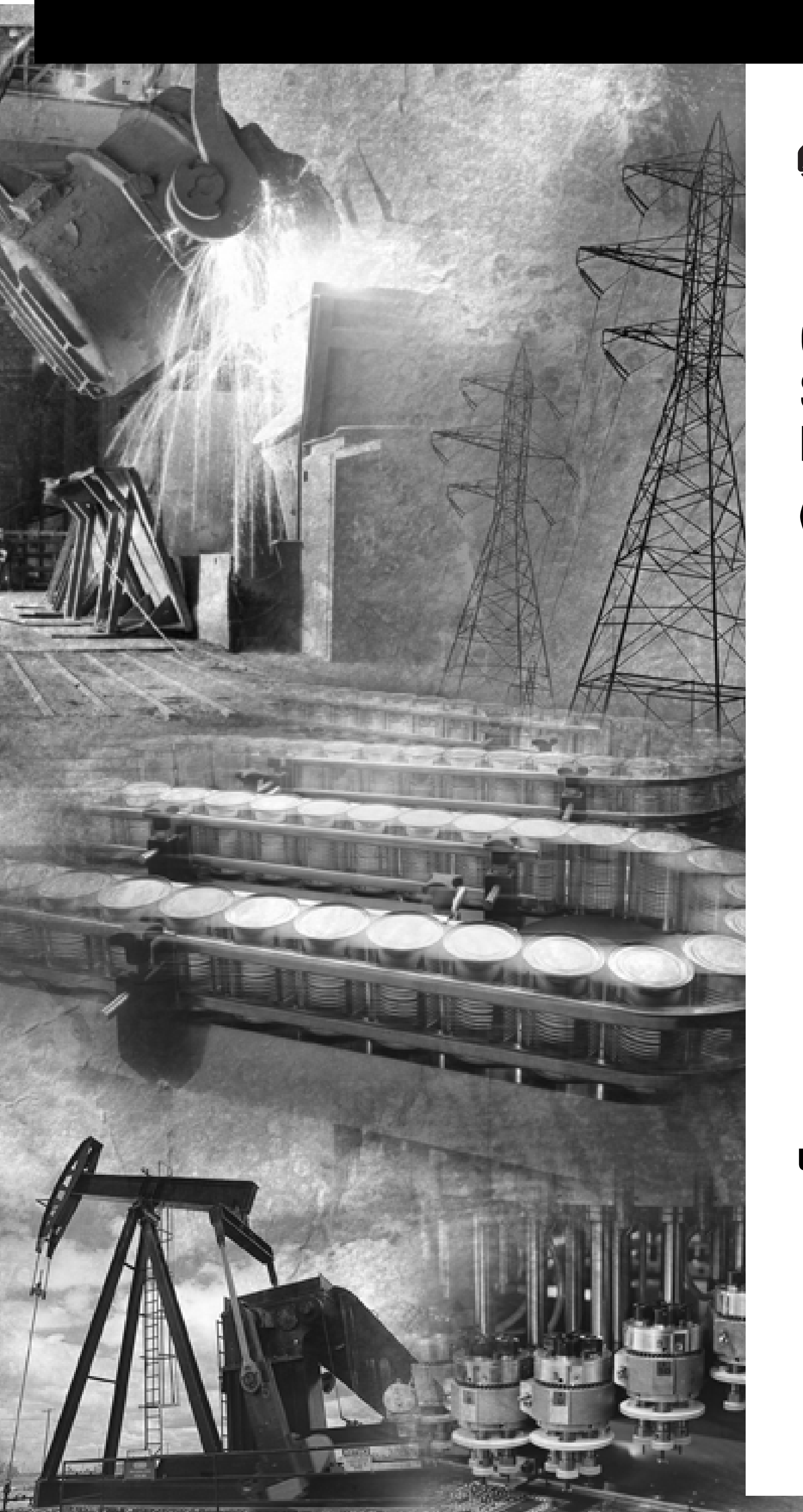
Allen-Bradley



Compact™ High Speed Counter Module

(Catalog Number 1769-HSC)

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION	Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

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

Preface

Who Should Use This Manual P-1
How to Use This Manual P-1
Conventions Used in This Manual P-2
Rockwell Automation Support P-3

Chapter 1
Module Overview

Module Overview 1-1
Hardware Features 1-3

Chapter 2
Module Operation

Counter Defaults 2-1
Module Operation Block Diagrams 2-2
Number of Counters 2-4
Summary of Available Counter Configurations 2-4
Input Filtering 2-6
Operational Mode Selection 2-7
Input Frequency 2-14
Counter Types 2-14
Modifying Count Value 2-15
Rate/Timer Functionality 2-18
Output Control 2-23

Chapter 3
Installation and Wiring

Compliance to European Union Directives 3-1
Power Requirements 3-2
General Considerations 3-2
System Assembly 3-6
Mounting 3-7
Replacing the Module within a System 3-10
Field Wiring Connections 3-11

Chapter 4
Module Configuration, Output, and Input Data

Configuring the Module 4-1
Configuration Array 4-2
Output Array 4-21
Input Array 4-31

Chapter 5
Diagnostics and Troubleshooting

Safety Considerations 5-1
Module Operation vs. Counter Operation 5-3
Counter Defaults 5-3
Module Diagnostics 5-4
Non-Critical vs. Critical Module Errors 5-5
Module Error Definition 5-5
Error Codes 5-8
Contacting Rockwell Automation 5-14

Appendix A
Specifications

General Specifications A-1
Input Specifications. A-2
Output Specifications A-2
Throughput and Timing A-3
Rate Accuracy A-3
Temperature Derating A-4

Appendix B
1769-HSC Module with CompactLogix Controllers and an Allen-Bradley 845F Encoder

System Diagram B-1
845F Encoder Wiring to the 1769-HSC B-1
Purpose B-2
Scope B-2
Adding a 1769-HSC High Speed Counter Module into a CompactLogix System B-2
Configuring Your 1769-HSC Module B-6
Monitoring the Current Count Value and Verifying Output Operation. B-10

Appendix C
1769-HSC Module with MicroLogix 1500 Controllers and an Allen-Bradley 845F Encoder

System Diagram C-1
845F Encoder Wiring to the 1769-HSC C-1
Purpose C-2
Scope C-2
Adding a 1769-HSC High Speed Counter Module into a MicroLogix 1500 System C-2
Configuring Your 1769-HSC Module C-5
Monitoring the Current Count and Verifying Output Operation. C-8

Appendix D
Programming Quick Reference

Configuration Array. D-2
Output Array D-4
Input Array D-5

Index

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- how to use this manual
- related publications
- conventions used in this manual
- Rockwell Automation support

Who Should Use This Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use Allen-Bradley Compact™ I/O and/or Micrologix™ 1500 or CompactLogix™ controllers.

How to Use This Manual

As much as possible, we organized this manual to explain, in a task-by-task manner, how to install, configure, program, operate and troubleshoot a control system using the 1769 High Speed Counter modules.

Manual Contents

If you want...	See
An overview of the module	Chapter 1
A description of module operation, including counters, inputs, and outputs.	Chapter 2
Installation and wiring guidelines	Chapter 3
Module addressing, configuration and status information	Chapter 4
Information on module diagnostics and troubleshooting	Chapter 5
Specifications	Appendix A
Programming and Configuration for CompactLogix	Appendix B
Programming and Configuration for MicroLogix 1500	Appendix C
Programming Quick Reference	Appendix D

Related Documentation

The table below provides a listing of publications that contain important information about Compact I/O, CompactLogix, and MicroLogix 1500 systems.

For	Read this document	Document number
A user manual containing information on how to install, use and program your MicroLogix 1500 controller	MicroLogix™ 1500 User Manual	1764-UM001A-US-P
A user manual containing information on how to install, use, and program your CompactLogix processor.	CompactLogix™ User Manual	1769-UM007C-EN-P
A user manual containing information on how to install, and use your 1769-ADN DeviceNet Adapter.	DeviceNet Adapter User Manual	1769-UM001A-US-P
An overview of 1769 Compact Discrete I/O modules	1769 Compact Discrete Input/Output Modules Product Data	1769-2.1
An overview of the MicroLogix 1500 System, including 1769 Compact I/O.	MicroLogix™ 1500 System Overview	1764-SO001B-EN-P
In-depth information on grounding and wiring Allen-Bradley programmable controllers.	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1

If you would like a manual, you can:

- download a free electronic version from the internet at **www.theautomationbookstore.com**
- purchase a printed manual by:
  - contacting your local distributor or Rockwell Automation representative
  - visiting **www.theautomationbookstore.com** and placing your order
  - calling 1.800.963.9548 (USA/Canada) or 001.330.725.1574 (Outside USA/Canada)

Conventions Used in This Manual

The following conventions are used throughout this manual:

- Bulleted lists (like this one) provide information not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.
- Text in this `font` indicates words or phrases you should type.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 Sales/Support Offices, 512 authorized distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreement

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in Chapter 5, Diagnostics and Troubleshooting first. Then call your local Rockwell Automation representative.

Your Questions or Comments on the Manual

If you find a problem with this manual, please notify us. If you have any suggestions for how this manual could be made more useful to you, please contact us at the address below:

Rockwell Automation
Automation Control and Information Group
Technical Communication, Dept. A602V
P.O. Box 2086
Milwaukee, WI 53201-2086

Module Overview

This chapter contains the following:

- module overview
- hardware features

Module Overview

The 1769-HSC is an intelligent counter module with its own microprocessor and I/O that is capable of reacting to high speed input signals. The module can interface with up to 2 channels of quadrature or 4 channels of pulse/count inputs. The signals received at the inputs are filtered, decoded, and counted. They are also processed to generate rate and time-between-pulses (pulse interval) data. Count and rate values can then be used to activate outputs based on user-defined ranges.

The module counts pulses at up to 1 MHz from devices such as proximity switches, pulse generators, turbine flowmeters, and quadrature encoders. The module has four on-board high speed switching outputs. These outputs can be under user program or direct module control, based on the count value or frequency.

The module is compatible with MicroLogix™ 1500(1) packaged controllers, CompactLogix™(2) modular PLCs, and the Series B(3) 1769-ADN DeviceNet Adapter.

Counters

The module is capable of counting pulses in either direction (forward/reverse, up/down, etc.). A maximum of four pulse counters (or 2 quadrature counters) are available. Each 32-bit counter can count to ±2 billion as a ring or linear counter. In addition to providing a count value, the module provides a rate value up to ±1 MHz, dependent upon the type of input. The rate value (as modified by scalar) is the input frequency to the counter. When the count value is increasing, the rate value is positive. When the count value is decreasing, the rate value is negative.

(1) 1764-LSP and 1764-LRP Series C, Firmware revision 6.0 and higher.

(2) Firmware versions prior to 11.0 require the use of Generic Profiles.

(3) Available Spring 2002.

Counters can also be reset or preset to any value between user-defined minimum and maximum values. Preset can be accomplished from the user program or at a Z input event. The Z input can also generate a capture value and/or freeze (gate) the counters.

Inputs

The module features six high-speed differential inputs labeled ±A0, ±B0, ±Z0, ±A1, ±B1, and ±Z1. These inputs support 2 quadrature encoders with ABZ inputs and/or up to 4 discrete count inputs. In addition, x1, x2, and x4 encoder configurations are provided to fully use the capabilities of high resolution quadrature encoders. The inputs can be wired for standard differential line driver output devices, as well as single-ended devices such as limit switches, photo eyes, and proximity sensors. Inputs are optically isolated from the bus and from one another, and have an operational range of 2.6 to 30V dc.

Outputs

Sixteen outputs are available: four on-board (real) and twelve virtual bits. All 16 outputs can be individually controlled by the module or by the user control program.

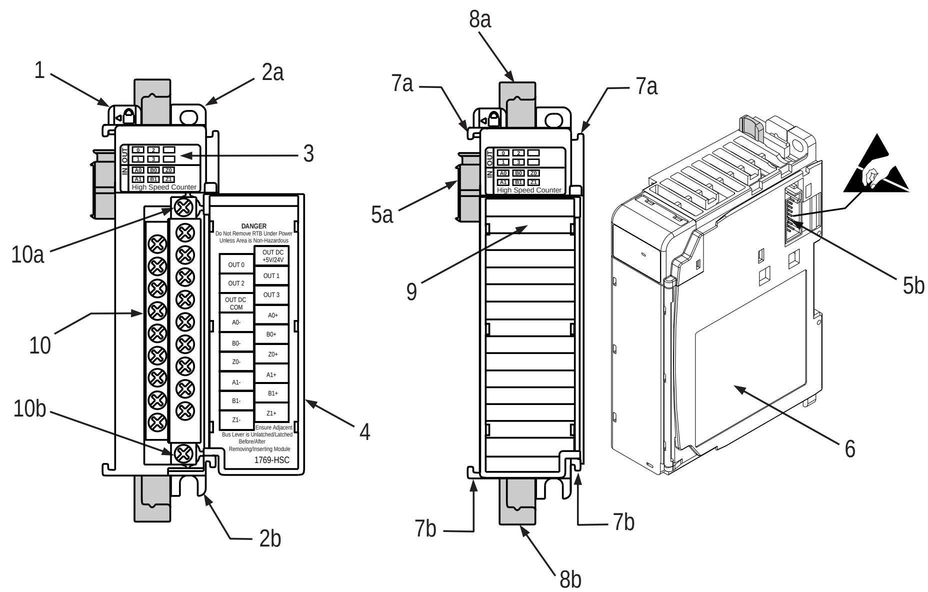
The 4 on-board (real) outputs are dc sourcing, powered by a user-supplied (5 to 30V dc) power source. These outputs are electronically protected from current overloads and short circuit conditions. Overcurrent status is monitored and fed back to the user program. Output states are determined by a combination of output data, configuration data, ranges, and overcurrent status.

See Output Control on page 2-23 for a description of how the module determines output status.

Hardware Features

The module's hardware features are illustrated below. Refer to Chapter 3 for detailed information on installation and wiring.

Figure 1.1 Hardware Features



Item	Description
1	bus lever
2a	upper panel mounting tab
2b	lower panel mounting tab
3	module status LEDs (6 Input, 4 Output, 1 Fuse, 1 OK)
4	module door with terminal identification label
5a	movable bus connector (bus interface) with female pins
5b	stationary bus connector (bus interface) with male pins
6	nameplate label
7a	upper tongue-and-groove slots
7b	lower tongue-and-groove slots
8a	upper DIN rail latch
8b	lower DIN rail latch
9	write-on label for user identification tags
10	removable terminal block (RTB) with finger-safe cover
10a	RTB upper retaining screw
10b	RTB lower retaining screw

LEDs

The front panel has a total of twelve indicator LEDs, as shown in Figure 1.1 on page 1-3.

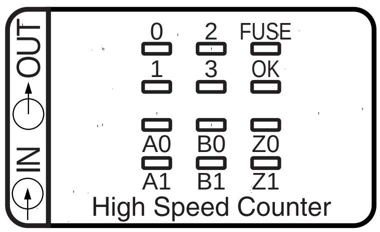


Table 1.1 Diagnostic Indicators

LED	Color	Indicates
0 OUT	Amber	ON/OFF logic status of output 0
1 OUT	Amber	ON/OFF logic status of output 1
2 OUT	Amber	ON/OFF logic status of output 2
3 OUT	Amber	ON/OFF logic status of output 3
FUSE	Red	Overcurrent
OK	Off	No power is applied.
	Red (briefly)	Performing self-test.
	Solid Green	OK, normal operating condition.
	Flashing Green	OK, module in Program or Fault mode.
	Solid Red or Amber	Hardware error. Cycle power to the module. If problem persists, replace the module.
	Flashing Red	Recoverable fault. Reconfigure, reset, or perform error recovery. See section on page 5-5, Non-Critical vs. Critical Module Errors. The OK LED flashes red for all of the error codes in Table 5.6.
A0	Amber	ON/OFF status of input A0
A1	Amber	ON/OFF status of input A1
B0	Amber	ON/OFF status of input B0
B1	Amber	ON/OFF status of input B1
Z0	Amber	ON/OFF status of input Z0
Z1	Amber	ON/OFF status of input Z1
ALL ON	Possible causes for all LEDs to be on: • Bus Error has occurred: Controller hard fault. Cycle power. • During Flash Upgrade of Controller: Normal. Do not cycle power during the Flash Upgrade. • All LEDs will flash on briefly during power-up. This is normal.	

Module Operation

This chapter contains information about:
- counter defaults
- module operation block diagrams
- number of counters
- input filtering
- input operational mode
- modifying count value or input signals
- counter types
- rate/timer functionality
- output control
- safe state control

Counter Defaults

When the module powers-up, all Output Array and Configuration Array values are set to their default values (see Chapter 4 or Appendix D for default values). All Input Array values are cleared. None of the module data is retentive through a power cycle.

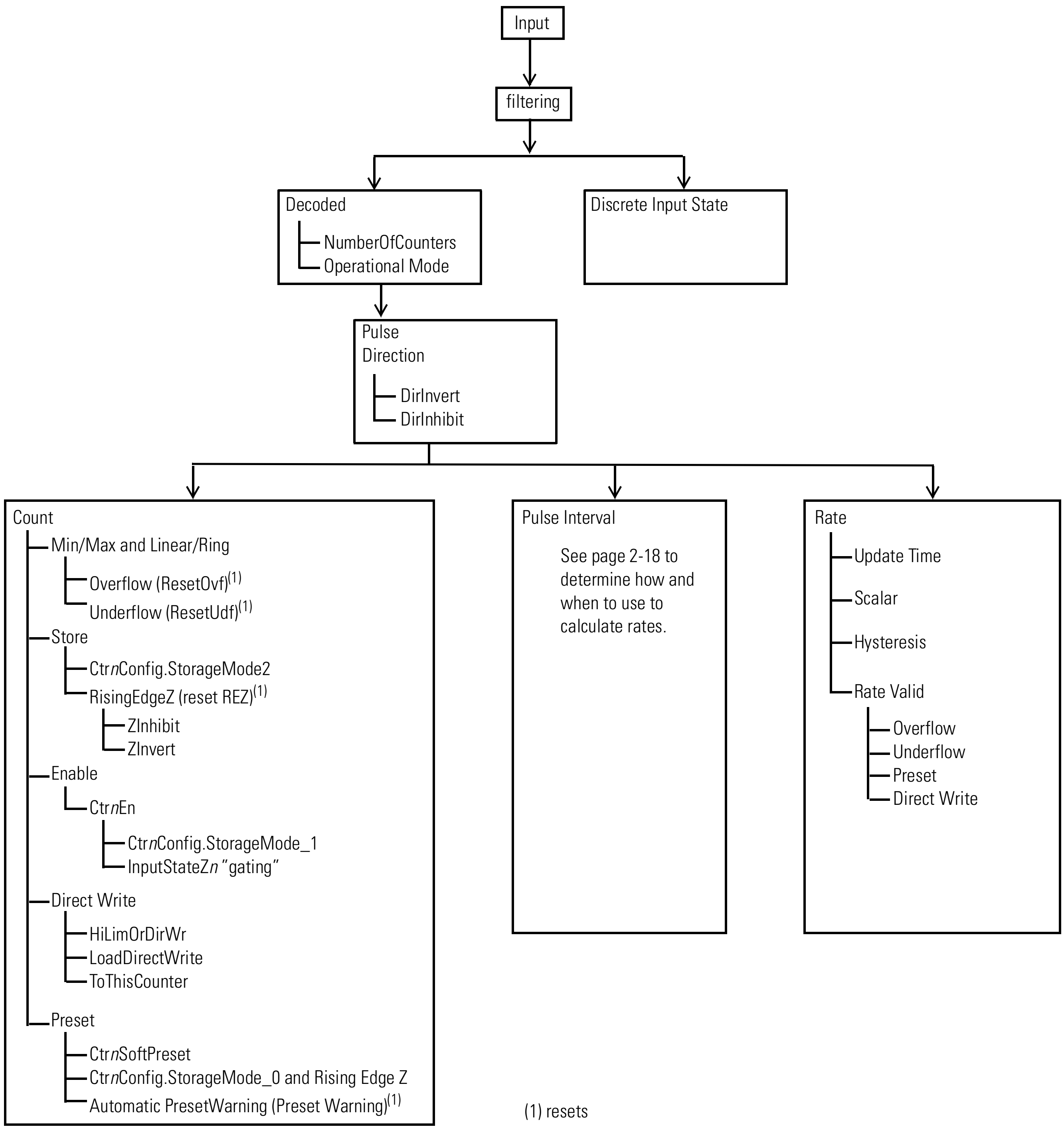
In effect, this means that power cycling clears the module:
- stored counts and configurations are lost
- faults and flags are cleared
- outputs are off

Module Operation Block Diagrams

To provide an overview of the module operation, the block diagrams indicate relationships between module functions and configuration parameters.

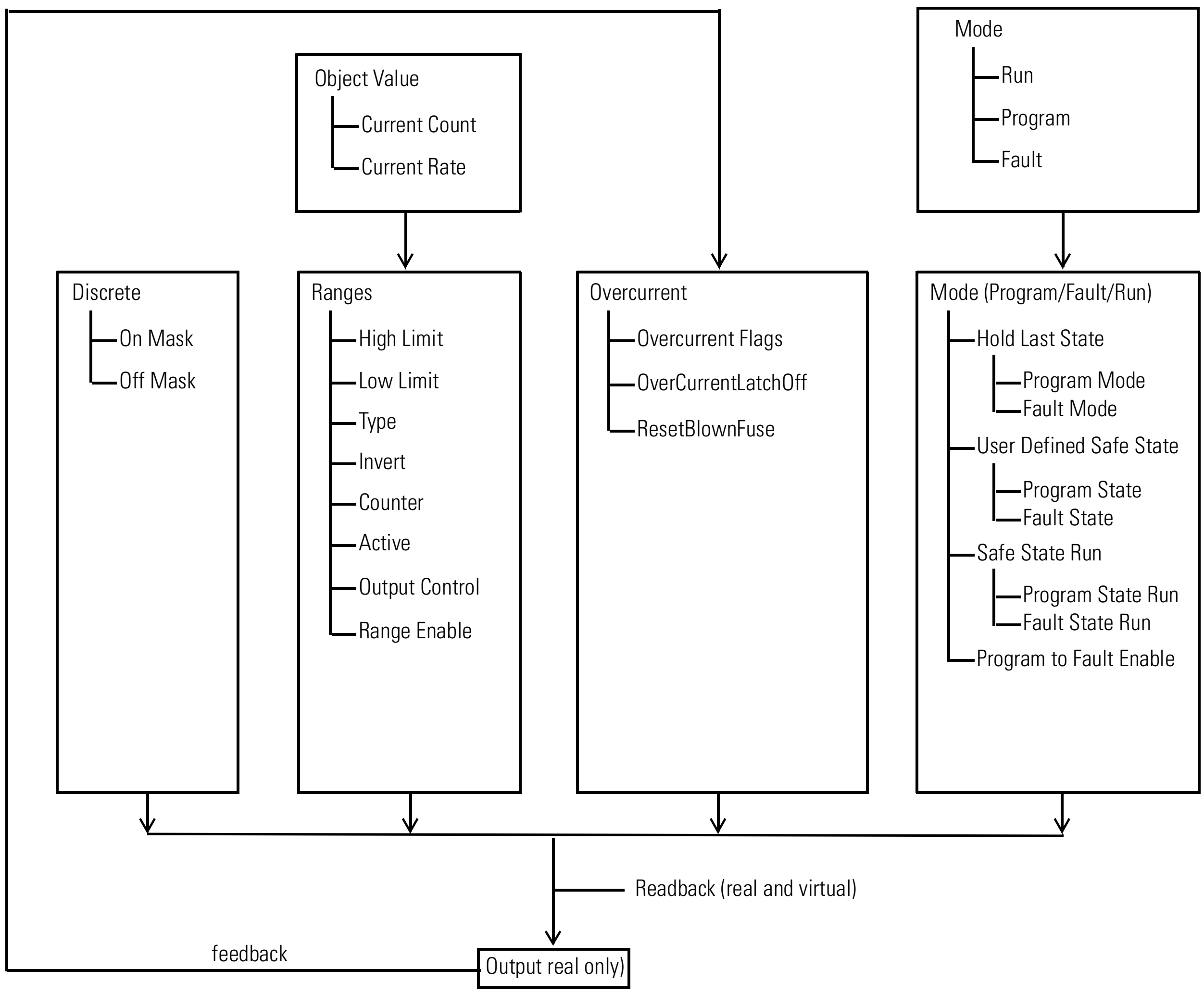
Inputs

The following diagram illustrates how the inputs function.



