User Manual

Allen-Bradley

5000 Series High-speed Counter Module in Logix5000 Control Systems

Catalog Number 5069-HSC2x0B4

Allen-Bradley • Rockwell Software

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.

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).

High-speed Counter Module Operating Modes

Chapter 3

Chapter 4

Configure the High-speed Counter Module

Appendix A

Troubleshoot Your Module

Appendix B

Module Tag Definitions

Index

Notes:

This manual describes how to use 5000 Series high-speed counter modules in Logix5000™ control systems.

Audience

This manual is intended for control engineers that design, install and monitor industrial automation systems.

Make sure that you are familiar with the following:

- Use of a controller in a Logix5000 control system
- Use of an EtherNet/IP network, if the high-speed counter module is installed in a remote location from the controller that is accessible via the | EtherNet/IP network
- Studio 5000 Logix Designer® environment

IMPORTANT Remember the following when you use 5000 series high-speed counter module:

- You cannot use 5000 Series high-speed counter modules with all Logix5000 controllers. For example, you can use the 5069 Compact I/O™ high-speed counter module with ControlLogix® 5580 controllers but not with ControlLogix 5570 controllers.

  For more information on which Logix5000 controllers that you can use with 5000 series high-speed counter modules, see the product description at http://www.ab.com.
- You must use the Logix Designer application, **version 28 or greater**, to configure the 5000 series high-speed counter modules.

Differences From Other High-speed Counter Modules

The 5069 Compact I/O high-speed counter functions similarly to other high-speed counter modules from Rockwell Automation. But the way to complete tasks is different in some cases.

For example, you enable the module counters and start counting as follows:

- 5069-HSC2xOB4 high-speed counter module - Set the O.Counter*xx*.Hold module output tag to 0.
- 1769-HSC high-speed counter module - Set the Ctr*n*En bit to 1.

Use this publication to learn how to use all of the functionality that the 5069-HSC2xOB4 high-speed counter module supports.

Additional Resources

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

Resource	Description
5069 Compact I/O Modules High-speed Counter Module Installation Instructions, publication 5069-IN005	Describes how to install and wire the 5069-HSC2x0B4 high-speed counter module
5069 Compact I/O EtherNet/IP Adapter Installation Instructions, publication 5069-IN003	Describes how to install and wire the 5069-AEN2TR EtherNet/IP adapter
EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication ENET-UM004	Describes how to use the 5069-AEN2TR EtherNet/IP adapter.
5069 Compact I/O Modules Specifications Technical Data, publication 5069-TD001	Provides specifications, wiring diagrams, and module block diagrams 5069 Compact I/O modules
5000 Series Digital I/O Modules in Logix5000 Control Systems User Manual, publication 5000-UM004	Describes how to use 5000 Series digital I/O modules.
5000 Series Analog I/O Module in Logix5000 Control Systems User Manual, publication 5000-UM005	Describes how to use 5000 Series analog I/O modules.
Integrated Architecture and CIP Sync Configuration Application Technique, publication IA-AT003	Describes how to configure CIP Sync with Integrated Architecture™ products and applications
Logix5000 Controllers Tasks, Programs, and Routines Programming Manual, publication 1756-PM005	Describes how to set up controller tasks and the programs and routines for the proper execution of these tasks.
Electronic Keying in Logix5000 Control Systems Application Technique, publication LOGIX-AT001.	Describes how to use Electronic Keying.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation® industrial system.
Product Certifications website, http://www.ab.com	Provides declarations of conformity, certificates, and other certification details.

You can view or download Rockwell Automation publications at http://www.rockwellautomation.com/literature/.

To order paper copies of technical documentation, contact your local Allen-Bradley distributor or Rockwell Automation sales representative.

High-speed Counter Module in a Logix5000 Control System

The 5069 Compact I/O™ high-speed counter module counts incoming pulses from pulse generators, counters, limit switches, and other devices at a high rate of speed. The module returns the count to a controller. The module can also use module windows to activate module outputs immediately.

IMPORTANT You cannot use 5069 Compact I/O high-speed counter modules with all Logix5000 controllers.

For more information on which Logix5000 controllers that you can use with 5069 Compact I/O high-speed counter modules, see the product description at http://www.ab.com.

The 5069 Compact I/O high-speed counter module uses a removable terminal block (RTB) to connect field-side wiring. You use the Logix Designer application to configure the module.

The 5069 Compact I/O high-speed counter module uses the producer/consumer network communication model. This communication is an intelligent data exchange between modules and other system devices in which each module produces data without first being polled.

Before You Begin

Before you use a 5069 Compact I/O high-speed counter module, you must complete the following:

- Install an EtherNet/IP network.
- Install a Logix5000™ controller with a connection to the EtherNet/IP network.
- Install a 5069 Compact I/O system.

  For more information on how to install a 5069 Compact I/O system, see the 5069 Compact I/O EtherNet/IP Adapter Installation Instructions, publication 5069-IN003.

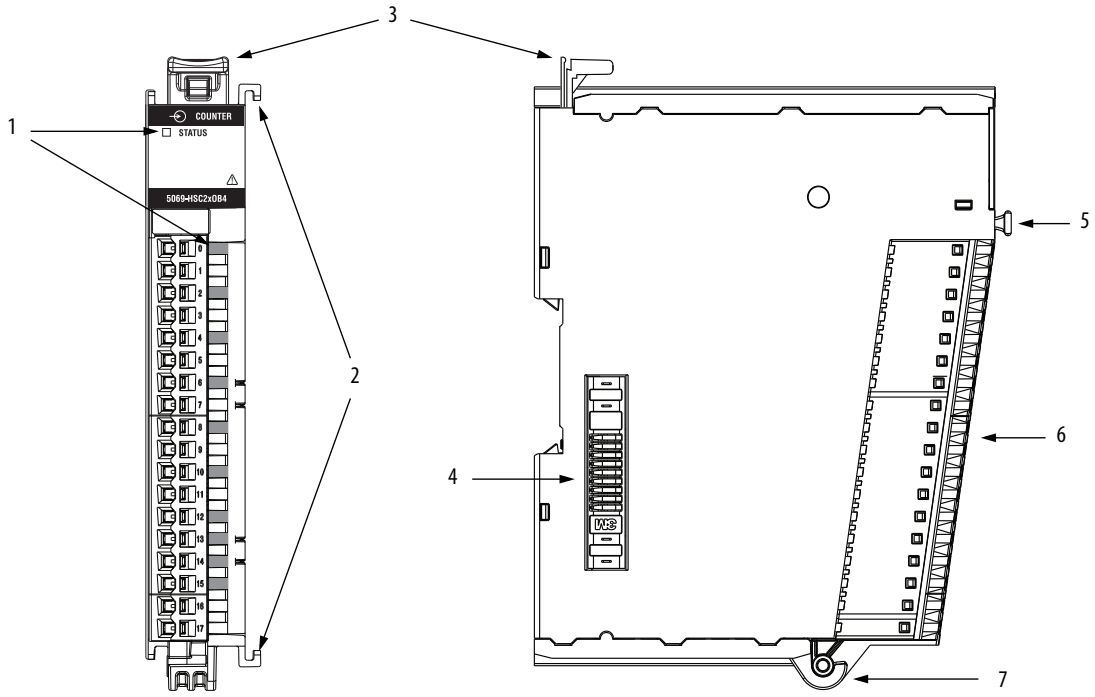
- Make sure that you have enough 5069 removable terminal blocks (RTBs) to satisfy your application needs.\

IMPORTANT	RTBs are not included with your module purchase.

Module Overview

Figure 1 shows the 5069 Compact I/O high-speed counter module.

Figure 1 - 5069 Compact I/O High-speed Counter Module



Item	Description
1	**Status indicators** - Displays the status of communication, module health, and input/output devices. Indicators help with troubleshooting anomalies.
2	**Interlocking side pieces** - Securely installs 5069 Compact I/O modules in the system.
3	**DIN rail latch** - Secures the module on the DIN rail.
4	**MOD power bus and SA power bus connectors** - Pass system-side and field-side power across the internal circuitry of the I/O modules in a 5069 Compact I/O system. The connectors are isolated from each other.
5	**RTB handle** - Anchors the RTB on the module.
6	**RTB** - Provides a wiring interface for the module.
7	**RTB lower tab** - Hooks RTB onto the module to begin installation.

Ownership

Every high-speed counter module in a Logix5000 control system must be owned by a controller, also known as the owner-controller. When a 5069 Compact I/O high-speed counter module is used in a Logix5000 control system, the owner-controller performs the following:

- Stores configuration data for the module.
- Can reside in a location different from the 5069 Compact I/O system.
- Sends the I/O module configuration data to define module behavior and begin operation in the control system.

5069 Compact I/O high-speed counter modules must continuously maintain communication with its owner-controller during normal operation. The 5069 Compact I/O high-speed counter module is limited to one owner-controller that performs the functions that are listed previously. Other controllers can establish Listen-Only connections to the 5069 Compact I/O high-speed counter module.

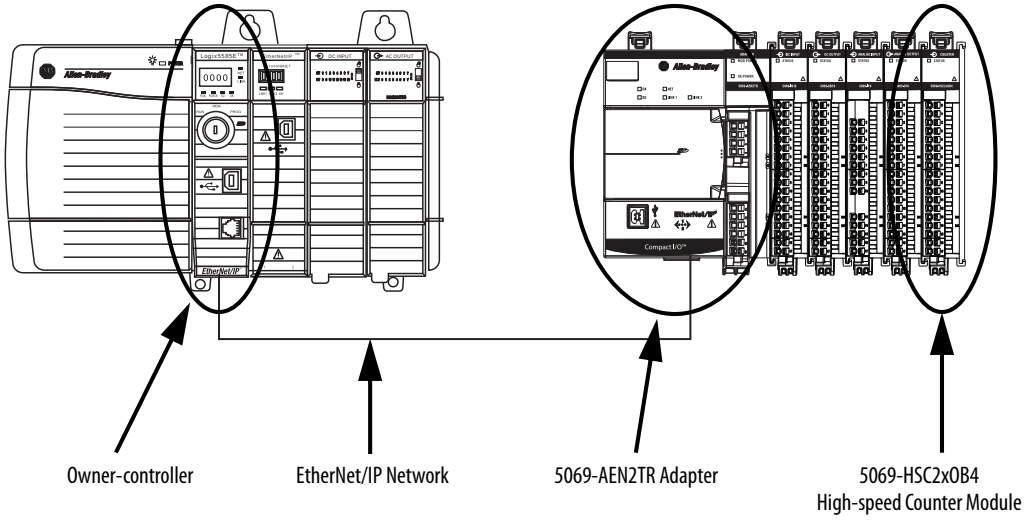
If necessary, different Logix5000 controllers can own different 5069 Compact I/O modules that reside in the same chassis. For example, one 1756-L85E controller can operate as the owner-controller for a 5069 Compact I/O module in slot 1 of a 5069 Compact I/O system. Another 1756-L85E controller can operate as the owner-controller for a 5069 Compact I/O module in slot 2 of the same 5069 Compact I/O system.

Construct a 5069 Compact I/O System

At minimum, the system must include a 5069-AEN2TR EtherNet/IP adapter. The system can contain other 5069 Compact I/O modules.

Figure 2 shows the owner-controller in 1756 ControlLogix chassis that is connected to the 5069 Compact I/O high-speed counter module via an EtherNet/IP network.

Figure 2 - 5069 Compact I/O Modules in a Logix5000 Control System



5069 Compact I/O System Power

The 5069-AEN2TR EtherNet/IP adapter provides system-side and field-side power to a 5069 Compact I/O system.

- System-side power that powers the 5069 Compact I/O system and lets modules transfer data and execute logic.
- Field-side power that powers field-side devices that are connected to some 5069 Compact I/O modules.

For more information on how to power a 5069 Compact I/O system, see the EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication ENET-UM004

Configure a 5069 Compact I/O System

You must create a Logix Designer application project for the Logix5000 controller that owns the 5069 Compact I/O high-speed counter module. The project includes module configuration data for the 5069 Compact I/O high-speed counter module.

The Logix Designer application transfers the project to the owner-controller during the program download. Data is then transferred to the 5069 Compact I/O high-speed counter module over the EtherNet/IP network.

The 5069 Compact I/O high-speed counter module can operate immediately after receiving the configuration data.

Connections

During module configuration, you must define the module. Among the Module Definition parameters, you must choose a connection type for the module. A connection is a real-time data transfer link between the owner-controller and the module that occupies the slot that the configuration references.

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

Because part of module configuration includes a slot in the 5069 Compact I/O system, the owner-controller checks for the presence of a module there. If a module is detected, the owner-controller sends the configuration. One of the following occurs:

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

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

The owner-controller monitors its connection with a module. Any break in the connection, for example, the loss of power to the 5069 Compact I/O system, causes a fault. The Logix Designer application monitors the fault status tags to indicate when a fault occurs on a module.

Connection Types Available with a 5069 Compact I/O High-speed Counter Module

When configuring a 5069 Compact I/O high-speed counter module, you must define the module. Connection is a required parameter when you define a module. The choice determines what data is exchanged between the owner-controller and the module.

Table 1 describes the connection types that you can use with a 5069 Compact I/O high-speed counter module.

Table 1 - Connections - 5069 Compact I/O High-speed Counter Module

Connection Type	Description
Data with Events	The module returns the following to the owner-controller: • General fault data • Event fault data • Input data • Event input data • Output data • Event output data
Data	The module returns the following to the owner-controller: • General fault data • Input data • Output data
Listen Only	When a Listen Only connection is used, another controller owns the module. A controller that makes a Listen Only connection to the module does not write configuration for the module. It merely listens to the data exchanged with the owner-controller. **IMPORTANT**: If a controller uses a Listen Only connection, the connection must use the Multicast option. For more information on Listen Only connections, see Listen-only Mode on page 19. In this case, all other connections to the module, for example, the connection to the owner-controller must also use the Multicast option.

You choose a Connection option in the General category on the Module Definition dialog box. For more information on how to define the module, page 72.

Requested Packet Interval

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

You set the RPI value during initial module configuration and can adjust it as necessary after module operation has begun. Valid RPI values are 0.2...750 ms.

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.

The 5069 Compact I/O high-speed counter module has two RPIs that can be used to send data. Only one rate is required. The available RPI types are as follows:

- Data RPI - Rate at which the module sends data.

  This RPI is **mandatory** regardless of what connection type you choose during module configuration.

- EventData RPI - Rate at which event data is sent.

  This RPI is only mandatory when you choose the **Data with Events** connection type. The EventData RPI is not used with other connection types.

The Connection types are described in Table 1 on page 14.

Connection Over EtherNet/IP

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

The 5069 Compact I/O high-speed counter module uses one of the following methods to transmit data:

- Multicast - Data is sent to all network devices
- Unicast - Data is sent to a specific controller depending on the module configuration

  Unicast is the default setting. We recommend that you use Unicast because it reduces network bandwidth usage.

For more information on Unicast and Multicast connections, see the EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication ENET-UM004.

Module Input Operation

Logix5000 controllers do not poll the 5069 Compact I/O high-speed counter module for input data. Instead, the modules send their input data, that is, count and status data, to the backplane at the time that is defined in the RPI.

At the RPI, not only does the module send input data to the controller, but also the controller sends data to the module inputs. For example, the controller sends data to command the module to unlatch alarms or enable alarms.

Module to Controller Input Data Transmission

The following events occur when the module sends input data to the controller at the RPI.

1. The module scans its inputs for data.
2. The module transmits the input data to the 5069 Compact I/O system backplane.
3. The 5069-AEN2TR EtherNet/IP adapter in the 5069 Compact I/O system sends the input data over the EtherNet/IP network.
4. One of the following:
   - If the owner-controller is directly connected to the EtherNet/IP network, it receives the input data from the network without need for a communication module.
   - If the owner-controller is connected to the EtherNet/IP network through another communication module, the module transmits the data to its backplane and the controller receives it.

Controller to Module Inputs Data Transmission

The following events occur when the owner-controller sends data to the module inputs:

1. One of the following:
   - If the controller is directly connected to the EtherNet/IP network, it broadcasts the data to the network.

     In this case, proceed to step 3.

   - If the controller is connected to the EtherNet/IP network through a communication module, the controller transmits the data to its backplane.

     In this case, continue at step 2.

2. The communication module transmits the data to the EtherNet/IP network.
3. The 5069-AEN2TR EtherNet/IP adapter in the 5069 Compact I/O system receives the data from the EtherNet/IP network and transmits it to the 5069 Compact I/O system backplane.
4. The 5069 Compact I/O high-speed counter module receives the data from the backplane and behaves as dictated by its configuration.

Trigger Events

A 5069 Compact I/O high-speed counter module counter can trigger as many as four events. The module can also trigger an Event task to execute in the owner-controller. The event task lets you execute a section of logic immediately when an event occurs.

For more information on event triggers, see Events on page 31.

For more information on event tasks, see the Logix5000 Controllers Tasks, Programs, and Routines Programming Manual, publication 1756-PM005.

Module Output Operation

The 5069 Compact I/O high-speed counter module outputs receive data from module windows that control the outputs. After the module outputs receive the data, they behave as commanded, for example, change from the Off state to the On state.

You use module windows to control module outputs. Input data, that is, count or frequency values, is used with the window configuration to dictate output behavior.

The following events occur when module windows send data to module outputs.

1. The window receives input data from the module counter to which it is tied.
2. Based on its configuration, the window changes the output behavior.
3. The module outputs operate as commanded to by the window controlling it.

For example, you configure window 00 as follows:

- Tied to counter 00.
- Controls output 00 and output 01.
- Output on value = 2000.
- Output off value = 5000.

When the number of counts reaches 2000, the window commands output 00 and output 01 to turn on. When the number of counts reaches 5000, the window commands the outputs to turn off.

If necessary, you can **override the state of module outputs**. For more information on how to override the state of module outputs, see Override Inputs on page 61.

Listen-only Mode

Any controller in the system can listen to the input data from a 5069 Compact I/O high-speed counter module even if the controller does not own the module.

During the I/O configuration process, you can specify a Listen Only connection. For more information on Connection options when configuring your system, see Table 1 on page 14.

When you choose a Listen Only connection, the controller and module establish communication without the controller sending configuration data. In this instance, another owner-controller owns the 5069 Compact I/O high-speed counter module.

IMPORTANT Remember the following:

- 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 to the owner-controller, must also use the Multicast option.
- 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
- 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.

Use 5069-ARM and 5069-FPD Modules

The following 5069 modules are available for unique purposes in a 5069 I/O system:

- 5069-ARM Address Reserve Module
- 5069-FPD Field Potential Distributor

5069-ARM Address Reserve Module

The 5069-ARM address reserve module reserves a node address in a 5069 Compact I/O system. The module remains installed until you insert another 5069 Compact I/O module into the same location.

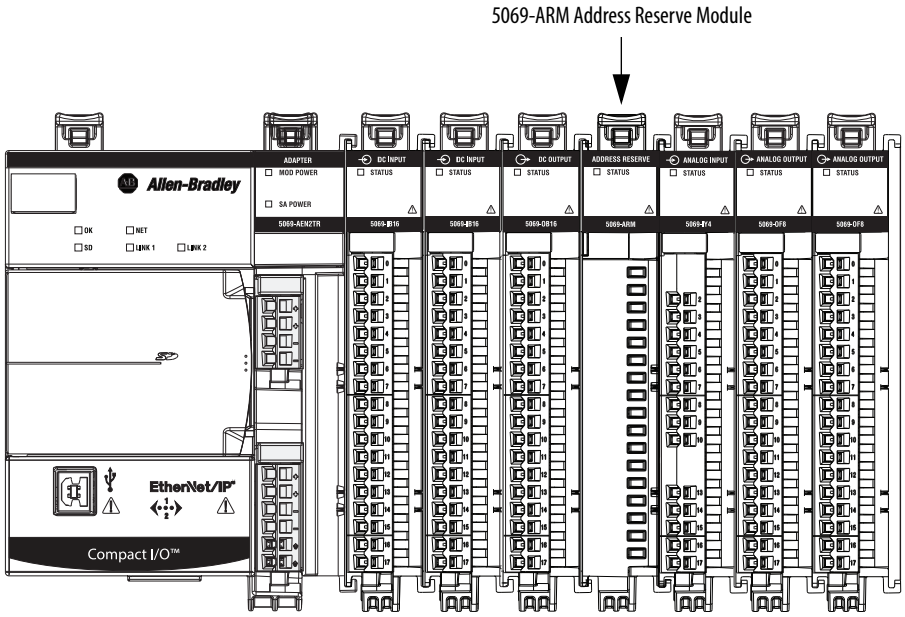
For example, your application can require the use of a 5069-HSC2xOB4 module in a specific node location. The module is typically installed when you install the 5069 Compact I/O system. In this case, however, the required 5069-HSC2xOB4 module is not available for insertion.

To install 5069 Compact I/O modules, you attach them to left-most device in the system. The node addresses increment as each module is installed. To make sure that the 5069-HSC2xOB4 module is installed in the correct location later, you install a 5069-ARM address reserve module during initial system installation.

When the required 5069-HSC2xOB4 module is available, you remove the 5069-ARM address reserve module and replace it with the 5069-HSC2xOB4 module. Thus, you insert the module in the correct node address location.

Figure 3 shows a 5069 Compact I/O system that uses a 5069-ARM address reserve module to reserve a node address.

Figure 3 - 5069 Compact I/O System with 5069-ARM Address Reserve Module



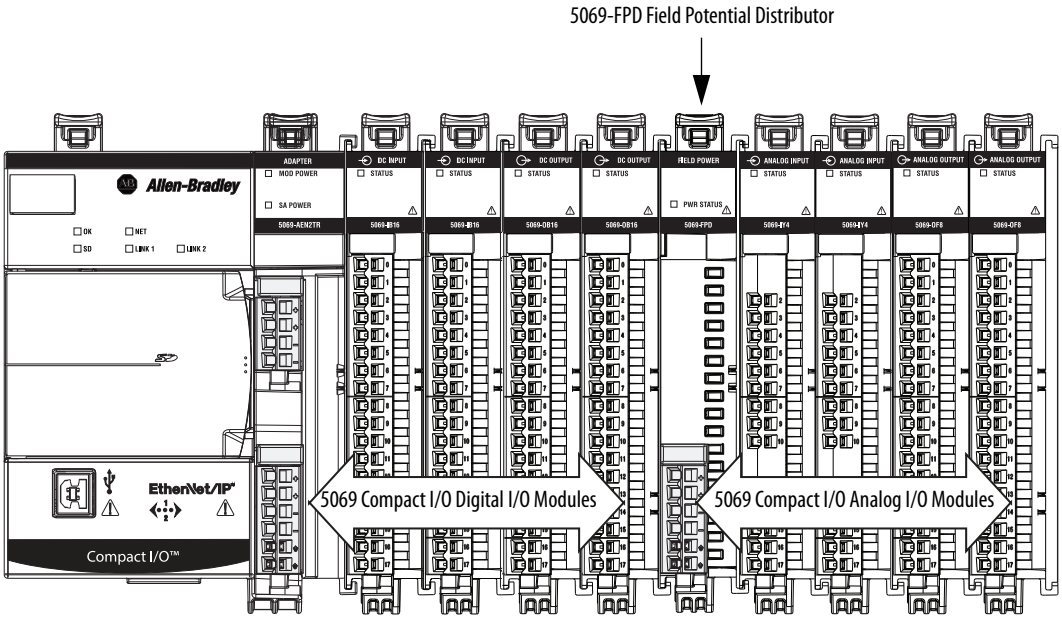
5069-FPD Field Potential Distributor

The 5069-AEN2TR EtherNet/IP adapter is the primary source of field-side power in the system. However, you can use a 5069-FPD field potential distributor to break field-side power distribution in a 5069 Compact I/O system.

Field-side power begins at the 5069 Compact I/O EtherNet/IP adapter and passes across the internal circuitry of the 5069 Compact I/O modules to the right. The field potential distributor blocks the passage of field-side power to the left of the distributor and functions as a new field-side power source for the modules to the right.

Figure 4 shows a 5069 Compact I/O system that includes a field potential distributor. In this example, the field potential distributer is used to isolate digital I/O modules from analog I/O modules regarding field-side power.

Figure 4 - 5069 Compact I/O System with 5069-FPD Field Potential Distributor



For more information on how to power a 5069 Compact I/O system, see the EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication ENET-UM004.

Protected Operations

To ensure the secure operation of your 5069 Compact I/O high-speed counter module, operations that can disrupt module operation are restricted based on the module operating mode. Table 2 describes the restrictions.

