 50 ms
Input00.InputOffOnFilter	SINT	Determines the amount of time that a signal must be in the On state before the input data indicates the On state.	<ul style="list-style-type: none"> • 0 = 0 ns • 7 = 10 μs • 8 = 20 μs • 9 = 50 μs • 10 = 100 μs • 11 = 200 μs • 12 = 500 μs • 13 = 1 ms • 14 = 2 ms • 15 = 5 ms • 16 = 10 ms

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 17 = 20 ms 18 = 50 ms
Input00.InputOnOffFilter	SINT	Determines the amount of time that a signal must be in the Off state before the input data indicates the Off state.	<ul style="list-style-type: none"> 0 = 0 ns 7 = 10 µs 8 = 20 µs 9 = 50 µs 10 = 100 µs 11 = 200 µs 12 = 500 µs 13 = 1 ms 14 = 2 ms 15 = 5 ms 16 = 10 ms 17 = 20 ms 18 = 50 ms
Outputxx.FaultMode	BOOL	<p>Determines the output action when a connection fault occurs.</p> <p>At the fault occurrence, the output holds its last state or transitions to the value set in the Fault Value parameter.</p> <p>The channel continues the Fault Mode for the length of time set in the Fault Value State Duration parameter.</p> <hr/> <p>IMPORTANT: FaultModeLocalControl takes precedence over this tag.</p> <hr/>	<ul style="list-style-type: none"> 0 = Transition to user-defined value 1 = Hold Last State
Outputxx.FaultValue	BOOL	<p>Value to which the output changes if the following events exist:</p> <ul style="list-style-type: none"> Fault Mode = 0 Either of the following: <ul style="list-style-type: none"> Controller is in Run mode and the connection is lost Controller is in Program mode, the connection is lost, and the ProgamToFaultEn tag is set 	<ul style="list-style-type: none"> 0 = Off 1 = On
Outputxx.ProgMode	BOOL	Determines output action when the controller transitions to Program mode or the connection to the module is inhibited.	<ul style="list-style-type: none"> 0 = Transition to user-defined value 1 = Hold Last State

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
		<p>At the transition to Program mode, the output holds its last state or transitions to the value set in the Program Value parameter.</p> <hr/> <p>IMPORTANT: ProgModeLocalControl takes precedence over this tag.</p> <hr/>	
Outputxx.ProgValue	BOOL	<p>Value to which the output changes if the following events exist:</p> <ul style="list-style-type: none"> • Program Mode = 0 • Module transitions to Program mode 	<ul style="list-style-type: none"> • 0 = Off • 1 = On
Outputxx.FaultFinalState	BOOL	<p>Value to which the output changes if the following events exist:</p> <ul style="list-style-type: none"> • Connection is lost • Time that is defined by the Fault State Duration parameter has been exceeded 	<ul style="list-style-type: none"> • 0 = Off • 1 = On
Outputxx.ProgramToFaultEn	BOOL	<p>Determines output action if a connection faults while the module is in a Program state.</p> <p>The output can remain in a Program mode or transition to a safe state for Fault mode.</p> <p>If the output remains in a Program mode, the Final Fault State parameter is ignored.</p>	<ul style="list-style-type: none"> • 0 = Stay in Program Mode • 1 = Go to Communication Fault Mode

Table 28. Configuration Tags (continued)

Name	Data Type	Definition	Valid Values
Outputxx.NoLoadEn	BOOL	No Load Diagnostics is not supported.	0 = Disable
Outputxx.FaultValueStateDuration	SINT	Determines the length of time the Communication Fault Mode state is held before the FaultFinalState is applied.	<ul style="list-style-type: none"> 0 = Hold forever 1, 2, 5, or 10 seconds
Outputxx.FaultModeLocalControl	BOOL	Determines whether the encoder controls the output when a connection fault occurs.	<ul style="list-style-type: none"> 0 = Encoder control is disabled 1 = Encoder control is enabled
Outputxx.ProgModeLocalControl	BOOL	Determines whether the encoder controls the output when the controller transitions to Program mode or the connection to the module is inhibited.	<ul style="list-style-type: none"> 0 = Encoder control is disabled 1 = Encoder control is enabled
Outputxx.TriggerSelect	BOOL	Specifies the input that triggers the pulse function.	<ul style="list-style-type: none"> -1 = No external trigger is selected. However, you can use O.Outputxx.Trigger tag to trigger the pulse. 0 = Input00 activates the output pulse. Input00 is only available when it is not used as a HSC counter function trigger or SSI store trigger.
Outputxx.InvertTrigger	BOOL	Indicates whether the output pulse is triggered on the rising or falling edge.	<ul style="list-style-type: none"> 0 = Output pulse is triggered on the rising edge. 1 = Output pulse is triggered on the falling edge.
Pwmxx.Enable	BOOL	Determines whether the output is Pulse Width Modulation or Digital Output.	<ul style="list-style-type: none"> 0 = Point is used as Digital Output 1 = Point is used as Pulse Width Modulation Output
Pwmxx.CycleCountEn	BOOL	Indicates whether to stop generating pulse cycles after a defined number or continue generating pulse cycles until stopped.	<ul style="list-style-type: none"> 0 = Continue generating pulse cycles until stopped 1 = Generate a defined number of pulse cycles
Pwmxx.ExecuteAllCycles	BOOL	Determines the PWM action when O.Outputxx.Data transitions to 0. Only applicable when C.Pwmxx.CycleCountEn = 1.	<ul style="list-style-type: none"> 0 = Stop generating pulses immediately 1 = Continue generating pulses until O.Pwmxx.CycleCountLimit is reached
Pwmxx.EnforcePTODutyCycle	BOOL	Indicates whether to enforce the PTO specific duty cycle, which is fixed at 50%.	<ul style="list-style-type: none"> 0 = O.Pwmxx.DutyCycle is used 1 = 50% duty cycle is used. O.Pwmxx.DutyCycle is ignored.
Pwmxx.IdleState	BOOL	Determines the state of the output when PWM is idle.	<ul style="list-style-type: none"> 0 = When PWM is idle, the output is Off. It also means that when PWM is running, the active state per O.Pwmxx.DutyCycle is the output in On state. 1 = When PWM is idle, the output is On. It also means that when PWM is running, the active state per O.Pwmxx.DutyCycle is the output in Off state.

Input Tag Definitions

Table 29. Input Tags

Name	Data Type	Definition	Valid Values
RunMode	BOOL	The module's operating state	<ul style="list-style-type: none"> 0 = Idle – It means one of the following: <ul style="list-style-type: none"> Connection is not up. Connection has been opened but the module has not started producing data for the connection. Because the controller is in Program Mode, the module is not applying new output tag data but is applying the Program Mode state of the channel instead. 1 = Run – It means the following: <ul style="list-style-type: none"> Connection is up. Module is producing data for the connection. Output tag data is being applied.
ConnectionFaulted	BOOL	Indicates if a connection to the target is running. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.	<ul style="list-style-type: none"> 0 = Connection is running. 1 = Connection is not running.
DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics are active. 1 = One or more diagnostics are active or the prognostics threshold is reached.
DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected.	-128...+127 The value of 0 is skipped except during module powerup.
Windowxx.InWindow	BOOL	Indicates if the selected HSC/SSI data is in the specified window. For the definition of "in the window", see Control Outputs with Windows on page .	<ul style="list-style-type: none"> 0 = Value is outside the specified window 1 = Value is inside the specified window
Windowxx.NotANumber	BOOL	Indicates that the last value received for the window data was not a number. In this case, the last valid window data is used.	<ul style="list-style-type: none"> 0 = Last window data received was a number 1 = Last window data received was not a number
Counter00.Fault	BOOL	Indicates that counter data is inaccurate and cannot be trusted for use in the application. If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
		<p>IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>	<p>Typically, this tag is set when the Counterxx.QuadratureErrorCount tag is any number other than 0.</p> <p>The Counterxx.QuadratureErrorCount tag only applies when the module is operating in the X1, X2, or X4 Quadrature mode.</p>
Counter00.Uncertain	BOOL	<p>Indicates that the counter data can be inaccurate but the degree of inaccuracy is not known.</p> <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>	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data <p>The following are causes of uncertain data:</p> <ul style="list-style-type: none"> Measured frequency is too high. That is, the Measured frequency $> 1.2 * f_{MAX}$ ($f_{MAX} = 1$ Mhz). Measured period is too long. That is, the Measured period $> T_{P-MAX}$. <p>Where $T_{P-MAX} = 1/(\text{Input pulse frequency, min})$</p> <p>If the period is longer than T_{P-MAX}, the module assumes that the input frequency is 0.</p> <ul style="list-style-type: none"> Measured pulse width is too short. That is, the pulse width is less than the module's rated minimum pulse width, T_{PW-MIN} ($T_{PW-MIN} = 125$ ns). If duration of the number of pulses that is configured in the Counter00:AveragOverPulses tag is greater than T_{P-MAX}. <p>In this case, the average frequency and average pulse width are reported based on the length of the complete and incomplete pulses that are measured and the Counter00.PartialAvgFrequency tag is set.</p>
Counter00.Direction	BOOL	Indicates the count direction.	<ul style="list-style-type: none"> 0 = Count up 1 = Count down
Counter00.StoredDirection	BOOL	Indicates the count direction when the Store function is triggered.	<ul style="list-style-type: none"> 0 = Count up 1 = Count down
Counter00.RevolutionCount	INT	<p>The number of Rollover and Rollunder transitions</p> <p>This value increments when a Rollover occurs and decrements when a Rollunder occurs.</p>	Any value