Outputs

The following diagram illustrates how the outputs function.



Number of Counters

The module has six input points: A0, B0, Z0, A1, B1, and Z1. Through these inputs, the module can function with 1, 2, 3, or 4 counters depending upon the number of counters and the operational mode configuration of the input points.

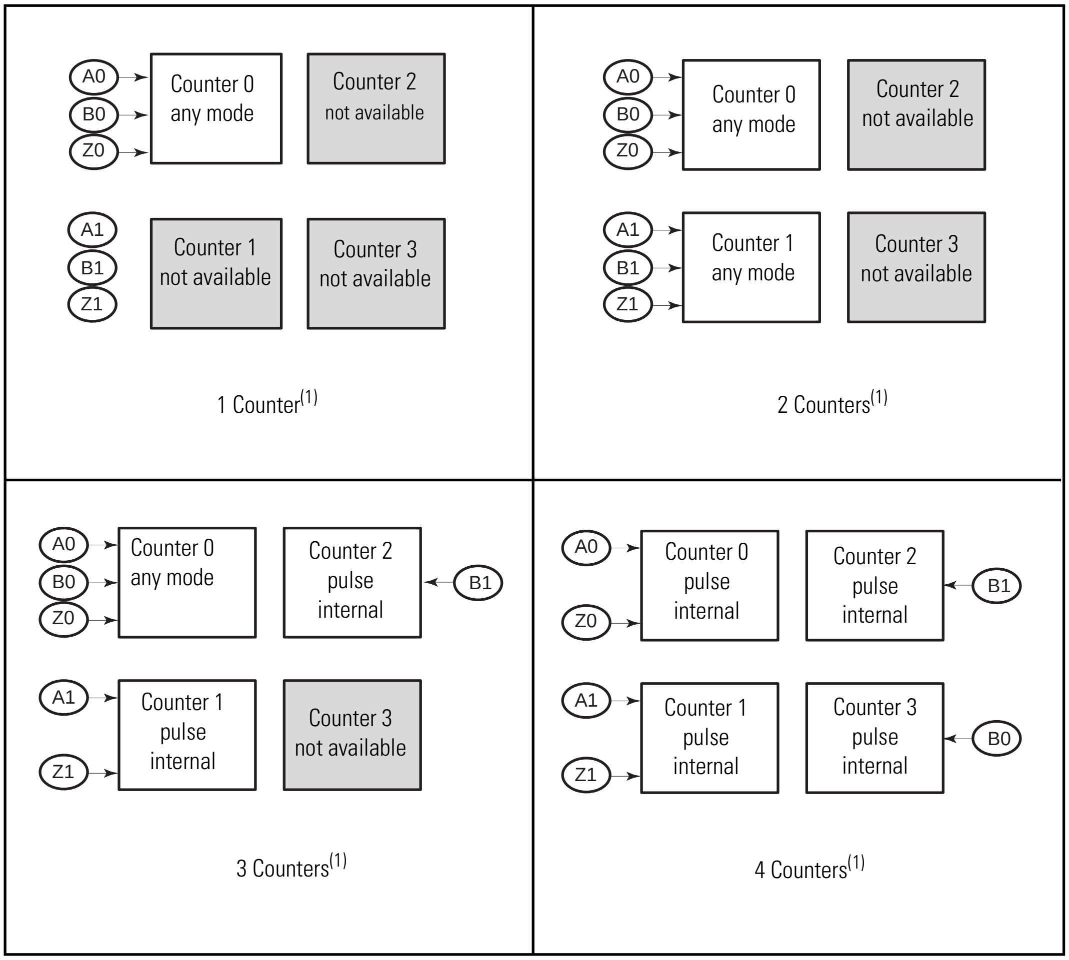
Summary of Available Counter Configurations

The table below summarizes the input configurations available for all counters, based on the number of counters.

Number of Counters	Counter	Operational Mode	Gate or Preset Functionality
1 Counter	0	Any	All
	1 through 3	Not available	
2 Counters	0	Any	All
	1	Any	All
	2 and 3	Not Available	
3 Counters	0	Any	All
	1	Pulse/Internal Direction	All
	2	Pulse/Internal Direction	None
	3	Not available	
4 Counters	0	Pulse/Internal Direction	All
	1	Pulse/Internal Direction	All
	2	Pulse/Internal Direction	None
	3	Pulse/Internal Direction	None

The counter options and operating modes are summarized in Figure 2.1.

Figure 2.1 Summary of Available Counters



(1) The number of counters is defined by the NumberOfCounters bits in word 0 of the Configuration Array.

Input Filtering

In many industrial environments, high frequency noise can be inadvertently coupled to the sensor wires. The module can help reject some noise by means of built-in filters. Inputs are filtered by means of user-selectable, low-pass filters(1) set up during module configuration.

The available nominal pulse width filters are:

Table 2.1 Available Filters

Input	Filter
A0, A1, B0, B1, Z0, Z1	5 ms, 500 µs, 10 µs, no filter

The filters are selected for each input in the Filter Selection word of the Module Configuration Array.

TIP

The input state bits (InputStateA0 through InputStateZ1) reflect the filter's inputs, but are NOT affected by the signal inhibit or invert operations described on page 2-7.

Table 2.2 Filter Pulse Width and Frequency

Nominal Filter Settings		Maximum Guaranteed Blocked Pulse Width		Minimum Guaranteed Pass Pulse Width	
Pulse Width	**Equivalent Frequency(1)**	**Pulse Width**	**Equivalent Frequency(1)**	**Pulse Width**	**Equivalent Frequency(1)**
no filter	1 MHz	n/a	n/a	250 ns	2 MHz
10 µs	50 kHz	7.4 µs	67.5 kHz	25 µs	20 kHz
500 µs	1 kHz	370 µs	1.35 kHz	1.25 ms	400 Hz
5 ms	100 Hz	3.7 ms	135 Hz	12.5 ms	40 Hz

(1) Equivalent frequency assumes a perfect 50% duty cycle and are for reference purposes only. Hence, the no-filter setting is guaranteed to pass 4 MHz even though the module's maximum is 1 MHz. This allows the sensor and wiring to attenuate the pulse to 25% duty cycle while the module maintains pulse recognition.

IMPORTANT

The built-in filters are simple, averaging, low-pass filters. They are designed to block noise pulses of width equal to the values presented in Table 2.2. Applying full amplitude, 50% duty cycle signals that are of frequency above the selected filter's threshold frequency may result in an average value signal of sufficient amplitude to turn the input on. A transition from no input to the full amplitude, 50% duty cycle signal (or back to no signal) may result in inadvertent input transitions.

(1) Low-pass filters block frequencies above the threshold frequency.

Operational Mode Selection

A count channel's operational mode configuration selection determines how the A and B inputs cause a counter channel to increment or decrement. The six available mode selections are:

- Pulse/External Direction Input
- Pulse/Internal Direction Input
- Up and Down Pulse Input
- X1 Quadrature Encoder Input
- X2 Quadrature Encoder Input
- X4 Quadrature Encoder Input

IMPORTANT The operational mode selection is limited by the number of counters selected.

- With 2 counters selected, Counters 0 and 1 can be assigned any operational mode.
- With 3 counters selected, Counter 0 can be assigned any mode, but Counters 1 and 2 can only be configured as pulse/internal direction.
- With 4 counters selected, all counters must be configured for the pulse/internal direction mode.

See the Figure 2.1 on page 2-5 for the operational modes available for the counters, based on the number of counters configured.

Direction Inhibit and Direction Invert Output Control Bits

These bits apply to all of the counter modes.

TIP When set, the Direction Inhibit bit disables any physical input from affecting count direction.

When set, the Direction Invert bit changes the direction of the counter in all operational modes.

When Direction Inhibit is set, then Direction Invert *is* the direction.

Pulse/External Direction Mode Selection

In this mode, the B input controls the direction of the counter, as shown in Figure 2.2 on page 2-8. If the B input is low (0), the counter increments on the rising edges of input A. If the input B is high (1), the counter decrements on the rising edges of input A.

TIP

Two Output Control bits allow you to modify the operation of the B input from your control program or during configuration. The Direction Inhibit bit, when set (1), disables the operation of the B input.

The Direction Invert bit, when set (1), reverses the operation of the B input, but only if the Direction Inhibit bit is not set. If the Direction Inhibit bit is set, then the Direction Invert bit controls counter direction.

When the Direction Inhibit bit is set (1):

- and Direction Invert = 0, count direction is up (forward)
- and Direction Invert = 1, count direction is down (reversed)

Figure 2.2 Pulse/External Direction Mode (Direction Inhibit = 0, Direction Invert = 0)

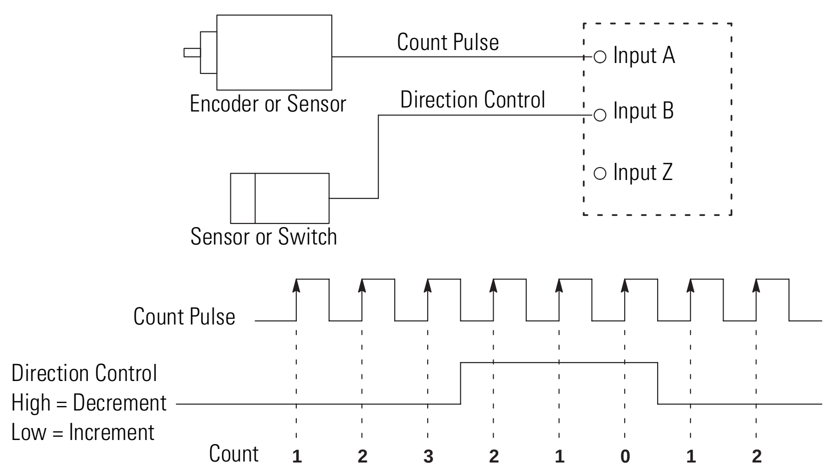


Table 2.3 Pulse External Direction Counting

Direction Inhibit Bit	Direction Invert Bit	Input A (Count)	Input B (Direction)	Change in Count Value
0	0	↑	0 or open	+1
		↑	1	-1
		0, 1, ↓	don’t care	0
0	1	↑	0 or open	-1
		↑	1	+1
		0, 1, ↓	don’t care	0
1	0	↑	0 or open	+1
		↑	1	+1
		0, 1, ↓	don’t care	0
1	1	↑	0 or open	-1
		↑	1	-1
		0, 1, ↓	don’t care	0

See Direction Inhibit and Direction Invert Output Control Bits page 2-7 for more information.

Pulse/Internal Direction Mode Selection

When the Pulse/Internal Direction mode is selected, the status of the Direction Invert bit, as controlled by the user program, determines the direction of the counter. The counter increments on the rising edge of the module’s A input when the Direction Invert bit is reset (0). The counter decrements on the rising edge of the A input when the Direction Invert bit is set (1).

Table 2.4 Pulse Internal Direction Counting - Counters 0 and 1

Direction Inhibit Bit	Direction Invert Bit	Input A (Count)	Input B	Change in Count Value
don’t care	0	↑	don’t care	+1
		0, 1, ↓	don’t care	0
don’t care	1	↑	don’t care	-1
		0, 1, ↓	don’t care	0

Table 2.5 Pulse Internal Direction Counting - Counters 2 and 3

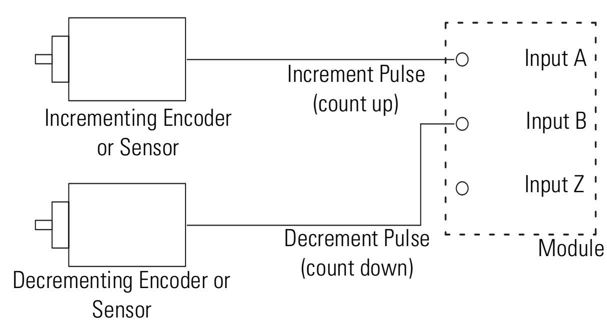
Direction Inhibit Bit	Direction Invert Bit	Input A	Input B (Count)	Change in Count Value
don't care	0	don't care	↑	+1
		don't care	0, 1, ↓	0
don't care	1	don't care	↑	-1
		don't care	0, 1, ↓	0

Up and Down Pulses Mode Selection

In this mode, the counter channel increments on the rising edge of pulses applied to input A and decrements on the rising edge of pulses applied to input B. When set, the Direction Inhibit bit causes both A and B to increment. When set, the Direction Invert bit causes B to increment and A to decrement. When the Direction Invert and Direction Inhibit bits are both set, both A and B decrement.

TIP When both inputs transition simultaneously or near simultaneously, the net result is no change to the count value.

Figure 2.3 Up and Down Pulse Mode (Direction Inhibit = 0, Direction Invert = 0)



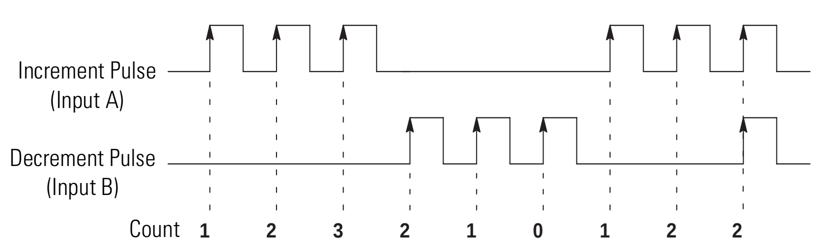


Table 2.6 Up and Down Counting

Direction Inhibit Bit	Direction Invert Bit	Input A (Count)	Input B (Direction)	Change in Count Value
0	0	↑	0, 1, ↓	+1
		0, 1, ↓	↑	-1
		↑	↑	0
0	1	↑	0, 1, ↓	-1
		0, 1, ↓	↑	+1
		↑	↑	0
1	0	↑	0, 1, ↓	+1
		0, 1, ↓	↑	+1
		↑	↑	0
1	1	↑	0, 1, ↓	-1
		0, 1, ↓	↑	-1
		↑	↑	0

X1 Quadrature Encoder Mode Selection

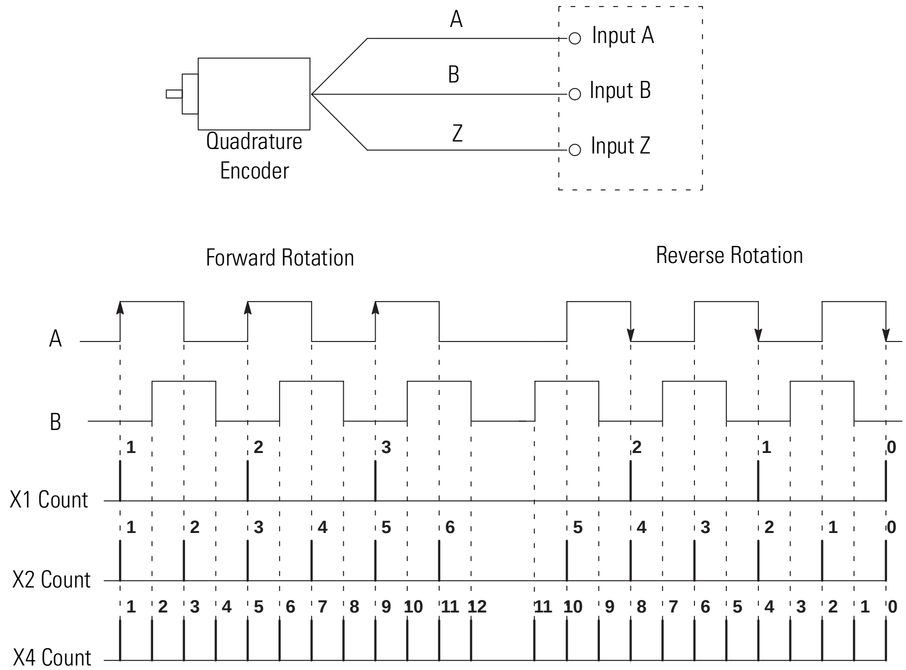
In this mode, when a quadrature encoder is attached to inputs A and B, the count direction is determined by the phase relation of inputs A and B. If A leads B, the counter increments. If B leads A, the counter decrements. In other words, when B is low, the count increments on the rising edge of input A and decrements on the falling edge of input A. If 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 2.4.

TIP

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

See also: Direction Inhibit and Direction Invert Output Control Bits on page 2-7 and their effect on Quadrature signals on page 2-13.

Figure 2.4 Quadrature Encoder Modes (Direction Inhibit = 0, Direction Invert = 0)



X2 Quadrature Encoder Mode Selection

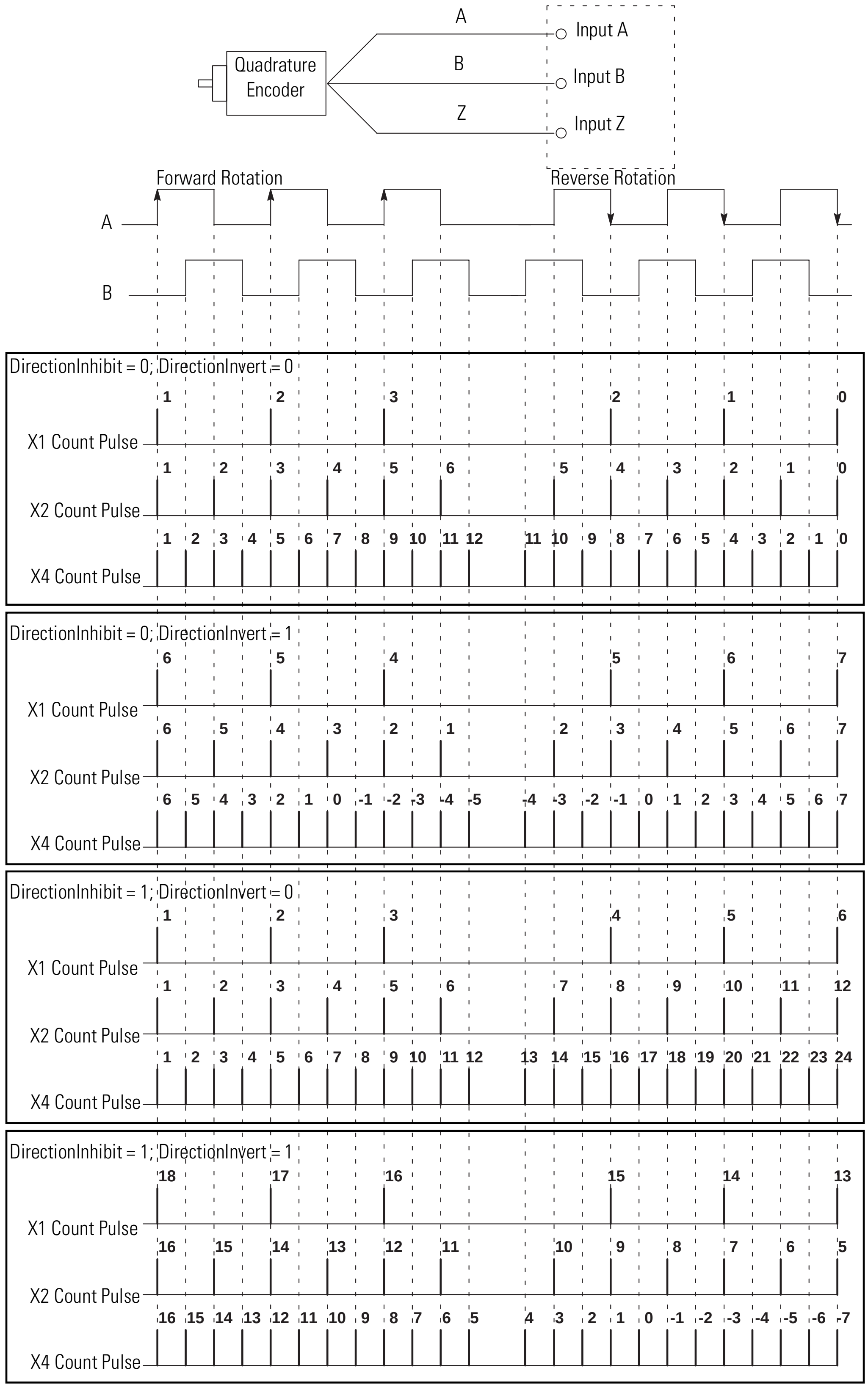
The X2 Quadrature Encoder mode operates much like the X1 Quadrature Encoder except that the resolution is doubled as shown in Figure 2.4 on page 2-12.

X4 Quadrature Encoder Mode Selection

The X4 Quadrature Encoder mode operates much like the X1 Quadrature Encoder except that the resolution is quadrupled as shown in Figure 2.4 on page 2-12.

The following diagram shows how Direction Inhibit and Direction Invert affect the counter.

Figure 2.5 Operation Using Various Direction Inhibit and Direction Invert Settings



Input Frequency

Maximum input frequency is determined by the input configuration as shown in the table below.

Input Configuration	Input Frequency
X4 Quadrature Encoder	250 kHz
X2 Quadrature Encoder	500 kHz
All Other Configurations	1 MHz

See Table 2.2 for additional details.

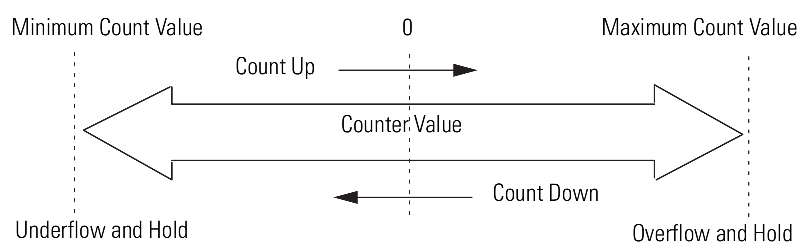
Counter Types

Each of the four possible counters can be configured to stop counting and set a flag at its limits (linear counter) or to roll over and set a flag at its limits (ring counter). A counter's limits are programmed by the Ctr*n*MaxCount and Ctr*n*MinCount words in the Module Configuration Array. Both types are described below.