Table 2 - Protected Operations on a 5069 Compact I/O High-speed Counter Module

Current Module Operation	Activity						
	Firmware Update Request	Module Reset Request	Connection Request	Configuration Change	Connection or Data Format Change	Electronic Keying Change	RPI Change
Connection not running	Accepted						
Connection running	Rejected		Accepted(1)	Accepted(2)	Not allowed(3)	Accepted(4)	
Firmware update is in process	Rejected						

(1) Only requests for Listen Only connections are accepted.

(2) Configuration change is accepted in the following scenarios:
- Changes are made in the Module Properties dialog box and you click Apply.
- Changes are made in the Configuration tags and you send a Reconfigure Module MSG to the module.

(3) The difference between Rejected and Not allowed is that rejected activities can be attempted in the Logix Designer application but do not take effect. The activities that are not allowed, that is, attempts to change the Connection or Data Format used, are prevented from occurring in the Logix Designer application.
For example, if a module reset request is made, the Logix Designer application executes the request and alerts you that it was rejected. If a data format change is attempted, the Logix Designer application does not execute the attempted change and alert you that it was not allowed. In the case, if the change is attempted online, the Module Definition dialog box field that changes the data format is disabled.

(4) The change occurs after the connection is closed and reopened. You can close and reopen the connection in the following ways:
- Change the project while it is offline and download the updated project before going online again.
- Change the project while it is online and click Apply or OK in the Module Properties dialog box. In this case, before the change is made, a dialog box alerts you of the ramifications before the change is made.

High-speed Counter Module Features

Purpose of the 5069 Compact I/O High-speed Counter Module

The 5069 Compact I/O™ high-speed counter module uses the following interactively:

- Two counters
- Six inputs
- Eight windows
- Four outputs

The two counters are constituted of six differential inputs. The counters count pulses from devices such as encoders, proximity switches, and photoelectric sensors. The counts are presented as an accumulated count or frequency.

The signals that are received at the inputs are filtered, decoded, and counted. The module generates the rate and time-between-pulses, that is, the pulse interval, data. Count and frequency values can activate module outputs through user-defined windows.

The windows can trigger output behavior so that the output responds to input conditions at a high speed. For example, the input-to-output response is 10 μS maximum. You can control the outputs with the user-program or the module windows, based on the count value or frequency.

General Module Features

The 5069 Compact I/O high-speed counter module supports the following module-wide features:

- Module Data Quality Reporting
- Software Configurable
- Fault and Status Reporting
- Module Inhibiting
- Electronic Keying
- Producer-Consumer Communication
- Module Firmware

Module Data Quality Reporting

The 5069 Compact I/O high-speed counter module indicates the quality of channel data that is returned to the owner-controller. Data quality represents accuracy. Levels of data quality are reported via module input tags.

The following input tags indicate the level of data quality:

- I.Counter*xx*.Fault - This tag indicates that the counter data is **inaccurate** and **cannot be trusted** for use in your application. Do not use the data for control.

  If the module sets this tag to 1, you must troubleshoot the module to correct the cause of the inaccuracy.

  Typically, this tag is set when the Counter*xx*.QuadratureErrorCount tag is any number other than 0. The Counter*xx*.QuadratureErrorCount tag only applies when the high-speed counter module is operating in the X1, X2, or X4 Quadrature mode.

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

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

  The following conditions **set** the Counter*xx*.Uncertain tag to 1:

  – 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 > $T_{P\text{-}MAX}$.

    If the pulses are longer than $T_{P\text{-}MAX}$, the module assumes the input frequency is 0. However, the module cannot determine if there are no pulses being measured because no pulses exist or because the module is filtering a high frequency signal.

    $T_{P\text{-}MAX}$ = 10 s

  – Measured pulse width is too short. That is, the pulse width is less than the module's rated minimum pulse width, $T_{PW\text{-}MIN}$.

    $T_{PW\text{-}MIN}$ = 125 ns

  – If the period defined in the Counter*xx*:AverageOverPulses configuration tag is greater than $T_{P\text{-}MAX}$.

    In this case, the average frequency and average pulse width are reported based on the length of the complete pulses measured.

  The following conditions **do not set** the I.Counter*xx*.Uncertain tag to 1:

  – Zero frequency setting is out of range
  – Missing pulse is out of range
  – Overflow, Underflow, Load, Windows are out range
  – A quadrature B (AQB) fault states
  – Data signal is slightly outside the channel operating range.

We recommend that you monitor these tags in your program to make sure that the application is operating as expected with accurate channel input data.

IMPORTANT	Once the condition that causes the Fault or Uncertain tag to change to 1 is removed, the tag automatically resets to 0. The Logix Designer application controllers these 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.

Software Configurable

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

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

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

Fault and Status Reporting

The 5069 Compact I/O high-speed counter module reports fault and status data along with channel data.

IMPORTANT	Faults are only applicable when a counter is operating in one of the Quadrature modes. That is, X1 Quadrature mode, X2 Quadrature mode, or X4 Quadrature mode, as described in High-speed Counter Module Operating Modes on page 39.

Fault and status data is reported in the following ways:

- Logix Designer application
- Module status indicators

For more information on fault reporting, see Appendix A, Troubleshoot Your Module on page 89.

Module Inhibiting

Module inhibiting lets you indefinitely suspend a connection, including Listen Only connections, between an owner-controller and high-speed counter module without removing the module from the configuration. This process lets you temporarily disable a module, such as to perform maintenance.

You can use module inhibiting in the following ways:

- You write a configuration for an I/O module but inhibit the module to prevent it from communicating with the owner-controller. The owner does not establish a connection and the configuration is not sent to the module until the connection is uninhibited.

- In your application, a controller already owns a module, has downloaded the configuration to the module, and is exchanging data over the connection between the devices.

  In this case, you can inhibit the module and the connection to the module does not exist.

IMPORTANT	Whenever you inhibit a 5069 Compact I/O high-speed counter module that is ProgMode enabled, it enters Program mode, and all outputs change to the state configured for Program mode. For example, if an output module is configured so that the state of the outputs transition to zero during Program mode, whenever that module is inhibited, outputs transition to zero.

You can use module inhibiting in these instances:

- You want to update a 5069 Compact I/O high-speed counter module, for example, update the module firmware revision. Use the following procedure.
  a. Inhibit the module.
  b. Perform the update.
  c. Uninhibit the module.

- You use a program that includes a module that you do not physically possess yet. You do not want the controller to look for a module that does not yet exist. In this case, you can inhibit the module in your program until it physically resides in the proper slot.

To see where to inhibit a 5069 Compact I/O high-speed counter module, see page 77.

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, digital I/O module.
Product Code	The specific type of the product. The Product Code maps to a catalog number.
Major Revision	A number that represents the functional capabilities of a device.
Minor Revision	A number that represents behavior changes in the device.

The following Electronic Keying options are available.

Keying Option	Description
Compatible Module	Lets the installed device accept the key of the device that is defined in the project when the installed device can emulate the defined device. With Compatible Module, you can typically replace a device with another device that has the following characteristics: • Same catalog number • Same or higher Major Revision • Minor Revision as follows: – If the Major Revision is the same, the Minor Revision must be the same or higher. – If the Major Revision is higher, the Minor Revision can be any number.
Disable Keying	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. **ATTENTION**: Be extremely cautious when using Disable Keying; if used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We **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.
Exact Match	Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.

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

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

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

More Information

For more detailed information on Electronic Keying, see Electronic Keying in Logix5000 Control Systems Application Technique, publication LOGIX-AT001.

Producer-Consumer Communication

The 5069 Compact I/O high-speed counter module can produce data without a controller polling them first by using Producer-Consumer communication. 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.

IMPORTANT	Keep in mind, **only one controller** can own the 5069 Compact I/O high-speed counter module. The 5069 Compact I/O high-speed counter module does not support multiple owners of the same module. Other controllers must use a Listen Only connection to the module.

Module Firmware

The 5069 Compact I/O high-speed counter module is manufactured with module firmware installed. If updated module firmware revisions are available in the future, you can update the firmware.

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

You access updated firmware files at the Rockwell Automation® Product Compatibility and Download Center (PCDC). A link to the PCDC is available at http://www.ab.com,

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.

Module Input Features

The 5069 Compact I/O high-speed counter module inputs support the following features:

- Alarm Latching
- Events
- Input Filtering
- Missing Pulse Detection
- Zero Frequency Detection

Alarm Latching

When enabled, Alarm Latching let you latch a counter alarm in the set position once the alarm is triggered and remain set. The alarm remains set, even if the condition that causes it to occur disappears, until the alarm is unlatched.

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

- Zero Frequency Alarm
- Missing Pulse Alarm

Enable Latching

You can enable alarm latching in the following ways:

- Module Properties dialog box - You can latch alarms on the *Alarms* category. To see where to latch an alarm on the Module Properties dialog box, see page 82.

- Module tags - The alarm type determines which tag to change.

  For more information on module tags and how to use them, see Appendix B, Module Tag Definitions on page 97.

Unlatch Alarms

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

You can use the module tags to unlatch an alarm. The alarm type determines which tag to change.

For more information on module tags and how to use them, see Appendix B, Module Tag Definitions on page 97.

Alarm Latching and Unlatching While Online

Before you can latch or unlatch alarms when your Logix Designer application project is online, you must inhibit the connection to the module. You uninhibit the connection after the changes are made.

Events

You can use the Event feature to trigger up to four events and trigger an Event task to execute in the program logic. You must complete the following tasks in the Logix Designer application to use the Event feature:

- Enable the event
- Define the event
- Define the event triggers

Enable the Event

You must enable an event to use it. By default, events are disabled. For more information on how to enable an event, see Events Category on page 85.

Define the Event

You can use as many as three user-defined data values to define an event. The following data values are available:

- Stored Count
- Scaled Count
- Scaled Stored Count
- Frequency
- 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 Acceleration Average

You define events when you define a module during module configuration in the Logix Designer application. For more information, see Table 17 on page 76.

Define the Event Triggers

You must define one or more triggers for an event. The following triggers are available:

- Windows00...Windows07 - Windows are used on an individual basis. That is, eight Window triggers are available.
- Counter Load
- Counter Store
- Counter Reset
- Counter Direction
- Counter Rollunder
- Counter Rollover

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:

- Count Not In Window/Count In Window - Used with the Window*xx* event triggers.
- Low/High - Used with the Counter *xxx* event triggers.

You can also complete the following tasks:

- Latch an event
- Enable the independent point trigger option
- Configure how the input transition is used with the triggers.

To see where to configure the Events feature, see page 85.

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. Table 3 lists the filter settings and accuracy.

Table 3 - Input Filter Selections

Filter Setting	Filter Accuracy
0 ns	± 10 ns
100 ns	± 10 ns
200 ns	± 10 ns
500 ns	± 10 ns
1 µs	± 10 ns
2 µs	± 10 ns
5 µs	± 10 ns
10 µs	± 10 ns
20 µs	± 1 µs
50 µs	± 1 µs
100 µs	± 1 µs
200 µs	± 1 µs
500 µs	± 25 µs
1 ms	± 50 µs
2 ms	± 125 µs
5 ms	± 250 µs
10 ms	± 500 µs
20 ms	± 1 ms
50 ms	± 1 ms

To see where to set the input filter parameters, see page 81.

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.

   You set the Missing Pulse Alarm Limit on the Alarms category for the correct counter in the Module Properties dialog box. To see where to set the limit, see page 82.

2. Set the O.Counter*xx*.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.

   For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

When the Missing Pulse Alarm is triggered, the I.Counter*xx*.MissingPulseAlarm module 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.Counter*xx*.ZeroFrequencyAlarmLimit tag to establish the minimum frequency that is required to be measured at the input before which the Zero Frequency Alarm is triggered.

When the Zero Frequency Alarm is triggered, the I.Counter*xx*.ZeroFrequencyAlarm tag goes to 1.

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.

Module Output Features

The 5069 Compact I/O high-speed counter module outputs support the following features:

No Load Detection

No Load Detection detects when a wire is disconnected from an output that is turned off. When a No Load condition is detected, the I.Output*xx*.NoLoad tag goes to 1.

For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

Short Circuit Protection

Short Circuit Protection prevents damage that can result from the presence of greater current at an output than the maximum current level the channel can handle.

When a short circuit condition is detected, the channel turns off and the I.Output*xx*.ShortCircuit tag goes to one. The channel can turn on again in the future.

For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

Connection Fault Handling

You can configure 5069 Compact I/O high-speed counter module output behavior when a connection fault occurs, that is, the connection between the owner-controller and the high-speed counter module breaks.

You must define the following:

- Immediate output behavior when the connection breaks.
- Length of time that the output behaves as defined.
- Output behavior if the connection remains broken when the length of time that is defined previously expires.

Output Behavior Immediately After a Connection Fault

When the connection between an owner-controller and high-speed counter module breaks, the module output can behave in the following ways:

- Transition to a user-defined value, that is, turn on or off. - Default configuration is for the output to turn off.

- Hold its last state

  If you configure the module output to hold its last state, it remains at that state value until the following occurs:
  - The connection to the owner-controller is re-established.
  - The output returns to normal operation, as defined in the module configuration.

The output state remains as commanded if Fault State Duration is set to Forever.

If the Fault State Duration is set to 1, 2, 5, or 10 seconds the output state changes to a user-configurable Final Fault State after the specified time period elapses. For more information, see Final Fault State Value on page 36.

Fault State Duration After Connection Fault

If you configure the output to transition to a specific value after the connection breaks, you must define how long the output remains at the specified value before it transitions to a Final Fault 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 Fault State Duration time expires, the output transitions to user-defined Final Fault State Value.

Final Fault State Value

The Final Fault State Value defines the value to which the output goes after the Fault State Duration time expires.

Output State Once Connection is Reestablished

Once the connection between the owner-controller and high-speed counter module is reestablished, the output resumes normal operation.

High-speed Counter Module Operating Modes

This chapter describes how the 5069 Compact I/O™ high-speed counter module operates. We strongly suggest that you review this information before you use your module.

Module Overview

The 5069 Compact I/O high-speed counter module offers the following during normal operation:

- Counters
- Windows
- Inputs
- Outputs

Counters

Two Signed 32-bit counters are available on the 5069 Compact I/O high-speed counter module, that is, Counter 0 and Counter 1. The counters use the extended counter functionality. Module counters can operate in the following modes:

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

For more information on how to use the modes, see Module Counter Use on page 51.

Windows

Eight windows are available on the module, including four hardware-based and four firmware-based windows. You tie windows to one of the two module counters and to as many as four module outputs.

The windows control output behavior. For example, the window configuration determines when outputs turn On or Off and if the outputs use hysteresis. The windows can operate in different modes.

For more information on windows and how to use them, see the following:

- Windows on page 38
- Module Window Use on page 52.

Inputs

Six, high-speed differential inputs are available on the module. The inputs support two quadrature encoders with A, B, and Z inputs. You can also use X1, X2, and X4 encoder configurations to employ the capabilities of high-resolution quadrature encoders. Inputs are optically isolated from the bus and have a 3...32V DC operating range.

You can wire the inputs to use differential line drive output devices, for example differential or single-ended encoders. You can also use discrete devices, such as proximity sensors or photoelectric sensors.

For more information on module input features and how to use the inputs, see the following:

- Module Input Features on page 30
- Module Input Use on page 60

Outputs

Four outputs are available on the module. The outputs are DC sourcing and powered by a user-supplied power source. The outputs are optically isolated from the bus and have an 18...32V DC operating range. You can wire the outputs to use discrete output devices.

The outputs are electronically protected from short circuit and current overload conditions. Short Circuit status is monitored and fed back to the user program. A combination of output data, configuration data, ranges, and short circuit status determine output states.

For more information on module output features and how to use the outputs, see the following:

- Module Output Features on page 35
- Module Output Use on page 62

High-speed Counter Module Operating Modes

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

- Count/External Direction Mode
- Count/Internal Direction Mode
- Up and Down Pulses Mode
- X1 Quadrature Encoder Mode
- X2 Quadrature Encoder Mode
- X4 Quadrature Encoder Mode

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

You configure the counter mode in the Module Definition dialog box in the Logix Designer application. For more information about the Module Definition dialog box, see Module Definition on page 73.

IMPORTANT When the 5069 Compact I/O high-speed counter module powers up, at initial power-up or in a power cycle sequence, the following occurs:

- Output array values are set to their default values.
- Configuration array values are set to their default values.
- Input array values are cleared.
- Stored counts and configurations are cleared.
- Faults and flags are cleared.
- Outputs turn off.

Before you learn about the modes within which a 5069 Compact I/O high-speed counter module can operate, you must learn conceptual information about the module.

Counter Enable/Disable

You can enable or disable the counter with the O.Counter*xx*.Hold module output tag. The module begins counting as soon as it is enabled.

- To enable the counter and let counting occur, set the tag to 0, the default.
- To disable the counter and stop counter, set the tag to 1.

  When the counter is disabled, the count value at the time the counter was disabled is held, regardless of incoming input data.

For more information on how to use the module tags, see Appendix B, Module Tag Definitions on page 97.

Counter and Input Relationship

Table 4 describes the relationship between the module counters and inputs.

Table 4 - Counters and Inputs

Counter Input	Counter Number	Input Terminal Number[1]	Role in Modes		
			Count/Direction[2]	Up and Down Pulse	Quadrature
A	0	0	The Count input.	The Up input.	The count inputs.
	1	6			
B	0	2	The Direction input. In the Count/Internal Direction mode, you cannot use the B input for other purposes.	The Down input.	
	1	8			
Z	0	4	You can tie the following functions of each counter to the Z input: • Hold • Load • Reset • Store Load and Reset are mutually exclusive.		
	1	10			

(1) To see how input terminal numbers and organized on the module, see Figure 17 on page 89.

(2) The Count/External Direction Input and Count/Internal Direction Input selections constitute this mode.

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 either of the following ways in the Logix Designer application:

- Module Properties dialog box - To see where to set the Invert Counter Direction feature, see page 80.
- Module tags - Change the C.Counter*xx*.InvertDirection tag. A tag value of 0 disables the feature, and a tag value of 1 enables the feature.

  For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

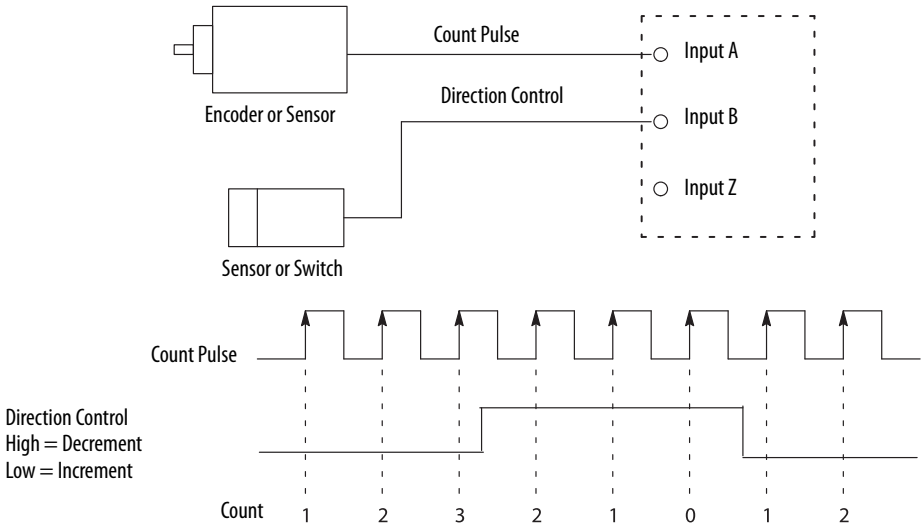
How the feature changes count direction is specific to the operating modes. The operating mode descriptions in the rest of this chapter include how the Invert Counter Direction feature affects the mode.

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 5 - Count/External Direction Mode (Invert Counter Direction Disabled)



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.

IMPORTANT Counting on a rising edge does not occur in the following conditions:

- The count up and count down pulses are asynchronous to each other.
- The rising edge of one count pulse can occur simultaneously as the falling edge of the other.

Table 5 - Count/External Direction Counting

Invert Counter Direction Setting	Input A (count)	Input B (direction)	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	0 or open	1
	↑	1	-1
	↑	↑	0
	0, 1	N/A	0
Enabled (InvertDirection tag = 1)	↑	0 or open	-1
	↑	1	1
	↑	↑	0
	0, 1	N/A	0

Count/Internal Direction Mode

In this mode, the Invert Counter Direction setting determines the direction of the counter.

- The counter increments on the rising edge of the input A when Invert Counter Direction is disabled.
- The counter decrements on the rising edge of the input A when Invert Counter Direction is enabled.

Table 6 - Count/Internal Direction Counting - Counters 0 and 1

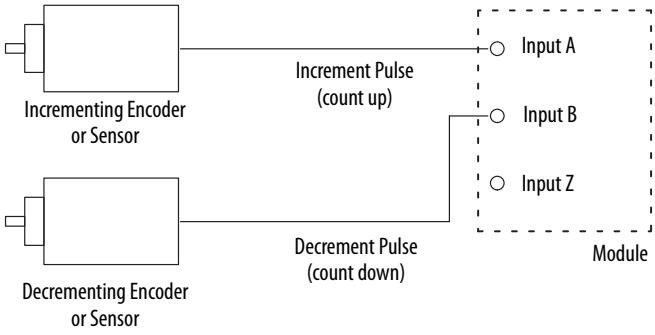
Invert Counter Direction Setting	Input A (count)	Input B	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	N/A	1
	0, 1		0
Enabled (InvertDirection tag = 1)	↑		-1
	0, 1		0

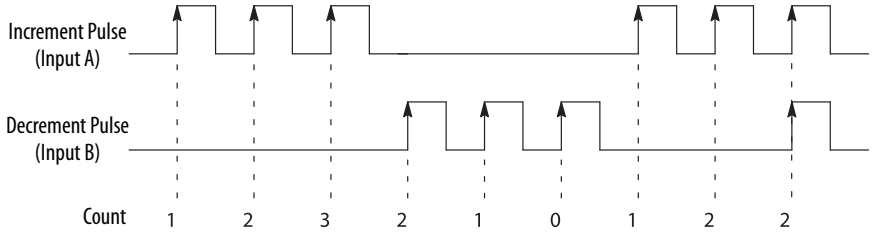
Up and Down Pulses Mode

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 6 - 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.

When the Invert Counter Direction and Direction Inhibit are both enabled, the counter decrements on the rising edge of pulses that are applied to input A or input B.

IMPORTANT Counting on a rising edge does not occur in the following conditions:

- The count up and count down pulses are asynchronous to each other.
- The rising edge of one count pulse can occur simultaneously as the falling edge of the other.

Table 7 - Up and Down Counting

Invert Counter Direction Setting	Input A (increment)	Input B (decrement)	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	0, 1	1
	0, 1	↑	-1
	↑	↑	0
Enabled (InvertDirection tag = 1)	↑	0, 1	-1
	0, 1	↑	1
	↑	↑	0

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. If input A leads input B, the counter increments. If input B leads input A, the counter decrements.

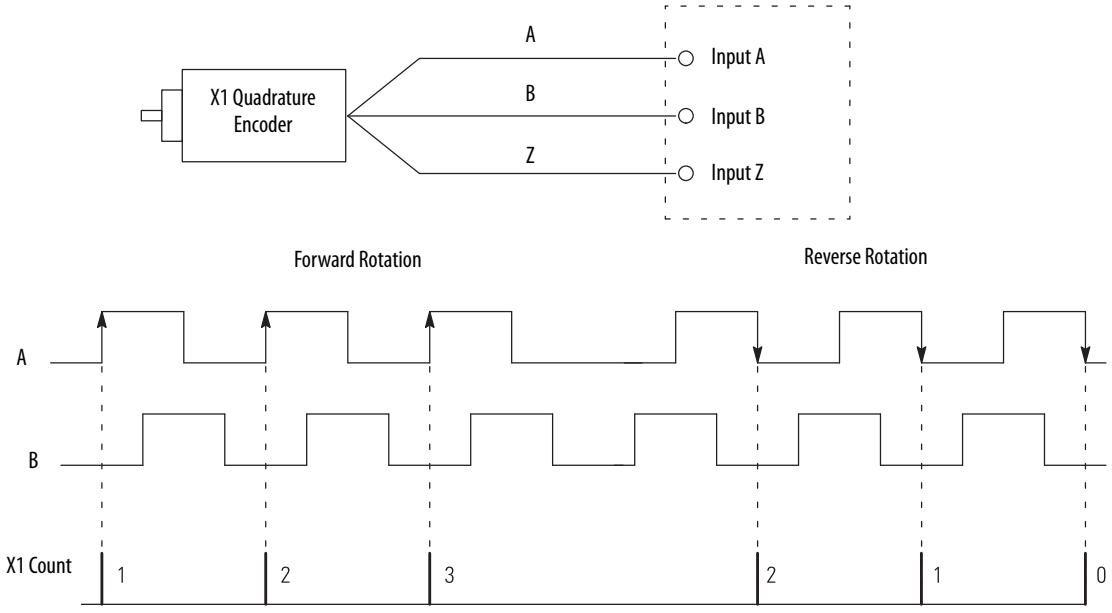
When **Invert Counter Direction is disabled**, the following occurs:

- If input B is low, the counter increments on the rising edge of input A and decrements on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 7.