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
Counter00.Count	DINT	The value of the counter The counter mode that is selected determines how the input transitions are counted.	Any value
Counter00.Frequency	REAL	Frequency of the input transitions measured by the counter This frequency is measured on Input A except in the Up/Down mode. For more information, see the PointMax High-speed Counter/SSI encoder user manual, publication 5034-UM006 .	Any value
Counter00.PulseWidth	REAL	On-state pulse width of the last pulse received measured in microseconds when the input is not inverted. Off-state pulse width of the last pulse received in microseconds when the input is inverted. This pulse width is measured on Input A except in Up/Down mode. For more information, see the PointMax High-speed Counter/SSI encoder user manual, publication 5034-UM006 .	Any value
Counter00.Accel	REAL	The acceleration of the pulses measured by the counter This acceleration is measured on Input A except in Up/Down mode. For more information, see the PointMax High-speed Counter/SSI encoder user manual, publication 5034-UM006 .	Any value
Counter00.FrequencyAvg	REAL	The average frequency	Any value
Counter00.PulseWidthAvg	REAL	The average pulse width measured in microseconds	Any value
Counter00.AccelAvg	REAL	The average acceleration	Any value
Counter00.ScaledCount	REAL	The count value that is scaled to engineering units	Any value
Counter00.ScaledFrequency	REAL	The frequency value that is scaled to engineering units	Any value
Counter00.ScaledAccel	REAL	The acceleration value that is scaled to engineering units	Any value
Counter00.ScaledFrequencyAvg	REAL	The average frequency that is scaled to engineering units	Any value
Counter00.ScaledAccelAvg	REAL	The average acceleration value that is scaled to engineering units	Any value

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
Counter00.StoredRevolutionCount	INT	The Revolution Count value when the Store function is triggered	Any value
Counter00.StoredCount	DINT	The count value when the Store function is triggered	Any value
Counter00.StoredFrequency	REAL	The frequency value when the Store function is triggered	Any value
Counter00.StoredPulseWidth	REAL	The pulse width value when the Store function is triggered	Any value
Counter00.StoredAccel	REAL	The acceleration when the Store function is triggered	Any value
Counter00.ScaledStoredCount	REAL	The stored count value that is scaled to engineering units	Any value
Counter00.ScaledStoredFrequency	REAL	The stored frequency that is scaled to engineering units	Any value

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
Counter00.ScaledStoredAccel	REAL	The stored acceleration that is scaled to engineering units	Any value
Counter00.Rollover	BOOL	Indicates that the counter counted up to Rollover value and then has continued counting from the Rollunder.	<ul style="list-style-type: none"> 0 = Counter has not counted to the Rollover value and continued counting from the Rollunder value 1 = Counter counted to the Rollover value and has continued counting from the Rollunder value
Counter00.Rollunder	BOOL	Indicates that the counter counted down to Rollunder value and then has continued counting from the Rollover.	<ul style="list-style-type: none"> 0 = Counter has not counted down to the Rollunder value and continued counting from the Rollover value 1 = Counter has counted down to the Rollunder value and continued counting from the Rollover value
Counter00.RolloverLeqRollunder	BOOL	Indicates that the Rollover value is less than or equal to the Rollunder value and the module is using either the last valid or default Rollover and Rollunder values.	<ul style="list-style-type: none"> 0 = Rollover value is greater than the Rollunder value 1 = Rollover value is less than or equal to the Rollunder value
Counter00.FrequencyOvrerrange	BOOL	Indicates that the counter frequency exceeds the module's maximum rated frequency, and therefore cannot be tracking the signal properly.	<ul style="list-style-type: none"> 0 = Counter frequency does not exceed the module's maximum rated frequency 1 = Counter frequency exceeds the module's maximum rated frequency
Counter00.PartialAvgFrequency	BOOL	<p>Indicates that the counter's average frequency is complete or partial with respect to Counter00.AvgOverPulses.</p> <p>Average frequency can be partial when the number of pulses received in the maximum time window (T_{P-MAX}) is less than the required number of pulses for averaging (Counter00.AvgOverPulses).</p> <p>Only applicable when Counter00.AvgOverPulsesEn = 0.</p>	<ul style="list-style-type: none"> 0 = Counter's average frequency is complete and is based on the complete number of pulses with respect to Counter00.AvgOverPulses 1 = Counter's average frequency is partial and is based on the length of the complete and incomplete pulses that are measured within the T_{P-MAX} time window
Counter00.PartialAvgPulseWidth	BOOL	<p>Indicates that the counter's average pulse width is complete or partial with respect to Counter00.AvgOverPulses.</p> <p>Average pulse width can be partial when the number of pulses received in the maximum time window (T_{P-MAX}) is less than the required number of pulses for averaging (Counter00.AvgOverPulses).</p>	<ul style="list-style-type: none"> 0 = Counter's average pulse width is complete and is based on the complete number of pulses with respect to Counter00.AvgOverPulses 1 = Counter's average pulse width is partial and is based on the length of the complete and incomplete pulses that are measured within the T_{P-MAX} time window

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
		Only applicable when Counter00.AvgOverPulsesEn = 1.	
Counter00.MissingPulseAlarm	BOOL	Indicates that the period of a pulse was greater than the missing pulse timeout or was never completed.	<ul style="list-style-type: none"> 0 = Period of the pulse was normal 1 = Period of the pulse was greater than the missing pulse timeout or was never completed
Counter00.ZeroFrequencyAlarm	BOOL	Counter frequency is less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	<ul style="list-style-type: none"> 0 = Counter frequency is not less than the Zero Frequency Alarm limit. 1 = Counter frequency is less than the Zero Frequency Alarm limit.
Counter00.ZeroFrequencyAvgAlarm	BOOL	Counter average frequency is less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	<ul style="list-style-type: none"> 0 = Counter average frequency is not less than the Zero Frequency Alarm limit. 1 = Counter average frequency is less than the Zero Frequency Alarm limit.
Counter00.FrequencyAlarm	BOOL	Counter frequency has exceeded the Frequency Alarm limit.	<ul style="list-style-type: none"> 0 = Counter frequency has not exceeded the Frequency Alarm limit. 1 = Counter frequency has exceeded the Frequency Alarm limit.
Counter00.FrequencyAvgAlarm	BOOL	Counter average frequency has exceeded the Frequency Alarm limit.	<ul style="list-style-type: none"> 0 = Counter average frequency has not exceeded the Frequency Alarm limit. 1 = Counter average frequency has exceeded the Frequency Alarm limit.
Counter00.PulseWidthAlarm	BOOL	Indicates if the counter pulse width is greater than the maximum pulse width established by the Pulse Width Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered That is, the counter pulse width is not greater than the Pulse Width Alarm limit. 1 = Alarm is triggered That is, the counter pulse width is greater than the Pulse Width Alarm limit.
Counter00.PulseWidthAvgAlarm	BOOL	Indicates if the counter average pulse width is greater than the maximum pulse width established by the Pulse Width Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered That is, the counter average pulse width is not greater than the Pulse Width Alarm limit. 1 = Alarm is triggered That is, the counter average pulse width is greater than the Pulse Width Alarm limit.
Counter00.AccelAlarm	BOOL	Indicates that the counter acceleration rate exceeds the Acceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered That is, the counter acceleration rate does not exceed the Acceleration Alarm limit.