Linear Counter

The figure below describes linear counter operation. In linear operation, the current count (Ctr[*n*].CurrentCount) value remains between, or equal to, the user-programmed minimum count (Ctr*n*MinCount) and maximum count (Ctr*n*MaxCount) values. If the Ctr[*n*].CurrentCount value would go above (>) or below (<) these values, the counter stops counting, and an overflow/underflow bit is set. The overflow/underflow bits can be reset using the Ctr*n*ResetCounterOverflow and Ctr*n*ResetCounterUnderflow bits.

Figure 2.6 Linear Counter Diagram

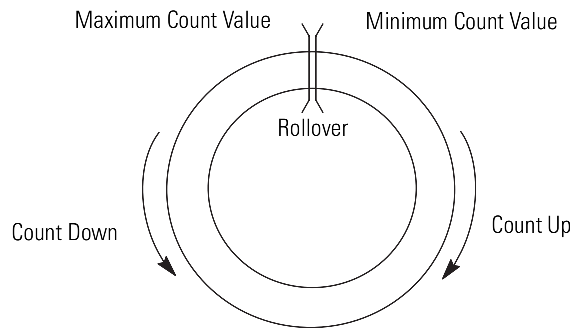


Pulses are not accumulated in an overflow/underflow state. The counter begins counting again when pulses are applied in the proper direction. For example, if you exceed the maximum by 1,000 counts, you do not need to apply 1,000 counts in the opposite direction before the counter begins counting down. The first pulse in the opposite direction decrements the counter.

Ring Counter

Figure 2.7 demonstrates ring counter operation. In ring counter operation, the current count (Ctr[*n*].CurrentCount) value changes between user-programmable minimum count (Ctr*n*MinCount) and maximum count (Ctr*n*MaxCount) values. If, when counting up, the counter reaches the Ctr*n*MaxCount value, it rolls over to the Ctr*n*MinCount value upon receiving the next count and sets the overflow bit. If, when counting down, the counter reaches the Ctr*n*MinCount value, it rolls under to the Ctr*n*MaxCount value upon receiving the next count and sets the underflow bit. These bits can be reset using the Ctr*n*ResetCounterOverflow and Ctr*n*ResetCounterUnderflow bits.

Figure 2.7 Ring Counter Diagram



Modifying Count Value

The count value (Ctr[*n*].CurrentCount) can be stored, reset, or preset using the Z input, CtrReset bit in the Configuration Array, control bits in the Output Array, or written over using a Direct Write command.

Table 2.7 Available Z Functions

Setting	For function
Store(1)	on rising edge of Z, store count in the Stored Count input word
Hold	while Z = 1, hold counter at its current value
Preset/Reset	on rising edge of Z, preset the count value to the value in the preset word

(1) If both a store and preset function are configured, the stored count is captured before the preset operation takes place.

IMPORTANT Because only the Z inputs are used for external gating and presetting, these functions are not available for Counters 2 and 3, which do not have Z inputs. All options are always available for Counters 0 and 1, regardless of input operational mode.

Counter Enable/Disable

The counter may be enabled or disabled using the Ctr*n*En control bit. Be aware that disabling the counter does not inhibit any current count loading functions (e.g. preset or direct write) or any Z function.

Z Input Functions

Store

The Z input can be used to capture the current count value even when the counter is counting at full 1 MHz speed.

Gate

The Z inputs can be used to gate (hold) the counter at its current value regardless of incoming A or B inputs. A gating function is typically one that allows pulses to reach the counter (gate open) or not (gate closed).

Z Preset

Preset can be programmed to occur based on the actions of the Z input signal.

Inhibit and Invert

The Z input signals may be inverted and/or inhibited, depending on the user configuration of the Ctr*n*ZInvert and Ctr*n*ZInhibit output control bits. If the signal is inhibited, the invert bit *is* the Z signal for the actions described above.

For an explanation of those bits, see Z Inv - Z Invert (CtrnZInvert) on page 4-25 and Z Inh - Z Inhibit (CtrnZInhibit) on page 4-25.

Direct Write

You can arbitrarily change the current count value (Ctr[*n*].CurrentCount) to the direct write control value (Range12To15[*n*].HiLimOrDirWr). This ability applies to ranges 12 through 15. The direct write value takes effect when the Load Direct Write bit (Range12To15[*n*].LoadDirectWrite) transitions from 0 to 1.

If you attempt to preset and load direct write to a counter at the same time, only the preset (Ctr*n*Preset) will take effect.

Preset/Reset

Preset sets the counter to a zero or non-zero value you define. Reset the counter by setting this value (Ctr*n*Preset) to zero.

Counter Reset

The CtrReset bit in the Configuration Array, when set, causes the following to occur when the system transitions to Run or the Inhibit Module bit transitions to 0:

- All counters are disabled and reset to zero.
- The Output Array is reset to default values until the ModConfig bit is set (1). The default value for the Output Array is all zeros.
- The Input Array counter Status Flags (Overflow, Underflow, RisingEdgeZ, RateValid, PresetWarning) are reset.
- The Input Array counter values (Current Count[(1)], StoredCount, CurrentRate and PulseInterval) are also reset to zero.
- All counts are lost and all outputs are turned off.

IMPORTANT For the most predictable results, you may want to clear the output image of the processor BEFORE performing a counter reset (CtrReset) to the 1769-HSC module.

This is because CtrReset does not change the processor's output image. CtrReset sets the 1769-HSC module's Output Array to all zero's. If any bit is set to 1 in the processor's output image, when sent to the module, it will be seen as a state transition and be acted upon.

Soft Preset

Preset can be programmed to occur by setting the appropriate output control bits via your control program. Setting the Ctr*n*SoftPreset bit in the Output Array causes the counter to be preset, changing the count to the value in Ctr*n*Preset.

(1) If zero is outside the MinCount and MaxCount limits set in the Configuration Array, then the Preset value is loaded into CurrentCount instead of zero. This also causes the PresetWarning bit to be set, which, in turn, sets the GenError bit.

Z Preset

Preset can be programmed to occur based on the actions of the Z input signal.

Autopreset

If the module is configured such that Ctr*n*MaxCount < Ctr[*n*].CurrentCount or Ctr*n*MinCount > Ctr[*n*].CurrentCount, then the module will automatically change Ctr[*n*].CurrentCount to the Ctr*n*Preset value and set the Ctr*n*PresetWarning bit.

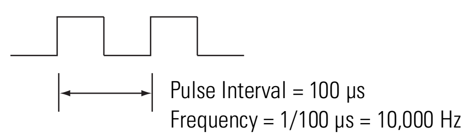
Rate/Timer Functionality

To ensure maximum accuracy, the module offers two different methods to calculate the rate:

- Per Pulse = 1/Pulse Interval
- Cyclic = Number of Pulses/User-Defined Time Interval

You select the method used, depending upon the pulse speed as defined below. These are continuously available regardless of input operational mode.

Pulse Interval Rate Calculation Method



The pulse interval rate method is very accurate for slower rates, i.e. when the pulse interval (or time between pulses) is large compared to the system clock timer (1 µs). A timer is used to measure the time between two successive pulses. The inverse of this value is the pulse interval rate. The pulse interval rate cannot be read directly from the module. It needs to be calculated. The calculation can be performed in the user control program.

This method is not as accurate for higher pulse rates. When the pulse interval shrinks, two factors can distort the per pulse calculation. If the pulse interval is close to the measuring timer's clock frequency, 1 MHz, the granularity of the time increments has a greater effect on rate inaccuracy. In addition, the rate may be calculated many times over the course of a single backplane scan. As a result, the rate data obtained at a backplane scan is only that of the very last pair of pulses and disregards the other rate calculations that may have occurred during that interval. This can result in rate inaccuracy if the pulses are unevenly spaced.

Cyclic Rate Calculation Method (Current Rate)

The module continuously calculates rates for each of its four possible counters, regardless of operational mode (e.g. up/down count). The 32-bit signed integer rate from each counter is reported in the Ctr[*n*].CurrentRate words of the Input Array.

In this method, the rates are calculated at the end of a counter's configured cycle time. This is configured via the Ctr*n*CyclicRateUpdateTime configuration word/menu. Valid entries are +1 to 32767 milliseconds. The number of net counts, net change in Ctr[*n*].CurrentCount, during that period is converted into a rate value, providing an average pulse rate.

The generalized rate calculation is: Rate = Δ count/ Δ time.

IMPORTANT	The rate calculation is based on net counts. If a counter goes up 500 counts and down 300 counts, the net count is +200. Therefore, changes in direction and speed affect the Ctr[*n*].CurrentRate value.

The cyclic method is better suited to high pulse rates.

Hysteresis Detection and Configuration

Because physical vibration can cause an encoder to generate pulses which you may not wish to consider as valid motion, a hysteresis value is used to eliminate a certain number of pulses in either direction as vibration-generated. These pulses are not used to calculate the Ctr[*n*].CurrentRate value. You program the minimum number of counts that are considered to be valid motion, using the Ctr*n*Hysteresis configuration word/menu. If the change in counts over the update time cycle is less than that minimum number of programmed counts, the Ctr[*n*].CurrentRate is reported as zero.

NOTE: This concept is not used to alter actual count values.

IMPORTANT	Hysteresis does not depend on the direction(s) of the change in count. Therefore, creeping, a slow change in count in one direction only, can also be reported as zero frequency when it falls below the hysteresis threshold.

Scalar

You can configure the Ctr*n*Scalar value to scale or convert the raw rate value to application-specific information, such as RPM (Revolutions Per Minute). Setting Ctr*n*Scalar to 1 leaves the rate value in cycles per second (Hertz).

The actual rate equation is:

$$\text{Current Rate} = \frac{1000 \times \Delta \text{ count}}{\text{CyclicRateUpdateTime} \times \text{Scalar}}$$

TIP	To configure the Ctr[*n*].CurrentRate value to show an RPM value, set Ctr*n*Scalar to (counts per revolution)/60.

For example, where Ctr0CyclicRateUpdateTime = 80, the encoder has 360 counts per revolution, and the change in Ctr[0].CurrentCount is 96:

$$\text{Scalar} = \frac{\text{360 counts/revolution}}{\text{60 sec/min}}$$

$$\text{RPM} = \frac{\text{1000 Cyclic Rate Update Time/sec x 96 counts}}{\text{80 Cyclic Rate Update Time x } \dfrac{\text{360 counts/revolution}}{\text{60 sec/min}}} = \text{200 RPM}$$

Rate Valid

The Ctr[*n*].RateValid bit indicates calculation integrity. When the bit is set, it indicates that the accompanying Ctr[n].CurrentRate value is accurate. The Ctr[*n*].RateValid bit is reset when the overflow or underflow events have occurred, i.e. at rising edges of Ctr[*n*].Overflow or Ctr[*n*].Underflow bits. It also happens when the count is abruptly modified via a preset (Ctr*n*SoftPreset, Ctr*n*CtrPresetWarning or Z based preset event) or direct write (Range12To15[*n*].LoadDirectWrite). When this occurs, the Ctr[*n*].CurrentRate value is frozen at the last known good value so that effects of erroneous rates will not propagate to range comparisons. The value remains frozen until the current cycle time plus one more cycle time are elapsed (this may be up to twice the Ctr*n*CyclicRateUpdateTime). If the overflow/underflow occurrence lasts for more than one cycle time, the value is frozen that entire time plus up to two more cycle times.

Ensure that another overflow/underflow, etc. does not happen during this recovery time. The rate will remain invalid until a full update time has occurred with no such events. If the Ctr[*n*].RateValid bit is seldom or never set, the Ctr*n*MinCount and Ctr*n*MaxCount values may be configured too close to each other.

Rate Method Selection

By knowing when to use each method, an optimal rate determination can be made.

TIP

Fractional rates are not reported by the module, but can be calculated from Ctr[*n*].PulseInterval in your control program.

The following information is provided to assist you in choosing the appropriate calculation method. In general you should consider the effect of having the count off by ±1 in each method at frequencies of interest to see if the resulting inaccuracy is acceptable.

Per Pulse Method Example

If the frequency of interest has 100 counts (of the 1 µs clock) between pulses, an error of 1 count results in a 1-in-100, or 1%, error. If there are 1000 counts between pulses, then the error is 1-in-1000, or 0.1%. Error for a variety of pulse values is shown below.

Table 2.8 Per Pulse Errors

Actual 1 µs Internal Pulses(1)	Reported Pulses	Real Frequency	Reported Frequency	% Error
2	1	500 kHz	1 MHz	100%
9	10	111 kHz	100 kHz	11.1%
101	100	9.901 kHz	10.000 kHz	1.00%
1001	1000	999 Hz	1000 Hz	0.10%
9,999	10,000	100.01 Hz	100.00 Hz	0.010%
99,999	100,000	10.00010 Hz	10.00000 Hz	0.001%

(1) 1.9999 can be rounded to 2 and so on.

Cyclic Method

Since the update time is programmable, there is more flexibility in choosing the correct fit when using the Cyclic Method.

Error estimates are shown below for a variety of update times.

Table 2.9 Maximum Cyclic Rate Errors

CyclicRate Update Time x Scalar	Frequency				
	100 Hz	1 kHz	10 kHz	100 kHz	1 MHz
1	n/a	n/a	20.02%	2.011%	0.210%
10	n/a	20.11%	2.020%	0.210%	0.030%
100	20.01%	2.110%	0.220%	0.031%	0.012%
1000	3.010%	0.310%	0.040%	0.013%	0.010%
10,000	1.210%	0.130%	0.022%	0.011%	0.010%

See also: Rate Accuracy graph on page A-3.

Output Control

All 16 outputs can be controlled by any of the 4 counters or by the user's control program, via the output mask function. Output states are determined by count, rate, ranges, mask configuration data, overcurrent status, and safe state settings and conditions.

The 16 outputs are made up of 4 real (physical) outputs and 12 virtual outputs. The status of the real and virtual outputs is available to the user program. The real outputs are electronically protected from overloads.

IMPORTANT To turn outputs on, you must use both the Output On Mask and the Output Off Mask.

Masks

Output On Mask

Using the Output On Mask, all of the module's outputs can be turned on directly by the user control program, like discrete outputs. A bit which is set in the mask turns on the corresponding real or virtual output.

Output Off Mask

The Output Off Mask has veto power over any output. It can turn any or all of the module's outputs off. When a bit in this mask is set to 0, the output will be turned off. Each bit is logically ANDed with the Output On Mask and masks of active and enabled ranges. If the bit in this mask is set to 1, the output may be turned on or off by the ranges, or the Output On Mask. The final result is available as the Readback.*n* bit.

Ranges

Up to 16 dynamically configurable ranges are available. Ranges activate outputs based on the current count value or the current rate value. Each range is programmed with a type, counter number, two limit values, an invert bit, and an output mask.

Each range is programmed with high and low limits for the chosen value. The range's invert bit indicates whether the range is active between or outside the range limits. When the chosen value fulfills the configuration parameters, the range is active as indicated in the Input Array. When a range is active and enabled (RangeEn.*n* = 1), the range turns on all outputs indicated by the Range Output Mask except those that are prevented from being enabled by the other factors such as Output Off Mask or Overcurrent. The status of a range is provided by the range active status word, where 1 equals range active and zero equals inactive.

TIP

Ranges can be disabled while the module is running using the RangeEn.*n* bit in the output file. However, even a disabled range will report when it is active or not. For example, an unprogrammed range has limits of 0, and points to the Ctr[0].CurrentCount value. If this value is 0, that range is reported as active.

Count Range

In a non-inverted count range, the outputs are active if the count value is within the user-defined range. In an inverted count range, the outputs are active if the count value is outside the user-defined range. Valid limits for the range are -2 billion and +2 billion regardless of programmed minimum and maximum values.

The example shows all ranges referring to one counter. The module is capable of individually assigning each range to any counter. Each counter can also have a combination of count and rate ranges.

Figure 2.8 Count Range Example

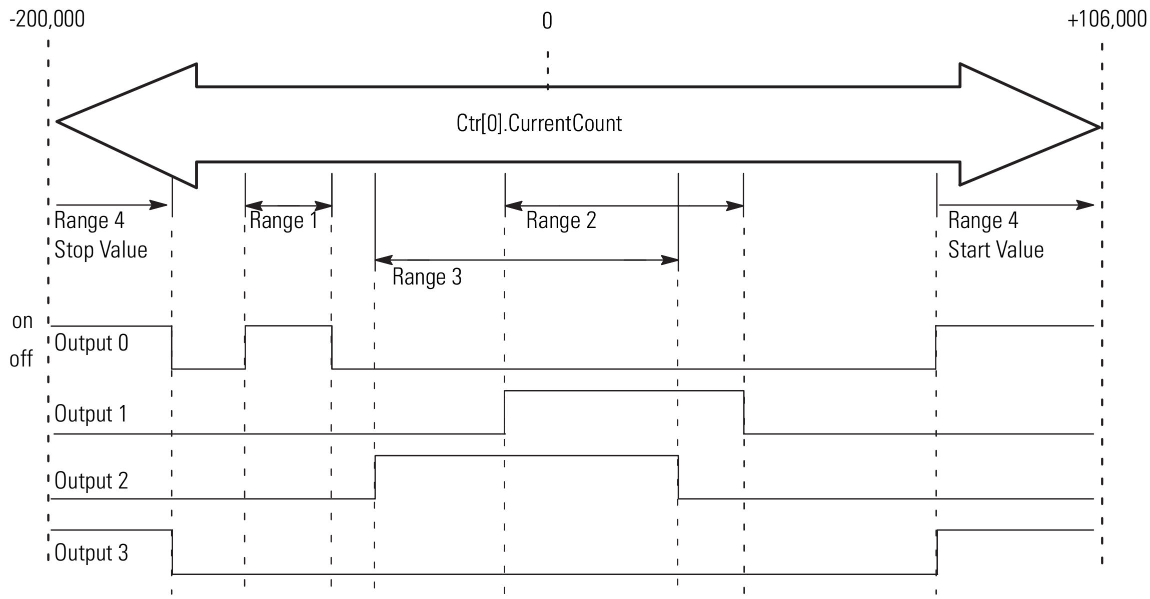


Table 2.10 Count Range Example Values

Range	Range Counter Number	Range Type(1)	Range Low Limit	Range High Limit	Range Invert Bit	Outputs(2) (Range[*n*].OutputControl word)																Outputs Affected
						15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	01	0	-7000	-5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	01	0	-1000	+4500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
3	01	0	-4000	+3000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
4	01	0	-9000	+9000	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0 and 3

(1) For Range Type, 0 = count range and 1 = rate range.

(2) Bits 0 through 3 are real outputs. Bits 4 through 15 are virtual outputs.

Rate Range

In a non-inverted rate range, the outputs are active if the rate measurement is within the user-defined range. In an inverted rate range, the outputs are active if the rate measurement is outside the user-defined range. The input rate can be up to 1 MHz in either direction.

The example shows all ranges referring to one counter. The module is capable of individually assigning each range to any counter. Each counter can also have a combination of count and rate ranges.

Figure 2.9 Rate Range Example

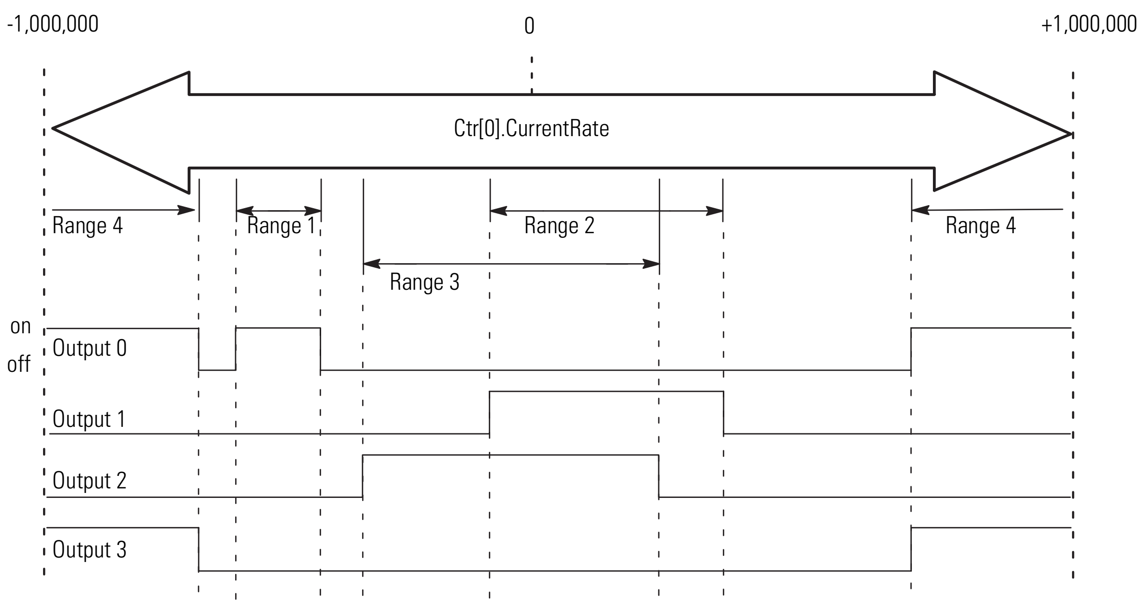


Table 2.11 Count Range Example Values

Range	Range Counter Number	Range Type(1)	Range Low Limit	Range High Limit	Range Invert Bit	Outputs(2) (Range[*n*].OutputControl word) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Outputs Affected
1	00	1	-7000	-5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	00	1	-1000	+4500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
3	00	1	-4000	+3000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
4	00	1	-20000	+20000	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0 and 3

(1) For Range Type, 0 = count range and 1 = rate range.

(2) Bits 0 through 3 are real outputs. Bits 4 through 15 are virtual outputs.

Overcurrent

If the module detects a real output point overcurrent condition, it reports it to the input file and turns off that output. You can also program the module to latch each of the four real outputs off, emulating a physical fuse, or to automatically reset. The 12 virtual outputs do not have this function.

When the OvercurrentLatchOff bit is set and an overcurrent situation occurs, even momentarily, the associated real output is latched off until the ResetBlownFuse bit transitions from 0 to 1.

If the OvercurrentLatchOff bit is reset and an overcurrent situation occurs, the output turns off for 1 second and is then retried (auto-reset). The module continues to attempt to turn the output back on until the overcurrent situation is no longer detected and the output is successfully turned back on.

IMPORTANT The outputs will be on momentarily while they are retried. The length of time they are on depends on the magnitude of the load.

Safe State Control

The 1769-HSC module combines the Hold Last State and User-Defined Safe State options with a safe state run alternative that allows the module to continue to control outputs under program or fault states[1]. These options are described below.

Only the physical outputs are affected by safe state settings and conditions. Virtual outputs, inputs, and counting are not affected by program or fault states.

Hold Last State (HLS)

This condition applies depending on the mode of the controller. When the hold last state option is set, the module holds the outputs at the state they were at just before the control system transitioned from Run to Program or Run to Fault.

HLS sets the module according to the values configured for Program Mode (described on page 4-9) and Output Fault Mode (described on page 4-10).

(1) The module continues to update the Input Array and count inputs in all modes. The operation of the outputs will vary according to mode and configuration and the capabilities of the controller or bus master.

User-Defined Safe State (UDSS)

In this configuration, the module sets the outputs to a user-defined safe state when the control system transitions from Run to Program or Run to Fault.