TIP 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.Counter*xx*.QuadratureErrorCount tag increments.

Figure 7 - X1 Quadrature Encoder Mode with Invert Counter Direction Disabled



When **Invert Counter Direction is enabled**, the following occurs:

- If input B is low, the counter decrements on the rising edge of input A and increments on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 8.

Figure 8 - X1 Quadrature Encoder Mode with Invert Counter Direction Enabled

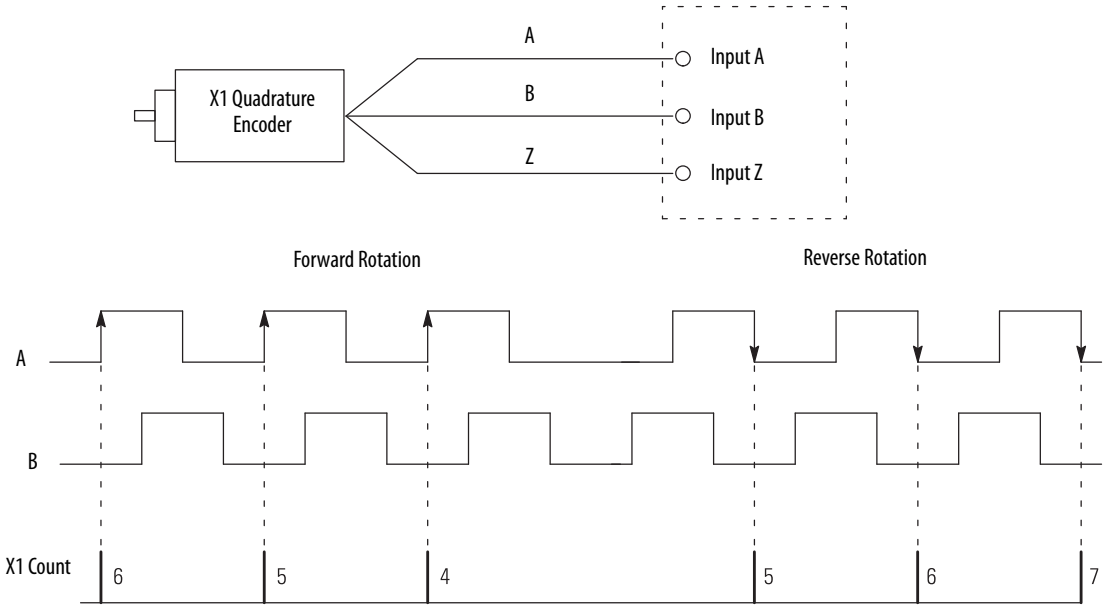


Table 8 - X1 Quadrature Direction Counting

Invert Counter Direction Setting	Input A	Input B	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	0	1
	↓	0	-1
	↑	1	0
	↓	1	
Enabled (InvertDirection tag = 1)	↑	0	-1
	↓	0	1
	↑	1	0
	↓	1	

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. If input A leads input B, the counter increments. If input B leads input A, the counter decrements.

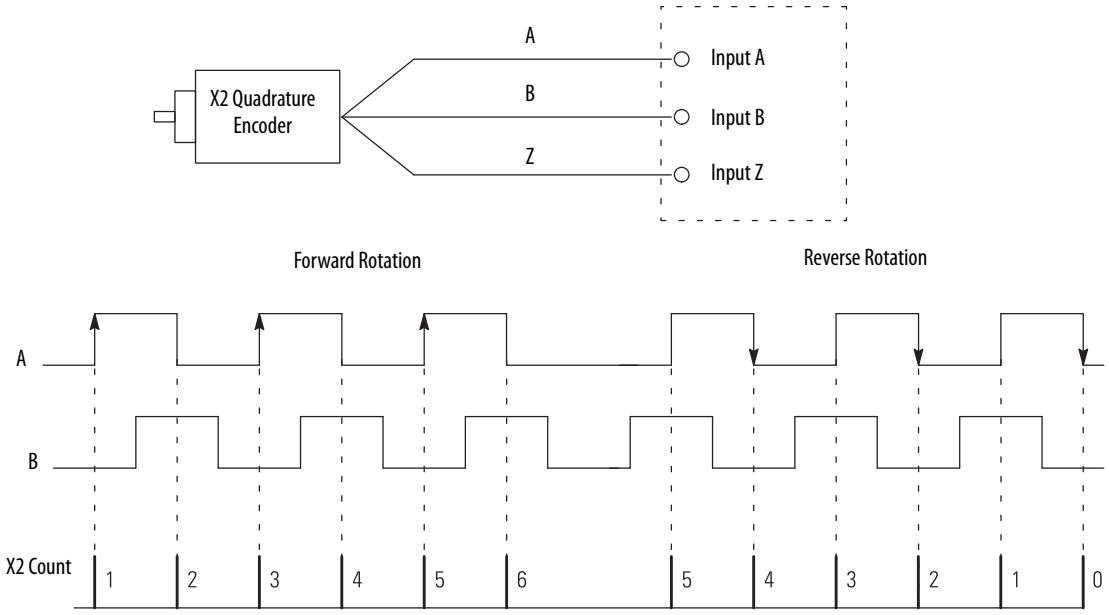
When **Invert Counter Direction is disabled**, the following occurs:

- If input B is low, the counter increments on the rising edge of input A and decrements on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 9.

TIP 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.Counter*xx*.QuadratureErrorCount tag increments.

Figure 9 - X2 Quadrature Encoder Mode with Invert Counter Direction Disabled



When **Invert Counter Direction is enabled**, the following occurs:

- If input B is low, the counter decrements on the rising edge of input A and increments on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 10.

Figure 10 - X2 Quadrature Encoder Mode with Invert Counter Direction Enabled

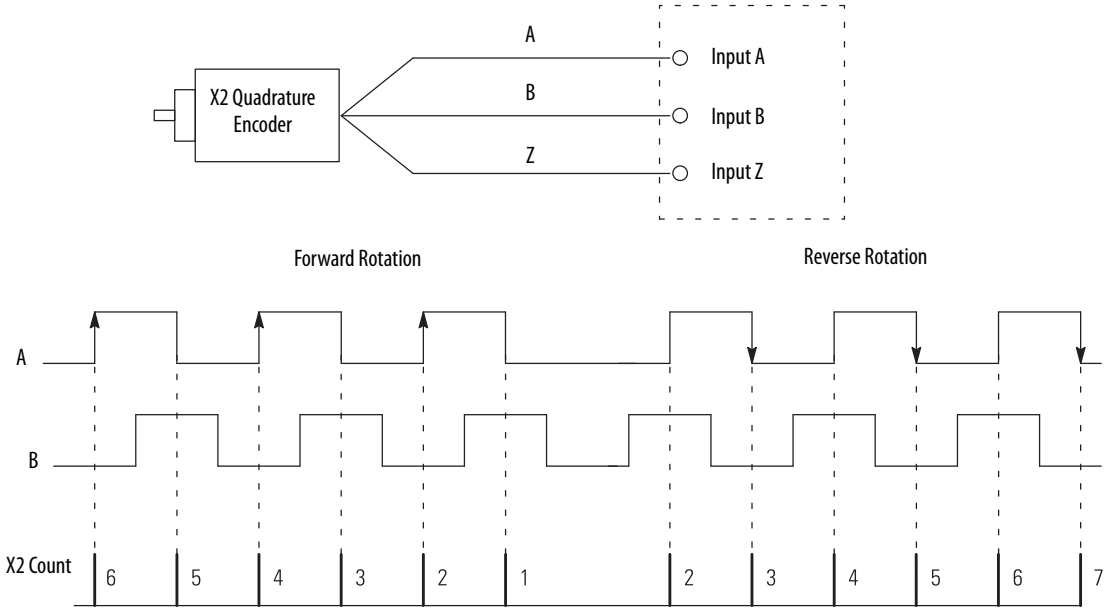


Table 9 - X2 Quadrature Direction Counting

Invert Counter Direction Setting	Input A	Input B	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	0	1
	↓	0	-1
	↑	1	0
	↓	1	
Enabled (InvertDirection tag = 1)	↑	0	-1
	↓	0	1
	↑	1	0
	↓	1	

X4 Quadrature Encoder Mode

The X4 Quadrature Encoder mode operates much like the X4 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. If input A leads input B, the counter increments. If input B leads input A, the counter decrements.

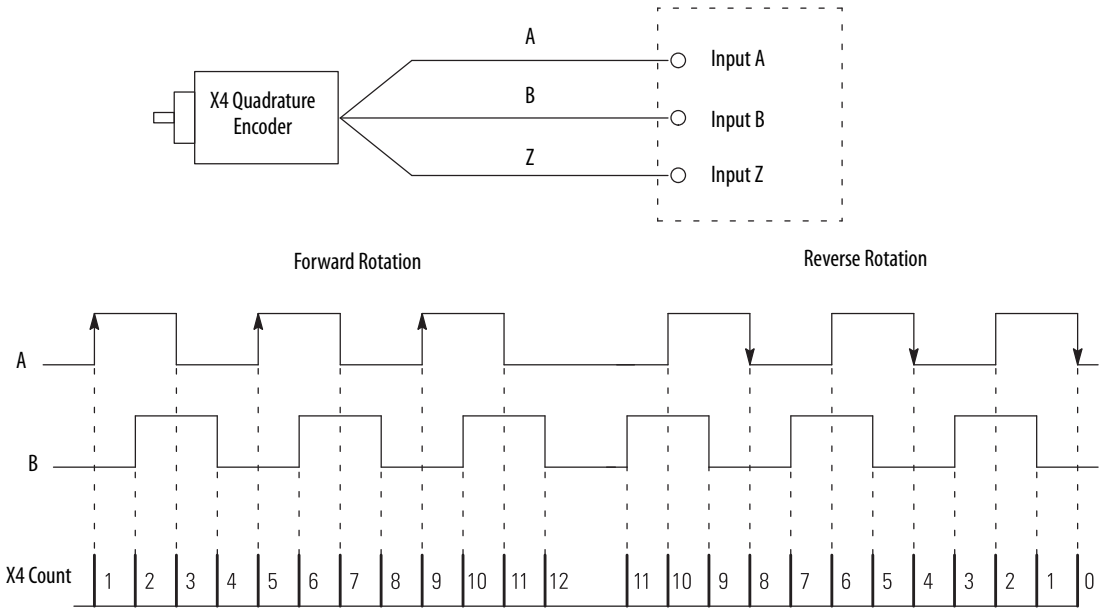
When **Invert Counter Direction is disabled**, the following occurs:

- If input B is low, the counter increments on the rising edge of input A and decrements on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 11.

TIP 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.Counter*xx*.QuadratureErrorCount tag increments.

Figure 11 - X4 Quadrature Encoder Mode with Invert Counter Direction Disabled



When **Invert Counter Direction is enabled**, the following occurs:

- If input B is low, the counter decrements on the rising edge of input A and increments on the falling edge of input A.
- If input B is high, all rising transitions on input A are ignored. The counter changes value only on one edge of input A as shown in Figure 12.

Figure 12 - X4 Quadrature Encoder Mode with Invert Counter Direction Enabled

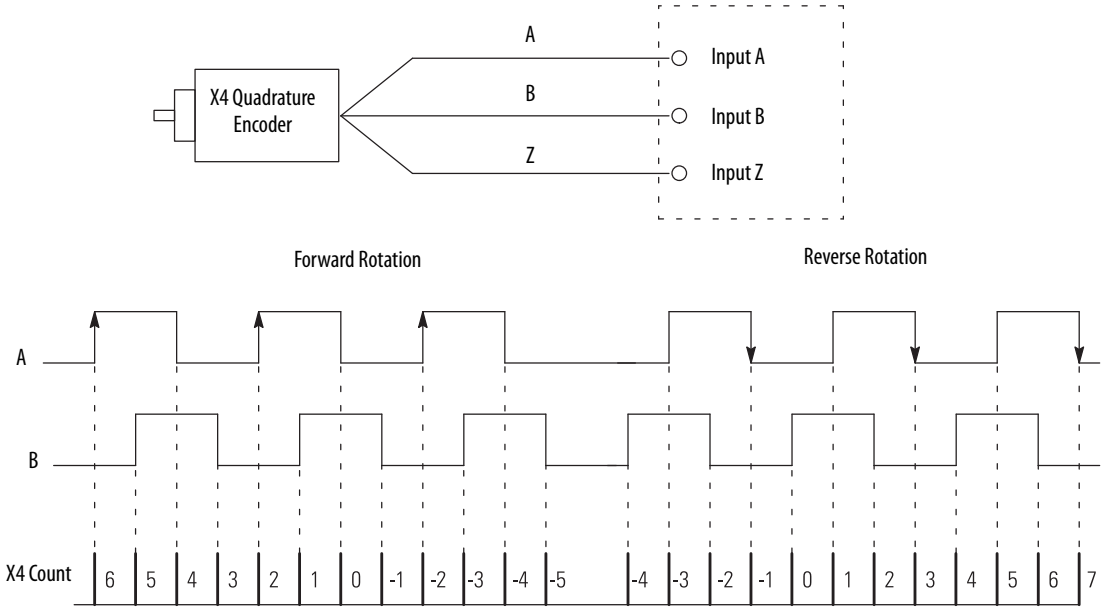


Table 10 - X4 Quadrature Direction Counting

Invert Counter Direction Setting	Input A	Input B	Change in Count Value
Disabled (InvertDirection tag = 0)	↑	0	1
	↓	0	-1
	↑	1	0
	↓	1	
Enabled (InvertDirection tag = 1)	↑	0	-1
	↓	0	1
	↑	1	0
	↓	1	

Module Counter Use

This section describes how to use the counters on a 5069 Compact I/O high-speed counter module.

Ring Counter Type

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

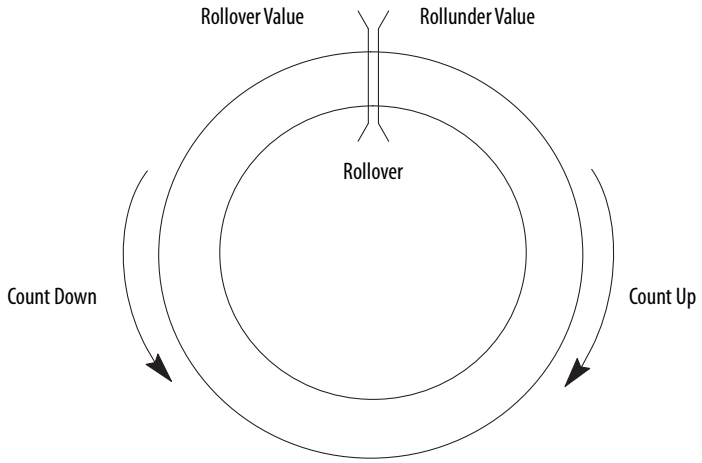
The values are represented in the module tags as follows:

- Current count = I.Counter*xx*.Count tag
- Rollover count = O.Counter*xx*.RolloverValue tag
- Rollunder count = O.Counter*xx*.RollunderValue tag

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 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.

Figure 13 - Ring Counter Diagram



Revolution Counter

Each counter supports a Revolution Counter that counts the Rollover and Rollunder transitions.

Module Window Use

The 5069 Compact I/O high-speed counter module has eight windows that you can configure to control outputs and associated window status tags.

The following types of windows are available on the 5069 Compact I/O high-speed counter module:

- Hardware-based windows - Windows 0...3
- Firmware-based windows - Windows 4...7

IMPORTANT Windows that are configured for Acceleration mode are firmware-based, regardless of the window number.

You can configure windows to operate in the following modes:

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

  Each value above scaled to user units, excluding pulse width

To see where to define the mode for a window, see page 75.

Tie Windows to Counters and Outputs

You can tie windows to the following:

- **Module counters** - You can tie a window to either counter but not both. You can tie as many as eight windows to a counter.

- **Module outputs** - You can tie a window to as many as four outputs. A window that is tied to an output controls the output and triggers its behavior as determined by the module configuration and system conditions.

  If any of the windows that are tied to an output is on, the output is on.

IMPORTANT The following are ways to control the output:
- Window - On module
- Override - Via user program

If a Window controls an output, the user program can override it.

If an Override controls an output, a Window **cannot** change it.

To see where to tie a window to a counter and outputs, see page 83.

Output State Change

You can configure a window to turn an output On or Off based on configurable count, frequency/pulse width/acceleration value designation and the module count. The state change, that is, from Off to On or On to Off, occurs in less than 10 µs for hardware-based windows.

To configure an output to turn On or Off, complete the following steps:

1. Tie the window to the desired counter.
2. Set the Output On value in the O.Window*xx*.On tag.

   This value represents the number of counts that must be reported to the window to trigger an output state change from Off to On.
3. Set the Output Off value in the O.Window*xx*.Off tag.

   This value represents the number of counts that must be reported to the window to trigger an output state change from On to Off.
4. Select the outputs that the window controls.

   To see where to complete the window configuration, see page 83.

For example, the following occurs when a window operates in Count mode:

- When the count value is within the range that is created by the O.Window*xx*.On and O.Window*xx*.Off tags, the output turns on.
- When the count value is outside the range that is created by the O.Window*xx*.On and O.Window*xx*.Off tags, the output turns 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.Window*xx*.HysteresisOn
- O.Window*xx*.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.

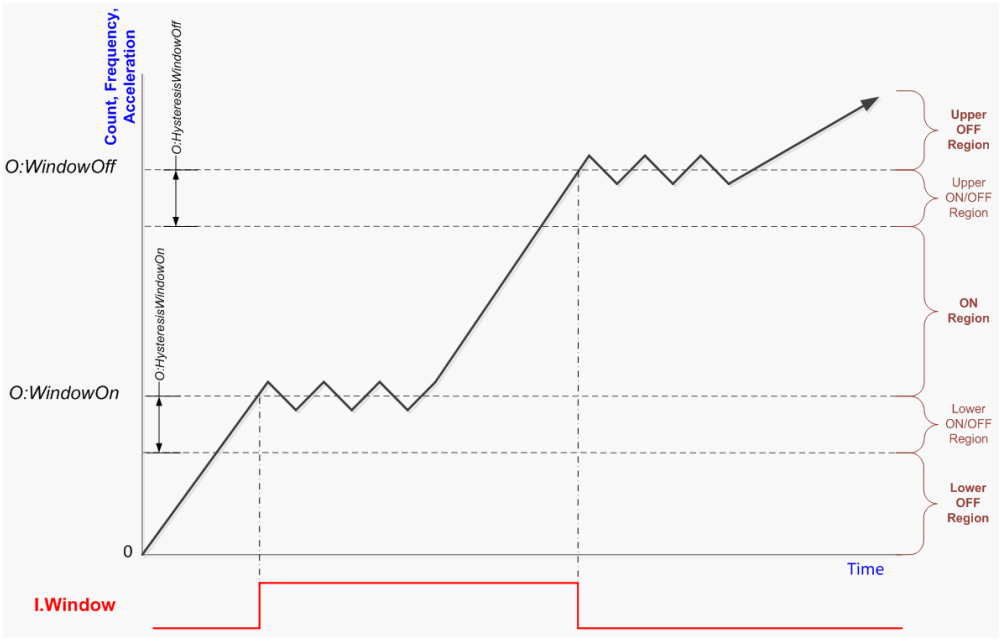
Hysteresis Example 1

Figure 14 shows a window with hysteresis as the source of the window, that is, count, frequency, or acceleration, varies near the window switching points.

The number of I.Window*xx*.InWindow tag transitions are reduced and O:Window*xx*.On tag < O:Window*xx*.Off tag. If O:Window*xx*.On tag > O:Window*xx*.Off tag, the I.Window*xx*.InWindow tag state is inverted.

The O:Window*xx*.HysteresisOn tag value is a negative offset from the O:Window*xx*.On tag value, and O:Window*xx*.HysteresisOff tag is a negative offset from the O:Window*xx*.Off tag

Figure 14 - Hysteresis Example 1



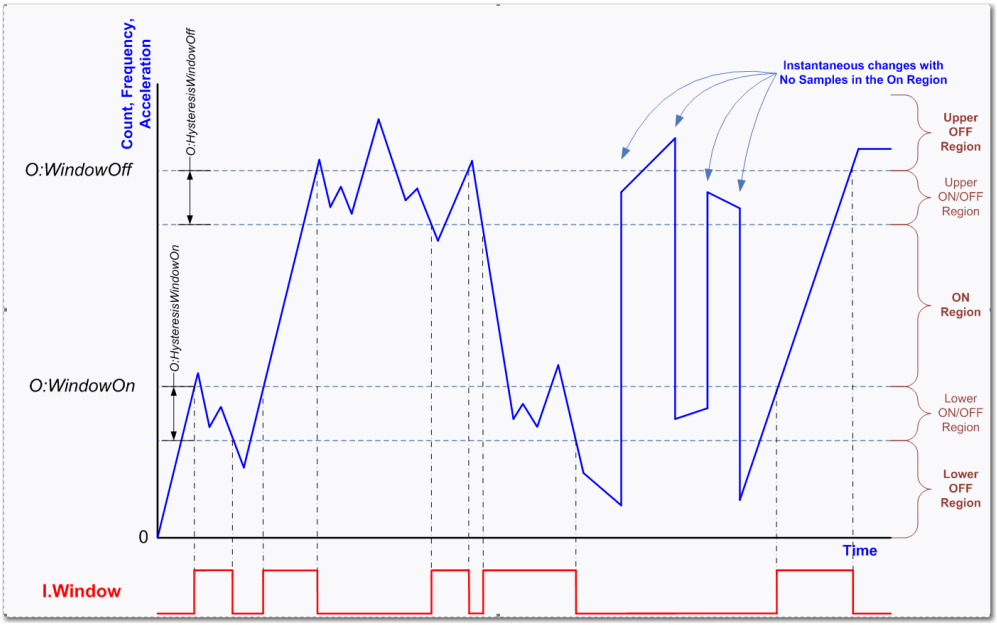
Hysteresis Example 2

Figure 15 is similar to the example on page 54. This example shows the window with hysteresis as the source of the window, that is, count, frequency, or acceleration, varies near the window switching points.

The number of I.Window*xx*.InWindow tag transitions are reduced and O:Window*xx*.On tag < O:Window*xx*.Off tag. If O:Window*xx*.On tag > O:Window*xx*.Off tag, the I.Window*xx*.InWindow tag state is inverted.

The O:Window*xx*.HysteresisOn tag is a negative offset from the O:Window*xx*.On tag, and the O:Window*xx*.HysteresisOff tag is a negative offset from the O:Window*xx*.Off tag.

Figure 15 - Hysteresis Example 2



The I.Window*xx*.InWindow tag does not always change state when the window source (count, frequency, or acceleration) equals the O:Window*xx*.On tag or the O:Window*xx*.Off tag.

The sampled widow source value falls into one of five regions. Table 11 defines the regions for when O:Window*xx*.On tag < O:Window*xx*.Off tag.

Table 11 - Regions (O:Window*xx*.On tag < O:Window*xx*.Off tag)

Region	In a region that is based on the count, frequency, or acceleration value X.
Upper OFF Region	X >= O:Window*xx*.Off tag
Upper ON/OFF Region	X >= (O:Window*xx*.Off tag – O:Window*xx*.HysteresisOff tag) and X < O:Window*xx*.Off tag
ON Region	X >= O:Window*xx*.On tag and X < (O:Window*xx*.Off tag – O:Window*xx*.HysteresisOff tag)
Lower ON/OFF Region	X >= (O:Window*xx*.On tag – O:Window*xx*.HysteresisOn tag) and X < O:Window*xx*.On tag
Lower OFF Region	X < (O:Window*xx*.On tag – O:Window*xx*.HysteresisOn tag)

The I.Window*xx*.InWindow tag value only changes when the sampled value is in an ON or OFF region. The I.Window*xx*.InWindow tag value does not change when the sampled value is in an ON/OFF region.