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 1 = Alarm is triggered <p>That is, the counter acceleration rate exceeds the Acceleration Alarm limit.</p>
Counter00.AccelAvgAlarm	BOOL	Indicates the counter average acceleration rate exceeds the Acceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered <p>That is, the counter average acceleration rate does not exceed the Acceleration Alarm limit.</p> <ul style="list-style-type: none"> 1 = Alarm is triggered <p>That is, the counter average acceleration rate exceeds the Acceleration Alarm limit.</p>
Counter00.DecelAlarm	BOOL	Indicates that the counter deceleration rate exceeds the Deceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered <p>That is, the counter deceleration rate does not exceed the Deceleration Alarm limit.</p> <ul style="list-style-type: none"> 1 = Alarm is triggered <p>That is, the counter deceleration rate exceeds the Deceleration Alarm limit.</p>
Counter00.DecelAvgAlarm	BOOL	Indicates that the counter average deceleration rate exceeds the Deceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm is not triggered <p>That is, the counter average deceleration rate does not exceed the Deceleration Alarm limit.</p> <ul style="list-style-type: none"> 1 = Alarm is triggered <p>That is, the counter average deceleration rate exceeds the Deceleration Alarm limit.</p>
Counter00.NotANumber	BOOL	Indicates that the last value received for the counter data was not a number. In this case, the last valid counter data is used.	<ul style="list-style-type: none"> 0 = Last counter data received was a number 1 = Last counter data received was not a number
Counter00.DataA	BOOL	Indicates the current Input A value. If O.OverrideDataAEn = 1, this tag reflects O.OverrideDataAValue.	<ul style="list-style-type: none"> 0 = Input A is Off 1 = Input A is On
Counter00.DataB	BOOL	Indicates the current Input B value. If O.OverrideDataBEn = 1, this tag reflects O.OverrideDataBValue.	<ul style="list-style-type: none"> 0 = Input B is Off 1 = Input B is On
Counter00.DataZ	BOOL	Indicates the current Input Z value.	<ul style="list-style-type: none"> 0 = Input Z is Off 1 = Input Z is On

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
		If O.OverrideDataZEn = 1, this tag reflects O.OverrideDataZValue.	
Counter00.DataAOverridden	BOOL	Input A data is overridden with O.OverrideDataAValue.	<ul style="list-style-type: none"> 0 = Input A data is not overridden 1 = Input A data is overridden
Counter00.DataBOverridden	BOOL	Input B data is overridden with the O.OverrideDataBValue.	<ul style="list-style-type: none"> 0 = Input B data is not overridden 1 = Input B data is overridden
Counter00.DataZOverridden	BOOL	Input Z data is overridden with the O.OverrideDataZValue.	<ul style="list-style-type: none"> 0 = Input Z data is not overridden 1 = Input Z data is overridden
Counter00.DataTrigger	BOOL	Indicates the current data value of the standard digital input, which is used as a trigger for counter functions. If O.OverrideDataTriggerEn = 1, this tag reflects the O.OverrideDataTriggerValue.	<ul style="list-style-type: none"> 0 = Input trigger is Off 1 = Input trigger is On
Counter00.DataTriggerOverridden	BOOL	DataTrigger data is overridden with O.OverrideDataTriggerValue.	<ul style="list-style-type: none"> 0 = Input trigger data was not overridden 1 = Input trigger data was overridden
Counter00.QuadratureErrorCount	SINT	Indicates the number of times that a quadrature counter error occurred.	Any value
Counter00.CountChangeIndicator	SINT	At every RPI, if the count value changes, the CountChangeIndicator value is incremented and reported.	Any value
SSI00.Fault	BOOL	Indicates that SSI data is inaccurate and cannot be trusted for use in the application. 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.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault
SSI00.Uncertain	BOOL	Indicates that the SSI data can be inaccurate but the degree of inaccuracy is not known. 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.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
SSI00.Data	DINT	Indicates the value of the data that is received from the SSI encoder, which is processed by the WordBitLength and TrailingBitLength tags.	Any value
SSI00.StoredData	DINT	Indicates the value of the data when the last Store function is triggered.	Any value
SSI00.RawData	DINT	Indicates the value of the raw data that is received from the SSI encoder.	Any value
SSI00.StoredRawData	DINT	Indicates the value of the raw data when the last Store function is triggered.	Any value
SSI00.Direction	BOOL	Indicates the direction of the encoder.	<ul style="list-style-type: none"> 0 = Moving up 1 = Moving down
SSI00.StoredDirection	BOOL	Indicates the direction of the encoder that is stored together with the data.	<ul style="list-style-type: none"> 0 = Moving up 1 = Moving down
SSI00.StoreInputData	BOOL	Indicates the current data of the digital Input when it is used as a Store function trigger. When no digital input is used as a Store function trigger, this member is always 0.	<ul style="list-style-type: none"> 0 = Off 1 = On
SSI00.StoreInputOverridden	BOOL	Indicates the override status of the digital input when it is used as a Store function trigger. When no digital input is used as a Store function trigger, this member is always 0.	<ul style="list-style-type: none"> 0 = Not overridden 1 = Overridden
Input00.Data	BOOL	Indicates the current Input00 value.	<ul style="list-style-type: none"> 0 = On 1 = Off
Input00.Fault	BOOL	Indicates that input data is inaccurate and cannot be trusted for use in the application. 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.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault <p>Typically, this tag is set when SA power loss is detected.</p>
Input00.Uncertain	BOOL	Indicates that the input data can be inaccurate but the degree of inaccuracy is not known. If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
		<p>IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.</p>	
Outputxx.Data	BOOL	Echo of the current output value	<ul style="list-style-type: none"> 0 = On 1 = Off
Outputxx.Fault	BOOL	<p>Indicates that output data is inaccurate and cannot be trusted for use in the application.</p> <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>	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault
Outputxx.Uncertain	BOOL	<p>Indicates that the output data can be inaccurate but the degree of inaccuracy is not known.</p> <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>	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data
Outputxx.TriggerMissed	BOOL	Indicates that at least one trigger is missed.	<ul style="list-style-type: none"> 0 = No trigger is missed. 1 = At least one trigger is missed because the delay and pulse of the previous trigger are not yet completed. <p>The output is not possible due to invalid PulseDelay or PulseActiveTime.</p>
EventStatus[x].EventDropped	BOOL	Indicates when an event has been discarded because events are occurring faster than they are being acknowledged.	<ul style="list-style-type: none"> 0 = An event status has not been dropped 1 = An event status has been dropped
EventStatus[x].CIPSyncValid	BOOL	<p>Indicates whether the module is synced with a 1588 master.</p> <p>A set bit alone cannot indicate that it is synced to the same master clock of the owner controller.</p>	<ul style="list-style-type: none"> 0 = CIP Sync is not available. 1 = CIP Sync is available.

Table 29. Input Tags (continued)

Name	Data Type	Definition	Valid Values
		You must compare the Grandmaster Clock ID of both the module and the owner controller.	
EventStatus[x].CIPSyncTimeout	BOOL	Indicates that the module was once synced with a 1588 master, but the module is not synced now due to a timeout.	<ul style="list-style-type: none"> 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. <p>The module is using its local clock and can be drifting away from the last known master time.</p>
EventStatus[x].EventReset	BOOL	When Eventxx.ResetEvent transitions from 0 to 1, EventStatus[x].EventReset transitions to 1 to indicate that the reset was received and completed. It stays 1 until Eventxx.ResetEvent transition to zero.	<ul style="list-style-type: none"> 0 = Do not reset 1 = Reset
EventStatus[x].EventsPending	SINT	The number of events currently queued in the module. A value greater than zero indicates that the controller is not currently keeping up with the rate of events.	All positive values
EventStatus[x].EventNumber	DINT	Running count of events, which increments by one for each new time event. The originator sets Eventxx.EventNumberAck to Eventxx.EventNumber to acknowledge receipt of the event. When the number of events reaches its maximum value and rolls over, it rolls over to 1, not 0.	All values
Pwmxx.Active	BOOL	Indicates whether the output is generating PWM pulses.	<ul style="list-style-type: none"> 0 = The output is not generating pulses 1 = The output is generating pulses
Pwmxx.Error	BOOL	Indicates whether there is an error with generating the PWM pulse.	<ul style="list-style-type: none"> 0 = There is no error present 1 = One of the following errors is present: <ul style="list-style-type: none"> 0.Pwmxx.CycleTime < minimum cycle time 0.Pwmxx.CycleTime > maximum cycle time 0.Pwmxx.DutyCycle * cycle time < minimum pulse width (100% - 0.Pwmxx.DutyCycle) * cycle time < minimum pulse width