UDSS sets the module according to the values configured for Output Program Value (described on page 4-10) and Output Fault Value (described on page 4-11).

Program State Run (PSR)

Program State Run allows you to specify that the output should continue to be controlled by the module as if it were in the Run state. That is, events on the module or changes in the Output image will affect the physical outputs without regard to the Program_HLS or UDSS state indicated. When this bit is set, the corresponding Out*n*ProgramMode and Out*n*ProgramValue bits are ignored.

PSR sets the module according to the value configured for Output Program State Run (described on page 4-9).

ATTENTION

Selecting this option will allow outputs to change state while ladder logic is not running. You must take care to assure that this does not pose a risk of injury or equipment damage when selecting this option.

IMPORTANT

The prescan initiated by some controllers could have an effect on the outputs. To overcome any changes in physical output states that may be caused by this, retentive output instructions (eg. latch, unlatch etc.) should be used when bit manipulations are done on the Output image of this module in ladder logic.

This applies to a wide range of bits when Program State Run is selected, since presetting a counter, enabling a range, changing a mask, and changing Module Configuration Array settings can cause ranges and outputs to change state.

Fault State Run (FSR)

Similar to Program State Run, Fault State Run allows you to specify, on a bit basis, that the output should continue to be controlled by the module as if it were Run state. That is, events on the module or changes in the Output image will affect the physical outputs without regard to the Program_HLS or UDSS state indicated. When this bit is set, the corresponding Program Mode and Program Value bits are ignored.

FSR sets the module according to the value configured for Output Fault State Run (described on page 4-10).

ATTENTION

Selecting this option will allow outputs to change state while ladder logic is not running. You must take care to assure that this does not pose a risk of injury or equipment damage when selecting this option.

IMPORTANT

The prescan initiated by some controllers could have an effect on the outputs. To overcome any changes in physical output states that may be caused by this, retentive output instructions (eg. latch, unlatch etc.) should be used when bit manipulations are done on the Output image of this module in ladder logic.

This applies to a wide range of bits when Fault State Run is selected, since presetting a counter, enabling a range, changing a mask, and changing Configuration Array settings can cause ranges and outputs to change state.

Program to Fault Enable (PFE)

The ProgToFaultEn bit allows you to select which data value (Program Value or Fault Value) to apply to the output when the Output State Logic state Prog_HLS changes to indicate Fault_HLS.

If PFE is 0, the module leaves the Program value applied. If PFE is set to 1, the Fault value is applied.

TIP

If the module is in a safe state such as Program or Fault which is configured to turn an output ON and excessive current is drawn from the output, the output will still turn off according to the programmed OverCurrentLatchOff bit configuration.

The module’ s Default Safe State configuration is all zero’s, resulting in the following:

- Program State = UDSS
- Program Value = OFF
- Program State Run = No
- Fault State = UDSS
- Fault Value = OFF
- Fault State Run = No
- PFE = leave program value applied.

Output Control Example

The following example illustrates the module’s output control flow. The following conditions are reflected in Table 2.12:

- Range 0 is enabled and active
- Range 1 is disabled
- Range 2 is enabled but not active
- an overcurrent condition exists on real output 3
- OvercurrentLatchOff is set
- the system is in Run mode

The table below illustrates the step-by-step logical operations that are performed to determine the final output state. For example, Range 1 values do not affect the output because Range 1 is disabled, and the Output Off Mask causes some of the outputs to change to zero because it takes priority over the range masks.

The output parameters shown in the table have been discussed in the previous sections.

Table 2.12 Output Control Example

Output Parameter	Mask Information				Logical Operation	Result(1)			
Range 0	0 0 0 1	0 1 1 0	1 1 0 1	0 0 0 1	OR	0 0 0 1	0 1 1 0	1 1 0 1	0 0 0 1
Range 1	0 0 1 0	1 1 1 1	1 1 1 1	0 0 1 0	OR	0 0 0 1	0 1 1 0	1 1 0 1	0 0 0 1
Range 2	0 1 0 0	0 0 0 0	0 0 0 0	1 1 0 0	OR	0 0 0 1	0 1 1 0	1 1 0 1	0 0 0 1
Output On Mask	0 1 0 0	1 0 1 0	1 0 1 0	1 0 0 0	OR	0 **1** 0 1	**1** 1 1 0	1 1 **1** 1	**1** 0 0 1
Output Overcurrent	- - - -	- - - -	- - - -	1 0 0 0	AND	0 1 0 1	1 1 1 0	1 1 1 1	**0** 0 0 1
Output Off Mask	1 1 1 1	0 0 0 0	1 1 1 1	1 1 0 0	AND	0 1 0 1	**0 0 0** 0	1 1 1 1	0 0 0 **0**
Program State Values	- - - -	- - - -	- - - -	1 1 1 1	Override	0 1 0 1	0 0 0 0	1 1 1 1	0 0 0 0
Fault State Values	- - - -	- - - -	- - - -	1 1 1 1	Override	0 1 0 1	0 0 0 0	1 1 1 1	0 0 0 0
Final Output State						0 1 0 1	0 0 0 0	1 1 1 1	0 0 0 0

(1) Bolded text indicates that these values have changed.

Readback/Loopback

The Readback/loopback function is the feedback of the module's outputs via its Input Array. This 16-bit image includes both real (4) and virtual (12) outputs.

If the module's output is OFF due to overcurrent, both the Overcurrent status flag and the Readback bit will indicate the condition being 1 and 0 respectively. Conversely, should the output be ON due to any module control (eg. UDSS), this will be indicated by Readback.

Installation and Wiring

This chapter tells you how to:

- determine the power requirements for the modules
- avoid electrostatic damage
- install the module
- wire the module's terminal block
- wire input devices

Compliance to European Union Directives

This product is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

The 1769-HSC module is tested to meet Council Directive 89/336/EEC Electromagnetic Compatibility (EMC) and the following standards, in whole or in part, documented in a technical construction file:

- EN 50081-2
  EMC – Generic Emission Standard, Part 2 - Industrial Environment

- EN 50082-2
  EMC – Generic Immunity Standard, Part 2 - Industrial Environment

This product is intended for use in an industrial environment.

Low Voltage Directive

This product is tested to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 – Equipment Requirements and Tests.

For specific information required by EN61131-2, see the appropriate sections in this publication, as well as the following Allen-Bradley publications:

- *Industrial Automation, Wiring and Grounding Guidelines for Noise Immunity*, publication 1770-4.1
- *Automation Systems Catalog*, publication B113

Power Requirements

The modules receive power through the Compact bus interface from the +5V dc/+24V dc system power supply. The maximum current drawn by the modules is shown in the table below.

Module Current Draw	at 5V dc	at 24V dc
	425 mA	0 mA

General Considerations

Compact I/O is suitable for use in an industrial environment when installed in accordance with these instructions. Specifically, this equipment is intended for use in clean, dry environments (Pollution degree 2[(1)]) and to circuits not exceeding Over Voltage Category II[(2)] (IEC 60664-1).[(3)]

(1) Pollution Degree 2 is an environment where, normally, only non-conductive pollution occurs except that occasionally a temporary conductivity caused by condensation shall be expected.

(2) Over Voltage Category II is the load level section of the electrical distribution system. At this level transient voltages are controlled and do not exceed the impulse voltage capability of the product's insulation.

(3) Pollution Degree 2 and Over Voltage Category II are International Electrotechnical Commission (IEC) designations.

Hazardous Location Considerations

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only. The following WARNING statement applies to use in hazardous locations.

WARNING

EXPLOSION HAZARD

- Substitution of components may impair suitability for Class I, Division 2.
- Do not replace components or disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
- Do not connect or disconnect components unless power has been switched off or the area is known to be non-hazardous.
- This product must be installed in an enclosure. All cables connected to the product must remain in the enclosure or be protected by conduit or other means.
- All wiring must comply with N.E.C. article 501-4(b).

Prevent Electrostatic Discharge

ATTENTION

Electrostatic discharge can damage integrated circuits or semiconductors if you touch the bus connector pins, terminal block, or devices on the circuit board. Follow these guidelines when you handle the module:

- Touch a grounded object to discharge static potential.
- Wear an approved wrist-strap grounding device.
- Do not touch the bus connector or connector pins.
- Do not touch circuit components inside the module.
- If available, use a static-safe work station.
- When it is not in use, keep the module in its static-shield box.

Remove Power

ATTENTION

Remove power before removing or inserting this module. When you remove or insert a module with power applied, an electrical arc may occur. An electrical arc can cause personal injury or property damage by:

- sending an erroneous signal to your system's field devices, causing unintended machine motion
- causing an explosion in a hazardous environment

Electrical arcing causes excessive wear to contacts on both the module and its mating connector and may lead to premature failure.

Selecting a Location

Reducing Noise

Most applications require installation in an industrial enclosure to reduce the effects of electrical interference. The module is highly susceptible to electrical noise. Electrical noise coupled to the inputs will reduce the performance (accuracy) of the module.

Group your modules to minimize adverse effects from radiated electrical noise and heat. Consider the following conditions when selecting a location for the module. Position the module:

- away from sources of electrical noise such as hard-contact switches, relays, and AC motor drives
- away from modules which generate significant radiated heat, such as the 1769-IA16. Refer to the module's heat dissipation specification.

In addition, route shielded, twisted-pair analog input and output wiring away from any high voltage I/O wiring.

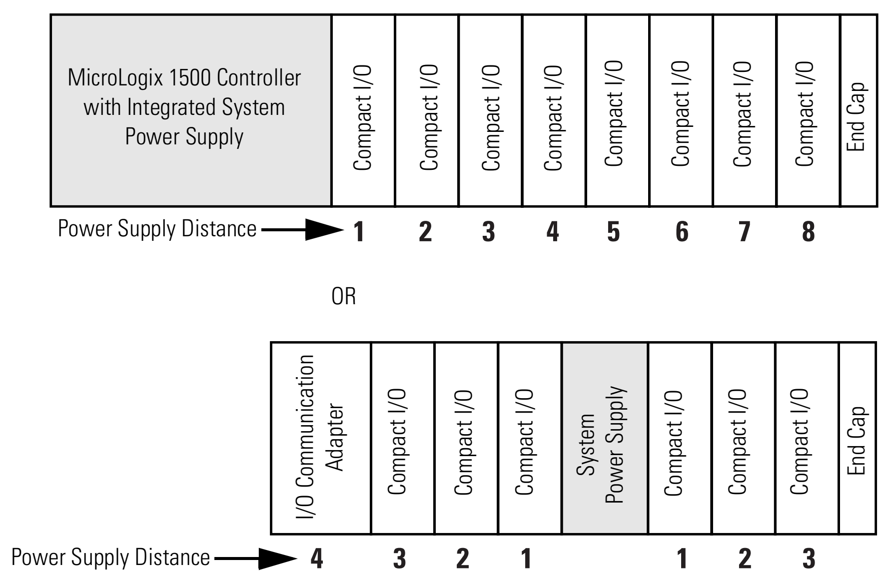
Protecting the Circuit Board from Contamination

The printed circuit boards of the modules must be protected from dirt, oil, moisture, and other airborne contaminants. To protect these boards, we recommend installing the system in an enclosure suitable for the environment. The interior of the enclosure should be kept clean and the enclosure door should be kept closed whenever possible.

Power Supply Distance

You can install as many modules as your power supply can support. However, the module has a power supply distance rating of 4, which means that it may not be located more than 4 modules away from the system power supply.

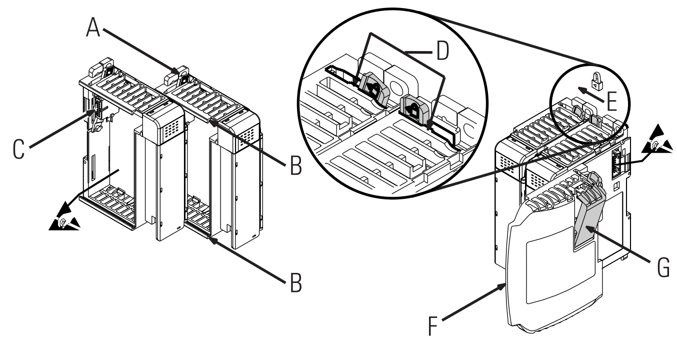
The illustration below provides an example showing how power supply distance is determined.



System Assembly

The module can be attached to an adjacent controller, power supply, or I/O module. For mounting instructions, see Panel Mounting on page 3-7, or DIN Rail Mounting on page 3-9. To work with a system that is already mounted, see Replacing the Module within a System on page 3-10.

The following procedure shows you how to assemble the Compact I/O system.



1. Disconnect power.

2. Check that the bus lever of the module (A) is in the unlocked (fully right) position.

3. Use the upper and lower tongue-and-groove slots (B) to secure the modules together.

4. Move the module back along the tongue-and-groove slots until the bus connectors (C) line up with each other.

5. Use your fingers or a small screw driver to push the bus lever back slightly to clear the positioning tab (D).

6. Move the module's bus lever fully to the left (E) until it clicks. Ensure it is locked firmly in place.

ATTENTION When attaching I/O modules, it is very important that the bus connectors are securely locked together to ensure proper electrical connection.

7. Attach an end cap terminator (F) to the last module in the system by using the tongue-and-groove slots as before.

8. Lock the end cap bus terminator (G).

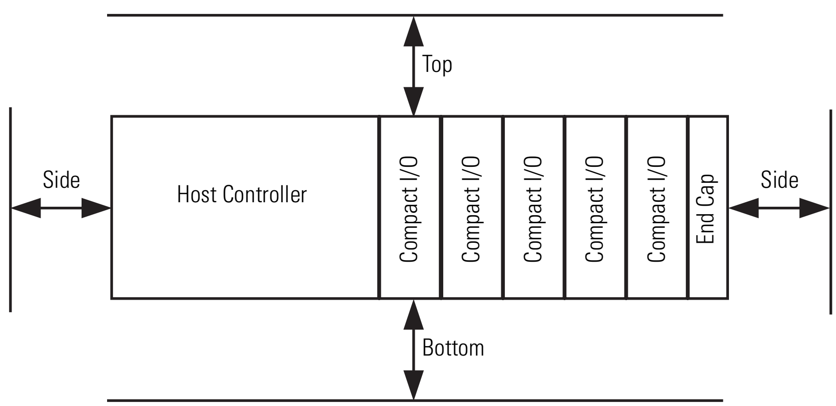
IMPORTANT	A 1769-ECR or 1769-ECL right or left end cap must be used to terminate the end of the serial communication bus.

Mounting

ATTENTION	During panel or DIN rail mounting of all devices, be sure that all debris (metal chips, wire strands, etc.) is kept from falling into the module. Debris that falls into the module could cause damage at power up.

Minimum Spacing

Maintain spacing from enclosure walls, wireways, adjacent equipment, etc. Allow 50 mm (2 in.) of space on all sides for adequate ventilation, as shown below:



Panel Mounting

Mount the module to a panel using two screws per module. Use M4 or #8 panhead screws. Mounting screws are required on every module.

Panel Mounting Using the Dimensional Drawing

NOTE: All dimensions are in mm (inches). Hole spacing tolerance: ±0.04 mm (0.016 in.).

Figure 3.1 Compact I/O with CompactLogix Controller and Power Supply

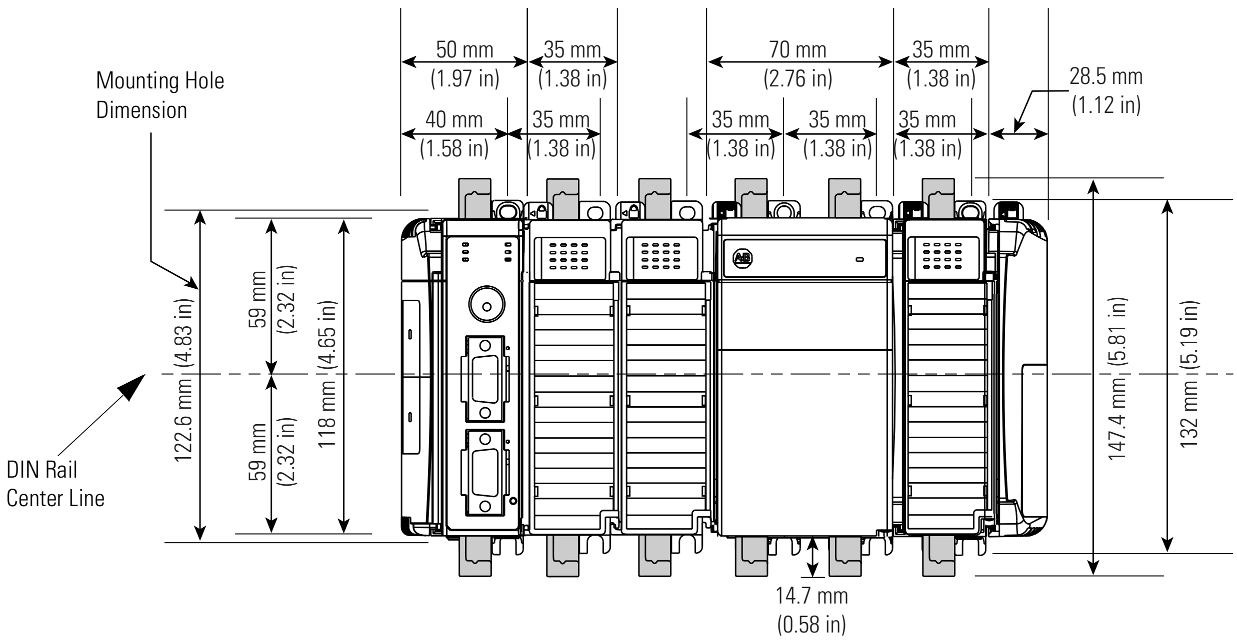
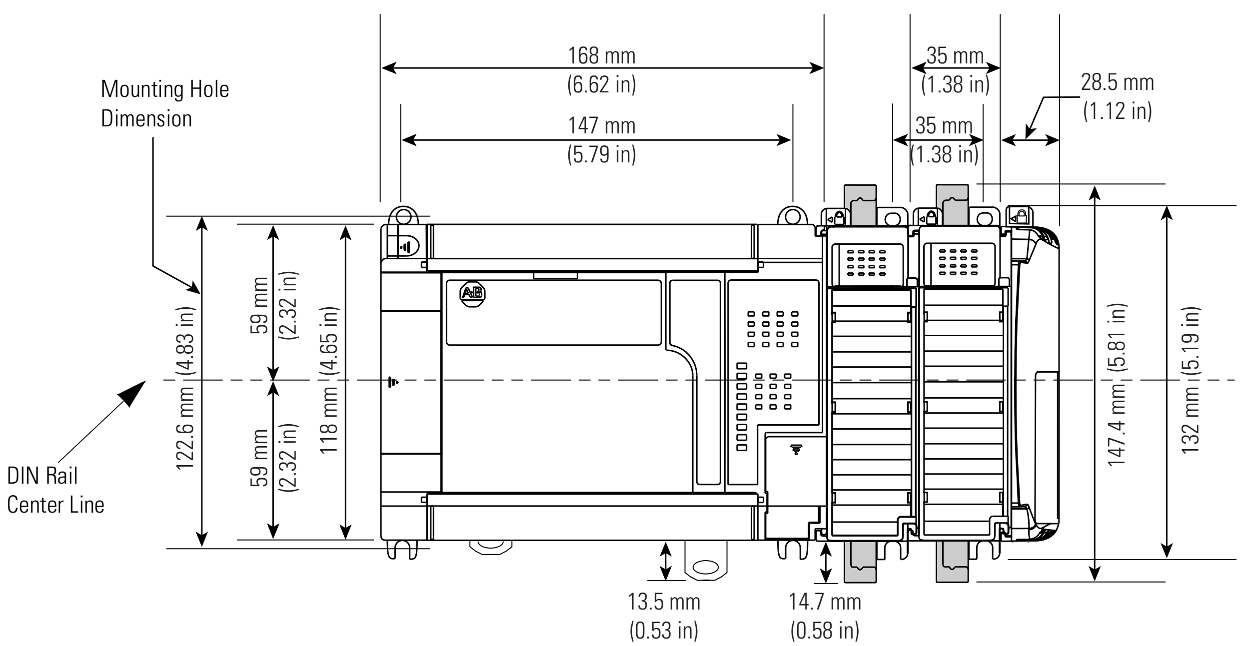


Figure 3.2 Compact I/O with MicroLogix 1500 Base Unit and Processor



Panel Mounting Procedure Using Modules as a Template

The following procedure allows you to use the assembled modules as a template for drilling holes in the panel. Due to module mounting hole tolerance, it is important to follow these procedures:

1. On a clean work surface, assemble no more than three modules.
2. Using the assembled modules as a template, carefully mark the center of all module-mounting holes on the panel.
3. Return the assembled modules to the clean work surface, including any previously mounted modules.
4. Drill and tap the mounting holes for the recommended M4 or #8 screw.
5. Place the modules back on the panel, and check for proper hole alignment.
6. Attach the modules to the panel using the mounting screws.

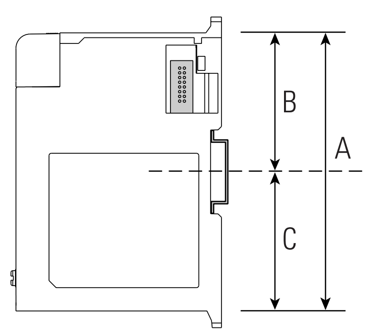
TIP If mounting more modules, mount only the last one of this group and put the others aside. This reduces remounting time during drilling and tapping of the next group.

7. Repeat steps 1 to 6 for any remaining modules.

DIN Rail Mounting

The module can be mounted using the following DIN rails:
35 x 7.5 mm (EN 50 022 - 35 x 7.5) or 35 x 15 mm (EN 50 022 - 35 x 15).

Before mounting the module on a DIN rail, close the DIN rail latches. Press the DIN rail mounting area of the module against the DIN rail. The latches will momentarily open and lock into place. DIN rail mounting dimensions are shown below.



Dimension	Height
A	118 mm (4.65 in.)
B	59 mm (2.325 in.)
C	59 mm (2.325 in.)

Replacing the Module within a System

The module can be replaced while the system is mounted to a panel (or DIN rail).

1. Remove power. See important note on page 3-4.

2. Remove terminal block or disconnect input/output wiring from the module.

3. Remove the upper and lower mounting screws from the module (or open the DIN latches using a flat-blade screwdriver).

4. On the module to be replaced and the right-side adjacent module (or end cap if the module is the last module in the bank), move the bus levers to the right (unlock) to disconnect the module from the adjacent modules.

5. Gently slide the disconnected module forward.

   If you feel excessive resistance, make sure that you disconnected the module from the bus and that you removed both mounting screws (or opened the DIN latches).

.

TIP It may be necessary to rock the module slightly from front to back to remove it, or, in a panel-mounted system, to loosen the screws of adjacent modules.

6. Before installing the replacement module, be sure that the bus lever on the right-side adjacent module is in the unlocked (fully right) position.

7. Slide the replacement module into the open slot.

8. Connect the modules together by locking (fully left) the bus levers on the replacement module and the right-side adjacent module or end cap.