When the sampled value transitions from one region to another region, the logic that is shown in Table 12 is followed. The **bolded** values show the I.Window*xx*.InWindow tag transitions.

Table 12 - Logic Followed When Sample Value Transitions Between Regions

		To Region				
From Region	**From Condition**	**Lower OFF Region**	**Lower ON/OFF Region**	**ON Region**	**Upper ON/OFF Region**	**Upper OFF Region**
Upper OFF Region	0	0	0	**1**	0	0
Upper ON/OFF Region	0	0	0	**1**	0	0
Upper ON/OFF Region	1	**0**	1	1	1	**0**
ON Region	1	**0**	1	1	1	**0**
Lower ON/OFF Region	1	**0**	1	1	1	**0**
Lower ON/OFF Region	0	0	0	**1**	0	0
Lower OFF Region	0	0	0	**1**	0	0

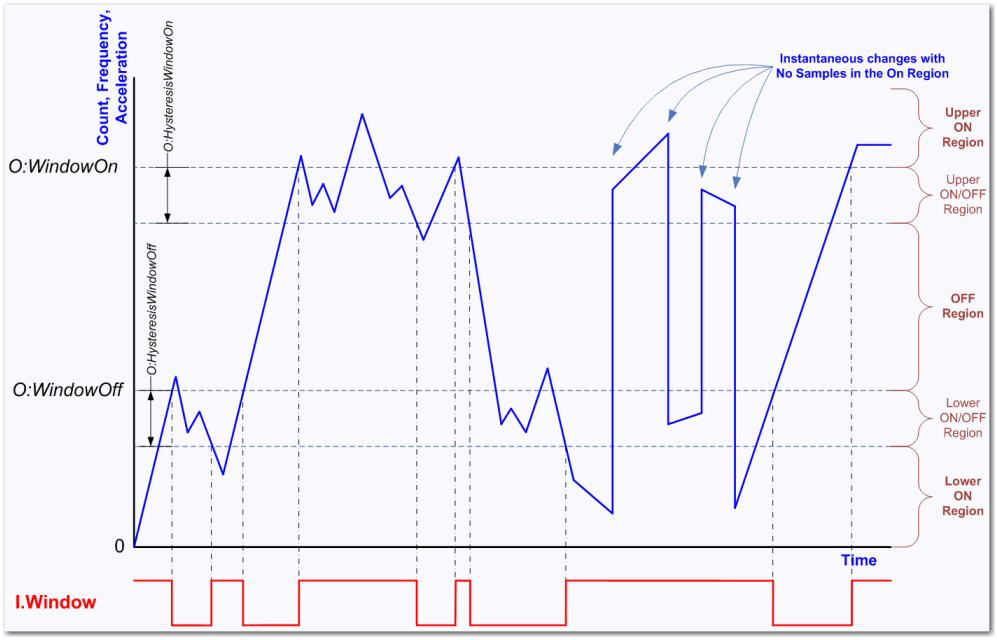
Hysteresis Example 3

Table 13 defines the regions for when O:Window*xx*.On tag > O:Window*xx*.Off tag.

Table 13 - Regions (O:Window*xx*.On tag > O:Window*xx*.Off tag)

Region	In a region that is based on the count, frequency, or acceleration value X.
Upper ON Region	X >= O:Window*xx*.On tag
Upper ON/OFF Region	X >= (O:Window*xx*.On tag – O:Window*xx*.HysteresisOn tag) and X < O:Windowxx.On tag
OFF Region	X >= O:Window*xx*.Off tag and X < (O:Window*xx*.On tag – O:Window*xx*.HysteresisOn tag)
Lower ON/OFF Region	X >= (O:Window*xx*.Off tag – O:Window*xx*.HysteresisOff tag) and X < O.WindowOff
Lower ON Region	X < (O:Window*xx*.Off tag – O:Window*xx*.HysteresisOff tag)

Figure 16 - Hysteresis Example 3



The I.Window*xx*.InWindow tag value only changes when the sampled value is in an ON or OFF region. The I.Window*xx*.InWindow tag value does not change when the sampled value is in an ON/OFF region.

When the sampled value transitions from one region to another region, the logic that is shown in the following table is followed. The **bolded** values show the I.Window*xx*.InWindow tag transitions.

		To Region				
From Region	**From Condition**	**Lower ON Region**	**Lower ON/OFF Region**	**OFF Region**	**Upper ON/OFF Region**	**Upper ON Region**
Upper ON Region	1	1	1	**0**	1	1
Upper ON/OFF Region	1	1	1	**0**	1	1
Upper ON/OFF Region	0	**1**	0	0	0	**1**
OFF Region	0	**1**	0	0	0	**1**
Lower ON/OFF Region	0	**1**	0	0	0	**1**
Lower ON/OFF Region	1	1	1	**0**	1	1
Lower ON Region	1	1	1	**0**	1	1

Manipulate Count Value

The count value that is indicated the I.Counter*xx*.Count tag. You can use the Z input functions to manipulate the count value. The following are available Z input functions:

- Store
- Hold
- Load
- Reset

You can configure the Z input function on the Module Definition dialog box or in the module tags associated with each function. For example, you can change the O:Counter*xx*.Store tag to configure the Z input for Store.

The Z input can be configured as active high or active low by inverting the input. You can configure the Z input to be inverted on the Module Definition dialog box or by changing the C.Counter*xx*.InvertInputZ tag to 1.

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.Counter*xx*.<StoredTag> tag.

You can trigger the Store function with the O.Counter*xx*.Store tag or the Z input. You can use the Z-input to capture the current count value even when the counter is counting at full 1 MHz speed.

Hold

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

Load

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

To load a user-defined count value, change the O.Counter*xx*.Load from 0 to 1. When the tag changes, the value of the O.Counter*xx*.LoadCountValue tag is copied to the I.Counter*xx*.Count tag.

If the O.Counter*xx*.Load tag is 1 and you need to load a new count value, you must change the O.Counter*xx*.Load tag to 0 and back to 1.

Reset

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

To reset the counter, change the O.Counter*xx*.Reset tag to 1. If the O.Counter*xx*.Reset tag is 1 and you need to load a new count value, you must change the O.Counter*xx*.Reset tag to 0 and back to 1.

When the Counter*xx*.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

The O.Counter*xx*.Reset tag or the Z input can set and trigger the Reset function.

Invert Z Input

The Z-input signals can be inverted. To see where to invert the Z input signal, see page 81.

Scalar

The Scalar function lets you perform the following:

- Scale count, frequency, and acceleration values to user units by multiplying their values in pulses by the Scaling value.
- Scales position, velocity, and acceleration/deceleration values to user units. Units are pulses per user unit.

Module Input Use

The 5069 Compact I/O high-speed counter module has six, high-speed differential inputs. The inputs support two quadrature encoders with A, B, and Z inputs. In addition, x1, x2, and x4 encoder configurations are provided to use the capabilities of high-resolution quadrature encoders. Inputs are optically isolated from the bus and have an operating range of 3...32V DC.

You can wire the inputs to use differential line drive output devices, for example differential or single-ended encoders. You can also use discrete devices, such as proximity sensors or photoelectric sensors.

The 5069 Compact I/O high-speed counter module uses six inputs. Three of the inputs are used with each counter. Inputs A0, B0 and Z0 are used with counter 0. Inputs A1, B1, and Z1 are used with counter 1.

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

The maximum input frequency is 1 MHz. Effectively, this value translates to 4 MHz in the Quadrature X4 mode.

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.Counter*xx*.OverrideDataxEn - Set the tag to 1 to enable the O.Counterxx.OverrideDataxValue to override the input state.
- O.Counter*xx*.OverrideDataxValue - Set the tag to the desired value.

For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

Module Output Use

You can use module windows or program logic to control the 5069 Compact I/O high-speed counter module outputs. The following can determine an output state:

- Count - Window-controlled outputs only
- Frequency - Window-controlled outputs only
- Short circuit status
- Safe state settings and conditions
- Pulse width - Window-controlled outputs only
- Acceleration - Window-controlled outputs only

For more information on how to use a module window to turn an output On or Off, see Output State Change on page 53.

Output Mode In Program Mode, Fault Mode or Upon Communication Failure

You must also configure an output to go to a commanded state if any of the following occurs:

- Controller in Program Mode - You can command the output to hold its last state, turn off, or turn on.
- Controller in Fault Mode - You can command the output to hold its last state, turn off, or turn on.
- Communication fails between the owner-controller and the 5069 Compact I/O high-speed counter module - You can command the output to Program Mode or Fault Mode.

  You must configure parameters that apply if the controller transitions to fault mode due to a communication failure and stays in fault mode beyond a user-defined period.

  For more information, see Connection Fault Handling on page 35.

To see where to configure the output state when the controller enters program mode or fault mode or if communication between the owner-controller and the module fails, see page 84.

Overriding Outputs

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

For more information about module tags, see Appendix B, Module Tag Definitions on page 97.

Notes:

Configure the High-speed Counter Module

This chapter describes how to configure your 5069 Compact I/O™ high-speed counter module in a Logix Designer application project. You can use the default module configuration or edit the module configuration.

IMPORTANT Consider the following:

- You must use the Logix Designer application, **version 28 or greater**, to configure the 5069 Compact I/O high-speed counter module. Version 28 or greater is slightly different from previous programming software versions. For example, in some cases, instead of tabs across the top of the Module Properties dialog box, the application uses categories on the left side of the dialog box.
- This chapter does not explain the user-configurable module features that you can edit on different screens in your Logix Designer application project.

  For detailed information about module features, see the following:
  - Chapter 2, High-speed Counter Module Features on page 23
  - Chapter 3, High-speed Counter Module Operating Modes on page 37

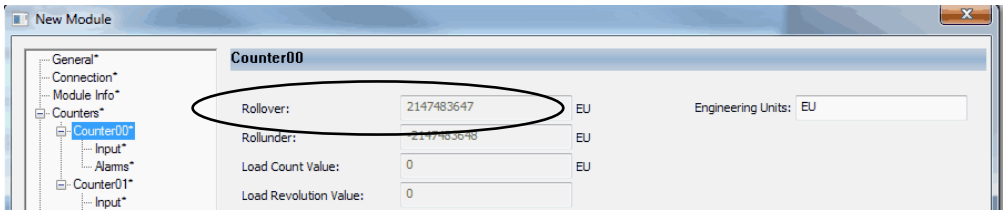
Configuration Methods

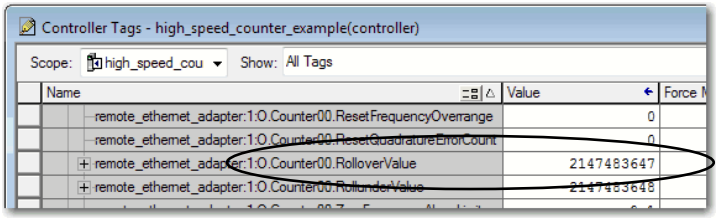
You configure the 5069 Compact I/O high-speed counter module with the Logix Designer application. The following configuration methods are available:

- Module Properties dialog box
- Module tags

You can configure most parameters with either method. You can configure some parameters with only the module tags. If a parameter is disabled, you must configure it with the module tags.

For example, the Rollover value for Counter 00 is read only on the Module Properties dialog box. You must change the O.Counter00.RolloverValue tag to change the Rollover value.





Before You Begin

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

1. Create a Logix Designer application project.

   The example in this chapter uses a 1756-L85E ControlLogix® controller.

2. Add a 5069-AEN2TR EtherNet/IP adapter to the project.

   For more information on how to add a 5069-AEN2TR EtherNet/IP adapter, see the EtherNet/IP Communication Modules in 5000 Series Systems User Manual, publication ENET-UM004.

Create a New Module

After you create a Logix Designer application project that includes a 5069-AEN2TR EtherNet/IP adapter, you can use the following methods to add modules to the project.

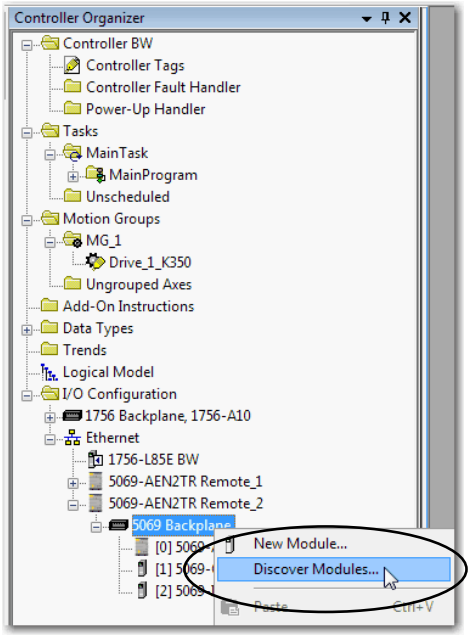
- Discover Modules
- New Module

Discover Modules

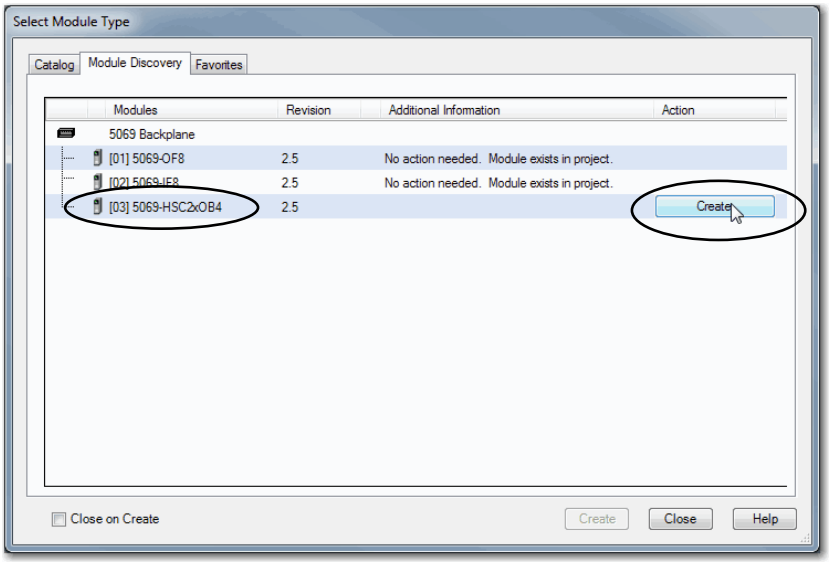
To add a module using Discover Modules, perform the following steps.

1. Go online with your Logix Designer application.
2. Right-click the 5069-AEN2TR EtherNet/IP adapter and choose Discover Modules.

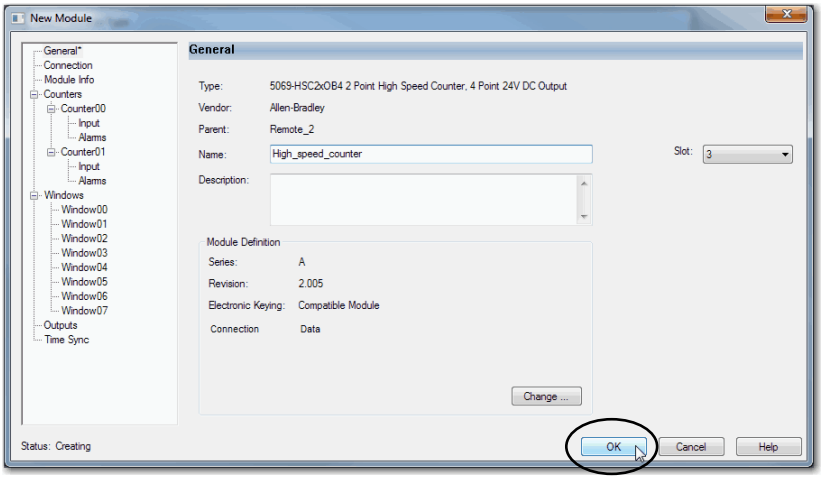
   The Logix Designer application automatically detects available modules that are connected to the backplane.



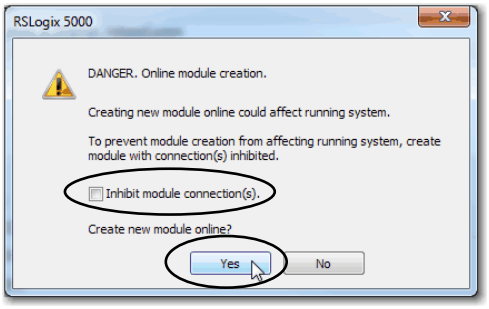
3. At the Select Module Type window, click Create to add the discovered module to your project.



4. At the New Module window, configure the module properties and click OK.



5. At the warning dialog box, make sure that Inhibit module connection(s) is selected and click Yes.

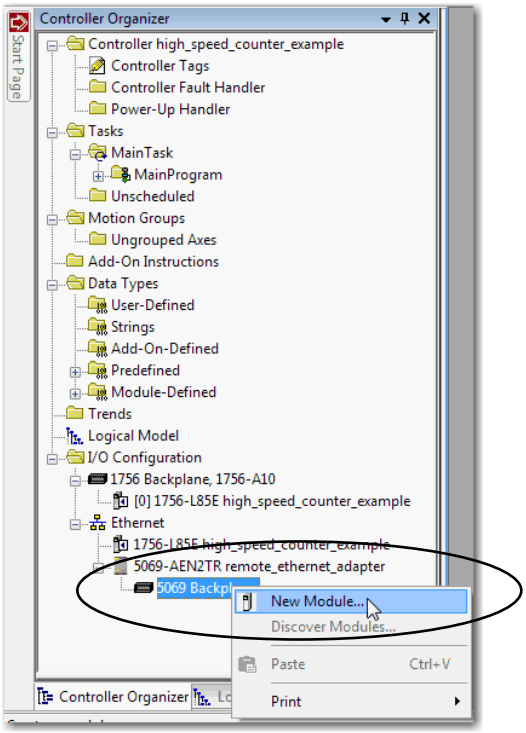


6. Close the Select Module Type dialog box.

New Module

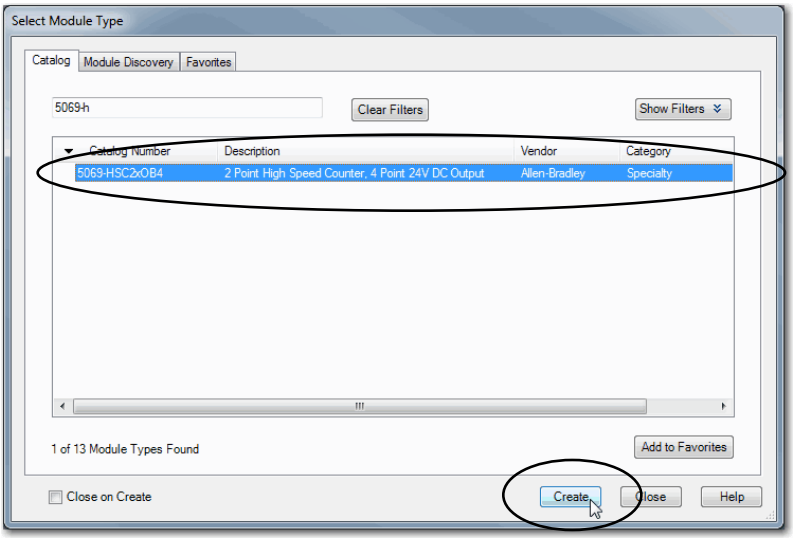
To add a module using New Module, perform the following steps.

1. Right-click I/O Configuration and choose New Module.



2. Select the module and click Create.

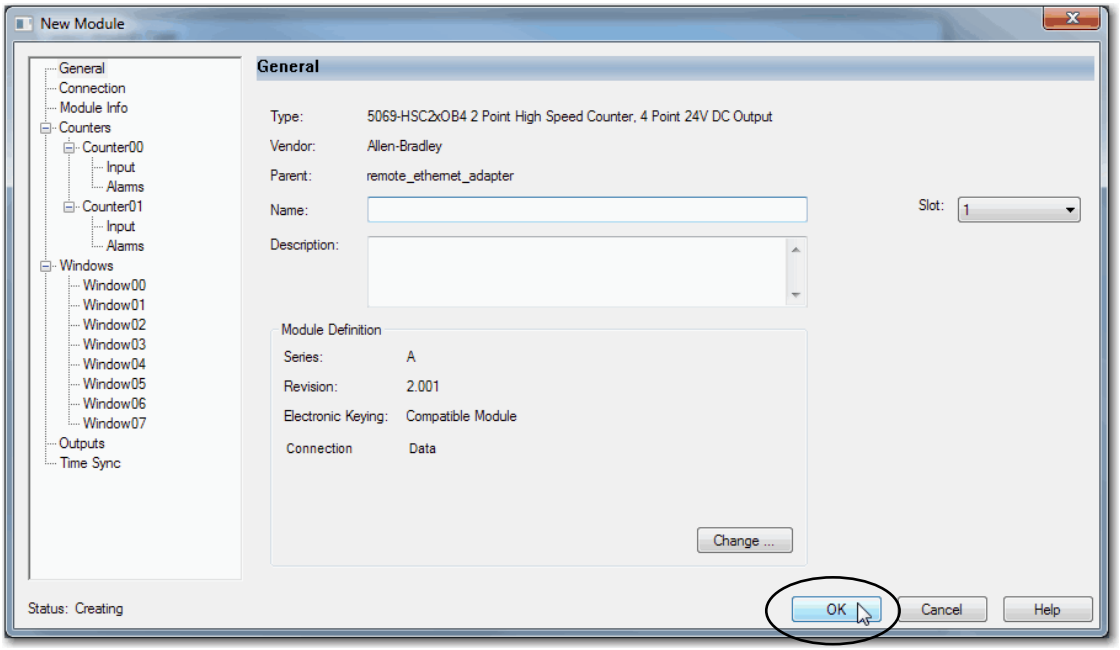
   TIP Type the first few characters of the module catalog number. The list of available modules is reduced.



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

3. Click OK to use the default configuration, if desired.

   Make changes on the categories that are listed on the left side of the dialog box to create a custom module configuration.



Edit the Module Configuration

This section describes how to edit the default module configuration when you add the module to the project. You can also change the module configuration after you add it to the project.

To change module configuration after you add it to the project, double-click the catalog number in the I/O Configuration tree or right-click on the catalog number and choose Properties.

The following categories are available:

- General Category
- Connection Category
- Module Info Category
- Counters Category
- Windows Category
- Outputs Category
- Events Category
- Time Sync Category

IMPORTANT Remember, feature fields that are dimmed in the Logix Designer application dialog boxes are read-only. You must use the module tags to configure the features.

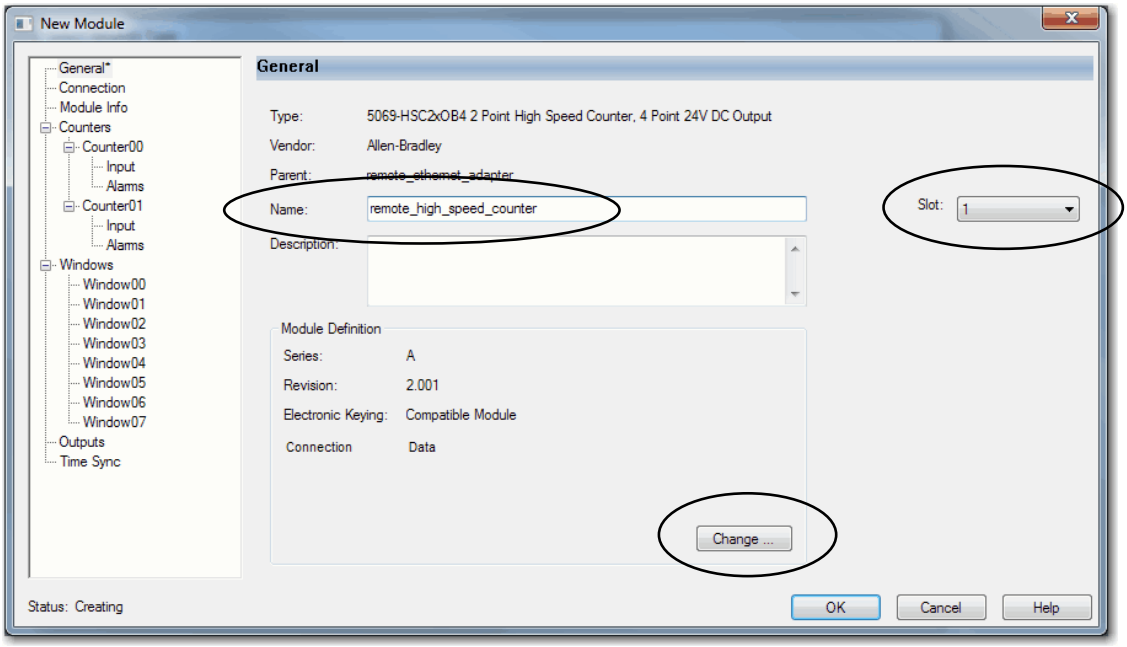
For more information on module tags, see Appendix B, Module Tag Definitions on page 97.