Output Tag Definitions

Table 30. Output Tags

Name	Data Type	Definition	Valid Values
Windowxx.On	DINT (If window is configured for instantaneous count) REAL	Displays the value that defines one side of the window threshold. The outputs (if tied to the window) and I.Windowxx.InWindow tag switch state at this threshold according to the diagram Figure 14: Windows Behavior Example on page 33 .	Any value
Windowxx.Off	DINT (If window is configured for instantaneous count) REAL		
Windowxx.HysteresisOn	DINT (If window is configured for instantaneous count) REAL	The negative offset from the Window.On tag value to apply hysteresis to	Any value If the value ≤ 0 , hysteresis is disabled.
Windowxx.HysteresisOff	DINT (If window is configured for instantaneous count) REAL	The negative offset from the Window.Off tag value to apply hysteresis to	Any value If the value ≤ 0 , hysteresis is disabled.
Counter00.Reset	BOOL	Resets Count and Revolution values to zero when a transition from 0 to 1 occurs.	<ul style="list-style-type: none"> 0 = Do not reset values to zero 1 = Reset values to zero
Counter00.Hold	BOOL	Pause the counter when the value is 1.	<ul style="list-style-type: none"> 0 = Resume counting 1 = Pause counting
Counter00.Load	BOOL	Changes the Count and Revolution values to the value of the corresponding LoadCountValue and	<ul style="list-style-type: none"> 0 = Do not change the value 1 = Change the value

Table 30. Output Tags (continued)

Name	Data Type	Definition	Valid Values
		LoadRevolutionValue tags when a transition from 0 to 1 occurs.	
Counter00.Store	BOOL	Copies the counter value to the corresponding stored value tag when a transition from 0 to 1 occurs.	<ul style="list-style-type: none">0 = Do not copy the counter value1 = Copy the counter value
Counter00.Direction	BOOL	When the counter mode is count with internal direction, set the counter direction with this value.	<ul style="list-style-type: none">0 = Count up1 = Count down

Table 30. Output Tags (continued)

Name	Data Type	Definition	Valid Values
		IMPORTANT: When the counter is operating in counter/external direction, up/down pulses, or quadrature mode, this tag is ignored.	
Counter00.RollOverAck	BOOL	Clears the Rollover tag.	<ul style="list-style-type: none"> 0 = Does not clear the Rollover tag 1 = Clears the Rollover tag
Counter00.RollunderAck	BOOL	Clears the Rollunder tag.	<ul style="list-style-type: none"> 0 = Does not clear the Rollunder tag 1 = Clears the Rollunder tag
Counter00.FrequencyAlarmUnlatch	BOOL	Unlatches the Frequency Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.FrequencyAvgAlarmUnlatch	BOOL	Unlatches the Frequency Average Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.PulseWidthAlarmUnlatch	BOOL	Unlatches the Pulse Width Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.PulseWidthAvgAlarmUnlatch	BOOL	Unlatches the Pulse Width Average Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.ZeroFrequencyAlarmUnlatch	BOOL	Unlatches the Zero Frequency Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.ZeroFrequencyAvgAlarmUnlatch	BOOL	Unlatches the Zero Frequency Average Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.MissingPulseAlarmEn	BOOL	Enables the Missing Pulse Alarm.	<ul style="list-style-type: none"> 0 = Alarm is disabled 1 = Alarm is enabled
Counter00.MissingPulseAlarmUnlatch	BOOL	Unlatches the Missing Pulse Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.AccelAlarmUnlatch	BOOL	Unlatches the Acceleration Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.DecelAlarmUnlatch	BOOL	Unlatches the Deceleration Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.AccelAvgAlarmUnlatch	BOOL	Unlatches the Acceleration Average Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched 1 = Alarm is unlatched
Counter00.DecelAvgAlarmUnlatch	BOOL	Unlatches the Deceleration Average Alarm.	<ul style="list-style-type: none"> 0 = Alarm remains latched

Table 30. Output Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 1 = Alarm is unlatched
Counter00.ResetFrequencyOverrange	BOOL	If Frequency Overrange is set, this tag resets the Frequency Overrange to zero.	<ul style="list-style-type: none"> 0 = Frequency Overrange is not reset 1 = Frequency Overrange is reset
Counter00.ResetQuadratureErrorCount	BOOL	Resets the Quadrature Error count.	<ul style="list-style-type: none"> 0 = Count is not reset 1 = Count is reset
Counter00.RolloverValue	DINT	When the counter counts up to the value of this tag, it is set to the Rollunder value when the next pulse is received.	Any value
Counter00.RollunderValue	DINT	When the counter counts down to the value of this tag, it is set to the Rollover value when the next pulse is received.	Any value
Counter00.ZeroFrequencyAlarmLimit	REAL	Value that the frequency or average frequency must be less than to trigger the Zero Frequency Alarm.	Any value
Counter00.LoadCountValue	DINT	Preset number of counts that can be loaded in the Counterxx.Count input tag.	Any value
Counter00.LoadRevolutionValue	INT	Preset number of counts that can be loaded in the Counterxx.RevolutionCount input tag.	Any value
Counter00.OverrideDataAEn	BOOL	Enables Input A to be overridden by the OverrideDataAValue tag.	<ul style="list-style-type: none"> 0 = Input state is not overridden 1 = Input state is overridden
Counter00.OverrideDataBEn	BOOL	Enables Input B to be overridden by the OverrideDataBValue tag.	<ul style="list-style-type: none"> 0 = Input state is not overridden 1 = Input state is overridden
Counter00.OverrideDataTriggerEn	BOOL	Enables trigger input to be overridden by the OverrideDataTriggerValue tag. Only applicable when the standard digital input is selected as the trigger of one or more counter functions (that is, Store, Reset, Load, Hold).	<ul style="list-style-type: none"> 0 = Input state is not overridden 1 = Input state is overridden
Counter00.OverrideDataZEn	BOOL	Enables input Z to be overridden by the OverrideDataZValue tag.	<ul style="list-style-type: none"> 0 = Input state is not overridden 1 = Input state is overridden
Counter00.OverrideDataAValue	BOOL	The value that is applied to Input A when the OverrideDataAEn tag is set.	Any value
Counter00.OverrideDataBValue	BOOL	The value that is applied to Input B when the OverrideDataBEn tag is set.	Any value
Counter00.OverrideDataTriggerValue	BOOL	The value that is applied to the input trigger when the OverrideDataTriggerEn tag is set.	Any value

Table 30. Output Tags (continued)

Name	Data Type	Definition	Valid Values
		Only applicable when the standard digital input is selected as the trigger of one or more counter functions (that is, Store, Reset, Load, Hold).	
Counter00.OverrideDataZValue	BOOL	The value that is applied to Input Z when the OverrideDataZEn tag is set.	Any value
Counter00.MissingPulseAlarmLimit	DINT	Sets the length of time that must elapse between detecting the leading edge of consecutive pulses before the Missing Pulse Alarm is triggered.	<ul style="list-style-type: none"> 0 = Missing Pulse Alarm is disabled 1 or greater = The time, in microseconds, allowed before the Missing Pulse Alarm is triggered
SSI00.Store	BOOL	Triggers the Store function with a transition from 0 to 1.	<ul style="list-style-type: none"> 0 = Does not trigger the Store function 1 = Triggers the Store function
SSI00.OverrideStoreInputEn	BOOL	If Input00 is used as the Store function trigger, this tag enables overriding Input00 with the OverrideStoreInputValue tag	<ul style="list-style-type: none"> 0 = Store function trigger is not overridden 1 = Store function trigger is overridden
SSI00.OverrideStoreInputValue	BOOL	If Input00 is used as the Store function trigger, this value is applied to the input when the OverrideStoreInputEn tag is set.	Any value
Outputxx.OverrideDataEn	BOOL	Enables the output state to be overridden by the OverrideDataValue tag.	<ul style="list-style-type: none"> 0 = Override is disabled. 1 = Override is enabled.
Outputxx.OverrideDataValue	BOOL	The value at the output point when OverrideDataEn is enabled	<ul style="list-style-type: none"> 0 = Off 1 = On
Outputxx.Data	BOOL	Indicates the current digital output value applied at the screw.	<ul style="list-style-type: none"> 0 = Off 1 = On
Outputxx.Trigger	BOOL	Triggers the output pulse.	A rising edge of this bit triggers the output pulse. The Outputxx.PulseDelay and Outputxx.PulseActiveTime still apply.
Outputxx.ResetTriggerMissed	BOOL	Resets I.Outputxx.TriggerMissed bit.	A rising edge of ResetTriggerMissed bit clears the I.Outputxx.TriggerMissed bit.
Outputxx.PulseDelay	REAL	Specifies the delay time in seconds between the trigger event and the start of the output pulse.	All positive values
Outputxx.PulseActiveTime	REAL	Specifies the active duration in seconds for the generated output pulse.	All positive values
Pwmxx.CycleTime	REAL	Indicates the period in seconds of one pulse cycle.	10 μ s...30 s
Pwmxx.DutyCycle	REAL	Indicates in percentage of the pulse cycle time when output is in the active state.	0...100%