9. Replace the mounting screws (or snap the module onto the DIN rail).

10. Replace the terminal block or connect input/output wiring to the module.

Field Wiring Connections

System Wiring Guidelines

Consider the following when wiring your system:

General

- Make sure the system is properly grounded.
- Input and output channels are isolated from the 1769 Compact bus. Input channels are isolated from one another; output channels are not.
- Shielded cable is required for high-speed input signals A, B, and Z. Use individually shielded, twisted-pair cable for lengths up to 300 meters (1000 feet).
- Group this module and other low voltage DC modules away from AC I/O or high voltage DC modules.
- Route field wiring away from any other wiring and as far as possible from sources of electrical noise, such as motors, transformers, contactors, and ac devices.
- Routing field wiring in a grounded conduit can reduce electrical noise.
- If field wiring must cross ac or power cables, ensure that they cross at right angles.

Terminal Block

- To ensures optimum accuracy, limit overall cable impedance by keeping cable as short as possible. Locate the module as close to input devices as the application permits.
- Tighten terminal screws with care. Excessive tightening can strip a screw.

Grounding

- This product is intended to be mounted to a well-grounded mounting surface such as a metal panel. Additional grounding connections from the module's mounting tabs or DIN rail (if used) are only required when the mounting surface is non-conductive and cannot be grounded.
- Keep shield connection to ground as short as possible.
- Ground the shield drain wire at the 1769-HSC input end only.
- Refer to *Industrial Automation Wiring and Grounding Guidelines*, Allen-Bradley publication 1770-4.1, for additional information.

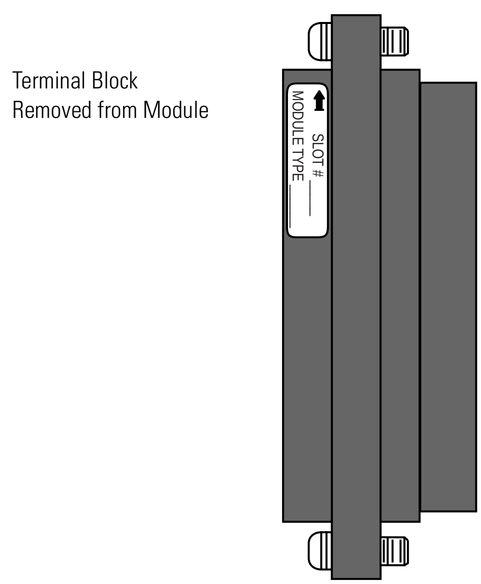
Considerations for Reducing Noise

In high noise environments, the 1769-HSC inputs may accept “false” pulses, particularly when using low frequency input signals with slowly sloping pulse edges. To minimize the effects of high frequency noise on low frequency signals, perform the following:

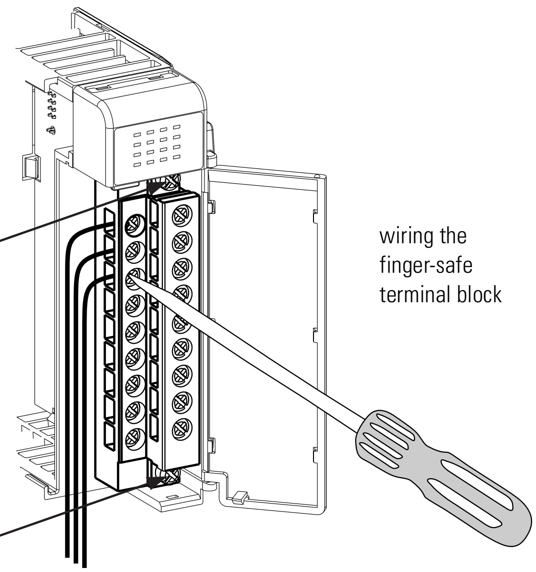
- Identify and remove noise sources.
- Route input cabling away from noise sources.
- Use your programming software to select low-pass filters on input signals. Filter values depend on the application and can be determined empirically.
- Use devices which output differential signals, such as differential encoders, to minimize the possibility that a noise source will cause a false input.

Removing and Replacing the Terminal Block

When wiring the module, you do not have to remove the terminal block. If you remove the terminal block, use the write-on label located on the side of the terminal block to identify the module location and type.



To remove the terminal block, loosen the upper and lower retaining screws. The terminal block will back away from the module as you remove the screws. When replacing the terminal block, torque the retaining screws to 0.46 Nm (4.1 in-lbs).



Wiring the Finger-Safe Terminal Block

When wiring the terminal block, keep the finger-safe cover in place.

1. Loosen the terminal screws to be wired.
2. Route the wire under the terminal pressure plate. You can use the bare wire or a spade lug. The terminals accept a 6.35 mm (0.25 in.) spade lug.

TIP The terminal screws are non-captive. Therefore, it is possible to use a ring lug [maximum 1/4 inch o.d. with a 0.139 inch minimum i.d. (M3.5)] with the module.

3. Tighten the terminal screw making sure the pressure plate secures the wire. Recommended torque when tightening terminal screws is 0.68 Nm (6 in-lbs).

TIP If you need to remove the finger-safe cover, insert a screwdriver into one of the square, wiring holes and gently pry the cover off. If you wire the terminal block with the finger-safe cover removed, you will not be able to put it back on the terminal block because the wires will be in the way.

Wire Size and Terminal Screw Torque

Each terminal accepts up to two wires with the following restrictions:

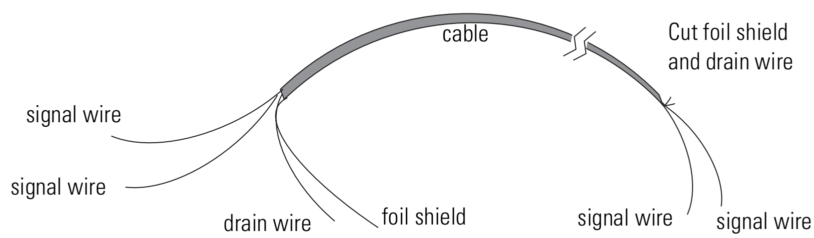
Wire Type		Wire Size	Terminal Screw Torque	Retaining Screw Torque
Solid	Cu-90°C (194°F)	#14 to #22 AWG	0.68 Nm (6 in-lbs)	0.46 Nm (4.1 in-lbs)
Stranded	Cu-90°C (194°F)	#16 to #22 AWG	0.68 Nm (6 in-lbs)	0.46 Nm (4.1 in-lbs)

Wiring the Modules

ATTENTION

To prevent shock hazard, care should be taken when wiring the module to signal sources. Before wiring any module, disconnect power from the system power supply and from any other source to the module.

After the module is properly installed, follow the wiring procedure below. To ensure proper operation and high immunity to electrical noise, always use shielded wire.



To wire your module follow these steps.

1. At each end of the cable, strip some casing to expose the individual wires.

2. Trim the signal wires to 2-in. (5 cm) lengths. Strip about 3/16 in. (5 mm) of insulation away to expose the end of the wire.

ATTENTION

Be careful when stripping wires. Wire fragments that fall into a module could cause damage at power up.

3. At the 1769-HSC input end of the cable, twist the drain wire and foil shield together, bend them away from the cable, and apply shrink wrap. Ground the shield at this end.

4. At the other end of the cable, cut the drain wire and foil shield back to the cable and apply shrink wrap.

5. Connect the signal wires to the terminal block. Connect the other end of the cable to the input device.

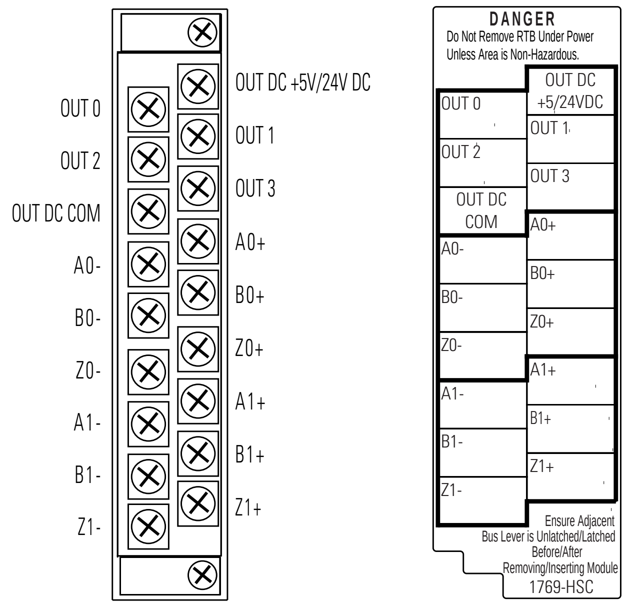
6. Repeat steps 1 through 5 for each channel on the module.

Terminal Door Label

A removable, write-on label is provided with the module. Remove the label from the door, mark the identification of each terminal with permanent ink, and slide the label back into the door. Your markings (ID tag) will be visible when the module door is closed.

Terminal Block Wiring

The input and output terminals are illustration in the figure below. Both inputs and outputs are isolated from the 1769 Compact bus.



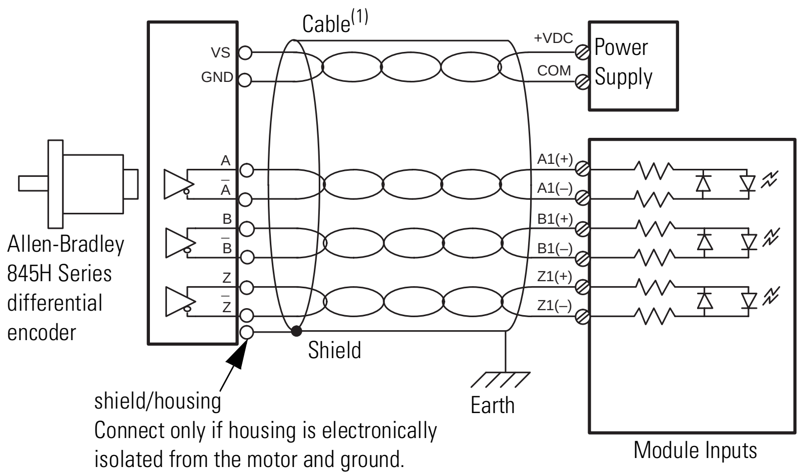
Wiring Diagrams

Inputs

The module utilizes differential inputs. Therefore, two input terminals are required for each input point. For example, the A0+ and A0- terminals are required for input point A0. Each input point is isolated from other input points, the 1769 Compact bus, and the entire output terminal group.

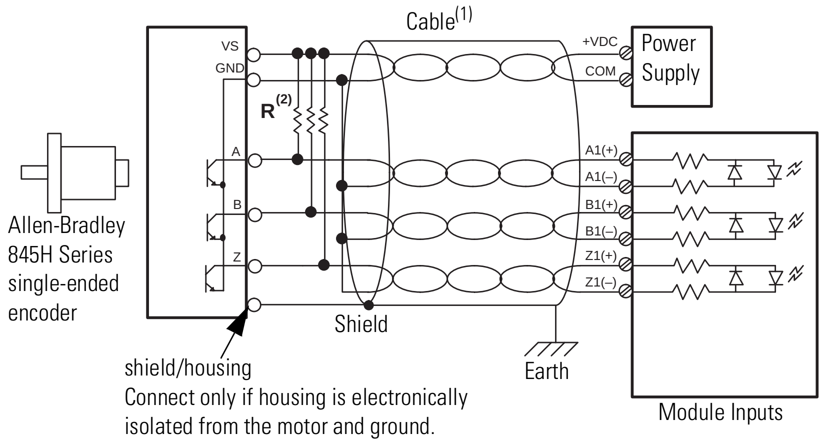
The inputs are compatible with standard differential line driver output devices as well as single-ended devices such as limit switches, photo-eyes, and proximity sensors. Examples of differential and single-ended circuits are shown in the following figures.

Figure 3.3 Differential Encoder Wiring



(1) Refer to your encoder manual for proper cable type. The type of cable used should be twisted pair, individually shielded cable with a maximum length of 300m (1000 ft.).

Figure 3.4 Single-Ended Encoder Wiring



(1) Refer to your encoder manual for proper cable type. The type of cable used should be twisted-pair, individually shielded cable with a maximum length of 300m (1000 ft.).

(2) External resistors are required if they are not internal to the encoder. The pull-up resistor (R) value depends on the power supply value. The table below shows the maximum resistor values for typical supply voltages. To calculate the maximum resistor value, use the following formula:

$$R = \frac{(Vdc - Vmin)}{Imin}$$

where:

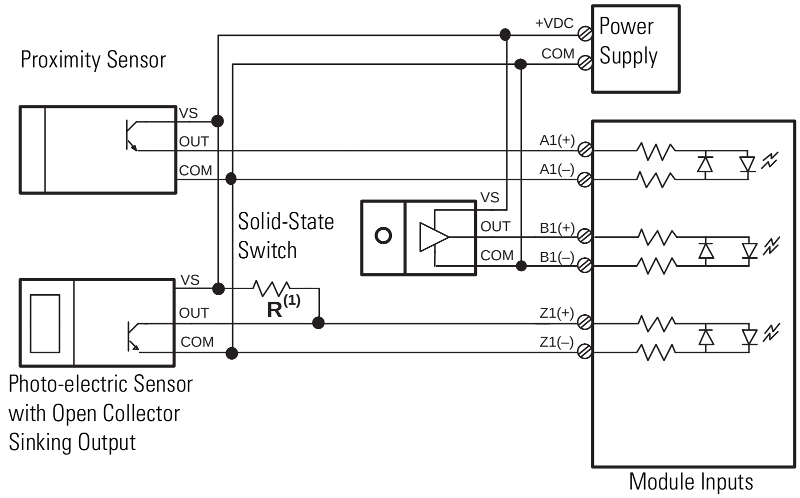
R = maximum pull-up resistor value
Vdc = power supply voltage
Vmin = 2.6V dc
Imin = 6.8 mA

Power Supply Voltage (Vdc)	Maximum Pull-up Resistor Value (R)(1)
5V dc	352 Ω
12V dc	1382 Ω
24V dc	3147 Ω

(1) Resistance values may change, depending upon your application.

The minimum resistor (R) value depends on the current sinking capability of the encoder. Refer to your encoder's documentation.

Figure 3.5 Discrete Device Wiring



(1) External resistors are required if they are not internal to the sensor. The pull-up resistor (R) value depends on the power supply value. The table below shows the maximum resistor values for typical supply voltages. To calculate the maximum resistor value, use the following formula:

$$R = \frac{(Vdc - Vmin)}{Imin}$$

where:

R = maximum pull-up resistor value
Vdc = power supply voltage
Vmin = 2.6V dc
Imin = 6.8 mA

Power Supply Voltage (Vdc)	Maximum Pull-up Resistor Value (R)(1)
5V dc	352 Ω
12V dc	1382 Ω
24V dc	3147 Ω

(1) Resistance values may change, depending upon your application.

The minimum resistor (R) value depends on the current sinking capability of the sensor. Refer to your sensor's documentation.

Outputs

The four output terminals must be powered by a user-supplied external source. User Power range is from +5 to +30V dc. See Output Specifications on page A-2 for voltage and current levels. There is no isolation between the outputs, but the outputs are isolated from the inputs and the 1769 Compact bus.

Electronic Protection

The electronic protection of the 1769-HSC has been designed to provide protection for current overload and short circuit conditions. The protection is based on a thermal cut-out principle. In the event of a short circuit or current overload condition on an output channel, that channel will turn off within milliseconds after the thermal cut-out temperature has been reached.

Overcurrent Autoreset Operation

The module detects overcurrent situations and reports them to the backplane in the Out*n*OverCurrent bits of the Input Array. When the overcurrent condition is detected, the outputs are turned off.

The module can latch outputs off in order to emulate the behavior of a physical fuse. Use the OvercurrentLatchOff bit to enable or disable this feature. When the OvercurrentLatchOff bit is set and an overcurrent situation occurs (even momentarily) the physical output will be latched off until the ResetBlownFuse bit is cycled from off to on (rising edge triggered). During the latched off time, the Readback.*n* bit in the Input Array also shows that the output is off.

If the OvercurrentLatchOff bit is not set, the output will be turned off for 1 second and then be retried (if still directed to be on). Retries will repeat until the overcurrent situation is corrected.

Only the 4 physical outputs can be latched off. The virtual outputs are not affected.

IMPORTANT During the retry period, the physical output and the Readback.*n* bits will be on briefly (until the overcurrent causes them to shut off again). Take this into consideration and configure your system accordingly.

TIP Short-circuits and overload conditions should be corrected as soon as possible. Damage may occur if short-circuits or overload conditions are allowed for extended periods.

Transistor Output Transient Pulses

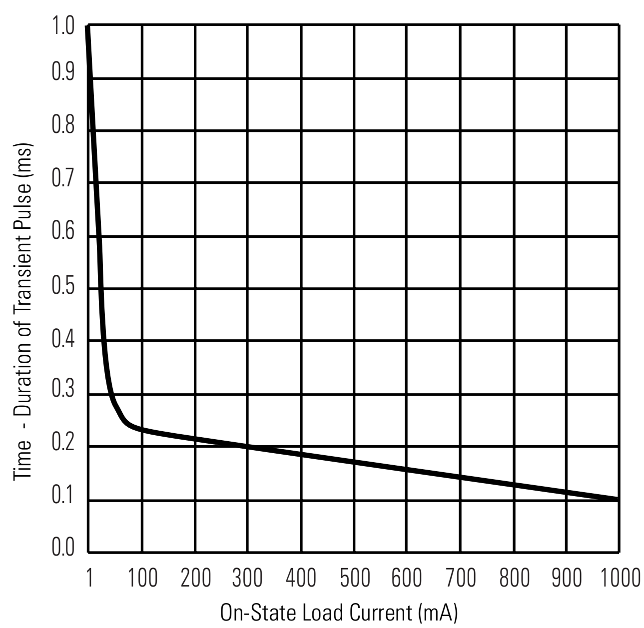
The maximum duration of the transient pulse occurs when minimum load is connected to the output. However, for most applications, the energy of the transient pulse is not sufficient to energize the load.

ATTENTION

A transient pulse occurs in transistor outputs when the external DC supply voltage is applied to the output common terminals (e.g. via the master control relay). The sudden application of voltage creates this transient pulse. This condition is inherent in transistor outputs and is common to solid state devices. A transient pulse can occur regardless of the controller having power or not. Refer to your controller's user manual to reduce inadvertent operation.

The graph below illustrates that the duration of the transient is proportional to the load current. Therefore, as the on-state load current increases, the transient pulse decreases. Power-up transients do not exceed the time duration shown below, for the amount of loading indicated, at 60°C (140°F).

Figure 3.6 Transient Pulse Duration as a Function of Load Current



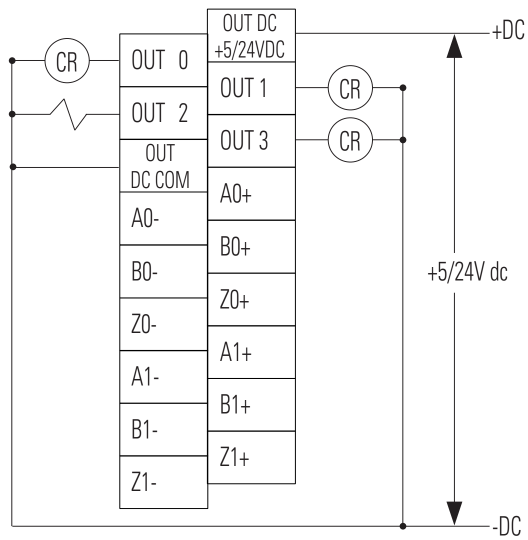
Output wiring is illustrated in the following diagram.

Figure 3.7 Output Wiring

Basic wiring[1] of output devices[2] to the module is shown below.

ATTENTION

- Miswiring of the module to an AC power source or applying reverse polarity will damage the module.
- Be careful when stripping wires. Wire fragments that fall into a module could cause damage at power up. Once wiring is complete, ensure the module is free of all metal fragments.



(1) Recommended Surge Suppression - The module has built-in suppression which is sufficient for most applications, however, for high-noise applications, use a 1N4004 diode reverse-wired across the load for transistor outputs switching 24V dc inductive loads. For additional details, refer to Industrial Automation Wiring and Grounding Guidelines, Allen-Bradley publication 1770-4.1.

(2) Sourcing Output - Source describes the current flow between the I/O module and the field device. Sourcing output circuits supply (source) current to sinking field devices. Field devices connected to the negative side (DC Common) of the field power supply are sinking field devices. Field devices connected to the positive side (+V) of the field supply are sourcing field devices. *Europe:* DC sinking input and sourcing output module circuits are the commonly used options.

Module Configuration, Output, and Input Data

After installation of the 1769-HSC module, you must configure it for operation, using the programming software compatible with the controller (for example, RSLogix 500 or RSLogix 5000).

TIP

Normal counter configuration is done using programming software. In that case, it is not necessary to know the meaning of the bit location. However, some systems allow configuration to be changed by the control program. Refer to your controller's documentation for details.

Information on programming the module using specific controllers and software is contained in the following Appendices.

Appendix	Controller	Software
Appendix B	CompactLogix Controller	RSLogix 5000
Appendix C	MicroLogix 1500 Controller	RSLogix 500

Configuring the Module

The module uses three arrays: Configuration Array, Output Array, and Input Array. You configure the module by establishing settings in the Configuration and Output Arrays. The Input Array shows the data that the module sends to the controller.

IMPORTANT

Both the Configuration Array and Output Array settings affect the module configuration. Changing certain configuration parameters from defaults may necessitate changing other values to avoid configuration errors.

Configuration Array

The Configuration Array, which consists of 118 words, allows you to specify how the module's counters will function. The default value is all zeros with the exception of:

- NumberofCounters (see page 4-8)
- Ctr*n*MaxCount (see page 4-11)
- Ctr*n*MinCount (see page 4-12)
- Ctr*n*Scalar (see page 4-14)
- Ctr*n*CyclicUpdateTime (see page 4-14)

TIP

Normal counter configuration is done using programming software. In that case, it is not necessary to know the bit location. However, some systems allow configuration to be changed by the control program. Refer to your controller's documentation for details.

IMPORTANT

When changing configuration values, verify that only valid configurations are created for the module. For example, changing NumberofCounters from its default of 1 to 0 requires that Ctr1MinCount and Ctr1MaxCount also be set to 0, etc.

See Table 5.6 "Configuration Error Codes" on page 5-9 if you encounter configuration errors.

Word 0 contains general configuration bits. Word 1 contains the filter settings. Words 2 through 5 refer to the physical outputs. Words 6 through 45 are counter configuration words. Words 46 through 117 are range configuration words. More detailed descriptions of the configuration words and bits follow the Configuration Array below.

IMPORTANT

Certain values (noted below) cannot be changed while a counter(s) or range(s) is enabled. Attempting to do so will cause a configuration error and the entire Configuration Array will be rejected until the error is eliminated.