General Category

The General category appears first when you create module.

You use this category to complete the following tasks:

- Name the module.
- Make sure the Slot number that is used in the module configuration matches the slot number in which the module is installed on the system.
- Describe the module.
- Access the Module Definition.



Module Definition

Click Change ... to access the configurable parameters that define the 5069 Compact I/O high-speed counter module. Configuration choices on individual categories can affect the number and type of choices on available on other categories. For example, if you choose Connection: Listen Only, the remaining categories become read-only.

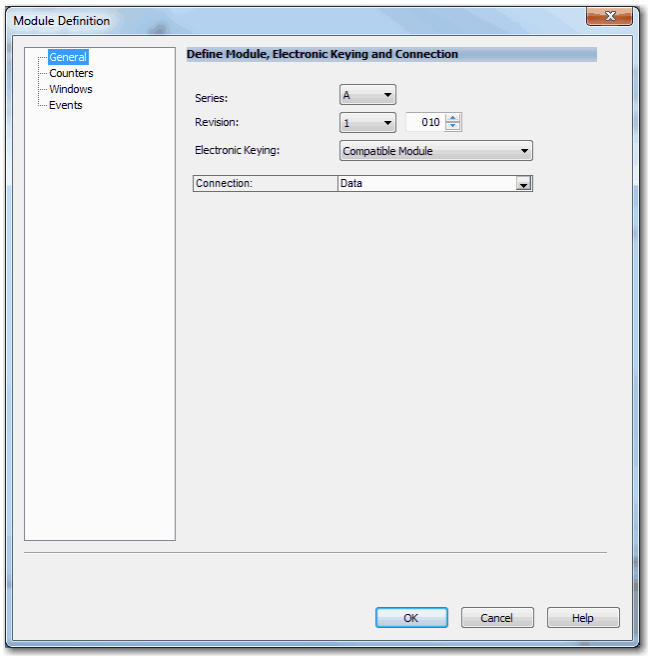
The following Module Definition categories are available:

- General - Table 14 describes the parameters on the General category.

Table 14 - General Category Parameters

Parameter	Definition	Available Choices
Series	Module hardware series	Module-specific
Revision	Module firmware revision, including major and minor revision levels	Module-specific
Electronic Keying	Software method by which you reduce the possibility of that you use the wrong device in a control system. For more information, see the following: • Electronic Keying on page 28 • Electronic Keying in Logix5000 Control Systems Application Technique, publication LOGIX-AT001	Exact Match Compatible Module Disable Keying
Connection(1)	Determines the following for the module type that you configure: • Available configuration categories and related parameters in the remainder of the Module Definition • Available configuration categories and related parameters on the Module Properties dialog box • Data type transferred between the module and the controller • Which tags are generated when configuration is complete	Data Data with Events Listen Only

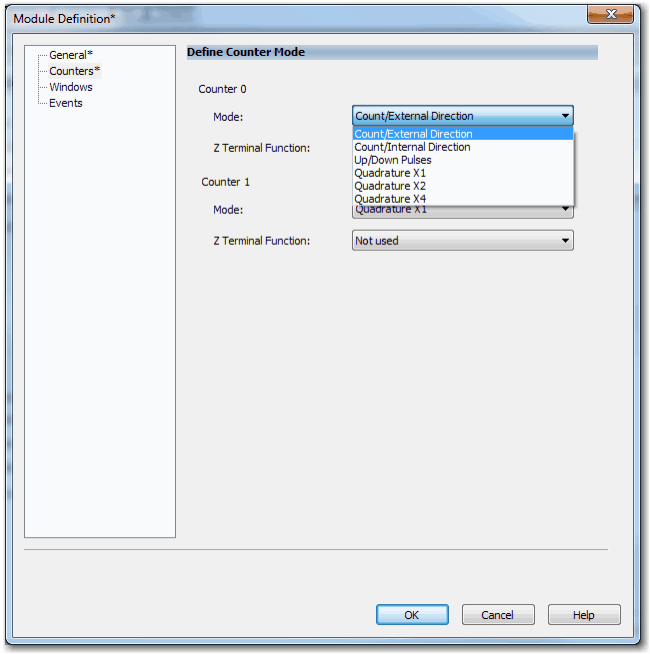
(1) For more information on Connections, see page 13

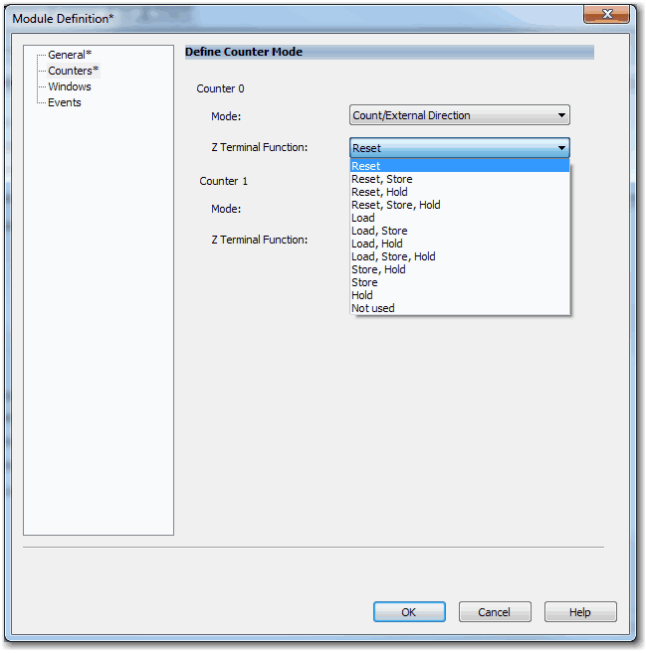


- Counters - Table 15 describes the parameters on the Counters category.

Table 15 - Counters Category Parameters

Parameter	Definition	Available Choices
Counter *x* - Mode	Defines the mode in which the counter operates.	Count/External Direction Count/Internal Direction Up/Down Pulses Quadrature X1 Quadrature X2 Quadrature X4
Counter *x* - Z Terminal Function	Defines the terminal function that is used with the Counter mode chosen in the previous parameter.	Reset Reset, Store Reset, Hold Reset, Store, Hold Load Load, Store Load, Hold Load, Store, Hold Store, Hold Store Hold Not used

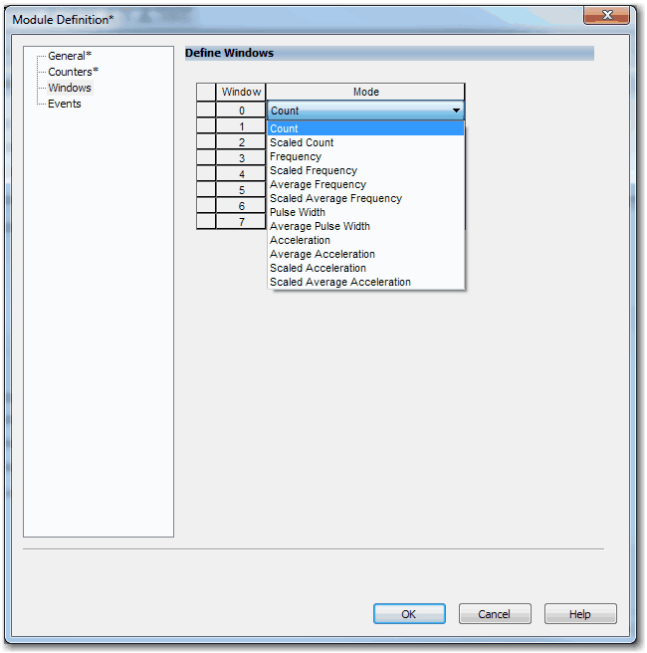




- Windows - Table 16 describes the parameters on the Windows category.

Table 16 - Windows Category Parameters

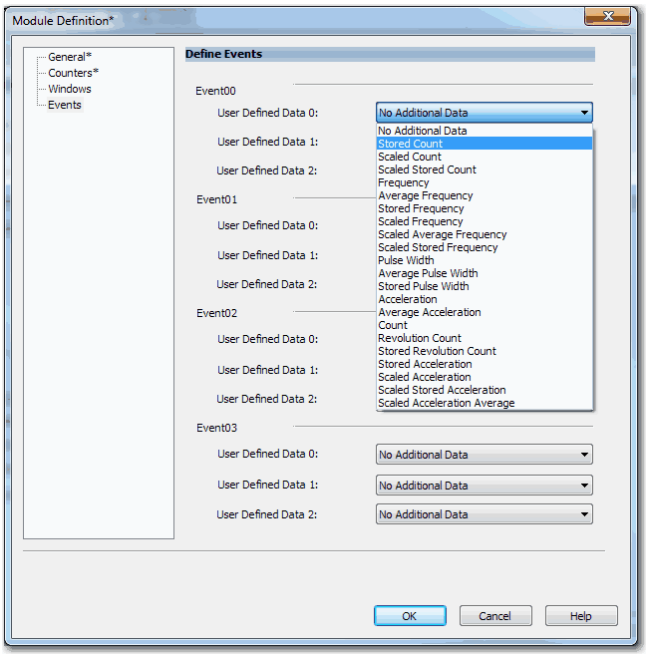
Parameter	Definition	Available Choices
Window *x*/ Mode	Defines the mode in which a Window operates when tied to a module counter.	Count Scaled Count Frequency Scaled Frequency Average Frequency Scaled Average Frequency Pulse Width Average Pulse Width Acceleration Average Acceleration Scaled Acceleration Scaled Average Acceleration



- Events - Table 17 describes the parameters on the Events category.

Table 17 - Events Definition Parameters

Parameter	Definition	Available Choices
Event*xx* - 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.	Stored Count Scaled Count Scaled Stored Count Frequency 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 Acceleration Average



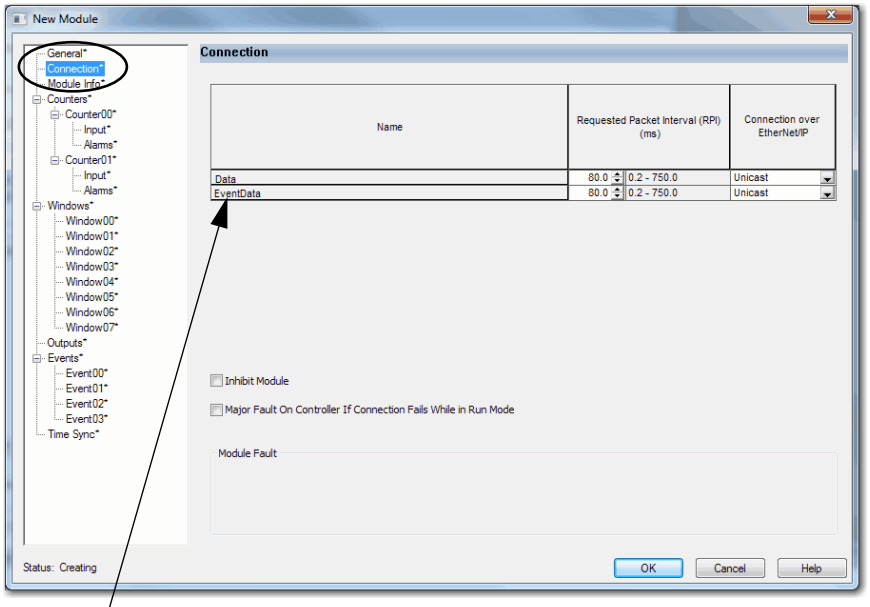
IMPORTANT You must use the Data with Events Connection options to use the Events category.

Connection Category

The Connection tab lets you complete the following tasks:

- Set the RPI rate. For more information the RPI, see page 14.
- Set the connection type to use on the EtherNet/IP network. For more information, see page 15.
- Inhibit the module. For more information on how to inhibit the module, see page 27.
- Configure whether a connection failure while the controller is in Run module causes a major or minor fault.

TIP The Module Fault area of the Connection category is useful during module troubleshooting. For more information on the Module Fault area, see page 94



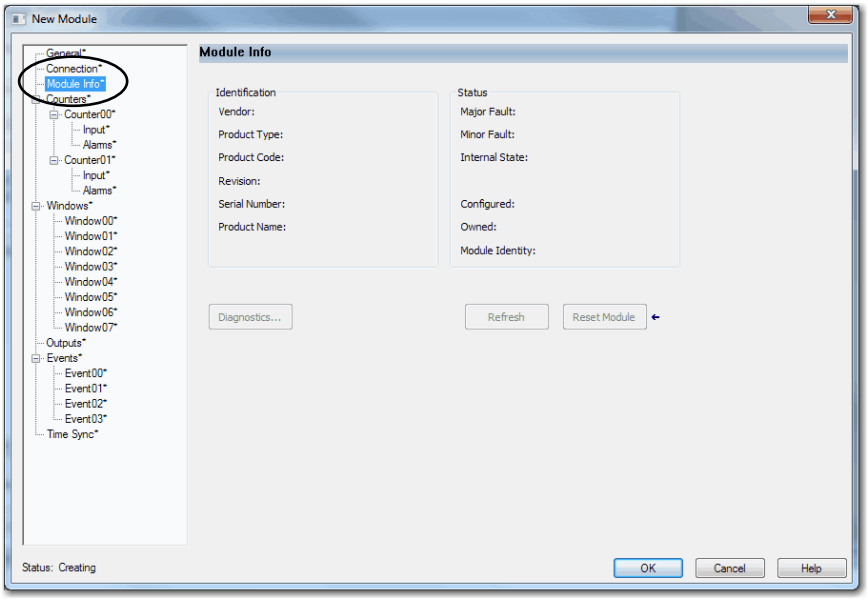
IMPORTANT: The EventData line only appears if you use the Data with Events Connection type in the Module Definition.

Module Info Category

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

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

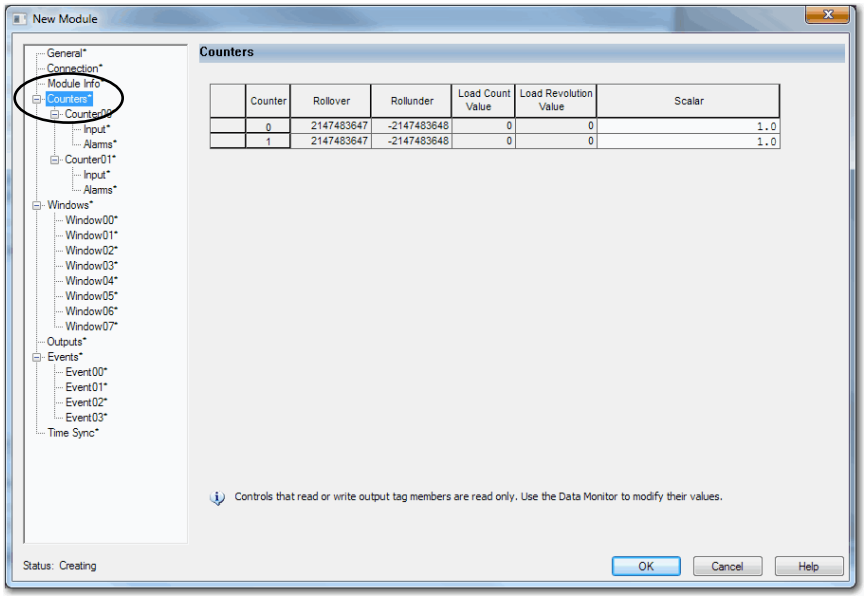
TIP The data on this tab comes directly from the module. If you use a Listen Only connection type in the module definition, this tab is not available in the future.



Counters Category

IMPORTANT This category is not available if you use the Listen Only Connection type.

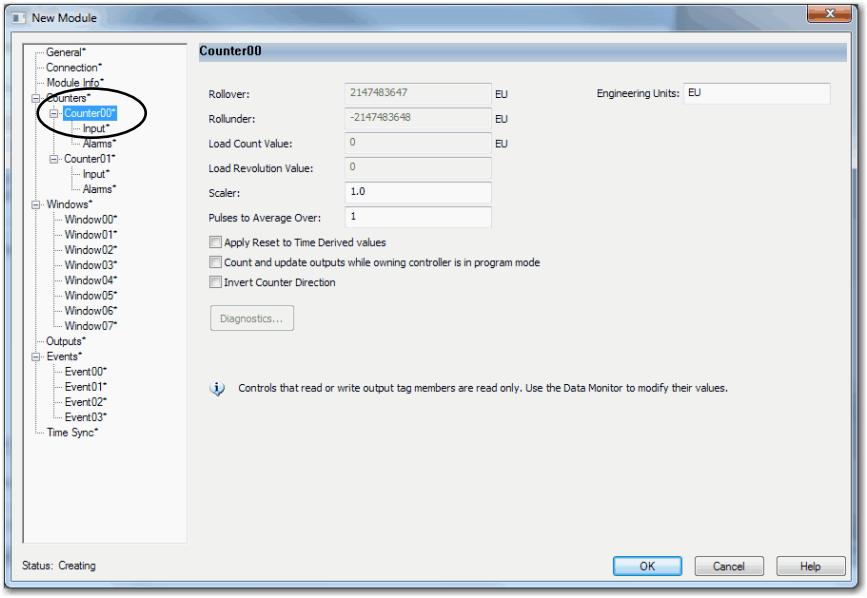
The Counters category shows a summary of the module counters configuration.



Counterxx Category

The Counter*xx* category shows the counter configuration, for example, the count number at which a rollover occurs.

When the project is online, the Diagnostics ... button is enabled so you can retrieve diagnostic information.



For more information on how to use the module counters, see the following:

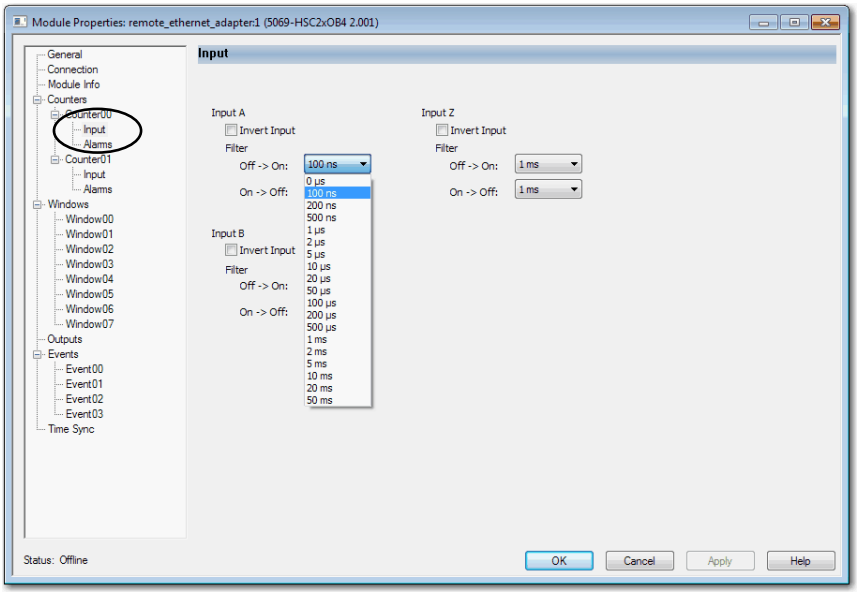
- Counters on page 37
- Module Counter Use on page 51

Input Category

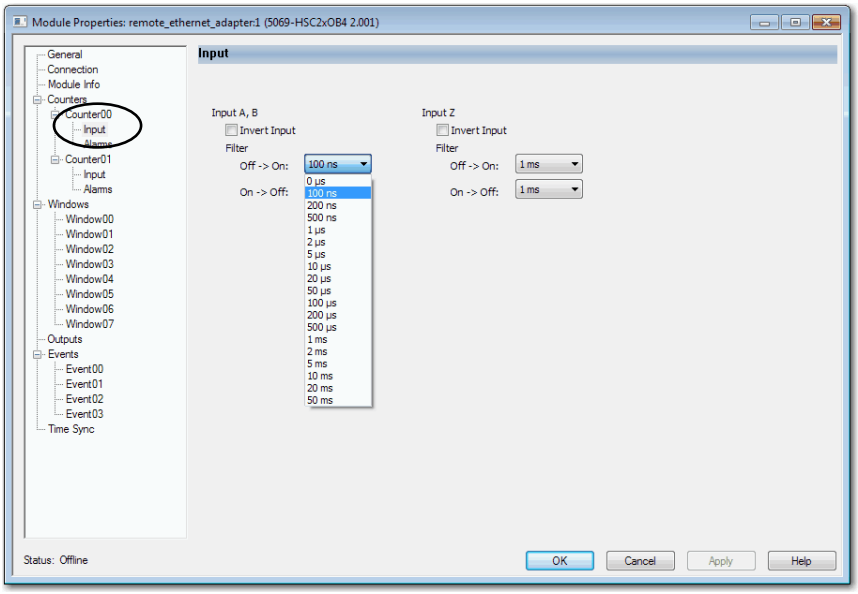
The Input category displays the Invert Input option and available Off to On or On to Off Filter Times for each input channel.

IMPORTANT The Input category dialog box looks different depending on which counter mode you choose in the Module Definition parameters.

Count/Direction and Up/Down modes



Quadrature mode

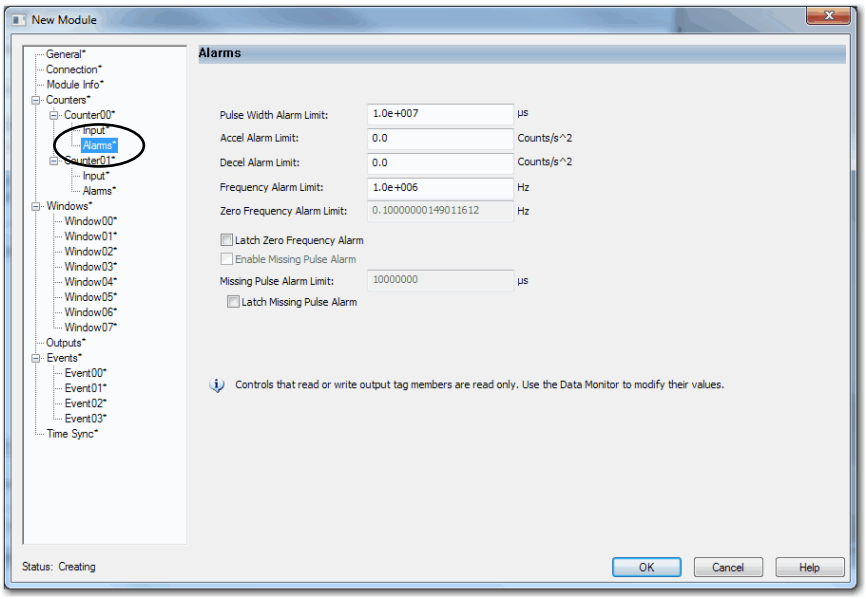


For more information on how to use the module inputs, see the following:

- Inputs on page 38
- Module Input Use on page 60

Alarms Category

The Alarms category displays the counter alarm options.

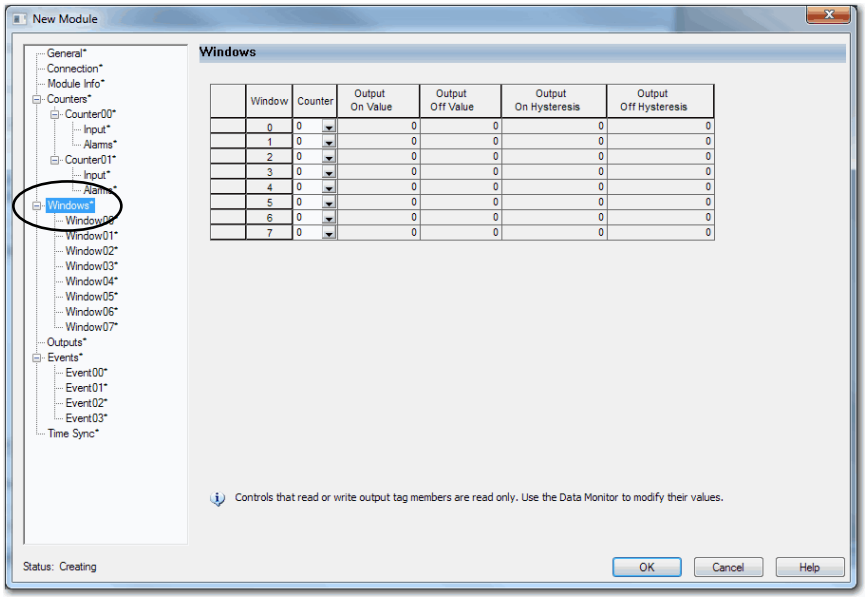


For more information on the module alarms, see Chapter 2, High-speed Counter Module Features on page 23.