Table 30. Output Tags (continued)

Name	Data Type	Definition	Valid Values
Pwmxx.CycleCountLimit	DINT	Determines the number of pulse cycles to be generated when there is a rising edge in the data. Only applicable when C.Pwmxx.CycleCountEn = 1.	0...2,147,483,647

Event Input Tag Definitions

Table 31. Event Input Tags

Name	Data Type	Definition	Valid Values
RunMode	BOOL	The input's operating state when the event is triggered.	<ul style="list-style-type: none"> 0 = Idle 1 = Run
ConnectionFaulted	BOOL	Indicates if a connection to the target is running. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.	<ul style="list-style-type: none"> 0 = Connection is running 1 = Connection is not running.
DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics are active. 1 = One or more diagnostics are active or the prognostics threshold is reached.
DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected.	-128...+127 The value of 0 is skipped except during module powerup.
Eventxx.Fault	BOOL	Indicates that the event data is inaccurate and cannot be trusted for use in the application. If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault
Eventxx.Uncertain	BOOL	Indicates that the event data can be inaccurate but the degree of inaccuracy is not known.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data

Table 31. Event Input Tags (continued)

Name	Data Type	Definition	Valid Values
		If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.	
Eventxx.InvalidConfiguration	BOOL	Indicates that there is an error in the Event Trigger definition.	<ul style="list-style-type: none"> 0 = There is no error in the Event Trigger definition. 1 = There is an error in the Event Trigger definition.
Eventxx.EventDropped	BOOL	Indicates whether an event has been discarded because events are occurring faster than they are being acknowledged.	<ul style="list-style-type: none"> 0 = An event has not been discarded. 1 = An event has been discarded.
Eventxx.EventRising	BOOL	Indicates whether an event triggered when a state change resulted in an event pattern being matched.	<ul style="list-style-type: none"> 0 = Event was not triggered 1 = Event was triggered
Eventxx.EventFailing	BOOL	Indicates whether an event triggered when a state change resulted in an event pattern no longer being matched.	<ul style="list-style-type: none"> 0 = Event was not triggered 1 = Event was triggered
Eventxx.CIPSyncValid	BOOL	<p>Indicates whether the module was synced with a 1588 master when the event was triggered.</p> <p>A set bit alone cannot indicate that it is synced to the same master clock of the owner controller.</p> <p>You must compare the Grandmaster Clock ID of both the module and the owner controller.</p>	<ul style="list-style-type: none"> 0 = CIP Sync was not available 1 = CIP Sync was available
Eventxx.CIPSyncTimeout	BOOL	Indicates that the module was once synced with a 1588 master, but the module is not synced now due to a timeout when the event was triggered.	<ul style="list-style-type: none"> 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. <p>The module is using its local clock and can be drifting away from the last known master time.</p>
Eventxx.EventsPending	SINT	<p>The number of events currently queued in the module.</p> <p>A value greater than zero indicates that the controller is not currently keeping up with the rate of events.</p>	All positive values
Eventxx.EventNumber	DINT	<p>Running count of events, which increments by one for each new time event.</p> <p>The originator sets Eventxx.EventNumberAck to Eventxx.EventNumber to acknowledge receipt of the event.</p> <p>When the number of events reaches its maximum value and rolls over, it rolls over to 1, not 0.</p>	All values

Table 31. Event Input Tags (continued)

Name	Data Type	Definition	Valid Values
Eventxx.EventTimestamp	LINT	The time that the event occurred	All positive values
Eventxx.MissingPulseAlarm	BOOL	Indicates that when the event was triggered, the period of a pulse was greater than the missing pulse timeout or was never completed.	<ul style="list-style-type: none"> 0 = Period of the pulse was normal 1 = Period of the pulse was greater than the missing pulse timeout or was never completed
Eventxx.ZeroFrequencyAlarm	BOOL	Indicates that when the event was triggered, the counter frequency was less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	<ul style="list-style-type: none"> 0 = Counter frequency was not less than the Zero Frequency Alarm limit 1 = Counter frequency was less than the Zero Frequency Alarm limit
Eventxx.ZeroFrequencyAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average frequency was less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	<ul style="list-style-type: none"> 0 = Counter average frequency was not less than the Zero Frequency Alarm limit 1 = Counter average frequency was less than the Zero Frequency Alarm limit
Eventxx.FrequencyAlarm	BOOL	Indicates that when the event was triggered, the counter frequency had exceeded the Frequency Alarm limit.	<ul style="list-style-type: none"> 0 = Counter frequency had not exceeded the Frequency Alarm limit 1 = Counter frequency exceeded the Frequency Alarm limit
Eventxx.FrequencyAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average frequency had exceeded the Frequency Alarm limit.	<ul style="list-style-type: none"> 0 = Counter average frequency had not exceeded the Frequency Alarm limit 1 = Counter average frequency exceeded the Frequency Alarm limit
Eventxx.PulseWidthAlarm	BOOL	Indicates that when the event was triggered, the counter pulse width was greater than the maximum pulse width established by the Pulse Width Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered That is, the counter pulse width was not greater than the Pulse Width Alarm limit. 1 = Alarm was triggered That is, the counter pulse width was greater than the Pulse Width Alarm limit.
Eventxx.PulseWidthAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average pulse width was greater than the maximum pulse width established by the Pulse Width Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered That is, the counter average pulse width was not greater than the Pulse Width Alarm limit. 1 = Alarm was triggered That is, the counter average pulse width was greater than the Pulse Width Alarm limit.
Eventxx.AccelAlarm	BOOL	Indicates that when the event was triggered, the counter acceleration rate exceeded the Acceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered That is, the counter acceleration rate did not exceed the Acceleration Alarm limit.

Table 31. Event Input Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 1 = Alarm was triggered <p>That is, the counter acceleration rate exceeded the Acceleration Alarm limit.</p>
Eventxx.AccelAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average acceleration rate exceeded the Acceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered <p>That is, the counter average acceleration rate did not exceed the Acceleration Alarm limit.</p> <ul style="list-style-type: none"> 1 = Alarm was triggered <p>That is, the counter average acceleration rate exceeded the Acceleration Alarm limit.</p>
Eventxx.DecelAlarm	BOOL	Indicates that when the event was triggered, the counter deceleration rate exceeded the Deceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered <p>That is, the counter deceleration rate did not exceed the Deceleration Alarm limit.</p> <ul style="list-style-type: none"> 1 = Alarm was triggered <p>That is, the counter deceleration rate exceeded the Deceleration Alarm limit.</p>
Eventxx.DecelAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average deceleration rate exceeded the Deceleration Alarm limit, triggering this alarm.	<ul style="list-style-type: none"> 0 = Alarm was not triggered <p>That is, the counter average deceleration rate did not exceed the Deceleration Alarm limit.</p>