Table 4.1 Configuration Array

<table>
<tr><th rowspan="2">Word</th><th colspan="16">Bit</th><th rowspan="2">Function</th></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td>0</td><td colspan="6">Not Used</td><td colspan="2">NumberOf Counters</td><td colspan="3">Not Used</td><td>PFE</td><td colspan="2">Not Used</td><td>Ctr Rst</td><td>OCLO</td><td>General Configuration Bits</td></tr>
<tr><td>1</td><td colspan="2">Filter_Z1</td><td>Not Used</td><td colspan="2">Filter_B1</td><td>Not Used</td><td colspan="2">Filter_A1</td><td colspan="2">Filter_Z0</td><td>Not Used</td><td colspan="2">Filter_B0</td><td>Not Used</td><td colspan="2">Filter_A0</td><td>Filter Selection</td></tr>
<tr><td>2</td><td colspan="8">Not Used</td><td>Out3 PSR</td><td>Out2 PSR</td><td>Out1 PSR</td><td>Out0 PSR</td><td>Out3 PM</td><td>Out2 PM</td><td>Out1 PM</td><td>Out0 PM</td><td>Output Program Mode and Output Program State Run</td></tr>
<tr><td>3</td><td colspan="12">Not Used</td><td>Out3 PV</td><td>Out2 PV</td><td>Out1 PV</td><td>Out0 PV</td><td>Output Program Value</td></tr>
<tr><td>4</td><td colspan="8">Not Used</td><td>Out3 FSR</td><td>Out2 FSR</td><td>Out1 FSR</td><td>Out0 FSR</td><td>Out3 FM</td><td>Out2 FM</td><td>Out1 FM</td><td>Out0 FM</td><td>Output Fault Mode and Output Fault State Run</td></tr>
<tr><td>5</td><td colspan="12">Not Used</td><td>Out3 FV</td><td>Out2 FV</td><td>Out1 FV</td><td>Out0 FV</td><td>Output Fault Value</td></tr>
<tr><td>6</td><td colspan="16" rowspan="2">Ctr0MaxCount</td><td rowspan="2">Counter 0 Maximum Count</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td colspan="16" rowspan="2">Ctr0MinCount</td><td rowspan="2">Counter 0 Minimum Count</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td><td colspan="16" rowspan="2">Ctr0Preset</td><td rowspan="2">Counter 0 Preset</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td colspan="16">Ctr0Hysteresis</td><td>Counter 0 Hysteresis</td></tr>
<tr><td>13</td><td colspan="16">Ctr0Scalar</td><td>Counter 0 Scalar</td></tr>
<tr><td>14</td><td colspan="16">Ctr0CyclicRateUpdateTime</td><td>Counter 0 Cyclic Rate Update Time</td></tr>
<tr><td>15</td><td colspan="3">Not Used</td><td>Lin-ear</td><td>Not Used</td><td colspan="3">Storage Mode</td><td colspan="5">Not Used</td><td colspan="3">Operational Mode</td><td>Counter 0 Configuration Flags</td></tr>
<tr><td>16</td><td colspan="16" rowspan="2">Ctr1MaxCount</td><td rowspan="2">Counter 1 Maximum Count</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td><td colspan="16" rowspan="2">Ctr1MinCount</td><td rowspan="2">Counter 1 Minimum Count</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td><td colspan="16" rowspan="2">Ctr1Preset</td><td rowspan="2">Counter 1 Preset</td></tr>
<tr><td>21</td></tr>
<tr><td>22</td><td colspan="16">Ctr1Hysteresis</td><td>Counter 1 Hysteresis</td></tr>
<tr><td>23</td><td colspan="16">Ctr1Scalar</td><td>Counter 1 Scalar</td></tr>
<tr><td>24</td><td colspan="16">Ctr1CyclicRateUpdateTime</td><td>Counter 1 Cyclic Rate Update Time</td></tr>
<tr><td>25</td><td colspan="3">Not Used</td><td>Lin-ear</td><td>Not Used</td><td colspan="3">Storage Mode</td><td colspan="5">Not Used</td><td colspan="3">Operational Mode</td><td>Counter 1 Configuration Flags</td></tr>
<tr><td>26</td><td colspan="16" rowspan="2">Ctr2MaxCount</td><td rowspan="2">Counter 2 Maximum Count</td></tr>
<tr><td>27</td></tr>
<tr><td>28</td><td colspan="16" rowspan="2">Ctr2MinCount</td><td rowspan="2">Counter 2 Minimum Count</td></tr>
<tr><td>29</td></tr>
<tr><td>30</td><td colspan="16" rowspan="2">Ctr2Preset</td><td rowspan="2">Counter 2 Preset</td></tr>
<tr><td>31</td></tr>
<tr><td>32</td><td colspan="16">Ctr2Hysteresis</td><td>Counter 2 Hysteresis</td></tr>
<tr><td>33</td><td colspan="16">Ctr2Scalar</td><td>Counter 2 Scalar</td></tr>
<tr><td>34</td><td colspan="16">Ctr2CyclicRateUpdateTime</td><td>Counter 2 Cyclic Rate Update Time</td></tr>
</table>

Table 4.1 Configuration Array

Word	Bit 15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Function
35	Not Used			Lin-ear	Not Used												Counter 2 Configuration Flags
36	Ctr3MaxCount																Counter 3 Maximum Count
37																	
38	Ctr3MinCount																Counter 3 Minimum Count
39																	
40	Ctr3Preset																Counter 3 Preset
41																	
42	Ctr3Hysteresis																Counter 3 Hysteresis
43	Ctr3Scalar																Counter 3 Scalar
44	Ctr3CyclicRateUpdateTime																Counter 3 Cyclic Rate Update Time
45	Not Used			Lin-ear	Not Used												Counter 3 Configuration Flags
46	Range0To11[0].HighLimit																Range 0 High Limit
47																	
48	Range0To11[0].LowLimit																Range 0 Low Limit
49																	
50	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 0 Output Control
51	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 0 Configuration Flags
52	Range0To11[1].HighLimit																Range 1 High Limit
53																	
54	Range0To11[1].LowLimit																Range 1 Low Limit
55																	
56	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 1 Output Control
57	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 1 Configuration Flags
58	Range0To11[2].HighLimit																Range 2 High Limit
59																	
60	Range0To11[2].LowLimit																Range 2 Low Limit
61																	
62	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 2 Output Control
63	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 2 Configuration Flags
64	Range0To11[3].HighLimit																Range 3 High Limit
65																	
66	Range0To11[3].LowLimit																Range 3 Low Limit
67																	
68	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 3 Output Control
69	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 3 Configuration Flags
70	Range0To11[4].HighLimit																Range 4 High Limit
71																	

Table 4.1 Configuration Array

Word	Bit																Function
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
72	Range0To11[4].LowLimit																Range 4 Low Limit
73																	
74	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 4 Output Control
75	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 4 Configuration Flags
76	Range0To11[5].HighLimit																Range 5 High Limit
77																	
78	Range0To11[5].LowLimit																Range 5 Low Limit
79																	
80	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 5 Output Control
81	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 5 Configuration Flags
82	Range0To11[6].HighLimit																Range 6 High Limit
83																	
84	Range0To11[6].LowLimit																Range 6 Low Limit
85																	
86	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 6 Output Control
87	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 6 Configuration Flags
88	Range0To11[7].HighLimit																Range 7 High Limit
89																	
90	Range0To11[7].LowLimit																Range 7 Low Limit
91																	
92	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 7 Output Control
93	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 7 Configuration Flags
94	Range0To11[8].HighLimit																Range 8 High Limit
95																	
96	Range0To11[8].LowLimit																Range 8 Low Limit
97																	
98	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 8 Output Control
99	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 8 Configuration Flags
100	Range0To11[9].HighLimit																Range 9 High Limit
101																	
102	Range0To11[9].LowLimit																Range 9 Low Limit
103																	
104	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 9 Output Control
105	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 9 Configuration Flags
106	Range0To11[10].HighLimit																Range 10 High Limit
107																	
108	Range0To11[10].LowLimit																Range 10 Low Limit
109																	

Table 4.1 Configuration Array

Word	Bit 15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Function
110	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 10 Output Control
111	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 10 Configuration Flags
112	Range0To11[11].HighLimit																Range 11 High Limit
113																	
114	Range0To11[11].LowLimit																Range 11 Low Limit
115																	
116	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 09	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Range 11 Output Control
117	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr		Range 11 Configuration Flags

General Configuration Bits

Configuration Array Word 0	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
General Configuration Bits	Not Used						Numberof Counters		Not Used			PFE	Not Used		Ctr Reset	OCLO

OCLO - Overcurrent Latch Off (OverCurrentLatchOff)

When set, this bit causes the module to make any overcurrent activity latch the corresponding output off, simulating a physical fuse. When OCLO = 0, it automatically resets. The rising edge of RBF resets the output.

IMPORTANT Do not set this bit while a counter or range is enabled (Ctr0En, Ctr1En, Ctr2En, Ctr3En, or RangeEn set to 1). Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Counter Reset (CtrReset)

The CtrReset bit in the Configuration Array, when set, causes the following to occur when the system transitions to Run or the Inhibit Module bit transitions to 0:

- All counters are disabled and reset to zero.
- The Output Array is reset to default values until the ModConfig bit is set (1). The default value for the Output Array is all zeros.
- The Input Array counter Status Flags (Overflow, Underflow, RisingEdgeZ, RateValid, PresetWarning) are reset.
- The Input Array counter values (Current Count[1], StoredCount, CurrentRate and PulseInterval) are also reset to zero.
- All counts are lost and all outputs are turned off.

IMPORTANT	For most predictable results, you may want to clear the output image of the processor BEFORE performing a counter reset (CtrReset) to the 1769-HSC module. This is because CtrReset does not change the processor's output image. CtrReset sets the 1769-HSC module's Output Array to all zero's. If any bit is set to 1 in the processor's output image, when sent to the module, it will be seen as a state transition and be acted upon.

PFE - Program to Fault Enable (ProgToFaultEn)

This bit indicates what should happen when the bus controller indicates a change from one condition (Program mode) to another (Fault mode). If this bit is set (1), the safe state operation of all 4 real outputs changes to that identified by the Fault State and Fault Value words. If this bit is reset (0), the module continues with the operation identified by the Program State and Program Value words.

(1) If zero is outside the MinCount and MaxCount limits set in the Configuration Array, then the Preset value is loaded into CurrentCount instead of zero. This also causes the PresetWarning bit to be set, which, in turn, sets the GenError bit.

Number of Counters (NumberOfCounters)

This 2-bit value indicates whether the module uses 1 counter, 2 counters, 3 counters, or 4 counters. The default value is 1 (2 counters).

Table 4.2 Number of Counters Determination

Bit 01	Bit 00	Counters
0	0	1
0	1	2
1	0	3
1	1	4

IMPORTANT Do not set this value while a counter or range is enabled (Ctr0En, Ctr1En, Ctr2En, Ctr3En, or RangeEn set to 1). Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Filter Selection

Configuration Array Word 1	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Filter Selection	Filter_Z1		Not Used	FilterB1		Not Used	FilterA1		FilterZ0		Not Used	FilterB0		Not Used	FilterA0	

This value indicates the nominal filter frequency as shown in the table below.

Table 4.3 Filter Selection Settings

Filters and Corresponding Bits	**FilterA0**	**Bit 1 - FilterA0_1**	**Bit 0 - FilterA0_0**
	FilterB0	**Bit 4 - FilterB0_1**	**Bit 3 - FilterB0_0**
	FilterZ0	**Bit 7 - FilterZ0_1**	**Bit 6 - FilterZ0_0**
	FilterA1	**Bit 9 - FilterA1_1**	**Bit 8 - FilterA1_0**
	FilterB1	**Bit 12 - FilterB1_1**	**Bit 11 - FilterB1_0**
	FilterZ1	**Bit 15 - FilterZ1_1**	**Bit 14 - FilterZ1_0**
Nominal Frequency Settings	None	0	0
	0.01 ms minimum pulse width	0	1
	0.5 ms minimum pulse width	1	0
	5 ms minimum pulse width	1	1

IMPORTANT Do not set these bits while certain counters or ranges are enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Program Mode and Program State Run

Configuration Array Word 2	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Output Program Mode and Output Program State Run	Not Used								Out3 PSR	Out2 PSR	Out1 PSR	Out0 PSR	Out3 PM	Out2 PM	Out1 PM	Out0 PM

Program Mode (Out0ProgramMode through Out3ProgramMode)

The program mode bits configure the output for Hold Last State (HLS) or User-Defined Safe State (UDSS) during Program State.

- 1 = Hold Last State
- 0 = User-Defined Safe State

IMPORTANT Program Mode and Program State Run only apply to certain controllers. Refer to your controller's documentation for more information.

Program State Run (Out0ProgramStateRun through Out3ProgramStateRun)

Program State Run allows you to specify, on a bit basis, that the output should continue to be controlled by the module as if it were in the Run state. That is, events on the module or changes in the Output image will affect the physical outputs without regard to the Program_HLS or UDSS state indicated. When this bit is set, the corresponding Program Mode and Program Value bits are ignored.

ATTENTION Selecting this option will allow outputs to change state while ladder logic is not running. You must take care to assure that this does not pose a risk of injury or equipment damage when selecting this option.

IMPORTANT The prescan initiated by some controllers could have an effect on the outputs. To overcome any changes in physical output states that may be caused by this, retentive output instructions (eg. latch, unlatch etc.) should be used when bit manipulations are done on the Output image of this module in ladder logic.

This applies to a wide range of bits when Program State Run is selected, since presetting a counter, enabling a range, changing a mask, and changing Configuration Array settings can cause ranges and outputs to change state.

Output Program Value (Out0ProgramValue through Out3ProgramValue)

Configuration Array Word 3	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Output Program Value	Not Used												Out3 PV	Out2 PV	Out1 PV	Out0 PV

These bits are the values that will be applied to each of the real outputs when User-Defined Safe State (UDSS) is set as described above and the module is in Program state.

Output Fault Mode and Output Fault State Run

Configuration Array Word 4	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Output Fault Mode and Output Fault State Run	Not Used								Out3 FSR	Out2 FSR	Out1 FSR	Out0 FSR	Out3 FM	Out2 FM	Out1 FM	Out0 FM

Output Fault Mode (Out0FaultMode through Out3FaultMode)

These bits configure the output for Hold Last State or User-Defined Safe State during a Fault state.

- 1 = Hold Last State
- 0 = User-Defined Safe State

Output Fault State Run (Out0FaultStateRun through Out3FaultStateRun)

Similar to Program State Run, Fault State Run allows you to specify, on a bit basis, that the output should continue to be controlled by the module as if it were Run state. That is, events on the module or changes in the Output image will affect the physical outputs without regard to the Program_HLS or UDSS state indicated. When this bit is set, the corresponding Program Mode and Program Value bits are ignored.

ATTENTION

Selecting this option will allow outputs to change state while ladder logic is not running. You must take care to assure that this does not pose a risk of injury or equipment damage when selecting this option.

IMPORTANT The prescan initiated by some controllers could have an effect on the outputs. To overcome any changes in physical output states that may be caused by this, retentive output instructions (eg. latch, unlatch etc.) should be used when bit manipulations are done on the Output image of this module in ladder logic.

This applies to a wide range of bits when Fault State Run is selected, since presetting a counter, enabling a range, changing a mask, and changing Configuration Array settings can cause ranges and outputs to change state.

Output Fault Value (Out0FaultValue through Out3FaultValue)

Configuration Array Word 5		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
5	Output Fault Value	Not Used												Out3 FV	Out2 FV	Out1 FV	Out0 FV

These bits are the values that will be applied to each of the real outputs when User-Defined Safe State is set as described above and the module is in Fault state.

TIP Outputs are also affected by PFT above.

Counter Maximum Count (Ctr*n*MaxCount)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
6 7	Counter 0 Maximum Count	Ctr0MaxCount															
16 17	Counter 1 Maximum Count	Ctr1MaxCount															
26 27	Counter 2 Maximum Count	Ctr2MaxCount															
36 37	Counter 3 Maximum Count	Ctr3MaxCount															

This is the maximum count value allowed for counter (*n*). The count value cannot exceed this value. Allowable values are Ctr*n*MinCount + 1 through 2,147,483,647 (decimal).

The default value is +2,147,483,647 decimal for counters 0 and 1. The default value is 0 for counters 2 and 3.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Counter Minimum Count (Ctr*n*MinCount)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
8 9	Counter 0 Minimum Count	Ctr0MinCount															
18 19	Counter 1 Minimum Count	Ctr1MinCount															
28 29	Counter 2 Minimum Count	Ctr2MinCount															
38 39	Counter 3 Minimum Count	Ctr3MinCount															

This is the minimum count value allowed for counter (*n*). The count value cannot fall below this value. This value must be less than Ctr*n*MaxCount or a configuration error occurs. Allowable values are from -2,147,483,648 to Ctr*n*MaxCount - 1.

The default value is -2,147,483,648 decimal for counters 0 and 1. The default value is 0 for counters 2 and 3.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Counter Preset (Ctr*n*Preset)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
10 11	Counter 0 Preset	Ctr0Preset															
20 21	Counter 1 Preset	Ctr1Preset															
30 31	Counter 2 Preset	Ctr2Preset															
40 41	Counter 3 Preset	Ctr3Preset															

This value can be used to change the current count value of counter*n* on certain gate (Z*n*) events and when Ctr*n*SoftPreset is used.

Ctr*n*Preset must be greater than or equal to Ctr*n*MinCount and less than Ctr*n*MaxCount. The default value is zero.

Counter Hysteresis (Ctr*n*Hysteresis)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
12	Counter 0 Hysteresis	Ctr0Hysteresis															
22	Counter 1 Hysteresis	Ctr1Hysteresis															
32	Counter 2 Hysteresis	Ctr2Hysteresis															
42	Counter 3 Hysteresis	Ctr3Hysteresis															

The hysteresis value is the number of counts that should be disregarded in the calculation of the cyclic rate. If the count value changes by less than the hysteresis value, the rate is reported as zero, regardless of the actual rate at which the pulses are counted.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Counter Scalar (Ctr*n*Scalar)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
13	Counter 0 Scalar	Ctr0Scalar															
23	Counter 1 Scalar	Ctr1Scalar															
33	Counter 2 Scalar	Ctr2Scalar															
43	Counter 3 Scalar	Ctr3Scalar															

This value is used to scale the Rate value. The Rate value is divided by the Scalar value. The default value is 1 for counters 0 and 1. The default value is 0 for counters 2 and 3.

Ctr*n*Scalar may be used to determine RPM. To configure the Ctr[*n*].CurrentRate value to show an RPM value, set Ctr*n*Scalar to (counts per revolution)/60. See page 2-20 for more information.

IMPORTANT For any counter being used, do not set Scalar to a value less than one or a configuration error will occur.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Cyclic Rate Update Time (Ctr*n*CyclicRateUpdateTime)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
14	Counter 0 Cyclic Rate Update Time	Ctr0CyclicRateUpdateTime															
24	Counter 1 Cyclic Rate Update Time	Ctr1CyclicRateUpdateTime															
34	Counter 2 Cyclic Rate Update Time	Ctr2CyclicRateUpdateTime															
44	Counter 3 Cyclic Rate Update Time	Ctr3CyclicRateUpdateTime															

This value is used to set the cyclic rate update time for the CurrentRate calculation. The value indicates the time in milliseconds from 1 to 32767. An invalid number causes a configuration error. The default value is 10 for counters 0 and 1. The default value is 0 for counters 2 and 3.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

See page 2-19 for more information on cyclic rate.

Configuration Flags

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
15	Counter 0 Configuration Flags	Not Used			Linear	Not Used	Storage Mode			Not Used					Operational Mode		
25	Counter 1 Configuration Flags	Not Used			Linear	Not Used	Storage Mode			Not Used					Operational Mode		
35	Counter 2 Configuration Flags	Not Used			Linear	Not Used											
45	Counter 3 Configuration Flags	Not Used			Linear	Not Used											

*Operational Mode (Ctr*n*Config.OperationalMode_0 through Ctr*n*Config.OperationalMode_2)*

These bits apply to Counters 0 and 1 only.

This value determines how the A*0* or A*1* and B*0* or B*1* inputs are decoded when assigned to counter 0 or counter 1. See the following table.

Table 4.4 Operational Mode Settings

Set bit			For function
Ctr*n*Config.OperationalMode_2	**Ctr*n*Config.OperationalMode_1**	**Ctr*n*Config.OperationalMode_0**	
0	0	0	Pulse internal direction
0	0	1	Pulse external direction
1	0	0	Quadrature encoder X1
1	0	1	Quadrature encoder X2
1	1	0	Quadrature encoder X4
0	1	0	Up/Down Pulses
0	1	1	reserved
1	1	1	reserved

TIP

The Ctr1Config.OperationalMode bits are reserved if the Number of Counters equals 1. Attempting to set reserved bits will result in a configuration error.

IMPORTANT

Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

*Storage Mode (Ctr*n*Config.StorageMode_0 through Ctr*n*Config.StorageMode_2)*

These three bits apply to Counters 0 and 1 only. They define how the module interprets the Z input, as shown below. Each bit works independently. If bit 0 and bit 2 are set simultaneously, a Z event causes the Current Count Value to be stored and then preset.

Table 4.5 Storage Mode Settings

Set bit	For function
Ctr*n*Config.StorageMode_0	Stores the Current Count Value on the rising edge of Z to Ctr[*n*].StoredCount in the input file.
Ctr*n*Config.StorageMode_1	Holds the counter at its Current Count Value while Z = 1.
Ctr*n*Config.StorageMode_2	Presets the Current Count Value on the rising edge of Z.

IMPORTANT Z = internal Z. Internal Z is the version of the Z input pin as modified by the Output Array control bits Z Invert and Z Inhibit.

TIP The Ctr1Config.StorageMode bits are reserved if NumberofCounters_1 and NumberofCounters_0 are set to 00 (one counter). Attempting to set reserved bits will result in a configuration error.

IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Linear (Ctr0Config.Linear through Ctr3Config.Linear)

This bit indicates how the counter operates upon reaching a Ctr*n*MinCount or Ctr*n*MaxCount.

- 0 = Ring Counter
- 1 = Linear Counter

See page 2-14 for a description of ring and linear counter operation.

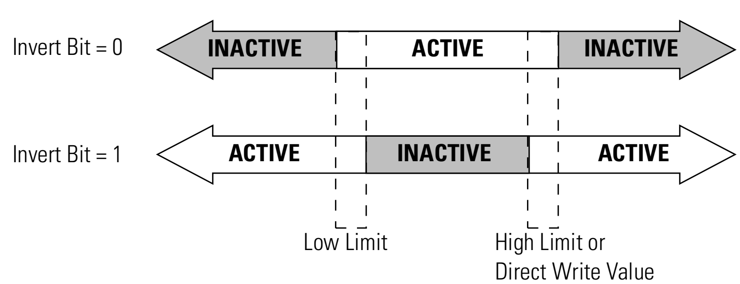
IMPORTANT Do not change this value while the counter is enabled. Attempting to do so will result in a BadModConfigUpdate error. See page 5-13 for a full list of prohibited settings.