Windows Category

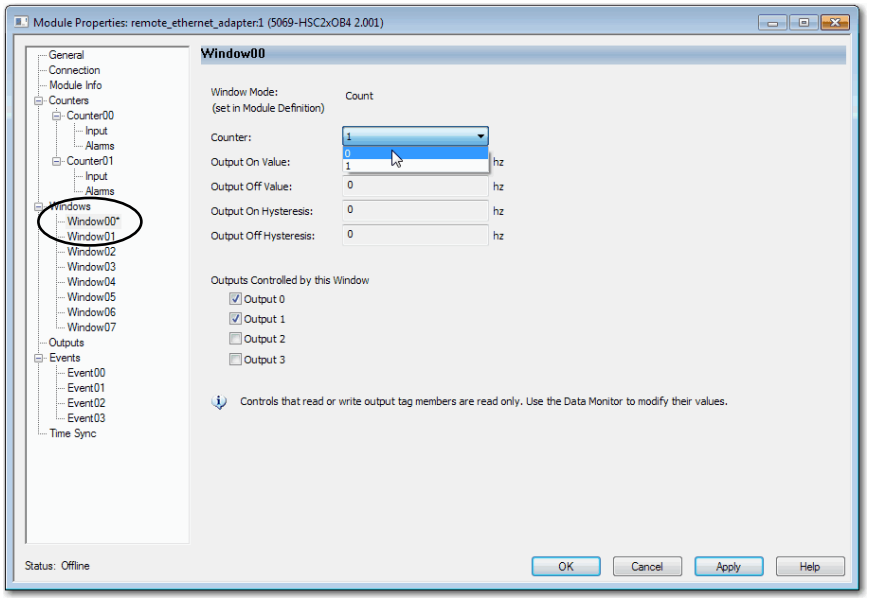
IMPORTANT This category is not available if you use the Listen Only Connection type.

The Windows category shows a summary of the module windows configuration.



Windowxx Category

The Window*xx* category shows the module windows configuration.



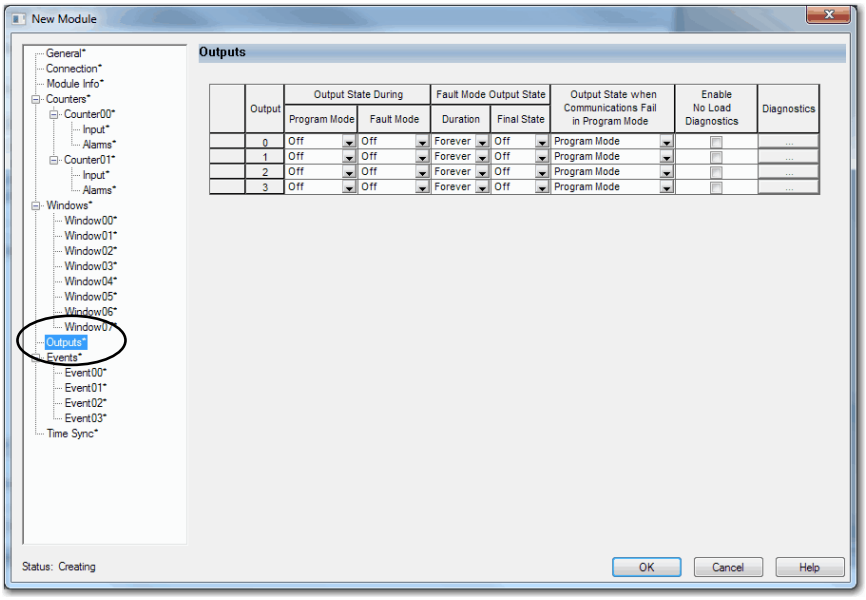
For more information on how to use the module windows, see the following:

- Windows on page 38
- Module Window Use on page 52

Outputs Category

IMPORTANT This category is not available if you use the Listen Only Connection type.

The Outputs category displays a summary of the module output configuration.



For more information on how to use the module outputs, see the following:

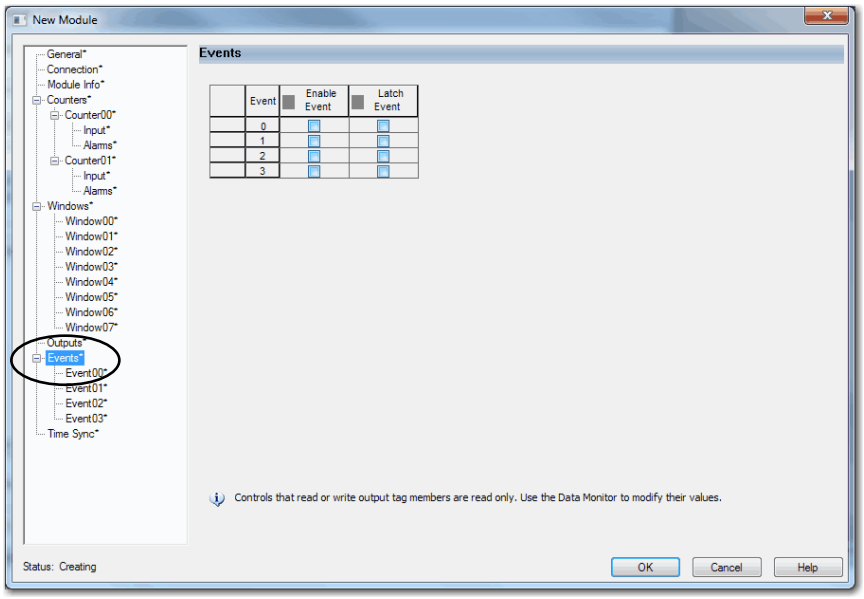
- Outputs on page 38
- Module Output Use on page 62

Events Category

The Events category is available only if you choose Data with Events for Connection in the Module Definition dialog box. Click the + sign next to the Events category to expand it.

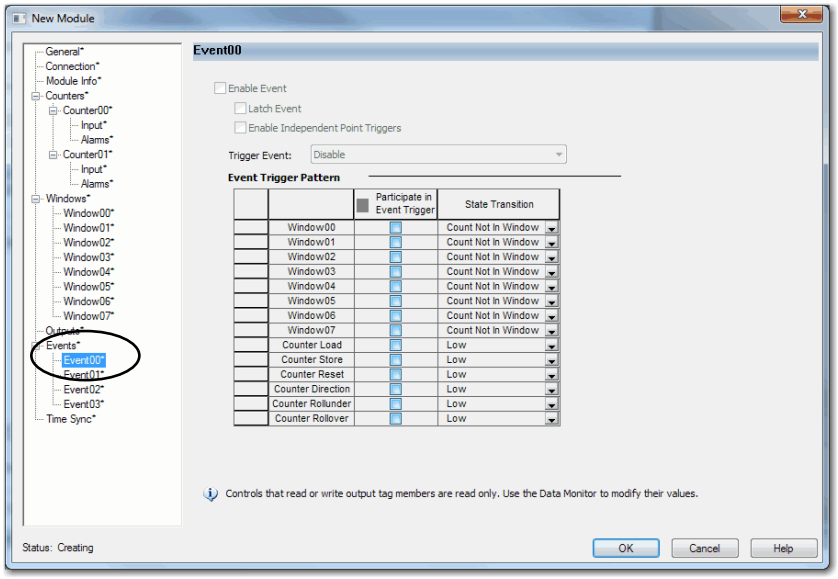
IMPORTANT You cannot configure events on the Module Properties dialog box. The parameters that are displayed are read-only.

You must use the Event Output tags to configure an event. For more information, see Configure an Event in the Event Output Tags on page 70.



Eventxx Category

The *Eventxx* category shows the configuration parameters for events.



Configure an Event in the Event Output Tags

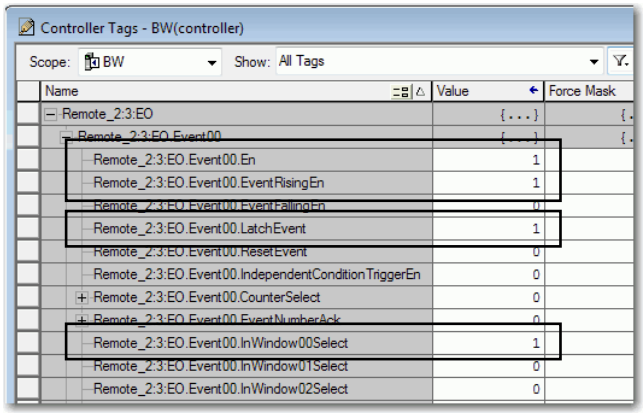
To configure an event, you must change the Event Output tags for the affected module via the Tag Monitor in the Logix Designer application. When you change the tags, the change is reflected on the Module Properties dialog box.

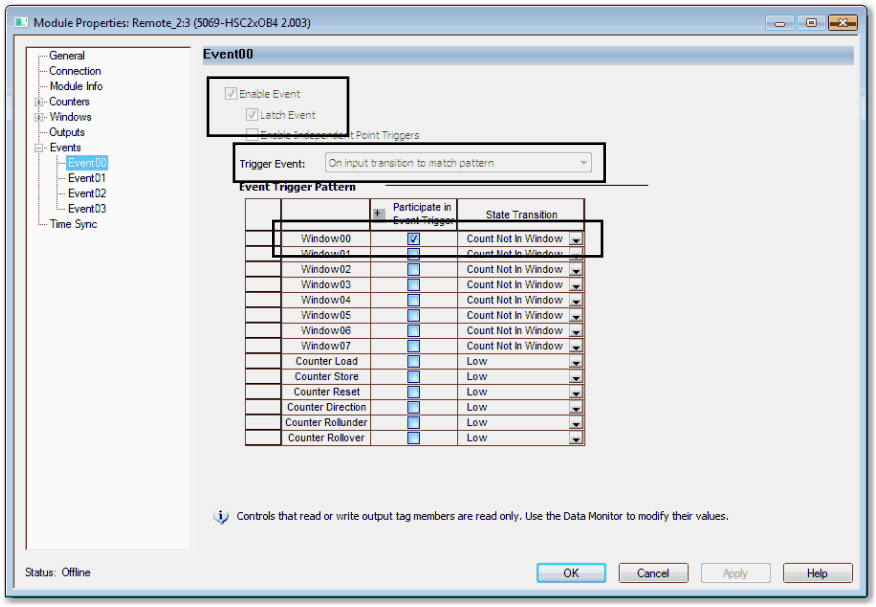
The following graphics show how tag values are reflected on the Module Properties. The following conditions are shown:

- Event is enabled.
- Window00 is configured to trigger the event.
- Event is latched.
- Trigger Event is the state transition of the count changing to a number outside the window that is established on the Window00 category.

These changes in the Event Output tags configure the event.

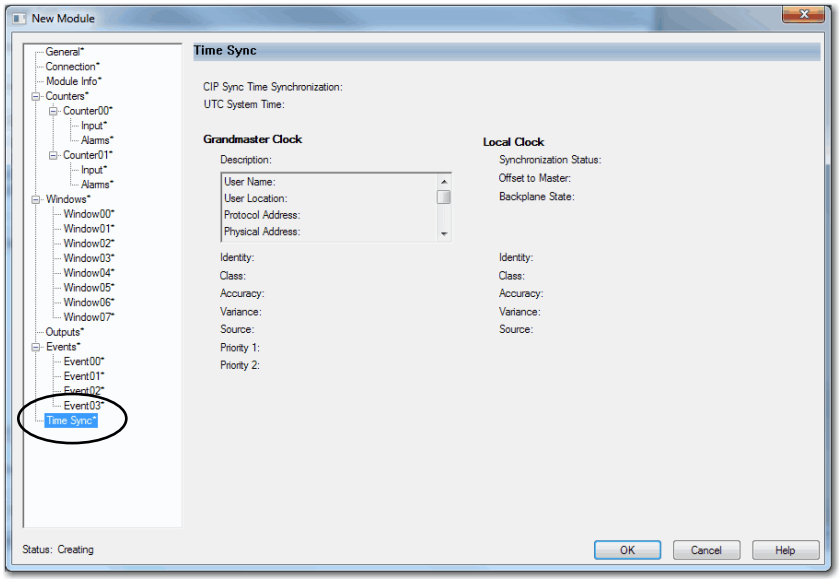
After the tags are changed, the related parameters on the Module Properties are updated automatically.





Time Sync Category

The Time Sync category shows time synchronization information when the project is online.



View the Module Tags

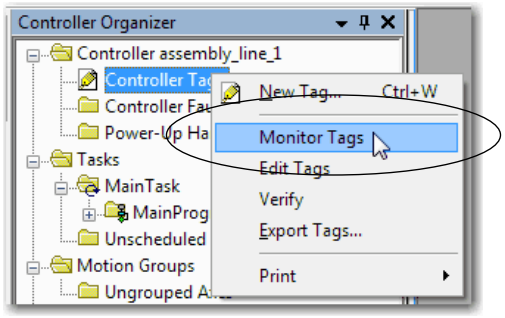
Module tags are created when you add a module to the Logix Designer application project. You can view in the Tag Editor.

Each configured feature on your module has a distinct tag that is available for use in the controller program logic. The Connection choice that you make when you define the module determines what tags are created.

For example, if you use a Listen Only Connection with a module, the Logix Designer application creates only Input tags for that module. For more information on module tags, see Appendix B, Module Tag Definitions on page 97.

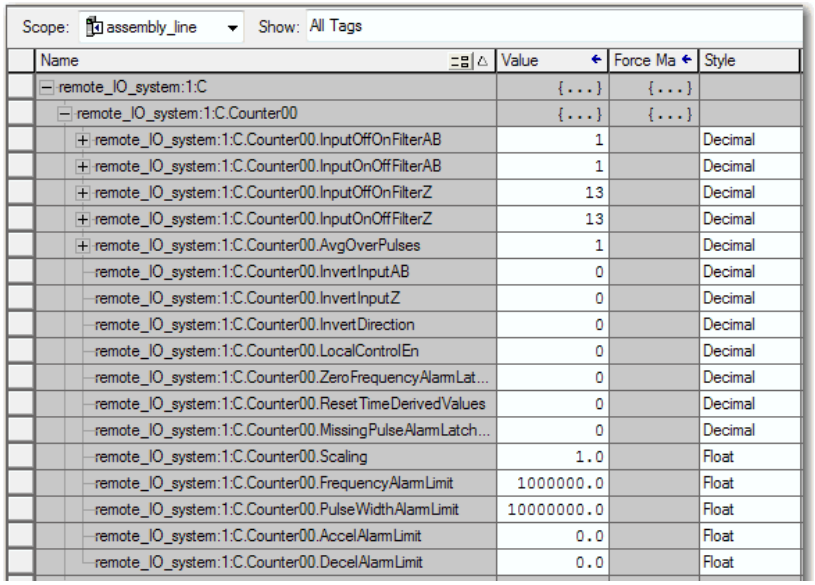
Complete the following steps to access the module tags.

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



   The Controller Tags dialog box appears with data.

2. Click the + symbols to view module tags.

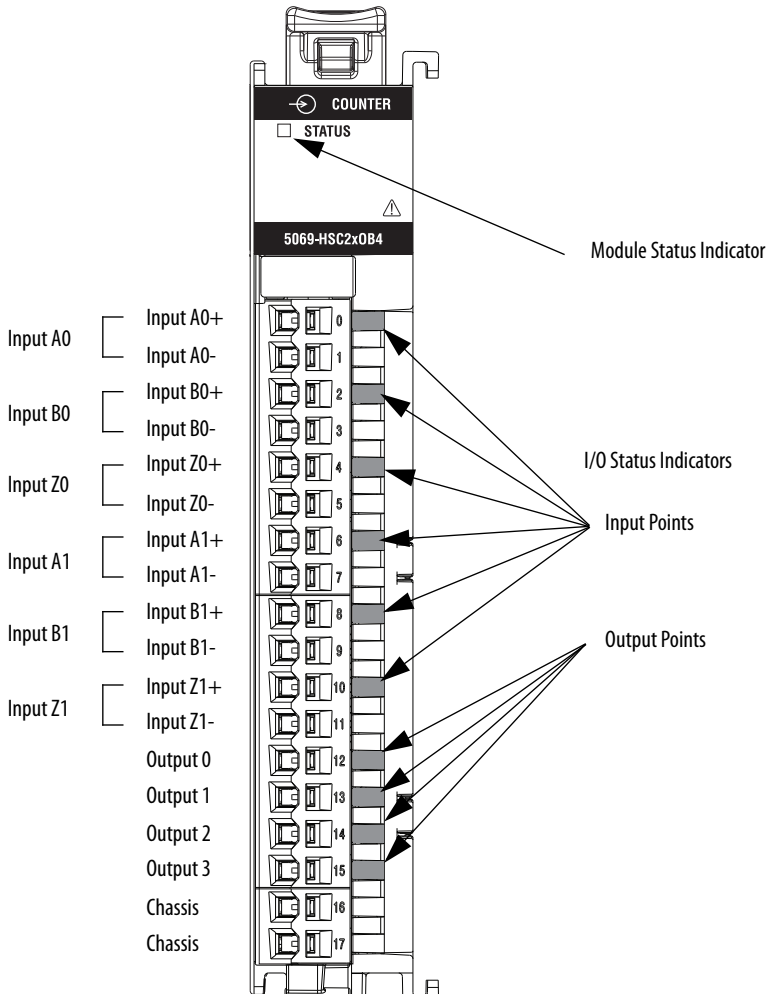


Troubleshoot Your Module

This section describes the status indicators on your 5000 Series high-speed counter module and methods to troubleshoot your application. Your high-speed counter module uses the following status indicators:

- Module (MOD) Status Indicator
- I/O Status Indicator

Figure 17 - 5069 Compact I/O High-speed Counter Module Status Indicators



Module Status Indicator

Table 18 describes the MOD Status indicator on the 5069 Compact I/O™ high-speed counter module.

Table 18 - Module (MOD) Status Indicator - 5069 Compact I/O High-speed Counter Module

Indicator State	Description	Recommended Action
Off	The module is not powered.	None if your application does not use the module If your application uses the module and it is expected to be operating, complete the following: • Confirm that the system is powered. • Confirm that the module is installed properly.
Steady green	The module is operating normally.	No action necessary.
Flashing green	The following conditions exist: • The module is powered. • The module does not have a connection to the controller. The lack of a connection can result from missing, incomplete, or incorrect module configuration.	Use the Logix Designer application to determine what is preventing a connection to the controller. The Connection category in the Module Properties for the module indicates the error information about what is preventing a connection to the controller.
Steady red	The module has experienced a nonrecoverable fault.	Replace the module.
Flashing red	One of the following conditions exist: • 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.	Complete one of the following: • Let the firmware update progress complete. • Reattempt a firmware update after one fails. • Use the Logix Designer application to determine the cause of the module fault. The Connection and Module Info categories of the module configuration indicate the fault type. To clear a recoverable fault, complete one of the following: – Cycle module power. – Click Reset Module in the Module Info category of the Module Properties in the Logix Designer application. If the fault does not clear after you cycle power and click Reset Module, contact Rockwell Automation Technical Support. • Use the Logix Designer application to determine if a connection has timed out. The Connection category in the Module Properties for the module indicates the module state, including if a connection has timed out. If a connection has timed out, determine the cause and correct it. For example, a cable failure can cause a connection timeout.
The following sequence: 1. Steady red for approximately 2 seconds 2. Flashing green	Module is powering up.	Wait for the power-up sequence to finish.

I/O Status Indicators

Table 19 describes the I/O Status indicator on 5069 Compact I/O high-speed counter module.

Table 19 - I/O Status Indicator- 5069 Compact I/O High-speed Counter Modules

Indicator State	Description		Recommended Action
	Input Point	**Output Point**	
Off	One of the following conditions exists: • The module is not powered. In this case, the MOD Status indicator is also Off. • The module is powered but no connection from the controller to module was established. In this case, the MOD Status indicator state is flashing green. Input points only - There is no signal on the input point. Output points only - The output is not powered.		Complete one of the following: • No action if your application does not use the input channel. • If you expect the module to be powered but it is not, complete the following: – Confirm that the system is powered. – Confirm that the module is installed properly. • If the module is powered but the channel is not operating as expected, use the Logix Designer application to confirm the following: – The channel is not disabled. – The channel has a connection to the controller. The Connection category in the Module Properties for the module indicates if the module is running or has experienced a fault. If the module experienced a fault, the Connection category indicates error information about the fault cause.
Steady yellow	The channel is operating normally.		No action necessary.
Steady red	N/A	One of the following conditions exists: • The module has experienced a nonrecoverable fault. In this case, the MOD Status indicator state is also steady red. • Short circuit on the output point.	Complete one of the following: • If a nonrecoverable fault exists, replace the module. • If short circuit condition exists, make necessary change to resolve the issue and resume normal operation.
Flashing red	N/A	Open Wire condition exists and the output point is not powered.	Reconnect wiring and power output point when needed.
Alternating yellow/red	N/A	Open Wire condition exists and the output point is powered.	Complete the following: 1. Remove power from the module. 2. Reconnect wiring. 3. Turn on power for the module.

Use the Logix Designer Application for Troubleshooting

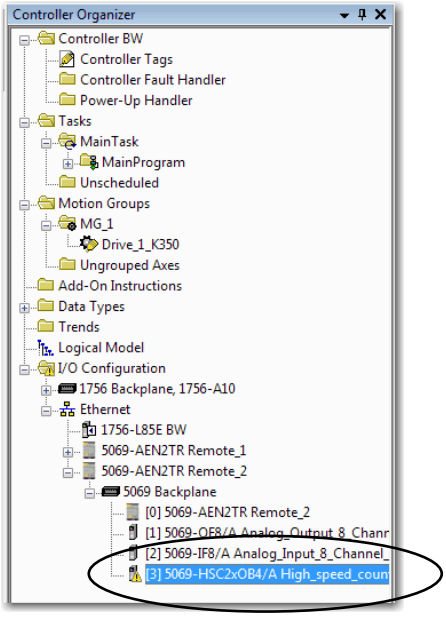
The Logix Designer application indicates the presence of fault conditions in the following ways:

- Warning Signal in the I/O Configuration Tree
- Status and Fault Information in Module Properties Categories
- Logix Designer Application Tag Editor

Warning Signal in the I/O Configuration Tree

As shown in Figure 18, a warning icon appears in the I/O Configuration tree when a fault occurs.

Figure 18 - Warning Signal



Status and Fault Information in Module Properties Categories

The Module Properties section in the Logix Designer applications includes a series of categories. The number of and types of categories varies by module type.

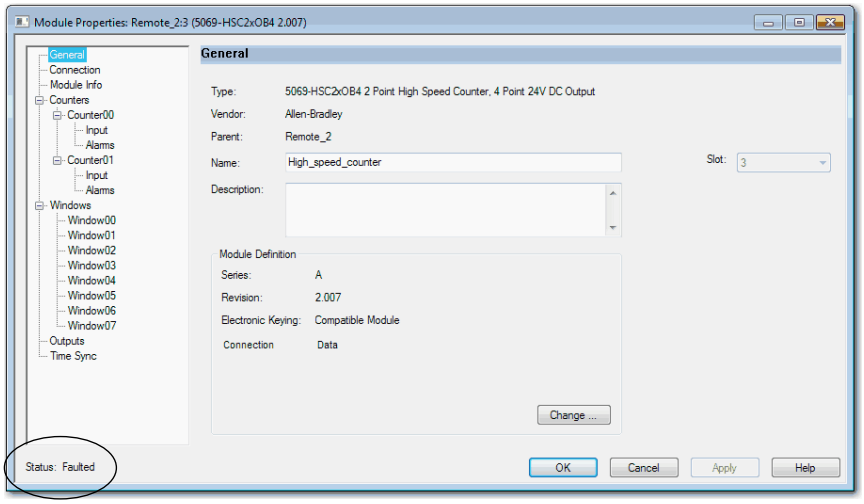
Each category includes options to configure the module or monitor the module status. The following are ways to monitor a module for faults:

- Module Status on General Category
- Module Fault Descriptions on Connection Category

Module Status on General Category

As shown in Figure 19, module status is indicated on the General category of the Modules Properties.