Table 31. Event Input Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 1 = Alarm was triggered <p>That is, the counter average deceleration rate exceeded the Deceleration Alarm limit.</p>
Eventxx.FrequencyOverrange	BOOL	Indicates that when the event was triggered, the counter frequency exceeded the module's maximum rated frequency, and therefore can not be tracking the signal properly.	<ul style="list-style-type: none"> 0 = Counter frequency did not exceed the module's maximum rated frequency. 1 = Counter frequency exceeded the module's maximum rated frequency.
Eventxx.Reset	BOOL	Indicates if the event is triggered by a counter reset.	<ul style="list-style-type: none"> 0 = The event is not triggered by a counter reset. 1 = The event is triggered by a counter reset.
Eventxx.Hold	BOOL	Indicates that when the event was triggered, the counter was in the Hold state.	<ul style="list-style-type: none"> 0 = Counter was counting. 1 = Counter was in the Hold state.
Eventxx.Load	BOOL	Indicates if the event is triggered by a counter loading.	<ul style="list-style-type: none"> 0 = The event is not triggered by a counter loading. 1 = The event is triggered by a counter loading.
Eventxx.Store	BOOL	Indicates if the event is triggered by a counter storing.	<ul style="list-style-type: none"> 0 = The event is not triggered by a counter storing. 1 = The event is triggered by a counter storing.
Eventxx.Direction	BOOL	Indicates the count direction when the event was triggered.	<ul style="list-style-type: none"> 0 = Counting up 1 = Counting down
Eventxx.StoredDirection	BOOL	Indicates the stored count direction when the event was triggered.	<ul style="list-style-type: none"> 0 = Counting up 1 = Counting down
Eventxx.Rollover	BOOL	Indicates the Rollover state when the event was triggered.	<ul style="list-style-type: none"> 0 = Counter had not counted to the Rollover value and continued counting from the Rollunder value. 1 = Counter had counted to the Rollover value and continued counting from the Rollunder value.
Eventxx.Rollunder	BOOL	Indicates the Rollunder state when the event was triggered.	<ul style="list-style-type: none"> 0 = Counter had not counted down to the Rollunder value and continued counting from the Rollover value. 1 = Counter had counted down to the Rollunder value and continued counting from the Rollover value.
Eventxx.DataA	BOOL	Indicates the Input A value when the event was triggered.	<ul style="list-style-type: none"> 0 = Input A was Off. 1 = Input A was On.

Table 31. Event Input Tags (continued)

Name	Data Type	Definition	Valid Values
		If <code>O.OverrideDataAEn = 1</code> , this tag reflects <code>O.OverrideDataAValue</code> .	
Eventxx.DataB	BOOL	Indicates the Input B value when the event was triggered. If <code>O.OverrideDataBEn = 1</code> , this tag reflects <code>O.OverrideDataBValue</code> .	<ul style="list-style-type: none"> 0 = Input B was Off. 1 = Input B was On.
Eventxx.DataZ	BOOL	Indicates the Input Z value when the event was triggered. If <code>O.OverrideDataZEn = 1</code> , this tag reflects <code>O.OverrideDataZValue</code> .	<ul style="list-style-type: none"> 0 = Input Z was Off. 1 = Input Z was On.
Eventxx.InWindowxx	BOOL	Indicates that the signal value was in the specified window when the event was triggered.	<ul style="list-style-type: none"> 0 = Signal value was not in the specified window. 1 = Signal value was in the specified window.
Eventxx.DataTrigger	BOOL	Indicates the data value of the digital input, which was used as a trigger for counter functions, when the event was triggered. If <code>O.OverrideDataTriggerEn = 1</code> , this tag reflects the <code>O.OverrideDataTriggerValue</code> .	<ul style="list-style-type: none"> 0 = Input was Off. 1 = Input was On.
Eventxx.StoreInputData	BOOL	Indicates the data of the digital input, which was used as a Store function trigger, when the event was triggered. When no digital input is used as a Store function trigger, this member is always 0.	<ul style="list-style-type: none"> 0 = Input was Off. 1 = Input was On.

Event Output Tag Definitions

Table 32. Event Output Tags

Name	Data Type	Definition	Valid Values
Eventxx.En	BOOL	Sets the corresponding event trigger definition to active. Events are triggered once when conditions match the definition.	<ul style="list-style-type: none"> 0 = Event trigger definition is not active and events are not triggered when conditions match the definition 1 = Event trigger definition is active and events are triggered when conditions match the definition
Eventxx.EventRisingEn	BOOL	Triggers an event each time a state change results in conditions that match the event trigger definition.	Values (EventRisingEn, EventFallingEn) represent: <ul style="list-style-type: none"> (0, 0): Event trigger is disabled. (1, 0): Event triggers on transitions to match the configured pattern.
Eventxx.EventFallingEn	BOOL	Triggers an event each time a state change results in conditions that do not match the event trigger definition.	

Table 32. Event Output Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> (0, 1): Event triggers on transitions to not match the configured pattern. (1, 1): Event triggers on transitions to either match or not match the configured pattern.
Eventxx.LatchEvent	BOOL	When set, events are latched until acknowledged. When not set, new events overwrite old events.	<ul style="list-style-type: none"> 0 = Event is not latched 1 = Event is latched
Eventxx.ResetEvent	BOOL	A transition from 0 to 1 resets all events and clears the event queue on the Eventxx.	<ul style="list-style-type: none"> 0 = Do not reset 1 = Reset
Eventxx.IndependentConditionTriggerEn	BOOL	Determines whether each condition indicated in the Event trigger definition can initiate an event independently.	<ul style="list-style-type: none"> 0 = Event is triggered when all participating conditions in the event trigger achieve their configured values 1 = Event is triggered when any participating condition in the event trigger achieves their configured value
Eventxx.CounterSelect	SINT	Indicates the counter with which the event is associated, if any.	0 = Counter00
Eventxx.SSISelect	SINT	Indicates the SSI with which the event is associated, if any.	0 = SSI00

Table 32. Event Output Tags (continued)

Name	Data Type	Definition	Valid Values
Eventxx.EventNumberAck	DINT	The controller writes back EventNumber into EventNumberAck to indicate receipt of the event.	All values
Eventxx.InWindowxxSelect	BOOL	Indicates that the specified window participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.ResetSelect	BOOL	Indicates that the counter's Reset function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.HoldSelect	BOOL	Indicates that the counter's Hold function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.LoadSelect	BOOL	Indicates that the counter's Load function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.StoreSelect	BOOL	Indicates that the counter's Store function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.RolloverSelect	BOOL	Indicates that the counter's Rollover function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.RollunderSelect	BOOL	Indicates that the counter's Rollunder function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.DirectionSelect	BOOL	Indicates that the counter's or SSI's Direction function participates in the Event trigger definition.	<ul style="list-style-type: none"> 0 = Does not participate in event trigger 1 = Participates in event trigger
Eventxx.InWindowxxValue	BOOL	If the specified window is selected in the Event trigger definition, this tag indicates the value to match after the status change.	<ul style="list-style-type: none"> 0 = InWindow value of 0 matches the event pattern

Table 32. Event Output Tags (continued)

Name	Data Type	Definition	Valid Values
			<ul style="list-style-type: none"> 1 = InWindow value of 1 matches the event pattern
Eventxx.ResetValue	BOOL	If the Reset function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Reset value of 0 matches the event pattern 1 = Reset value of 1 matches the event pattern
Eventxx.HoldValue	BOOL	If the Hold function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Hold value of 0 matches the event pattern 1 = Hold value of 1 matches the event pattern
Eventxx.LoadValue	BOOL	If the Load function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Load value of 0 matches the event pattern 1 = Load value of 1 matches the event pattern
Eventxx.StoreValue	BOOL	If the Store function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Store value of 0 matches the event pattern 1 = Store value of 1 matches the event pattern
Eventxx.RolloverValue	BOOL	If the Rollover function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Rollover value of 0 matches the event pattern 1 = Rollover value of 1 matches the event pattern
Eventxx.RollunderValue	BOOL	If the Rollunder function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Rollunder value of 0 matches the event pattern 1 = Rollunder value of 1 matches the event pattern
Eventxx.DirectionValue	BOOL	If the Direction function is selected in the Event trigger definition, this tag indicates that value that is to trigger the event.	<ul style="list-style-type: none"> 0 = Direction value of 0 matches the event pattern 1 = Direction value of 1 matches the event pattern

Diagnostic Assembly

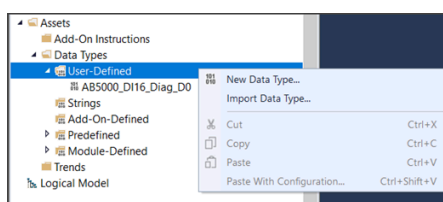
Diagnostic assembly helps you troubleshoot and diagnose the fault in your system.