Range High Limit (Range0To11[*n*].HighLimit) and Range Low Limit (Range0To11[*n*].LowLimit)

<table>
<tr><th colspan="2">Configuration Array Words</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td>46 and 47</td><td>Range 0 High Limit</td><td colspan="16">Range0To11[0].HighLimit</td></tr>
<tr><td>48 and 49</td><td>Range 0 Low Limit</td><td colspan="16">Range0To11[0].LowLimit</td></tr>
<tr><td>52 and 53</td><td>Range 1 High Limit</td><td colspan="16">Range0To11[1].HighLimit</td></tr>
<tr><td>54 and 55</td><td>Range 1 Low Limit</td><td colspan="16">Range0To11[1].LowLimit</td></tr>
<tr><td>58and 59</td><td>Range 2 High Limit</td><td colspan="16">Range0To11[2].HighLimit</td></tr>
<tr><td>60 and 61</td><td>Range 2 Low Limit</td><td colspan="16">Range0To11[2].LowLimit</td></tr>
<tr><td>64 and 65</td><td>Range 3 High Limit</td><td colspan="16">Range0To11[3].HighLimit</td></tr>
<tr><td>66 and 67</td><td>Range 3 Low Limit</td><td colspan="16">Range0To11[3].LowLimit</td></tr>
<tr><td>70 and 71</td><td>Range 4 High Limit</td><td colspan="16">Range0To11[4].HighLimit</td></tr>
<tr><td>72 and 73</td><td>Range 4 Low Limit</td><td colspan="16">Range0To11[4].LowLimit</td></tr>
<tr><td>76 and 77</td><td>Range 5 High Limit</td><td colspan="16">Range0To11[5].HighLimit</td></tr>
<tr><td>78 and 79</td><td>Range 5 Low Limit</td><td colspan="16">Range0To11[5].LowLimit</td></tr>
<tr><td>82 and 83</td><td>Range 6 High Limit</td><td colspan="16">Range0To11[6].HighLimit</td></tr>
<tr><td>84 and 85</td><td>Range 6 Low Limit</td><td colspan="16">Range0To11[6].LowLimit</td></tr>
<tr><td>88 and 89</td><td>Range 7 High Limit</td><td colspan="16">Range0To11[7].HighLimit</td></tr>
<tr><td>90 and 91</td><td>Range 7 Low Limit</td><td colspan="16">Range0To11[7].LowLimit</td></tr>
<tr><td>94 and 95</td><td>Range 8 High Limit</td><td colspan="16">Range0To11[8].HighLimit</td></tr>
<tr><td>96 and 97</td><td>Range 8 Low Limit</td><td colspan="16">Range0To11[8].LowLimit</td></tr>
<tr><td>100 and 101</td><td>Range 9 High Limit</td><td colspan="16">Range0To11[9].HighLimit</td></tr>
<tr><td>102 and 103</td><td>Range 9 Low Limit</td><td colspan="16">Range0To11[9].LowLimit</td></tr>
<tr><td>106 and 107</td><td>Range 10 High Limit</td><td colspan="16">Range0To11[10].HighLimit</td></tr>
<tr><td>108 and 109</td><td>Range 10 Low Limit</td><td colspan="16">Range0To11[10].LowLimit</td></tr>
<tr><td>112 and 113</td><td>Range 11 High Limit</td><td colspan="16">Range0To11[11].HighLimit</td></tr>
<tr><td>114 and 115</td><td>Range 11 Low Limit</td><td colspan="16">Range0To11[11].LowLimit</td></tr>
</table>

These values, which represent a count value or rate value, depending upon the programed Type, are used for range comparison. When the rate value is equal to Range0To11[*n*].HighLimit or Range0To11[*n*].LowLimit, Range*n* changes state, becoming either active or inactive, depending upon the setting of the Range0To11[*n*].Invert bit.

Object Value (Current Count or Current Rate)



TIP

Range0To11[*n*].HighLimit must be greater than Range0To11[*n*].LowLimit or a configuration error results.

Range Output Control (Range0To11[*n*].OutputControl)

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
50	Range 0 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
56	Range 1 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
62	Range 2 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
68	Range 3 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
74	Range 4 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
80	Range 5 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
86	Range 6 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
92	Range 7 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
98	Range 8 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
104	Range 9 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
110	Range 10 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
116	Range 11 Output Control	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0

These 16-bit words indicate which outputs should be enabled when a range is active. When range *n* is enabled, this word is combined with the other range output masks as described in Output Off Mask (OutputOffMask.0 through OutputOffMask.15) on page 4-23 and Output On Mask (OutputOnMask.0 through OutputOnMask.15) on page 4-22.

Range Configuration Flags

Configuration Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
51	Range 0 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
57	Range 1 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
63	Range 2 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
69	Range 3 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
75	Range 4 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
81	Range 5 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
87	Range 6 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
93	Range 7 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
99	Range 8 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
105	Range 9 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
111	Range 10 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	
117	Range 11 Configuration Flags	Not Used							Inv	Not Used			Type	Not Used		ToThisCtr	

ToThisCtr (Range0To11[n].ToThisCounter)

This 2-bit value indicates which counter is used in the range comparison for range *n*, as shown in the table below.

Table 4.6 Range Counter Number Determination

Bit 01	Bit 00	Counter
0	0	0
0	1	1
1	0	2
1	1	3

IMPORTANT If this value is greater than NumberOfCounters, a configuration error occurs.

Type (Range0To11[n].Type)

This bit indicates which type of value to use for the range comparison in range *n*. This value and Range0To11[*n*].ToThisCounter determine the current value that is used in range comparison as the rate or count value.

Range0To11[*n*].Type	Range Type
0	Count Value
1	Rate Value

Inv (Range0To11[n].Invert)

This bit indicates whether the range *n* should be active inside or outside the Range0To11[*n*].Low Limit and Range0To11[*n*].HighLimit window.

- 0 = The range *n* is active when the rate or count value is at or between Range0To11[*n*].Low Limit and Range0To11[*n*].HighLimit. When the range is active, the RangeActive.*n* bit is set. When the range is active and enabled, the outputs indicated in the Range Output Control word are activated.
- 1 = The range *n* is active when the rate or count value is lower than or equal to Range0To11[*n*].LowLimit or higher than or equal to Range0To11[*n*].HighLimit. When the range is active, the RangeActive.*n* bit is set. When the range is active and enabled, the outputs indicated in the Range Output Control word are applied.

TIP Ranges can be active in overflow, underflow, and rollover situations.

Output Array

The Output Array, which consists of 34 words, allows you to access the module's real-time output data to control the module. The default value is all zeros.

IMPORTANT The Output Array contains dynamic configuration data. The settings in the Output Array must be compatible with the settings in the Configuration Array.

For example, do not attempt to set Counter Control Bits for a given counter in the Output Array unless NumberOfCounters in the Configuration Array indicates that the counter is declared to be used.

IMPORTANT All *Not Used* bits (shaded in Table 4.7) must be set to 0 or the InvalidOutput bit in the Input Array will be set. When the InvalidOutput bit is set, the entire Output Array is rejected until an Output Array that does not have this error is sent.

Table 4.7 Output Array

Word	Bit																Function
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
0	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Output On Mask
1	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0	Output Off Mask
2	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	Range Enable
3	Not Used																Not Used
4	Not Used								RBF	Not Used							Reset Blown Fuse
5	Not Used						RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RU	RO	SP	En	Counter 0 Control Bits
6	Not Used						RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RU	RO	SP	En	Counter 1 Control Bits
7	Not Used						RPW	Not Used				D Inv	RU	RO	SP	En	Counter 2 Control Bits
8	Not Used						RPW	Not Used				D Inv	RU	RO	SP	En	Counter 3 Control Bits
9	Not Used																Not Used
10	Range12To15[0].HiLimOrDirWr																Range High Limit or Direct Write Value
11																	
12	Range12To15[0].LowLimit																Range Low Limit
13																	
14	Range12To15[0].OutputControl																Range Output Control
15	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr		Range Configuration Flags
16	Range12To15[1].HiLimOrDirWr																Range High Limit or Direct Write Value
17																	
18	Range12To15[1].LowLimit																Range Low Limit
19																	
20	Range12To15[1].OutputControl																Range Output Control
21	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr		Range Configuration Flags

Table 4.7 Output Array

Word	Bit																Function
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
22	Range12To15[2].HiLimOrDirWr																Range High Limit or Direct Write Value
23																	
24	Range12To15[2].LowLimit																Range Low Limit
25																	
26	Range12To15[2].OutputControl																Range Output Control
27	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr		Range Configuration Flags
28	Range12To15[3].HiLimOrDirWr																Range High Limit or Direct Write Value
29																	
30	Range12To15[3].LowLimit																Range Low Limit
31																	
32	Range12To15[3].OutputControl																Range Output Control
33	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr		Range Configuration Flags

Output On Mask (OutputOnMask.0 through OutputOnMask.15)

Output Array Word 0	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Output On Mask	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0

This word allows you to turn on any output, real or virtual, when the corresponding bit is set. This mask is logically OR'ed with the range masks but logically AND'ed with the Output Off Mask Word described on page 4-23.

Using the Output On Mask, all of the module's outputs can be turned on directly by the user control program, like discrete outputs. A bit which is set in the mask turns on the corresponding real or virtual output.

See "Output Control" on page 2-23 and "Output Control Example" on page 2-30 for more information about output determination.

TIP The corresponding Output Off Mask bit must be set to enable this bit.

Output Off Mask (OutputOffMask.0 through OutputOffMask.15)

Output Array Word 1	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Output Off Mask	Out 15	Out 14	Out 13	Out 12	Out 11	Out 10	Out 9	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0

This word turns OFF any output, real or virtual, when the corresponding bit is reset. This mask has veto power over all the Range masks and the Output On Mask described above. It is logically AND'ed with the results of those masks. See "Output Control" on page 2-23 and "Output Control Example" on page 2-30 for more information about output determination.

TIP

This mask can be overridden when a safe state is indicated.

Range Enable (RangeEn.0 through RangeEn.15)

Output Array Word 2	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Range Enable	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

When the bit corresponding to the range number is set, Range[*n*].OutputControl is applied whenever the range is active.

RBF - Reset Blown Fuse (ResetBlownFuse)

Output Array Word 4	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reset Blown Fuse	Not Used								RBF	Not Used						

When the OvercurrentLatchOff bit is set and an overcurrent condition has occurred, the real output remains off until this bit is cycled from 0 to 1 (rising edge).

Control Bits

Output Array Words 5 to 8	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Counter 0 Control Bits (Word 5)	Not Used						RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RU	RO	SP	En
Counter 1 Control Bits (Word 6)	Not Used						RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RU	RO	SP	En
Counter 2 Control Bits (Word 7)	Not Used						RPW	Not Used				D Inv	RU	RO	SP	En
Counter 3 Control Bits (Word 8)	Not Used						RPW	Not Used				D Inv	RU	RO	SP	En

The control bits for counter (*n*) are described below.

TIP

The order of precedence for the Preset and Direct Write actions is:

1. Preset
2. Direct Write

IMPORTANT

Setting any of the control bits under certain conditions of the NumberOfCounters value will result in the input error flag, Ctr[*n*].InvalidCounter. For more information, see IC - Invalid Counter (Ctr[1].InvalidCounter to Ctr[3].Invalid Counter) on page 4-39.

*En - Enable Counter (Ctr*n*En)*

This bit, when set (1), enables the inputs to be counted. When reset (0), this bit inhibits any activity of the A or B inputs from affecting the count, pulse interval, and rate values.

*SP - Soft Preset (Ctr*n*SoftPreset)*

A 0 to 1 transition of this bit causes counter (*n*) to be preset, changing the count to the value in Ctr*n*Preset.

*RCO - Reset Counter Overflow (Ctr*n*ResetCounterOverflow)*

A 0 to 1 transition of this bit causes the corresponding Ctr[*n*]Overflow bit to be reset.

*RCU - Reset Counter Underflow (Ctr*n*ResetCounterUnderflow)*

A 0 to 1 transition of this bit causes the corresponding Ctr[*n*]Underflow bit to be reset.

*D Inv - Direction Invert (Ctr*n*DirectionInvert)*

This bit, when set, inverts the direction of the counter (*n*). If the Ctr*n*DirectionInhibit bit is set when this bit is:

- 0 – the resulting direction is up, increasing counts
- 1 – the resulting direction is down, decreasing counts

*D Inh - Direction Inhibit (Ctr*n*DirectionInhibit)*

This bit, when set, inhibits the direction of the input signal from being used by the module.

*Z Inv - Z Invert (Ctr*n*ZInvert)*

When set, this bit inverts the Z*n* value. The Z*n* value is also affected by the Ctr*n*ZInhibit bit. If the Ctr*n*ZInhibit is set, the module uses Ctr*n*ZInvert for all internal Z activities, preset, hold and store. Input state Z*n* is not affected by this bit.

*Z Inh - Z Inhibit (Ctr*n*ZInhibit)*

When set, this bit inhibits the Z*n* state from being used by the module.

*RREZ - Reset Rising Edge Z (Ctr*n*ResetRisingEdgeZ)*

A 0 to 1 transition causes the Ctr[*n*].RisingEdgeZ bit to be reset.

*RPW - Reset Counter Preset Warning (Ctr*n*ResetCtrPresetWarning)*

A 0 to 1 transition causes the Ctr[*n*]PresetWarning bit to be reset.

Range High Limit or Direct Write Value (Range12To15[*n*].HiLimOrDirWr)

Output Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
10 and 11	Range 12 High Limit Direct Write Value	Range12To15[0].HiLimOrDirWr															
16 and 17	Range 13 High Limit Direct Write Value	Range12To15[1].HiLimOrDirWr															
22 and 23	Range 14 High Limit Direct Write Value	Range12To15[2].HiLimOrDirWr															
28 and 29	Range 15 High Limit Direct Write Value	Range12To15[3].HiLimOrDirWr															

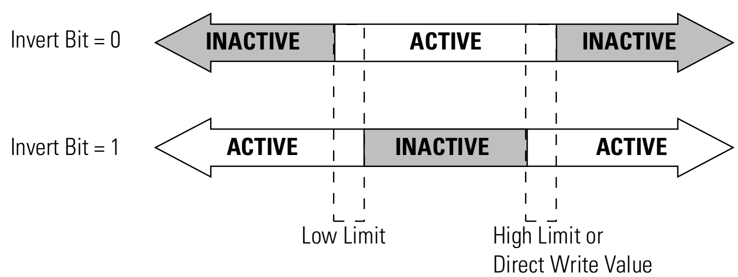
This value may be used in one of two ways, depending on the setting of the Load Direct Write (Range12To15[*n*].LoadDirectWrite) bit.

When Load Direct Write = 0

When Range12To15[*n*].LoadDirectWrite = 0, then Range12To15[*n*].HiLimOrDirWr is used in the range comparison (range represents a count value or a rate value according to the programmed range type, Range12To15[*n*].Type).

When the range value is equal to Range12To15[*n*].HiLimOrDirWr , Range*n* will change state. The range will become active or inactive depending on the Range12To15[*n*].Invert bit.

Range Value (Current Count or Current Rate)



TIP

Range12To15[*n*].HiLimOrDirWr must be higher than the Range12To15[*n*].LowLimit or the InvalidRangeLimit*n* error flag in the Input Array will be set.

TIP

Range12To15[*n*].HiLimOrDirWr may be higher than the maximum rate or count value. For example, when the object value is a rate, Range12To15[*n*].HiLimOrDirWr may be programmed in excess of 1,000,000 with no configuration error.

When Load Direct Write = 1

When Range12To15[*n*].LoadDirectWrite = 1, then Range12To15[*n*].HiLimOrDirWr is used to change the Ctr[*n*].CurrentCount to Range12To15[*n*].HiLimOrDirWr.

When the Range12To15[*n*].LoadDirectWrite bit transitions from 0 to 1, then Range12To15[*n*].HiLimOrDirWr is loaded into Ctr[*n*].CurrentCount (where *n* is the counter indicated in Range12To15[*n*].ToThisCounter).

TIP

When Ctr*n*SoftPreset and a Range12To15[*n*].LoadDirectWrite to counter *n* are indicated at the same time, only the Ctr*n*SoftPreset will occur. When more than one range indicates a Range12To15[*n*].LoadDirectWrite to a single counter, only the one from the lowest designated range will take effect.

Range Low Limit (Range12To15[*n*].LowLimit)

Output Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
12 and 13	Range 12 Low Limit	Range12To15[0].LowLimit															
18 and 19	Range 13 Low Limit	Range12To15[1].LowLimit															
24 and 25	Range 14 Low Limit	Range12To15[2].LowLimit															
30 and 31	Range 15 Low Limit	Range12To15[3].LowLimit															

This value is used in the range comparison. It is the complement of the Range12To15[*n*].HiLimOrDirWr value in setting the compare window.

When the rate or count value is equal to Range12To15[*n*].LowLimit, the range will change state – opposite of the action at Range12To15[*n*].HiLimOrDirWr. The range will become active or inactive depending on the Range12To15[*n*].Invert bit.

TIP

Range12To15[*n*].LowLimit must be lower than the Range12To15[*n*].HiLimOrDirWr or the InvalidRangeLimit*n* error flag in the Input Array will be set.

TIP

Like Range12To15[*n*].HiLimOrDirWr (see page 4-26) Range12To15[*n*].LowLimit may extend beyond the minimum rate or count value.

TIP

When Range12To15[*n*].LoadDirectWrite is set, Range12To15[*n*].LowLimit is ignored.

Range Output Control (Range12To15[*n*].OutputControl)

Output Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
14	Range 12 Output Control	Range12To15[0].OutputControl															
20	Range 13 Output Control	Range12To15[1].OutputControl															
26	Range 14 Output Control	Range12To15[2].OutputControl															
32	Range 15 Output Control	Range12To15[3].OutputControl															

This 16-bit word indicates which outputs should be on (corresponding bit set in this word) when a range is active. When Range*n* is enabled and active, Range12To15[*n*].OutputControl will be logically OR'ed with other Range12To15[*n*].OutputControl masks and the OutputOnMask.*n* etc., as described on page 4-22.

When Range12To15[*n*].LoadDirectWrite is set, Range12To15[*n*].OutputControl is ignored.

Range Configuration Flags (12 through 15)

Output Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
15	Range 12 Configuration Flags	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr	
21	Range 13 Configuration Flags	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr	
27	Range 14 Configuration Flags	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr	
33	Range 15 Configuration Flags	Not Used							Inv	Not Used		LDW	Type	Not Used		ToThisCtr	

ToThisCtr - Range Counter Number (Range12To15[n].ToThisCounter)

This 2-bit value indicates which counter will be used in the range comparison or Range12To15[*n*].LoadDirectWrite. The counter is indicated as follows:

Table 4.8 Range Counter Number Determination

Bit 01	Bit 00	Counter
0	0	0
0	1	1
1	0	2
1	1	3

If Range12To15[*n*].ToThisCounter is set to a number larger than NumberOfCounters in the Configuration Array, then the InvalidCtrAssignToRange*n* error bit in the Input Array will be set.

Type - RangeType (Range12To15[n].Type)

This bit value indicates which type of value to use for the range comparison in Range. That is, the Range12To15[*n*].ToThisCounter, from above, and this Range12To15[*n*].Type value determine the rate or count value, the current value which is compared to, for the range comparison. The type of value is indicated as follows:

- 0 = Count Value
- 1 = Rate Value

When Range12To15[*n*].LoadDirectWrite is set Range12To15[*n*].Type is ignored.

LDW - Load Direct Write (Range12To15[n].LoadDirectWrite)

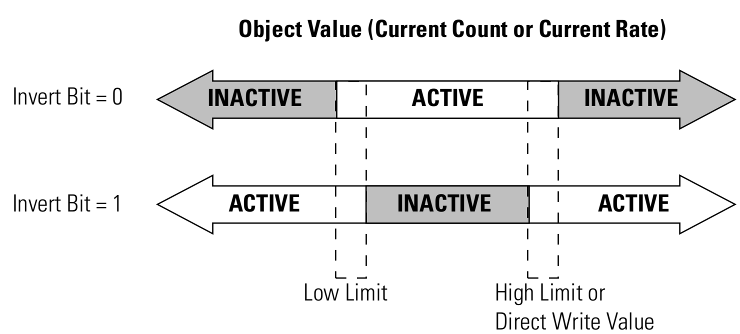
A 0 to 1 transition of this bit causes counter (*n*)'s current count value to change to the value of Range12To15[*n*].HiLimOrDirWr.

IMPORTANT The write occurs according to the internal timings of the module and the system. For the most predictable results, the counter should be disabled or stopped while performing this action.

IMPORTANT If both Ctr*n*SoftPreset and Range12To15[*n*].HiLimOrDirWr transition to 1 during the same Output Array update, only the Ctr*n*SoftPreset occurs. Range12To15[*n*].HiLimOrDirWr is ignored.

Inv - Range Invert (Range12To15[n].Invert)

Indicates the active portion of Range*n*. When Range12To15[*n*].Invert = 0, the outputs are activated when the range value is at or between the Range12To15[*n*].LowLimit and Range12To15[*n*].HiLimOrDirWr. When Range12To15[*n*].Invert = 1, the outputs are activated when the range is at or outside the range limits.



Input Array

The Input Array, which consists of 35 words, allows read-only access to the module's input data via word and bit access.The Input Array is described below. The functions are described in more detail in the sections following the table.

IMPORTANT During the non-run states (program and fault), the module continues to update the Input Array (continues counting, etc). Depending on the bus master, you may not see this.

TIP Status bits for a particular counter reflect the configuration settings for that counter. To receive valid status, the counter must be enabled and the module must have stored a valid configuration for that counter.

Table 4.9 Input Array

Word	Bit 15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Function
0	Not Used										Z1	B1	A1	Z0	B0	A0	Input State
1	Readback.0 through Readback.15																Readback
2	InvalidRangeLimit12 through InvalidRangeLimit15				InvalidCtrAssignToRange12 through InvalidCtrAssignToRange15				Gen Error	Invalid Output	Mod Config	Not Used	Out0Overcurrent through Out3Overcurrent				Status Flags
3	RangeActive.0 through RangeActive.15																Range Active
4	Ctr[0].CurrentCount																Counter 0 Current Count
5																	
6	Ctr[0].StoredCount																Counter 0 Stored Count
7																	
8	Ctr[0].CurrentRate																Counter 0 Current Rate
9																	
10	Ctr[0].PulseInterval																Counter 0 Pulse Interval
11																	
12	Not Used									C0PW	RV	Not Used	IDW	REZ	CUdf	COvf	Counter 0 Status Flags
13	Not Used																Not Used
14	Ctr[1].CurrentCount																Counter 1 Current Count
15																	
16	Ctr[1].StoredCount																Counter 1 Stored Count
17																	
18	Ctr[1].CurrentRate																Counter 1 Current Rate
19																	
20	Ctr[1].PulseInterval																Counter 1 Pulse Interval
21																	
22	Not Used									C1PW	RV	IC	IDW	REZ	CUdf	COvf	Counter 1 Status Flags
23	Not Used																Not Used
24	Ctr[2].CurrentCount																Counter 2 Current Count
25																	
26	Ctr[2].CurrentRate																Counter 2 Current Rate
27																	
28	Not Used									C2PW	RV	IC	IDW	Not Used	CUdf	COvf	Counter 2 Status Flags
29	Not Used																Not Used
30	Ctr[3].CurrentCount																Counter 3 Current Count
31																	
32	Ctr[3].CurrentRate																Counter 3 Current Rate
33																	
34	Not Used									C3PW	RV	IC	IDW	Not Used	CUdf	COvf	Counter 3 Status Flags

Input State (InputStateA0 through InputStateZ1)

Input Array Word 0	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Input State	Not Used										Z1	B1	A1	Z0	B0	A0

This word indicates the state of the real (physical) inputs after filtering.