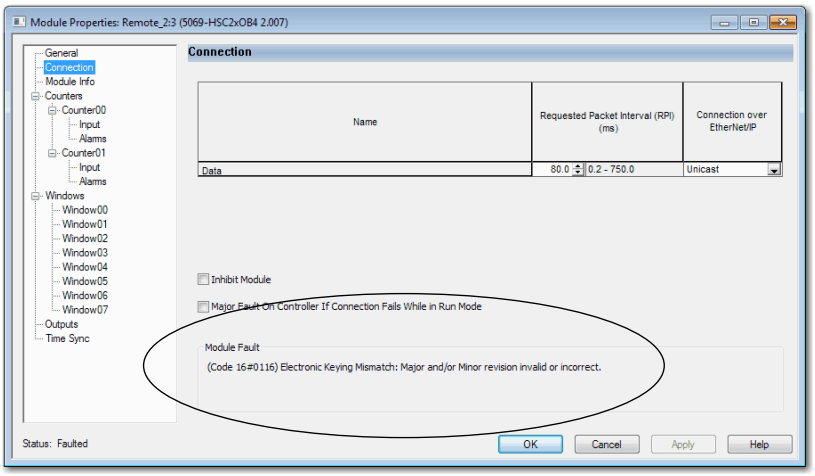
Figure 19 - Fault Message in Status Line



Module Fault Descriptions on Connection Category

A module fault description that includes an error code.

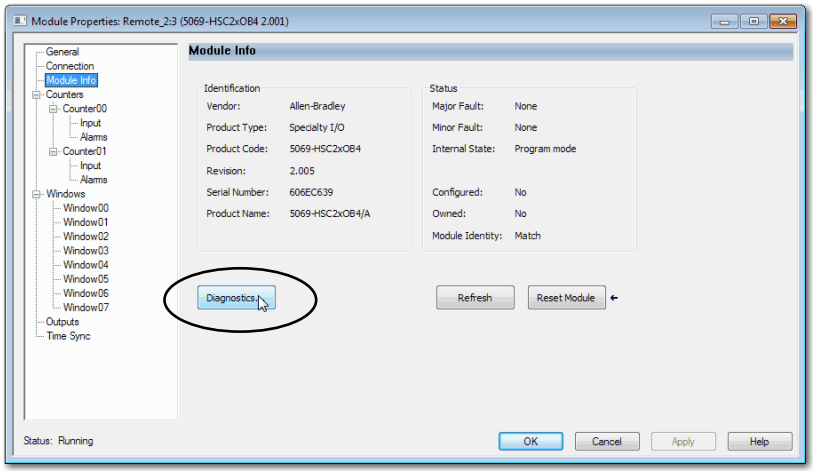
Figure 20 - Fault Description with Error Code

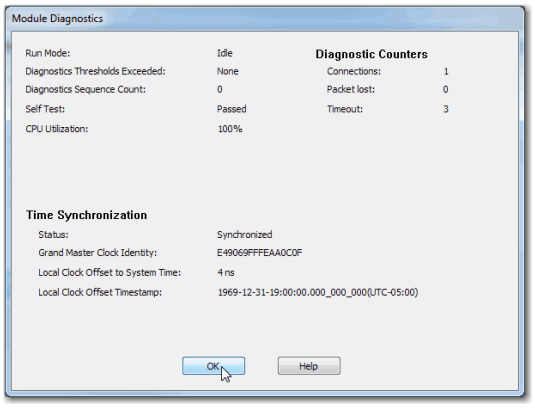


Module Diagnostics Dialog Box

Module Diagnostics are accessible from the Module Properties dialog box.

Figure 21 - Module Diagnostics

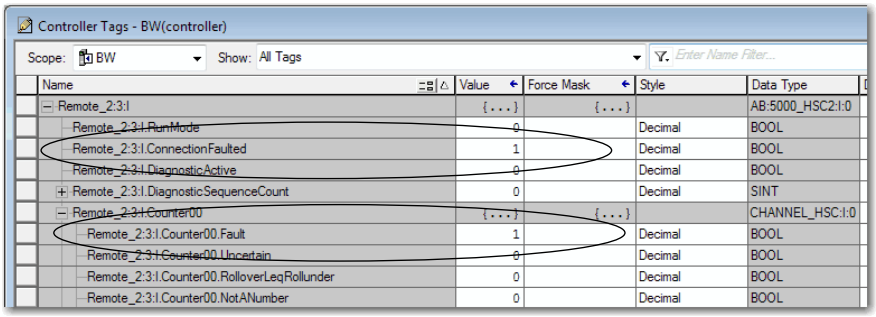




Logix Designer Application Tag Editor

As shown in Figure 22, fault conditions are indicated in the controller tags for the module.

Figure 22 - Fault Indication in Controller Tags



Notes:

Module Tag Definitions

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

The following types of tags are available:

- Configuration
- Input
- Output
- Event Input
- Event Output

The set of module tags created depends on the module type and Module Definition choices made during module configuration. For example, if you use a Listen Only Connection in the Module Definition, the Logix Designer application creates only Input tags for that module.

Tag Name Conventions

The module tag names use defined naming conventions. The conventions are as follows: (example tag name = ***C.Counterxx.InputOffOnFilterA***).

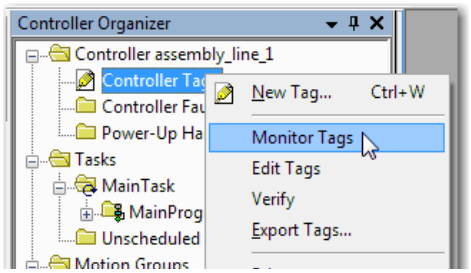
- remote_ethernet_adapter = name of the 5069-AEN2TR EtherNet/IP adapter in the 5069 Compact I/O system
- 1 = slot number
- C = Configuration tag type
- Counter*xx* = channel counter number
- InputOffOnFilterA = tag function

  In this case, InputOffOnFilterA represents a configurable parameter.

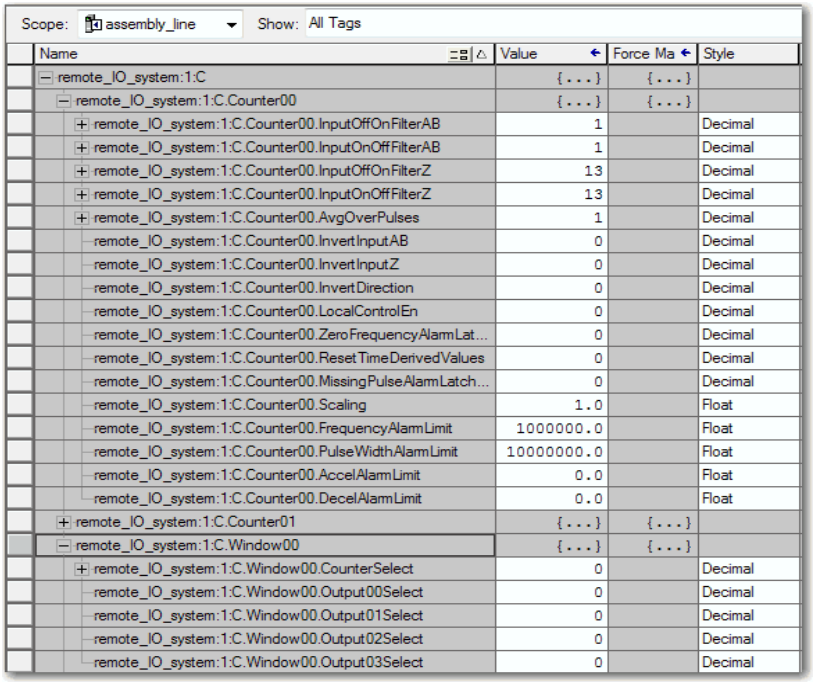
Access the Tags

You view tags from the Tag Editor. Complete the following steps.

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



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



Module Tags

This section describes all of the module tags available with the 5069-HSC2xOB4 high-speed counter module.

Configuration Tags

Table 20 describes the Configuration tags.

Table 20 - 5069-HSC2xOB4 High-speed Counter Module - Configuration Tags

Name	Size	Description	Valid Values
Counterxx.InputOffOnFilterA	SINT	Time that the signal must be in the On state before the A input data indicates the On state.	• 0 = 0 (20 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 = 50ms
Counterxx.InputOnOffFilterA	SINT	Time that the signal must be in the Off state before the A input data indicates the Off state.	• 0 = 0 (20 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 = 50ms

Table 20 - 5069-HSC2x0B4 High-speed Counter Module - Configuration Tags

Name	Size	Description	Valid Values
Counterxx.InputOffOnFilterB	SINT	Time that the signal must be in the On state before the B input data indicates the On state.	• 0 = 0 (20 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 = 50ms
Counterxx.InputOnOffFilterB	SINT	Time that the signal must be in the Off state before the B input data indicates the Off state.	• 0 = 0 (20 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 = 50ms
Counterxx.InputOffOnFilterZ	SINT	Time that the signal must be in the On state before the Z input data indicates the On state.	• 0 = 0 (20 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 = 50ms

Table 20 - 5069-HSC2x0B4 High-speed Counter Module - Configuration Tags

Name	Size	Description	Valid Values
Counterxx.InputOnOffFilterZ	SINT	Time that the signal must be in the Off state before the Z input data indicates the Off state.	• 0 = 0 (20 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 = 50ms
Counterxx.AvgOverPulses	INT	Number of pulses to average when calculating average frequency, pulse width, and rate.	1…32767
Counterxx.InvertInputA	BOOL	Invert the A input.	• 0 = Do not invert A input. • 1 = Invert A input.
Counterxx.InvertInputB	BOOL	Invert the B input.	• 0 = Do not invert B input. • 1 = Invert B input.
Counterxx.InvertInputZ	BOOL	Invert the Z input.	• 0 = Do not invert Z input. • 1 = Invert Z input.
Counterxx.InvertDirection	BOOL	Invert the counter direction.	• 0 = Do not invert the counter direction. • 1 = Invert the counter direction.
Counterxx.LocalControlEn	BOOL	Enables counter to continue counting when the owner-controller is in Program mode.	• 0 = Counter does not continue counting • 1 = Counter continues counting
Counterxx.ZeroFrequencyAlarmLatchEn	BOOL	Latches the Zero Frequency Alarm when set so that it does not clear until explicitly unlatched.	• 0 = Latching disabled (default) • 1 = Latching enabled
Counterxx.ResetTimeDerivedValues	BOOL	Indicates that a Reset sets frequency and average frequency, pulse width, and acceleration to zero.	• 0 = Reset does not reset Time-derived values. • 1 = Reset resets Time-derived values.
Counterxx.MissingPulseAlarmLatchEn	BOOL	Latches the Missing Pulse Alarm when set so that it does not clear until explicitly unlatched.	• 0 = Latching disabled (default) • 1 = Latching enabled
Counterxx.Scaling	REAL	One of the following based on how any Windows tied to the counter are configured in the module properties. • Count, frequency, and acceleration values will be scaled to user units by multiplying their values in pulses by Scaling. • Scales position, velocity, and acceleration/deceleration tag members to user units. Units are pulses per user unit.	Any value
Counterxx.FrequencyAlarmLimit	REAL	If the instantaneous or average frequency is greater than this value the Frequency Alarm is triggered.	Any positive value
Counterxx.PulseWidthAlarmLimit	REAL	Maximum pulse width 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.	Any positive value
Counterxx.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

Table 20 - 5069-HSC2x0B4 High-speed Counter Module - Configuration Tags

Name	Size	Description	Valid Values
Counterxx.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 negative value
Windowxx.CounterSelect	SINT	Counter with which the window is associated.	• 0 = Counter 0 • 1 = Counter 1
Windowxx.OutputxxSelect	BOOL	Window controls the indicated output, that is, any of Output00…Output03. If a Window controls the output, when the target value is in the window, the output is on.	• 0 = Window does not control output • 1 = Window does control output
Outputxx.FaultMode	BOOL	Determines output action when a connection fault occurs. At the fault occurrence, the output holds its last state or transitions to the value set in the Fault Value parameter. The channel continues the Fault Mode for the length of time set in the Fault Value State Duration parameter.	• 0 = Transition to user-defined value • 1 = Hold Last State (default)
Outputxx.FaultValue	BOOL	Value to which the output changes if the following events exist: • Fault Mode = 0 • Either of the following: – 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	• 0 = On • 1 = Off
Outputxx.ProgMode	BOOL	Determines output action when the controller transitions to Program mode or the connection to the module is inhibited. At the transition to Program mode, the output holds its last state or transitions to the value set in the Program Value parameter.	• 0 = Transition to user-defined value (default) • 1 = Hold Last State
Outputxx.ProgValue	BOOL	Value to which the output changes if the following events exist: • Program Mode = 0 • Module transitions to Program mode	• 0 = Off • 1 = On
Outputxx.FaultFinalState	BOOL	Value to which the output changes if the following events exist: • Connection is lost • Time defined by the Fault State Duration parameter has been exceeded	• 0 = Off • 1 = On
Outputxx.ProgramToFaultEn	BOOL	Determines output action if a connection faults while the module is in a Program state. The output can remain in a Program mode or transition to a safe state for Fault mode. If the output remains in a Program mode, the Final Fault State parameter is ignored.	• 0 = Remains in the Program state • 1 = Transitions to the safe state for the Fault mode
Outputxx.NoLoadEn	BOOL	Enable the output No Load diagnostic	• 0 = Disabled (default) • 1 = Enabled
Outputxx.FaultValueStateDuration	SINT	Determines the length of time that the channel stays in a Fault mode or that the Fault Value is held prior to the Final Fault State.	• 0 = Hold forever (default) • Any of the following: – 1, 2, 5, or 10 seconds

Input Tags

Table 21 describes the Input tags.

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
RunMode	BOOL	Input's operating state	• 0 = Idle • 1 = Run
ConnectionFaulted	BOOL	Indicates if a connection is working The module sets the value to 0 when connected. If the module is not connected, the controller overwrites the value with a 1.	• 0 = Connection running • 1 = Connection not running
DiagnosticsActive	BOOL	Indicates if any diagnostics are active	• 0 = No diagnostics active • 1 = One or more diagnostics are active
DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	-128...127 The value of 0 is skipped except during module power-up.
Counterxx.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. **IMPORTANT**: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.	• 0 = Good data • 1 = Bad data, causing fault Typically, this tag is set when the Counter*xx*.QuadratureErrorCount tag is any number other than 0. The Counter*xx*.QuadratureErrorCount tag only applies when the high-speed counter module is operating in the X1, X2, or X4 Quadrature mode.
Counterxx.Uncertain	BOOL	Indicates that the counter 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.	• 0 = Good data • 1 = Uncertain data The following are causes of uncertain data: – 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}. If the pulses are longer than TP-MAX, the module assumes the input frequency is 0. However, the module cannot determine if there are no pulses being measured because no pulses exist or because the module is filtering a high frequency signal. (T_{P-MAX} = 10 s) – 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 the period defined in the Counterxx:AverageOverPulses configuration tag is greater than TP-MAX. In this case, the average frequency and average pulse width are reported based on the length of the complete pulses measured.

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
Counterxx.RolloverLeqRollunder	BOOL	Indicates the Rollover value is less than the Rollunder value and the module is using either the last valid or default Rollover and Rollunder values.	• 0 = Rollover value is not less than the Rollunder value • 1 = Rollover value is less than the Rollunder value
Counterxx.MissingPulseAlarm	BOOL	Indicates the period of a pulse was greater than the missing pulse timeout or was never completed.	• 0 = Period of the pulse was normal • 1 = Period of the pulse was greater than the missing pulse timeout or was never completed.
Counterxx.ZeroFrequencyAlarm	BOOL	Counter frequency is less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	• 0 = Counter frequency is not less than the Zero Frequency Alarm limit. • 1 = Counter frequency is less than the Zero Frequency Alarm limit.
Counterxx.ZeroFrequencyAvgAlarm	BOOL	Counter average frequency is less than the Zero Frequency Alarm limit. Therefore, the frequency and rate are being reported as zero.	• 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.
Counterxx.FrequencyAlarm	BOOL	Counter frequency has exceeded the Frequency Alarm limit.	• 0 = Counter frequency has not exceeded the Frequency Alarm limit. • 1 = Counter frequency has exceeded the Frequency Alarm limit.
Counterxx.FrequencyAvgAlarm	BOOL	Counter's average frequency has exceeded the Frequency Alarm limit.	• 0 = Counter's average frequency has not exceeded the Frequency Alarm limit. • 1 = Counter's average frequency has exceeded the Frequency Alarm limit.
Counterxx.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.	• 0 = Alarm 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.
Counterxx.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.	• 0 = Alarm 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.
Counterxx.AccelAlarm	BOOL	Indicates the counter acceleration rate exceeds the Acceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter acceleration rate does not exceed the Acceleration Alarm limit. • 1 = Alarm is triggered. That is, the counter acceleration rate exceeds the Acceleration Alarm limit.

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
Counterxx.AccelAvgAlarm	BOOL	Indicates the counter average acceleration rate exceeds the Acceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter average acceleration rate does not exceed the Acceleration Alarm limit. • 1 = Alarm is triggered. That is, the counter average acceleration rate exceeds the Acceleration Alarm limit.
Counterxx.DecelAlarm	BOOL	Indicates the counter deceleration rate exceeds the Deceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter deceleration rate does not exceed the Deceleration Alarm limit. • 1 = Alarm is triggered. That is, the counter deceleration rate exceeds the Deceleration Alarm limit.
Counterxx.DecelAvgAlarm	BOOL	Indicates the counter average deceleration rate exceeds the Deceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter average deceleration rate does not exceed the Deceleration Alarm limit. • 1 = Alarm is triggered. That is, the counter average deceleration rate exceeds the Deceleration Alarm limit.
Counterxx.FrequencyOverrange	BOOL	Indicates the counter frequency exceeds the module's maximum rated frequency, and therefore can not be tracking the signal properly.	• 0 = Counter frequency does not exceed the module's maximum rated frequency. • 1 = Counter frequency exceeds the module's maximum rated frequency.
Counterxx.PartialAvgFrequency	BOOL	Indicates the counter's average frequency is complete or partial with respect to Counterxx.AvgOverPulses. Average frequency can be partial when the length of Counterxx.AvgOverPulses is greater than the T_{P-MAX} time window.	• 0 = Counter's average frequency is complete and is based on the Counterxx.AvgOverPulses tag complete pulses. • 1 = Counter's average frequency is partial and is based on the length of the complete and incomplete pulses measured within the T_{P-MAX} time window.
Counterxx.PartialAvgPulseWidth	BOOL	Indicates the counter's average pulse width is complete or partial with respect to the Counterxx.AvgOverPulses tag. Average pulse width can be partial when the length of the Counterxx.AvgOverPulses pulse widths is greater than T_{P-MAX} time window.	• 0 = Counter's average pulse width is complete and is based on the Counterxx.AvgOverPulses tag complete pulses widths. • 1 = Counter's average pulse width is partial and is based on the length of the complete and incomplete pulses widths measured within the T_{P-MAX} time window.
Counterxx.Direction	BOOL	Indicates the count direction.	• 0 = Counting up • 1 = Counting down
Counterxx.StoredDirection	BOOL	Indicates the count direction when the Store input transitioned from Off to On.	• 0 = Count up • 1 = Count down
Counterxx.Rollover	BOOL	Indicates the counter counted up to Rollover value and then has continued counting from the Rollunder.	• 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.

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
Counterxx.Rollunder	BOOL	Indicates the counter counted down to Rollunder value and then has continued counting from the Rollover.	• 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.
Counterxx.DataA	BOOL	Current input A value.	0 = Input A is off. 1 = Input A is on.
Counterxx.DataB	BOOL	Current input B value.	0 = Input B is off. 1 = Input B is on.
Counterxx.DataZ	BOOL	Current input Z value.	0 = Input Z is off. 1 = Input Z is on.
Counterxx.DataAOverridden	BOOL	Input A data was overridden with the OverrideDataA value.	• 0 = Input A data was not overridden. • 1 = Input A data was overridden.
Counterxx.DataBOverridden	BOOL	Input B data was overridden with the OverrideDataB value.	• 0 = Input B data was not overridden. • 1 = Input B data was overridden.
Counterxx.DataZOverridden	BOOL	Input Z data was overridden with the OverrideDataZ value.	• 0 = Input Z data was not overridden. • 1 = Input Z data was overridden.
Counterxx.Count	DINT	Number of input transitions by the counter.	Any
Counterxx.StoredCount	DINT	Count value when the Store input transitioned from Off to On.	Any
Counterxx.ScaledCount	REAL	Counter count value scaled to user units.	Any
Counterxx.ScaledStoredCount	REAL	Stored count value scaled to user units.	Any
Counterxx.RevolutionCount	INT	Number of Rollover and Rollunder transitions. This value increments when a Rollover occurs and decrements when a Rollunder occurs.	Any
Counterxx.StoredRevolutionCount	INT	Revolution Count value when the Store input transitioned from Off to On.	Any
Counterxx.Frequency	REAL	Frequency of the input transitions counted by the counter.	Any
Counterxx.FrequencyAvg	REAL	Average frequency of the input transitions counted by the counter.	Any
Counterxx.StoredFrequency	REAL	Frequency value when the Store input transitioned from Off to On.	Any
Counterxx.ScaledFrequency	REAL	Frequency of the input transitions counted by counter scaled to user units.	Any
Counterxx.ScaledFrequencyAvg	REAL	Average frequency of the input transitions counted by counter scaled to user units.	Any
Counterxx.ScaledStoredFrequency	REAL	Stored Frequency scaled to user units.	Any
Counterxx.PulseWidth	REAL	On-state pulse width of the last pulse received measured in microseconds.	Any
Counterxx.PulseWidthAvg	REAL	Average on-state pulse width of the last pulse received measured in microseconds.	Any
Counterxx.StoredPulseWidth	REAL	Pulse width value when the Store input transitioned from Off to On.	Any
Counterxx.QuadratureErrorCount	SINT	Indicates the number of times a quadrature counter entered the fault state.	Any
Counterxx.CountChangeIndicator	SINT	Indicates the number of count changes.	Any
Counterxx.Accel	REAL	Frequency change of the last two pulses captured divided by the period of the last pulse.	Any
Counterxx.AccelAvg	REAL	Average acceleration of the pulses counted by the counter.	Any
Counterxx.StoredAccel	REAL	Acceleration when the Store input transitioned from Off to On.	Any
Counterxx.ScaledAccel	REAL	Acceleration scaled to user units.	Any

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
Counterxx.ScaledAccelAvg	REAL	Average acceleration scaled to user units.	Any
Counterxx.ScaledStoredAccel	REAL	Acceleration when the Store input transitioned from Off to On scaled to user units.	Any
Windowxx.InWindow	BOOL	Indicates if the value is in the specified window.	• 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.	• 0 = Last window data received was a number • 1 = Last window data received was not a number
Outputxx.Data	BOOL	Output data.	• 0 = On • 1 = Off
Outputxx.Fault	BOOL	Indicates that output 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.	• 0 = Good data • 1 = Bad data, causing fault Typically, this tag is set when the Counter*xx*.QuadratureErrorCount tag is any number other than 0. The Counter*xx*.QuadratureErrorCount tag only applies when the high-speed counter module is operating in the X1, X2, or X4 Quadrature mode.
Outputxx.Uncertain	BOOL	Indicates that the output 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.	• 0 = Good data • 1 = Uncertain data The following are causes of uncertain data: – 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 $> T_{P\text{-}MAX}$. If the pulses are longer than TP-MAX, the module assumes the input frequency is 0. However, the module cannot determine if there are no pulses being measured because no pulses exist or because the module is filtering a high frequency signal. ($T_{P\text{-}MAX} = 10$ s) – Measured pulse width is too short. That is, the pulse width is less than the module's rated minimum pulse width, $T_{PW\text{-}MIN}$. ($T_{PW\text{-}MIN} = 125$ ns) – If the period defined in the Counterxx:AverageOverPulses configuration tag is greater than TP-MAX. In this case, the average frequency and average pulse width are reported based on the length of the complete pulses measured.
Outputxx.NoLoad	BOOL	The signal wire is disconnected from the output or the RTB is removed from the module.	• 0 = No Load condition does not exist • 1 = No Load condition exists. That is, a signal wire is disconnected from the output or the RTB is removed from the module.