Create User-defined Diagnostic Assembly Types

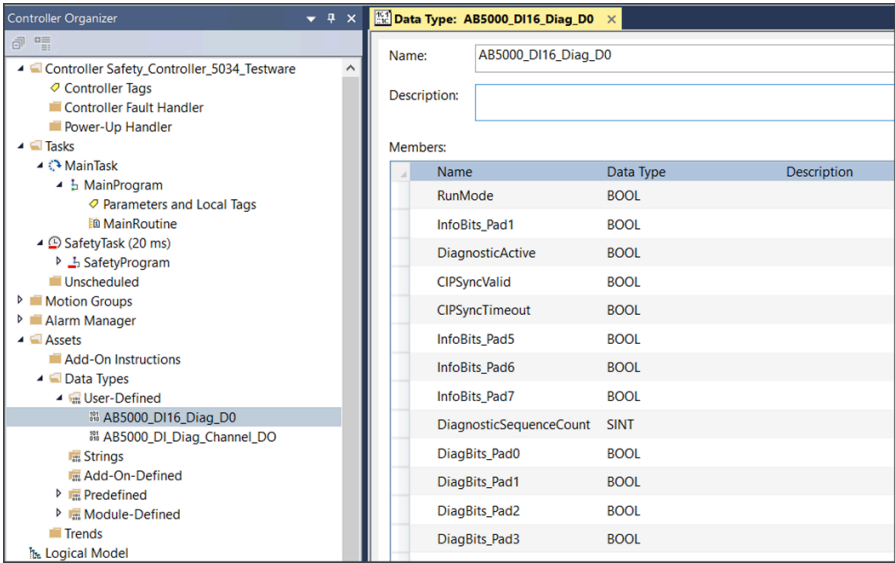
You can use the Studio 5000 Logix Designer application to create user-defined Diagnostic Assembly types.

Create user-defined diagnostic assembly by following the below steps:

1. From the Controller Organizer pane, go to Assets → Data types → User-Defined.
2. Right-click on the User-Defined folder and select New Data Type.



- 3. Add a Name and Description (optional) for your diagnostic assembly.
- 4. Under Members area, add the data members based on the diagnostic assembly detailed below.



IMPORTANT: The members indicated in the tables are arranged according to Data Alignment Rules of controllers. Strictly follow the data type and the sequence of the members that are indicated in the tables of this appendix. If the data type and the sequence are not followed, data misalignment may occur after executing Get Attribute Single Message (MSG) instruction.

Diagnostic Assembly

- 1. Diagnostic ENC Assembly (Combo)
 - Instance ID = 0x80B9 (32,953)
 - Size = 80 bytes

Table 33. Diagnostic Assembly Instance - 32,953

Name	Data Type	Size in Bytes
RunMode	BOOL	1
InfoBits_Pad1	BOOL	
DiagnosticActive	BOOL	
CIPSyncValid	BOOL	
CIPSyncTimeout	BOOL	
InfoBits_Pad5	BOOL	
InfoBits_Pad6	BOOL	
InfoBits_Pad7	BOOL	

Name	Data Type	Size in Bytes
DiagnosticSequenceCount	SINT	1
DiagBits_Pad0	BOOL	2
DiagBits_Pad1	BOOL	
DiagBits_Pad2	BOOL	
DiagBits_Pad3	BOOL	
DiagBits_Pad4	BOOL	
DiagBits_Pad5	BOOL	
DiagBits_Pad6	BOOL	
DiagBits_Pad7	BOOL	
DiagBits_Pad8	BOOL	
DiagBits_Pad9	BOOL	
BaseUnsupportedFault	BOOL	
BaseIDFault	BOOL	
FlashUpdateRequired	BOOL	
SelfTestFault	BOOL	
DiagBits_Pad14	BOOL	
DiagBits_Pad15	BOOL	
EncoderModes[2]	SINT	2
Pad	INT	2
LocalClockOffset	LINT	8
LocalClockOffsetTimestamp	LINT	8
GrandMasterClockID[8]	SINT	8
Counter00	See Diagnostic HSC Channel	24
SSI00	See Diagnostic SSI Channel	24

2. Diagnostic Digital Channels Assembly
 - Instance ID = 0x801C (32,796)
 - Size = 200 bytes

Table 34. Diagnostic Assembly Instance – 32,796

Name	Data Type	Size in Bytes
Pad1	LINT	8
Input00	See Diagnostic Digital Input Channel	48
Output00	See Diagnostic Digital Output Channel	48
Output01	See Diagnostic Digital Output Channel	48
Pad2	DINT[12]	48

Diagnostic Counters

Diagnostic Counters Assembly for I/O

- Instance ID = 0x301 (769)
- Size = 16 bytes

Table 35. Diagnostic Assembly Instance – 769

Name	Data Type	Size in Bytes
RunMode	BOOL	1
InfoBits_Pad1	BOOL	
DiagnosticActive	BOOL	
InfoBits_Pad3	BOOL	
InfoBits_Pad4	BOOL	
InfoBits_Pad5	BOOL	
InfoBits_Pad6	BOOL	
InfoBits_Pad7	BOOL	
DiagnosticSequenceCount	SINT	1
CIPConnections	INT	2
CIPLostPackets	DINT	4

Table 35. Diagnostic Assembly Instance – 769 (continued)

Name	Data Type	Size in Bytes
CIPTimeouts	DINT	4
CPUUtilization	INT	2
Pad	INT	2

Diagnostic Channel

This data type is retrieved as part of the diagnostic assembly instance.

1. Diagnostic HSC Channel
 - Size = 24 bytes

Table 36. Structure for Data Type Diagnostic HSC Channel

Name	Data Type	Size in Bytes
DataBits_Pad0	BOOL	2
Fault	BOOL	
Uncertain	BOOL	
DataBits_Pad3	BOOL	
DataBits_Pad4	BOOL	
DataBits_Pad5	BOOL	
FieldPowerOff	BOOL	
DataBits_Pad7	BOOL	
FrequencyOvrrange	BOOL	
DataBits_Pad9	BOOL	
DataBits_Pad10	BOOL	
DataBits_Pad11	BOOL	
DataBits_Pad12	BOOL	
DataBits_Pad13	BOOL	
DataBits_Pad14	BOOL	
DataBits_Pad15	BOOL	
DiagBits_Pad0...DiagBits_Pad15	INT	2

Name	Data Type	Size in Bytes
QuadratureErrorCount	SINT	1
Pad1	SINT	1
Pad2	SINT	1
Pad3	SINT	1
FieldPowerOnTimestamp	LINT	8
FieldPowerOffTimestamp	LINT	8

2. Diagnostic SSI Channel
 - Size = 24 bytes

Table 37. Structure for Data Type Diagnostic SSI Channel

Name	Data Type	Size in Bytes
DataBits_Pad0	BOOL	2
Fault	BOOL	
Uncertain	BOOL	
DataBits_Pad3	BOOL	
DataBits_Pad4	BOOL	
DataBits_Pad5	BOOL	
FieldPowerOff	BOOL	
DataBits_Pad7	BOOL	
DataBits_Pad8	BOOL	
DataBits_Pad9	BOOL	
DataBits_Pad10	BOOL	
DataBits_Pad11	BOOL	
DataBits_Pad12	BOOL	
DataBits_Pad13	BOOL	
DataBits_Pad14	BOOL	
DataBits_Pad15	BOOL	
DiagBits_Pad0...DiagBits_Pad15	INT	2

Name	Data Type	Size in Bytes
Pad0	SINT	1
Pad1	SINT	1
Pad2	SINT	1
Pad3	SINT	1
FieldPowerOnTimestamp	LINT	8
FieldPowerOffTimestamp	LINT	8

3. Diagnostic Digital Input Channel
 - Size = 48 bytes

Table 38. Structure for Data Type Diagnostic Digital Input Channel

Name	Data Type	Size in Bytes
DiagBits_Pad0	BOOL	2
Fault	BOOL	
Uncertain	BOOL	
DiagBits_Pad3	BOOL	
DiagBits_Pad4	BOOL	
DiagBits_Pad5	BOOL	
FieldPowerOff	BOOL	
DiagBits_Pad7	BOOL	
DiagBits_Pad8	BOOL	
DiagBits_Pad9	BOOL	
DiagBits_Pad10	BOOL	
DiagBits_Pad11	BOOL	
DiagBits_Pad12	BOOL	
DiagBits_Pad13	BOOL	
DiagBits_Pad14	BOOL	
DiagBits_Pad15	BOOL	
Pad1	INT	2

Name	Data Type	Size in Bytes
Pad2	DINT	4
Pad3	LINT	8
Pad4	LINT	8
FieldPowerOnTimestamp	LINT	8
FieldPowerOffTimestamp	LINT	8
Pad7	LINT	8

4. Diagnostic Digital Output Channel
 - Size = 48 bytes

Table 39. Structure for Data Type Diagnostic Digital Output Channel

Name	Data Type	Size in Bytes
DiagBits_Pad0	BOOL	2
Fault	BOOL	
Uncertain	BOOL	
DiagBits_Pad3	BOOL	
ShortCircuit	BOOL	
DiagBits_Pad5	BOOL	
FieldPowerOff	BOOL	
DiagBits_Pad7	BOOL	
DiagBits_Pad8	BOOL	
DiagBits_Pad9	BOOL	
DiagBits_Pad10	BOOL	
DiagBits_Pad11	BOOL	
DiagBits_Pad12	BOOL	
DiagBits_Pad13	BOOL	
DiagBits_Pad14	BOOL	
DiagBits_Pad15	BOOL	
Pad1	INT	2

Name	Data Type	Size in Bytes
Pad2	DINT	4
Pad3	LINT	8
ShortCircuitTimestamp	LINT	8
FieldPowerOnTimestamp	LINT	8
FieldPowerOffTimestamp	LINT	8
Pad7	LINT	8

Definitions for Diagnostic Assembly Members

Table 40. Definition of Members in Diagnostic Assembly

Name	Data Type	Definition	Valid Values
RunMode	BOOL	The module's operating state	<ul style="list-style-type: none"> 0 = Idle – It means one of the following: <ul style="list-style-type: none"> Connection is not up. Connection has been opened but the module has not started producing data for the connection. Module is not applying new output tag data because the controller is in Program Mode. 1 = Run – It means the following: <ul style="list-style-type: none"> Connection is up. Module is producing data for the connection. Output tag data is being applied or there is no output tag data for the connection (For example, all points configured as digital input without timestamp).
DiagnosticActive	BOOL	Indicates if any diagnostics are active or if the prognostics threshold is reached.	<ul style="list-style-type: none"> 0 = No diagnostics are active 1 = One or more diagnostics are active or the prognostics threshold is reached
CIPSyncValid	BOOL	Indicates whether the module is synced with a 1588 master. A set bit alone cannot indicate that it is synced to the same master clock of the owner controller.	<ul style="list-style-type: none"> 0 = CIP Sync is not available 1 = CIP Sync is available

Table 40. Definition of Members in Diagnostic Assembly (continued)

Name	Data Type	Definition	Valid Values
		You must compare the Grandmaster Clock ID of both the module and the owner controller.	
CIPSyncTimeout	BOOL	Indicates that the module was once synced with a 1588 master, but the module is not synced now due to a timeout.	<ul style="list-style-type: none"> 0 = A valid time master has not timed out 1 = A valid time master has timed out The module is using its local clock and can be drifting away from the last known master time.
DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected.	-128...+127 The value of 0 is skipped except during module powerup.
LocalClockOffset	LINT	The offset from the local clock to the system time. This value helps to detect steps in time. This value updates when a PTP update is received.	All values
LocalClockOffsetTimestamp	LINT	Indicates the time of the local clock offset.	A valid time or None if there is no recorded event time. Time format is YYYY-MMDD-HH:mm:ss_mmm_uuu_nnn(UTC-00:00). <ul style="list-style-type: none"> YYYY = year MM = month DD = day HH = hour (24 hour) mm = minutes SS = seconds mmm = milliseconds uuu = microseconds nnn = nanoseconds UTC-00:00 = Time zone
GrandMasterClockID	SINT[8]	The EUI-64 Identity of the CIP Sync Grandmaster clock the module is synced to	All values
Fault	BOOL	Indicates that the data is inaccurate and cannot be trusted for use in the application. If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy. <hr/> IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.	<ul style="list-style-type: none"> 0 = Good 1 = Bad, causing fault
Uncertain	BOOL	Indicates that the data can be inaccurate but the degree of inaccuracy is not known. If the tag is set to 1, you must troubleshoot the module to correct the cause of the inaccuracy.	<ul style="list-style-type: none"> 0 = Good data 1 = Uncertain data

Table 40. Definition of Members in Diagnostic Assembly (continued)

Name	Data Type	Definition	Valid Values
		IMPORTANT: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.	
FieldPowerOff	BOOL	Indicates that a field power loss condition exists on the channel.	<ul style="list-style-type: none"> 0 = Field power off condition does not exist 1 = Field power off condition exists
FieldPowerOnTimestamp	LINT	Indicates the time of the last field power turned on event.	<p>A valid time or None if there is no recorded event time.</p> <p>Time format is YYYY-MM-DD-HH:mm:ss.mmm.</p> <ul style="list-style-type: none"> YYYY = year MM = month DD = day HH = hour (24 hour) mm = minutes SS = seconds mmm = milliseconds
FieldPowerOffTimestamp	LINT	Indicates the time of the last field power turned off event.	<p>A valid time or None if there is no recorded event time.</p> <p>Time format is YYYY-MM-DD-HH:mm:ss.mmm.</p> <ul style="list-style-type: none"> YYYY = year MM = month DD = day HH = hour (24 hour) mm = minutes SS = seconds mmm = milliseconds
FrequencyOverrange	BOOL	Indicates whether the frequency is overrange.	<ul style="list-style-type: none"> 0 = Frequency is not overrange 1 = Frequency is overrange
QuadratureErrorCount	SINT	Indicates the number of times that a quadrature counter error occurred.	All values
ShortCircuit	BOOL	Indicates whether an output short circuit or overcurrent fault is present on the point.	<ul style="list-style-type: none"> 0 = Short Circuit condition does not exist 1 = Short Circuit condition exists
ShortCircuitTimestamp	LINT	Indicates the time of the last short circuit fault.	<p>A valid time or None if there is no recorded event time.</p> <p>Time format is YYYY-MM-DD-HH:mm:ss.mmm.</p> <ul style="list-style-type: none"> YYYY = year MM = month DD = day HH = hour (24 hour) mm = minutes SS = seconds mmm = milliseconds

Table 40. Definition of Members in Diagnostic Assembly (continued)

Name	Data Type	Definition	Valid Values
CIPConnections	INT	Indicates the number of CIP connections that are currently open.	0...24
CIPLostPackets	DINT	Indicates the running sum of the number of Sequenced Address Item Sequence Numbers that are skipped in Class 0 and Class 1 connections consumed by the adapter and its children.	0...2,147,483,647
CIPTimeouts	DINT	Indicates the running count of the number of connections that time out, both originated and targeted, to and through the adapter.	0...2,147,483,647
CPUUtilization	INT	Indicates the usage of the compute engine.	0...100%

Create Message Type User Tags

Create MESSAGE type user tags for requests and associated response user tags for each of the new user-defined diagnostic assembly types.

Name	Usage	Value	Force Mask	Style	Data Type
Digital_5034_IB16	Local	0		Decimal	BOOL
Digital_Diag_Resp	Local	(...)		(...)	AB5000_DI_Diag_Channel_DO
Digital_Diagnostic_Assembly	Local	(...)		(...)	MESSAGE

From the Controller Organizer pane, expand Tasks → MainTask → MainProgram

1. Create MESSAGE type user tags for each request.
2. Create associated response user tags for each new user-defined diagnostic assembly type.
3. Add the user tags to your ladder program.



4. Expand the message tag to open the message configuration dialog.
5. On the Configuration tab, select:
 - Service type: Get Attribute Single
 - Class: 4
 - Attribute: 3
 - Instance:
 - 32,953 (80B9h) Diagnostic ENC Assembly (Combo)
 - 32,796 (801h) Diagnostic Digital Channels Assembly
 - 769 (301h) Diagnostic Counters Base I/O Assembly
 - Destination element: User-defined data type suitable for the instance entered.

6. On the Communication tab, select the path to the module that you wish to send the messages to.
7. Download the project and set to Run mode.
You can monitor the user-defined tag values from the Program Parameters and Local Tags window, under the MainProgram task in the Controller Organizer pane.

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, 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)	Get help determining how products interact, check features and capabilities, and find associated firmware.	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 information on its website at rok.auto/pec.

Allen-Bradley, CompactLogix, ControlLogix, expanding human possibility, FactoryTalk, GuardLogix, Logix 5000, POINT I/O, PointMax, Rockwell Automation, Studio 5000 Logix Designer, and TechConnect are trademarks of Rockwell Automation, Inc.

CIP, CIP Sync, ControlNet, DeviceNet, and EtherNet/IP are trademarks of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

rockwellautomation.com — expanding **human possibility**

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2663 0600
ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608
UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800