Table 21 - 5069-HSC2x0B4 High-speed Counter Module - Input Tags

Name	Size	Description	Valid Values
Outputxx.ShortCircuit	BOOL	A Short Circuit or Overcurrent condition exists.	• 0 = No Short Circuit or Overcurrent condition exists • 1 = Short Circuit or Overcurrent condition exists
EventStatus[x].EventDropped	BOOL	Indicates if an event has been discarded because events are occurring faster than they are being acknowledged.	• 0 = Event was not discarded • 1 = Event was discarded
EventStatus[x].CIPSyncValid	BOOL	Indicates that the module was synchronized with a 1588 master when the event was triggered.	• 0 = Module was not synchronized with a 1588 master when the event was triggered. • 1 = Module was synchronized with a 1588 master when the event was triggered.
EventStatus[x].CIPSyncTimeout	BOOL	Module was previously synchronized with a 1588 master but due to a timeout was no longer synchronized when the event was triggered.	• 0 = Module has never been synchronized with a 1588 master when the event was triggered. • 1 = Module was synchronized with a 1588 master but due to a timeout was not synchronized when the event was triggered.
EventStatus[x].EventReset	BOOL	Indicates that the event was reset.	• 0 = Event was not reset • 1 = Event was reset
EventStatus[x].EventsPending	SINT	Number of events currently queued in the module. The module queues events when the owner-controller cannot keep up with the rate of events that are triggered.	• 0 = No events currently queued • One or greater = Number of events currently queued
EventStatus[x].EventNumber	DINT	Number of events that have been triggered.	Any

Output Tags

Table 22 describes the Output tags.

Table 22 - 5069-HSC2x0B4 High-speed Counter Module - Output Tags

Name	Size	Description	Valid Values
Counterxx.Reset	BOOL	Resets Count and Rollover values to zero.	• 0 = Do not reset values to zero • 1 = Reset values to zero
Counterxx.Hold	BOOL	Enables or disables the counter so counting can occur or not. **IMPORTANT**: When this tag is set to 1, the counter is disabled and the current count value is held regardless of incoming input data.	• 0 = Enables the counter and counting occurs (default) • 1 = Disables the counter and counting does not occur.
Counterxx.Load	BOOL	Changes Count value to the value of the LoadCountValue tag.	• 0 = Do not change Count value • 1 = Change Count value
Counterxx.Store	BOOL	Copies the Count value to the StoreCount tag	• 0 = Do not copy the Count value • 1 = Copy the Count value to the StoreCount tag
Counterxx.Direction	BOOL	Sets the counter direction. **IMPORTANT**: If the counter is operating in the Up/Down Pulses or Quadrature mode, or if the Direction input terminal is defined in the Count/Direction mode, this tag is ignored.	• 0 = Count up • 1 = Count down
Counterxx.RolloverAck	BOOL	Clears the Rollover tag	• 0 = Does not clear the Rollover tag • 1 = Clears the Rollover tag

Table 22 - 5069-HSC2x0B4 High-speed Counter Module - Output Tags

Name	Size	Description	Valid Values
Counterxx.RollunderAck	BOOL	Clears the Rollunder tag	• 0 = Does not clear the Rollunder tag • 1 = Clears the Rollunder tag
Counterxx.FrequencyAlarmUnlatch	BOOL	Unlatches the Frequency Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.FrequencyAvgAlarmUnlatch	BOOL	Unlatches the Frequency Average Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.PulseWidthAlarmUnlatch	BOOL	Unlatches the Pulse Width Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.PulseWidthAvgAlarmUnlatch	BOOL	Unlatches the Pulse Width Average Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.ZeroFrequencyAlarmUnlatch	BOOL	Unlatches the Zero Frequency Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.ZeroFrequencyAvgAlarmUnlatch	BOOL	Unlatches the Zero Frequency Average Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.MissingPulseAlarmEn	BOOL	Enables the Missing Pulse Alarm	• 0 = Alarm is disabled • 1 = Alarm is enabled
Counterxx.MissingPulseAlarmUnlatch	BOOL	Unlatches the Missing Pulse Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.AccelAlarmUnlatch	BOOL	Unlatches the Acceleration Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.DecelAlarmUnlatch	BOOL	Unlatches the Deceleration Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.AccelAvgAlarmUnlatch	BOOL	Unlatches the Acceleration Average Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.DecelAvgAlarmUnlatch	BOOL	Unlatches the Deceleration Average Alarm.	• 0 = Alarm remains latched • 1 = Alarm is unlatched
Counterxx.ResetFrequencyOverrange	BOOL	If Frequency Overrange is set, this tag resets the Frequency Overrange to zero.	• 0 = Frequency Overrange is not reset • 1 = Frequency Overrange is reset
Counterxx.ResetQuadratureErrorCount	BOOL	Resets the Quadrature Error count	• 0 = Count is not reset • 1 = Count is reset
Counterxx.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
Counterxx.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
Counterxx.ZeroFrequencyAlarmLimit	REAL	Value that the frequency or average frequency must be less than to trigger the Zero Frequency Alarm.	Any
Counterxx.LoadCountValue	DINT	Preset number of counts that can be loaded in the Counterxx.Count input tag.	Any
Counterxx.LoadRevolutionValue	DINT	Preset number of counts that can be loaded in the Counterxx.RevoltuionCount input tag.	Any
Counterxx.OverrideDataAEn	BOOL	Enables the A input to be overwritten by the OverrideDataAValue tag.	• 0 = Input state is not overwritten • 1 = Input state is overwritten
Counterxx.OverrideDataBEn	BOOL	Enables the B input to be overwritten by the OverrideDataBValue tag.	• 0 = Input state is not overwritten • 1 = Input state is overwritten

Table 22 - 5069-HSC2x0B4 High-speed Counter Module - Output Tags

Name	Size	Description	Valid Values
Counterxx.OverrideDataZEn	BOOL	Enables the Z input to be overwritten by the OverrideDataZValue tag.	• 0 = Input state is not overwritten • 1 = Input state is overwritten
Counterxx.OverrideDataAValue	BOOL	Value that is applied to the A input when the OverrideDataAEn tag is set.	Any
Counterxx.OverrideDataBValue	BOOL	Value that is applied to the B input when the OverrideDataBEn tag is set.	Any
Counterxx.OverrideDataZValue	BOOL	Value that is applied to the Z input when the OverrideDataZEn tag is set.	Any
Counterxx.MissingPulseAlarmLimit	DINT	Sets the length of time that must elapse between detecting the leading edge of consecutive pulses befor the Missing Pulse Alarm is triggered.	• 0 = Missing Pulse Alarm disabled • 1 or greater = Time, in microseconds, allowed before the Missing Pulse Alarm is triggered
Windowxx.On	REAL DINT (If window is configured for instantaneous count.)	The average, count, frequency, or pulse width at which the Window turns on.	Any For a window that is configured for frequency mode, the following applies: • If the value of this tag is less than the value of the ZeroFrequencyAlarmLimit output tag, value of the ZeroFrequencyAlarmLimit tag is used. For a window that is configured for Pulse Width mode, the following applies: • If the value of this tag is less than 1/ZeroFrequencyAlarmLimit output tag, 1/ ZeroFrequencyAlarmLimit tag is used. • If the value of this tag is $\leq$ 0, the minimum normalized positive real number is used.
Windowxx.Off	REAL DINT (If window is configured for instantaneous count.)	The acceleration, count, frequency, or pulse width at which the Window turns off.	Any For a window that is configured for frequency mode, the following applies: • If the value of this tag is less than the value of the ZeroFrequencyAlarmLimit output tag, value of the ZeroFrequencyAlarmLimit tag is used. For a window that is configured for Pulse Width mode, the following applies: • If the value of this tag is less than 1/ZeroFrequencyAlarmLimit output tag, 1/ ZeroFrequencyAlarmLimit tag is used. • If the value of this tag is $\leq$ 0, the minimum normalized positive real number is used.
Windowxx.HysteresisOn	REAL DINT (If window is configured for instantaneous count.)	Negative offset from the Window.On tag value to apply hysteresis to.	Any If the value $\leq$ 0, hysteresis is disabled.

Table 22 - 5069-HSC2x0B4 High-speed Counter Module - Output Tags

Name	Size	Description	Valid Values
Windowxx.HysteresisOff	REAL DINT (If window is configured for instantaneous count.)	Negative offset from the Window.Off tag value to apply hysteresis to.	Any If the value ≤ 0, hysteresis is disabled.
Outputxx.OverrideDataEn	BOOL	Enables the output state to be overridden by the OverrideDataValue tag.	• 0 = Override disabled • 1 = Override enabled
Outputxx.OverrideDataValue	BOOL	Value that is applied to the output when OverrideDataEn is set to 1.	Any

Event Input Tags

Table 23 describes the Event Input tags.

IMPORTANT The Logix Designer application project only creates the Event Input tags for a module if you use the Data with Events connection type in the module configuration.

For more information on configuring your 5069 Compact I/O™ high-speed counter module, see Chapter 4, Configure the High-speed Counter Module on page 65.

Table 23 - 5069-HSC2x0B4 High-speed Counter Module - Event Input Tags

Name	Size	Description	Valid Values
RunMode	BOOL	Input's operating state when the event is triggered.	• 0 = Idle • 1 = Run
ConnectionFaulted	BOOL	Indicates if a connection is working The module sets the value to 0 when connected. If the module is not connected, the controller overwrites the value with a 1.	• 0 = Connection running • 1 = Connection not running
DiagnosticActive	BOOL	Indicates if any diagnostics are active.	• 0 = No diagnostics active • 1 = One or more diagnostics are active
CIPSyncValid	BOOL	Module is currently synchronized with a 1588 master.	• 0 = Module is not synchronized with a 1588 master. • 1 = Module is synchronized with a 1588 master.
CIPSyncTimeout	BOOL	Module was previously synchronized with a 1588 master but no longer is due to a timeout.	• 0 = Module has never been synchronized with a 1588 master. • 1 = Module was synchronized with a 1588 master but no longer is due to a timeout.
DiagnosticSequenceCount	SINT	Increments for each time a distinct diagnostic condition is detected, and when a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.	-128…127 The value of 0 is skipped except during module power-up.

Table 23 - 5069-HSC2x0B4 High-speed Counter Module - Event Input Tags

Name	Size	Description	Valid Values
Eventxx.Fault	BOOL	Indicates that 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. **IMPORTANT**: Once the condition that causes the tag to change to 1 is removed, the tag automatically resets to 0.	• 0 = Good data • 1 = Bad data, causing fault Typically, this tag is set when the Counter*xx*.QuadratureErrorCount tag is any number other than 0. The Counter*xx*.QuadratureErrorCount tag only applies when the high-speed counter module is operating in the X1, X2, or X4 Quadrature mode.
Eventxx.Uncertain	BOOL	Indicates that the event 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.	• 0 = Good data • 1 = Uncertain data The following are causes of uncertain data: – 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 $> T_{P\text{-}MAX}$. If the pulses are longer than TP-MAX, the module assumes the input frequency is 0. However, the module cannot determine if there are no pulses being measured because no pulses exist or because the module is filtering a high frequency signal. ($T_{P\text{-}MAX} = 10$ s) – Measured pulse width is too short. That is, the pulse width is less than the module's rated minimum pulse width, $T_{PW\text{-}MIN}$. ($T_{PW\text{-}MIN} = 125$ ns) – If the period defined in the Counterxx:AverageOverPulses configuration tag is greater than TP-MAX. In this case, the average frequency and average pulse width are reported based on the length of the complete pulses measured.
Eventxx.InvalidConfiguration	BOOL	Indicates that there is an error in the Event Trigger definition.	• 0 = No error in the Event Trigger definition • 1 = Error in the Event Trigger definition
Eventxx.EventDropped	BOOL	Indicates if an event has been discarded because events are occurring faster than they are being acknowledged.	• 0 = Event was not discarded • 1 = Event was discarded
Eventxx.EventRising	BOOL	Indicates if an event triggered when an input transition results in an event pattern being matched.	• 0 = Event was not triggered • 1 = Event was triggered
Eventxx.EventFalling	BOOL	Indicates if an event triggered when an input transition resulted in an event pattern no longer being matched.	• 0 = Event was not triggered • 1 = Event was triggered
Eventxx.CIPSyncValid	BOOL	Indicates that the module was synchronized with a 1588 master when the event was triggered.	• 0 = Module was not synchronized with a 1588 master when the event was triggered. • 1 = Module was synchronized with a 1588 master when the event was triggered.

Table 23 - 5069-HSC2x0B4 High-speed Counter Module - Event Input Tags

Name	Size	Description	Valid Values
Eventxx.CIPSyncTimeout	BOOL	Module was previously synchronized with a 1588 master but due to a timeout was no longer synchronized when the event was triggered.	• 0 = Module has never been synchronized with a 1588 master. • 1 = Module was synchronized with a 1588 master but due to a timeout was not synchronized when the event was triggered.
Eventxx.EventsPending	SINT	Number of events currently queued in the module. The module queues events when the owner-controller cannot keep up with the rate of events that is occurring.	• 0 = No events currently queued • One or greater = Number of events currently queued
Eventxx.EventNumber	DINT	Number of events that had already occurred when the new event occurred.	Any
Eventxx.EventTimestamp	LINT	Time that the event occurred.	Any positive number
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	• 0 = Period of the pulse was normal when the event was triggered. • 1 = Period of the pulse was greater than the missing pulse timeout or was never completed when the event was triggered.
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.	• 0 = Counter frequency was not less than the Zero Frequency Alarm limit when the event was triggered. • 1 = Counter frequency was less than the Zero Frequency Alarm limit when the event was triggered.
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.	• 0 = Counter average frequency was not less than the Zero Frequency Alarm limit when the event was triggered. • 1 = Counter average frequency was less than the Zero Frequency Alarm limit when the event was triggered.
Eventxx.FrequencyAlarm	BOOL	Indicates that when the event was triggered, the counter frequency had exceeded the Frequency Alarm limit.	• 0 = Counter frequency had not exceeded the Frequency Alarm limit when the event was triggered. • 1 = Counter frequency exceeded the Frequency Alarm limit when the event was triggered.
Eventxx.FrequencyAvgAlarm	BOOL	Indicates that when the event was triggered, the counter's average frequency had exceeded the Frequency Alarm limit.	• 0 = Counter's average frequency had not exceeded the Frequency Alarm limit when the event was triggered. • 1 = Counter's average frequency has exceeded the Frequency Alarm limit when the event was triggered.
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.	• 0 = Alarm not triggered. That is, the counter pulse width was not greater than the Pulse Width Alarm limit when the event was triggered. • 1 = Alarm is triggered. That is, the counter pulse width was greater than the Pulse Width Alarm limit when the event was triggered.

Table 23 - 5069-HSC2x0B4 High-speed Counter Module - Event Input Tags

Name	Size	Description	Valid Values
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.	• 0 = Alarm not triggered. That is, the counter average pulse width was not greater than the Pulse Width Alarm limit when the event was triggered. • 1 = Alarm is triggered. That is, the counter average pulse width was greater than the Pulse Width Alarm limit when the event was triggered.
Eventxx.AccelAlarm	BOOL	Indicates that when the event was triggered, the counter acceleration rate exceeded the Acceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter acceleration rate did not exceed the Acceleration Alarm limit when the event was triggered. • 1 = Alarm is triggered. That is, the counter acceleration rate exceeded the Acceleration Alarm limit when the event was triggered.
Eventxx.AccelAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average acceleration rate exceeded the Acceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter average acceleration rate did not exceed the Acceleration Alarm limit when the event was triggered. • 1 = Alarm is triggered. That is, the counter average acceleration rate exceeded the Acceleration Alarm limit when the event was triggered.
Eventxx.DecelAlarm	BOOL	Indicates that when the event was triggered, the counter deceleration rate exceeded the Deceleration Alarm limit, triggering this alarm.	• 0 = Alarm not triggered. That is, the counter deceleration rate did not exceed the Deceleration Alarm limit when the event was triggered. • 1 = Alarm is triggered. That is, the counter deceleration rate exceeded the Deceleration Alarm limit when the event was triggered.
Eventxx.DecelAvgAlarm	BOOL	Indicates that when the event was triggered, the counter average deceleration rate exceeded the Deceleration Alarm limit, triggering this alarm.	• 0 = Alarm was not triggered. That is, the counter average deceleration rate did not exceed the Deceleration Alarm limit when the event was triggered. • 1 = Alarm was triggered. That is, the counter average deceleration rate exceeded the Deceleration Alarm limit when the event was triggered.
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.	• 0 = Counter frequency did not exceed the module's maximum rated frequency when the event was triggered. • 1 = Counter frequency exceeded the module's maximum rated frequency when the event was triggered.
Eventxx.Reset	BOOL	Indicates that when the event was triggered, the Count and Rollover values were reset to zero.	• 0 = Values were not reset to zero when the event was triggered. • 1 = Values were reset to zero when the event was triggered.

Table 23 - 5069-HSC2x0B4 High-speed Counter Module - Event Input Tags

Name	Size	Description	Valid Values
Eventxx.Hold	BOOL	Indicates that when the event was triggered, counting occurred	• 0 = Counting occurred when the event was triggered. • 1 = Counting did not occur when the event was triggered.
Eventxx.Load	BOOL	Indicates that when the event was triggered, the Count value was changed to the value of the LoadCountValue tag.	• 0 = Count value was not changed when the event was triggered. • 1 = Count value was changed when the event was triggered.
Eventxx.Store	BOOL	Indicates that the Count value was copied to the StoreCount tag when the event was triggered.	• 0 = Count value was not copied when the event was triggered. • 1 = Count value was copied to the StoreCount tag when the event was triggered.
Eventxx.Direction	BOOL	Indicates the count direction when the event was triggered.	• 0 = Counting up when the event was triggered. • 1 = Counting down when the event was triggered.
Eventxx.StoredDirection	BOOL	Indicates the count direction when the Store input terminal transitioned from Off to On when the event was triggered.	• 0 = Count up when the event was triggered. • 1 = Count down when the event was triggered.
Eventxx.Rollover	BOOL	Indicates that the counter had counted up to Rollover value and then continued counting from the Rollunder when the event was triggered.	• 0 = Counter had not counted to the Rollover value and continued counting from the Rollunder value when the event was triggered. • 1 = Counter had counted to the Rollover value and continued counting from the Rollunder value when the event was triggered.
Eventxx.Rollunder	BOOL	Indicates the counter had counted down to Rollunder value and then continued counting from the Rollover when the event was triggered.	• 0 = Counter had not counted down to the Rollunder value and continued counting from the Rollover value when the event was triggered. • 1 = Counter had counted down to the Rollunder value and continued counting from the Rollover value when the event was triggered.
Eventxx.DataA	BOOL	Current input A value when the event was triggered.	• 0 = Input A was off when the event was triggered. • 1 = Input A was on when the event was triggered.
Eventxx.DataB	BOOL	Current input B value when the event was triggered.	• 0 = Input B was off when the event was triggered. • 1 = Input B was on when the event was triggered.
Eventxx.DataZ	BOOL	Current input Z value when the event was triggered.	• 0 = Input Z was off when the event was triggered. • 1 = Input Z was on when the event was triggered.
Eventxx.InWindow00	BOOL	Indicates that the signal value was in the specified window when the event was triggered. Specified Window can be any of eight windows, that is, window00…window07.	• 0 = Signal value was not in the specified window when the event was triggered. • 1 = Signal value was in the specified window when the event was triggered.

Event Output Tags

Table 24 describes the Event Output tags.

IMPORTANT The Logix Designer application project only creates the Event Output tags for a module if you use the Data with Events connection type in the module configuration.

For more information on configuring your 5069 Compact I/O high-speed counter module, see Chapter 4, Configure the High-speed Counter Module on page 65.

Table 24 - 5069-HSC2x0B4 High-speed Counter Module - Event Output Tags

Name	Size	Description	Valid Values
Eventxx.En	BOOL	Sets the corresponding event trigger definition to active. Events are triggered once when conditions match the definition.	• 0 = Event trigger definition is not active • 1 = Event trigger definition is active
Eventxx.EventRisingEn	BOOL	Triggers an event each time a condition change results in conditions matching the event trigger definition.	• 0 = Event is not triggered. • 1 = Event is triggered.
Eventxx.EventFallingEn	BOOL	Triggers an event each time a condition change results in conditions no longer matching the event trigger definition.	• 0 = Event is not triggered. • 1 = Event is triggered.
Eventxx.LatchEvent	BOOL	Latches an event until it is acknowledged. A new event is lost if the previous event has not been acknowledged. When not set, new events overwrite old events.	• 0 = Event is not latched. In this case, new events overwrite old events. • 1 = Event is latched. In this case, new events are lost if the latched event has not been acknowledged.
Eventxx.ResetEvent	BOOL	Indicates the state of the Reset function when the event was triggered. That is, were the Count and Rollover value reset to 0.	• 0 = Reset function did not reset the Count and Rollover values to 0 when the event was triggered. • 1 = Reset function reset the Count and Rollover values to 0 when the event was triggered.
Eventxx.IndependentConditionTriggerEn	BOOL	Determines whether each condition indicated in the Event trigger definition can initiate an event independently.	• 0 = When all selected conditions achieve the configured values an event is triggered. • 1 = When any selected condition achieves the configured value an event is triggered.
Eventxx.CounterSelect	SINT	Counter with which the event is associated, if any.	• 0 = Counter 0 • 1 = Counter 1 • 0xFF = Not associated with any counter.

Table 24 - 5069-HSC2x0B4 High-speed Counter Module - Event Output Tags

Name	Size	Description	Valid Values
Eventxx.EventNumberAck	DINT	When an event is triggered, the controller writes back the value of EventNumber tag to this tag to indicate receipt of the event. All events with EventNumbers that occurred before EventNumberAck acknowledged.	Any
Eventxx.InWindowxxSelect	BOOL	Indicates that the specified window participates in the Event trigger definition for the event. Specified Window can be any of eight windows, that is, window00…window07.	• 0 = Window does not participate in the Event trigger definition • 1 = Window participates in the Event trigger definition
Eventxx.ResetSelect	BOOL	Indicates that the Counter Reset function participates in the Event trigger definition.	• 0 = Counter's Reset function does not participate in the Event trigger definition. • 1 = Counter's Reset function participates in the Event trigger definition.
Eventxx.HoldSelect	BOOL	Indicates that the counter's Hold function participates in the Event trigger definition.	• 0 = Counter's Hold function does not participate in the Event trigger definition. • 1 = Counter's Hold function participates in the Event trigger definition.
Eventxx.LoadSelect	BOOL	Indicates that the counter's Load function participates in the Event trigger definition.	• 0 = Counter's Load function does not participate in the Event trigger definition. • 1 = Counter's Load function participates in the Event trigger definition.
Eventxx.StoreSelect	BOOL	Indicates that the counter's Store function participates in the Event trigger definition.	• 0 = Counter's Store function does not participate in the Event trigger definition. • 1 = Counter's Store function participates in the Event trigger definition.
Eventxx.RolloverSelect	BOOL	Indicates that the counter's Rollover function participates in the Event trigger definition.	• 0 = Counter's Rollover function does not participate in the Event trigger definition. • 1 = Counter's Rollover function participates in the Event trigger definition.
Eventxx.RollunderSelect	BOOL	Indicates that the counter's Rollunder function participates in the Event trigger definition.	• 0 = Counter's Rollunder function does not participate in the Event trigger definition. • 1 = Counter's Rollunder function participates in the Event trigger definition.

Table 24 - 5069-HSC2x0B4 High-speed Counter Module - Event Output Tags

Name	Size	Description	Valid Values
Eventxx.DirectionSelect	BOOL	Indicates that the counter's Direction function participates in the Event trigger definition.	• 0 = Counter's Direction function does not participate in the Event trigger definition. • 1 = Counter's Direction function participates in the Event trigger definition.
Eventxx.InWindowxxValue	BOOL	If the specified window is selected in the Event trigger definition, this tag indicates that value that is to trigger the event. Specified Window can be any of eight windows, that is, window00…window07.	• 0 = Window value does not participate in the Event trigger definition. • 1 = Window value participates in the Event trigger definition.
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.	• 0 = Reset function value does not trigger the event. • 1 = Reset function value triggers the event.
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.	• 0 = Hold function value does not trigger the event. • 1 = Hold function value triggers the event.
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.	• 0 = Load function value does not trigger the event. • 1 = Load function value triggers the event.
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.	• 0 = Store function value does not trigger the event. • 1 = Store function value triggers the event.
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.	• 0 = Rollover function value does not trigger the event. • 1 = Rollover function value triggers the event.
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.	• 0 = Rollunder function value does not trigger the event. • 1 = Rollunder function value triggers the event.
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.	• 0 = Direction function value does not trigger the event. • 1 = Direction function value triggers the event.

Numerics

A

C

D

E

F

H

I

U

W

X

Z

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support you can find technical and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com/ for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/rockwellautomation/support/overview.page, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to help ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page.

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 5000-UM006B-EN-P - November 2015