- 1 = On
- 0 = Off

Readback (Readback.0 through Readback.15)

Input Array Word 1	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Readback	Readback.0 through Readback.15															

This input word reflects counter's module-directed status of all sixteen outputs, real and virtual.

- 1 = On
- 0 = Off

Status Flags

Input Array Word 2	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Status Flags	InvalidRangeLimit12 through InvalidRangeLimit15				InvalidCtrAssignToRange12 through InvalidCtrAssignToRange15				Gen Error	Invalid Output	Mod Config	Not Used	Out0Overcurrent through Out3Overcurrent			

Output Overcurrent (Out0Overcurrent to Out3OverCurrent)

The output overcurrent bits are set (1) when the module is in an overcurrent condition. These bits also show whether the output is latched off, because the output(s) remain in the off state and these bits remain on until the ResetBlownFuse bit is used.

Module Configured (ModConfig)

Word 2, bit 5 is set by the module after it has accepted all of the configuration data. When set (1), this bit confirms that the module received and accepted valid configuration data. When reset (0), this bit indicates that the module is still checking for errors or contains errors and the old configuration is still being used.

TIP The module takes up to 2 seconds to validate configuration data.

Invalid Output (InvalidOutput)

- 1 = an unused bit in the Output Array is set
- 0 = no unused bits in the Output Array are set

When this error occurs, the entire Output Array is rejected until an Output Array that does not have this error is sent.

Error (GenError)

When this bit is set (1), it indicates one or more of the following errors for the Input Array:

- Out*n*Overcurrent
- InvalidRangeLimit*n*
- InvalidCtrAssignToRange*n*
- InvalidOutput
- Ctr[*n*].Overflow
- Ctr[*n*].Underflow
- Ctr[*n*].InvalidDirectWrite
- Ctr[*n*].InvalidCounter
- Ctr[*n*].PresetWarning

where *n* indicates the counter number.

To determine which error has set the GenError bit, identify which bit is set. This could be done by using a subroutine to examine these bits in the Input Array.

TIP Ctr[*n*].RateValid does not set the GenError bit.

Invalid Counter Assigned to Range (InvalidCtrAssignToRange12 through InvalidCtrAssignToRange15)

InvalidCtrAssignToRange12 is set when the indicated range in the Output Array refers to a non-existent counter.

- It is set (1) when Range12To15[*n*].ToThisCounter > NumberOfCounters.
- It is cleared (0) when Range12To15[*n*].ToThisCounter ≤ NumberOfCounters.

When this error occurs, the entire Output Array is rejected until a valid configuration is detected.

Invalid Range Limit (InvalidRangeLimit12 through InvalidRangeLimit15)

This bit is set when the range limits are invalid according to the limitations indicated in Range12To15[*n*].HiLimOrDirWr and Range12To15[*n*].LowLimit in the Output Array.

- 1 = Range limits are invalid.
- 0 = no error

When this error occurs, the entire Output Array is rejected until a valid configuration is detected.

Range Active (RangeActive.0 through RangeActive.15)

Input Array Word 3	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Range Active	RangeActive.0 through RangeActive.15															

This word reflects the status of all of the ranges. When a count or rate meets the criteria programmed for a given range, the range is active.

- 1 = active
- 0 = inactive/false

TIP When the range is enabled and active, the output mask for that range is applied.

Current Count (Ctr[*n*].CurrentCount)

Input Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
4	Counter 0 Current Count	Ctr[0].CurrentCount															
5																	
14	Counter 1 Current Count	Ctr[1].CurrentCount															
15																	
24	Counter 2 Current Count	Ctr[2].CurrentCount															
25																	
30	Counter 3 Current Count	Ctr[3].CurrentCount															
31																	

This is the 32-bit count value from the counter.

Stored Count (Ctr[*n*].StoredCount)

Input Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
6	Counter 0 Stored Count	Ctr[0].StoredCount															
7																	
16	Counter 1 Stored Count	Ctr[1].StoredCount															
17																	

This is the last stored 32-bit value from counter (*n*). The count value is stored depending on the Ctr*n*Config.StorageMode and Z*n* inputs.

When a storage event occurs, the Ctr[*n*].RisingEdgeZ bit is set, indicating that the value is new. If more than one Z*n* occurs before the Ctr[*n*].RisingEdgeZ bit is reset (using the Ctr*n*ResetRisingEdgeZ bit), the Ctr[*n*].StoredCount word will contain only the last Ctr[*n*].StoredCount value. There is no indication that the data has been overwritten.

Current Rate (Ctr[0].CurrentRate to Ctr[3].CurrentRate)

Input Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
8	Counter 0 Current Rate	Ctr[0].CurrentRate															
9																	
18	Counter 1 Current Rate	Ctr[1].CurrentRate															
19																	
26	Counter 2 Current Rate	Ctr[2].CurrentRate															
27																	
32	Counter 3 Current Rate	Ctr[3].CurrentRate															
33																	

This 32-bit value is the current rate value, scaled by Ctr*n*Scalar, from the counter. This uses the Cyclic Rate Calculation Method (see page 2-19 for more information).

Rate-based ranges use this value for comparisons, even when the Ctr[*n*].RateValid bit is zero.

IMPORTANT	This value is only current when the Ctr[*n*].RateValid bit is set (1).

Pulse Interval (Ctr[0].PulseInterval and Ctr[1].PulseInterval)

Input Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
10	Counter 0 Pulse Interval	Ctr[0].PulseInterval															
11																	
20	Counter 1 Pulse Interval	Ctr[1].PulseInterval															
21																	

This is the time, in microseconds, between the last two pulses for the counter. The pulses indicated here are those transitions on which the count value can change. For example, in quadrature X1 mode, these are the successive rising edges of A only.

If more than two pulses have occurred since the value was last read, the value indicates only the time between the *last two pulses* that have been processed.

Status Flags

Input Array Words		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
12	Counter 0 Status Flags	Not Used									C0PW	RV	Not Used	IDW	REZ	CUdf	COvf
22	Counter 1 Status Flags	Not Used									C1PW	RV	IC	IDW	REZ	CUdf	COvf
28	Counter 2 Status Flags	Not Used									C2PW	RV	IC	IDW	Not Used	CUdf	COvf
34	Counter 3 Status Flags	Not Used									C3PW	RV	IC	IDW		CUdf	COvf

The status bits for the counter (*n*) are described below.

COvf - Count Overflow (Ctr[0].Overflow to Ctr[3].Overflow)

For linear counters, this bit is set when the counter is, or has been, in an overflow condition. For ring counters, this bit is set when the counter has rolled over. COvf is reset when the Ctr*n*ResetCountOverflow bit transitions from 0 to 1.

See Counter Types on page 2-14 for more information about linear and ring counters.

CUdf - Count Underflow (Ctr[0].Underflow to Ctr[3].Underflow)

For linear counters, this bit is set when the counter is, or has been, in an underflow condition. For ring counters, this bit is set when the counter has rolled under. CUdf is reset when the Ctr*n*ResetCountUnderflow bit transitions from 0 to 1.

See Counter Types on page 2-14 for more information about linear and ring counters.

REZ - Rising Edge Z (Ctr[0].RisingEdgeZ to Ctr[1].RisingEdgeZ)

This bit is set (1) when Z*n*, as modified by the Ctr*n*ZInvert and Ctr*n*ZInhibit bits, has a rising edge. It is reset (0) by a 0 to 1 transition of the Ctr*n*ResetRisingEdgeZ bit in the Output Array. *N* is equal to 0 or 1 depending upon which input is used, Z0 or Z1.

IDW - Invalid Direct Write (Ctr[0].InvalidDirectWrite to Ctr[3].InvalidDirectWrite)

This bit is set when the Range12To15[*n*].HiLimOrDirWr is invalid. (For example, if Ctr*n*MaxCount < Range12To15[*n*].HiLimOrDirWr or Range12To15[*n*].HiLimOrDirWr < Ctr*n*MinCount.)

When this error occurs, the entire Output Array is rejected until a valid configuration is detected.

IC - Invalid Counter (Ctr[1].InvalidCounter to Ctr[3].Invalid Counter)

When set (1) this bit indicates that an invalid control bit is set for the counter. Depending on the value of NumberOfCounters, the following errors will occur:

- If NumberOfCounters < 1, then setting one of the control bits for Counter 1 will result in input error flag Ctr[1].InvalidCounter.
- If NumberOfCounters < 2, then setting one of the control bits for Counter 2 will result in input error flag Ctr[2].InvalidCounter.
- If NumberOfCounters < 3, then setting one of the control bits for Counter 3 will result in input error flag Ctr[3].InvalidCounter.

When this error occurs, the entire Output Array is rejected until an Output Array that does not have this error is sent.

The control bits are shown on page 4-24.

RV - Rate Valid (Ctr[0].RateValid to Ctr[3].RateValid)

This bit is set (1) when the rate value indicated in Ctr[*n*].CurrentRate is current. When this bit is reset (0), Ctr[*n*].CurrentRate is frozen at the last known good value.

This bit is reset when the Ctr[*n*].Overflow or Ctr[*n*].Underflow bits have been set during the last Ctr*n*CyclicRateUpdateTime period. See page 2-21 for more Rate Valid reset conditions.

CnPW - Counter Preset Warning (Ctr[0].PresetWarning to Ctr[3].PresetWarning)

This bit is set when Ctr[*n*].CurrentCount has been forced, by the module, to the Ctr*n*Preset value. This will happen when a Configuration Array is accepted which sets:

Ctr*n*MinCount > Ctr[*n*].CurrentCount, or

Ctr*n*MaxCount < Ctr[*n*].CurrentCount.

This bit is reset by a 0 to 1 transition of the Ctr*n*ResetCtrPresetWarning bit in the Output Array.

TIP

You must manually reset C*n*PW, COvf, CUdf and REZ (but not IDW, RV or IC) to enable them to be set again.

Chapter 5

Diagnostics and Troubleshooting

This chapter describes troubleshooting the module. This chapter contains information on:

- safety considerations when troubleshooting
- module vs. counter operation
- the module's diagnostic features
- critical vs. non-critical errors
- error codes

Safety Considerations

Safety considerations are an important element of proper troubleshooting procedures. Actively thinking about the safety of yourself and others, as well as the condition of your equipment, is of primary importance.

The following sections describe several safety concerns you should be aware of when troubleshooting your control system.

ATTENTION

Never reach into a machine to actuate a switch because unexpected motion can occur and cause injury.

Remove all electrical power at the main power disconnect switches before checking electrical connections or inputs/outputs causing machine motion.

Indicator Lights

When any LED on the module is illuminated, it indicates that power is applied to the module.

Stand Clear of the Machine

When troubleshooting any system problem, have all personnel remain clear of the machine. The problem could be intermittent, and sudden unexpected machine motion could occur. Have someone ready to operate an emergency stop switch in case it becomes necessary to shut off power to the machine.

Program Alteration

There are several possible causes of alteration to the user program, including extreme environmental conditions, Electromagnetic Interference (EMI), improper grounding, improper wiring connections, and unauthorized tampering. If you suspect a program has been altered, check it against a previously saved program on an EEPROM or UVPROM memory module.

Safety Circuits

Circuits installed on the machine for safety reasons, like over-travel limit switches, stop push buttons, and interlocks, should always be hard-wired to the master control relay. These devices must be wired in series so that when any one device opens, the master control relay is de-energized, thereby removing power to the machine. Never alter these circuits to defeat their function. Serious injury or machine damage could result.

Module Operation vs. Counter Operation

The module performs operations at two levels:

- module level
- counter level

Module-level operations include functions such as power-up, configuration, and communication with a bus master, such as a MicroLogix 1500 controller.

Counter-level operations include counter-related functions, such as data conversion and overflow or underflow detection.

Internal diagnostics are performed at both levels of operation. When detected, module error conditions are immediately indicated by the module status LED. Both module hardware and configuration error conditions are reported to the controller. Counter overflow or underflow conditions are reported in the module's input data table. Module hardware errors are typically reported in the controller's I/O status file. Refer to your controller manual for details.

Counter Defaults

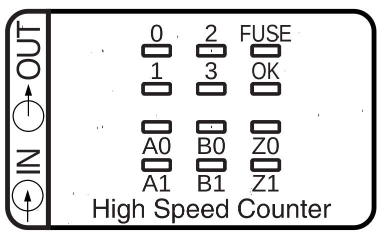
When the module powers-up, all Output Array and Configuration Array values are set to their default values (see Chapter 4 or Appendix D for default values). All Input Array values are cleared. None of the module data is retentive through a power cycle.

In effect, this means that power cycling clears the module:

- stored counts are lost
- faults and flags are cleared
- outputs are off

The bus master will attempt to write program data to the Output Array and Configuration Array.

Module Diagnostics



Power-up Diagnostics

At module power-up, a series of internal diagnostic tests are performed. These diagnostic tests must be successfully completed or the OK LED remains off and a module error results and is reported to the controller.

Table 5.1 Diagnostic Indicators

LED	Color	Indicates
0 OUT	Amber	ON/OFF logic status of output 0
1 OUT	Amber	ON/OFF logic status of output 1
2 OUT	Amber	ON/OFF logic status of output 2
3 OUT	Amber	ON/OFF logic status of output 3
FUSE	Red	Overcurrent
OK	Off	No power is applied.
	Red (briefly)	Performing self-test.
	Solid Green	OK, normal operating condition.
	Flashing Green	OK, module in Program or Fault mode.
	Solid Red or Amber	Hardware error. Cycle power to the module. If problem persists, replace the module.
	Flashing Red	Recoverable fault. Reconfigure, reset, or perform error recovery. See section on page 5-5, Non-Critical vs. Critical Module Errors. The OK LED flashes red for all of the error codes in Table 5.6.
A0	Amber	ON/OFF status of input A0
A1	Amber	ON/OFF status of input A1
B0	Amber	ON/OFF status of input B0
B1	Amber	ON/OFF status of input B1
Z0	Amber	ON/OFF status of input Z0
Z1	Amber	ON/OFF status of input Z1
ALL ON	Possible causes for all LEDs to be on: • Bus Error has occurred: Controller hard fault. Cycle power. • During Flash Upgrade of Controller: Normal. Do not cycle power during the Flash Upgrade. • All LEDs will flash on briefly during power-up. This is normal.	

Configuration Diagnostics

When a configuration is sent, the module performs a diagnostic check to see that the configuration is valid. This results in either a valid ModConfig bit or module configuration error. See Table 5.6 for configuration error codes.

Post Configuration Diagnostics

If the ModConfig bit in the input array is set, then the module has accepted the configuration. Now, on every scan, each channel status flag in the Input Array is examined. The Output Array is checked on each scan for compatibility with the Configuration Array.

Non-Critical vs. Critical Module Errors

Non-Critical Errors

Non-critical module errors are typically recoverable. Non-critical error conditions are indicated by the extended error code. See Table 5.6 Configuration Error Codes on page 5-9.

TIP The OK LED will be in a flashing red state for all of the error codes in Table 5.6.

Critical Errors

Critical module errors are conditions that prevent normal or recoverable operation of the system. When these types of errors occur, the system typically leaves the run or program mode and enters the fault mode of operation until the error can be dealt with. Critical module errors are indicated in Table 5.5 General Common Hardware Error Codes on page 5-8.

Module Error Definition

Module errors are expressed in two fields as four-digit Hex format with the most significant digit as “don’t care” and irrelevant. The two fields are “Module Error” and “Extended Error Information”. The structure of the module error data is shown below.

Table 5.2 Module Error Definition

“Don’t Care” Bits				Module Error			Extended Error Information								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hex Digit 4				**Hex Digit 3**				**Hex Digit 2**				**Hex Digit 1**			

Module Error Field

The purpose of the module error field is to classify module errors into three distinct groups, as described in the table below. The type of error determines what kind of information exists in the extended error information field. These types of module errors are typically reported in the controller's I/O status file. Refer to your controller manual for details.

Table 5.3 Module Error Types

Error Type	Module Error Field Value Bits 11 through 09 (Binary)	Description
No Errors	000	No error is present. The extended error field holds no additional information.
Hardware Errors	001	General and specific hardware error codes are specified in the extended error information field.
Configuration Errors	010	Module-specific error codes are indicated in the extended error field. These error codes correspond to options that you can change directly. For example, the input range or input filter selection.

Extended Error Information Field

Check the extended error information field when a non-zero value is present in the module error field. Depending upon the value in the module error field, the extended error information field can contain error codes that are module-specific or common to all 1769 modules.

TIP

If no errors are present in the module error field, the extended error information field will be set to zero.

Hardware Errors

General or module-specific hardware errors are indicated by module error code 1. See Table 5.5 General Common Hardware Error Codes on page 5-8.

Configuration Errors

If you set the fields in the configuration file to invalid or unsupported values, the module ignores the invalid configuration, generates a non-critical error, and keeps operating with the previous configuration.

Table 5.6 Configuration Error Codes on page 5-9 lists the possible module-specific configuration error codes defined for the module. Correct the error by providing proper configuration data to the module.

Table 5.4 describes configuration errors in more general terms.

Table 5.4 Error Conditions by Type of Configuration

Programming Words	Error Conditions
General Configuration Bits, Filters and Safe State Words	• Unused or Reserved bit(s) were set. • A counter or counters were running when the general configuration bits or filter and safe state words were sent.
Counter Configuration	• Unused or Reserved bit(s) were set. • Operational Mode is invalid for the counter. (NumberOfCounters may be incorrect.) • Operational Mode is invalid for the counter. (mode selection may be incorrect). • The selected counter was running when the configuration was sent. • Ctr*n*MaxCount ≤ Ctr*n*MinCount • Ctr*n*Hysteresis < 0 • Ctr*n*Scalar < 1 • Ctr*n*CyclicRateUpdateTime < 1 • The preset value is outside its valid range. (Ctr*n*Preset not equal to or between Ctr*n*MinCount or Ctr*n*MaxCount) • Counter was running when the minimum/maximum count value was changed.
Range Configuration	• Unused or Reserved bit(s) were set. • Range0to11[*n*].HighLimit ≤ Range0to11[*n*].LowLimit • Range0To11[*n*].ToThisCounter refers to a non-declared counter (Range0To11[*n*].ToThisCounter > NumberOfCounters)

Error Codes

The tables in this section explain the extended error codes for general common hardware errors, configuration errors, and runtime errors.

Table 5.5 General Common Hardware Error Codes

Error Type	Hex Equivalent(1)	Module Error Code	Extended Error Information Code	Description	Status of the OK LED(2)
		Binary	Binary		
No Error	X000	000	0 0000 0000	OK, normal operating condition.	Solid or flashing green
General Common Hardware Error	X200	001	0 0000 0000	General hardware error; no additional information	Solid red
	X201	001	0 0000 0001	Power-up reset state	Briefly red
	X202	001	0 0000 0010	Bus master incompatibility	Solid red
	X203	001	0 0000 0011	General hardware error	Solid red
	X20A	001	0 0000 1010	General microprocessor error	Solid red
	X20B	001	0 0000 1011	Microprocessor internal register error	Solid red
	X20C	001	0 0000 1100	Microprocessor special function register error	Solid red
	X20D	001	0 0000 1101	Microprocessor internal memory error	Solid red
	X20E	001	0 0000 1110	Microprocessor timer error	Solid red
	X20F	001	0 0000 1111	Microprocessor interrupt error	Solid red
	X210	001	0 0001 0000	Microprocessor watchdog error	Solid red
	X218	001	0 0001 1000	Firmware corrupt	Solid red
	X219	001	0 0001 1001	Firmware checksum error in non-volatile RAM	Solid red
	X21A	001	0 0001 1010	Firmware checksum error in RAM	Solid red
	X21E	001	0 0001 1110	External RAM test error	Solid red
	X21F	001	0 0001 1111	External RAM cell test error	Solid red
	X224	001	0 0010 0100	Gate array loading failed	Solid red
	X232	001	0 0011 0010	External watchdog error	Solid red

(1) X represents the “Don’t Care” digit.

(2) See Table 5.1 on page 5-4 for recommendation based on LED operation.

TIP The OK LED will be in a flashing red state for all of the error codes in Table 5.6.

Table 5.6 Configuration Error Codes

Hex Equivalent(1)	Module Error Code Binary	Extended Error Information Code Binary	Error	Description
X400	010	0 0000 0000	General Configuration Error	no additional information
X401	010	0 0000 0001	UnusedConfigBitSet	One or more of the unused module configuration bits are set.
X402	010	0 0000 0010	BadModConfigUpdate	Occurs when you attempt to change a forbidden module configuration parameter while a counter or range is still enabled. See Table 5.7 on page 5-13 for a list of the forbidden parameters.
X411	010	0 0001 0001	BadCounterNum_1	Nonzero configuration values were entered for Counter 1, when Counter 1 was not available.
X412	010	0 0001 0010	BadCounterNum_2	Nonzero configuration values were entered for Counter 2, when Counter 2 was not available.
X413	010	0 0001 0011	BadCounterNum_3	Nonzero configuration values were entered for Counter 3, when Counter 3 was not available.
X420	010	0 0010 0000	BadCounterMode_0	Operation Mode_0 is set to an invalid value. For example, value is reserved (011 or 111) or nonzero when NumberofCounters = 11.
X421	010	0 0010 0001	BadCounterMode_1	Operation Mode_1 is set to an invalid value. For example, value is reserved (011 or 111) or nonzero when NumberofCounters = 10 or 11.
X430	010	0 0011 0000	BadMin_0	Programmed Ctr0MinCount is greater than the Ctr0MaxCount.
X431	010	0 0011 0001	BadMin_1	Programmed Ctr1MinCount is greater than the Ctr1MinCount .
X432	010	0 0011 0010	BadMin_2	Programmed Ctr2MinCount is greater than the Ctr2MaxCount.
X433	010	0 0011 0011	BadMin_3	Programmed Ctr3MinCount is greater than the Ctr3MaxCount.
X440	010	0 0100 0000	BadPreset_0	The programmed Ctr0Preset is greater than the Ctr0MaxCount or less than the Ctr0MinCount.
X441	010	0 0100 0001	BadPreset_1	The programmed Ctr1Preset is greater than the Ctr1MaxCount or less than the Ctr1MinCount.
X442	010	0 0100 0010	BadPreset_2	The programmed Ctr2Preset is greater than the Ctr2MaxCount or less than the Ctr2MinCount.
X443	010	0 0100 0011	BadPreset_3	The programmed Ctr3Preset is greater than the Ctr3MaxCount or less than the Ctr3MinCount.

Table 5.6 Configuration Error Codes

Hex Equivalent[1]	Module Error Code Binary	Extended Error Information Code Binary	Error	Description
X450	010	0 0101 0000	BadHysteresis_0	The Ctr0Hysteresis value is invalid, i.e. less than zero.
X451	010	0 0101 0001	BadHysteresis_1	The Ctr1Hysteresis value is invalid, i.e. less than zero.
X452	010	0 0101 0010	BadHysteresis_2	The Ctr2Hysteresis value is invalid, i.e. less than zero.
X453	010	0 0101 0011	BadHysteresis_3	The Ctr3Hysteresis value is invalid, i.e. less than zero.
X460	010	0 0110 0000	BadScalar_0	The Ctr0Scalar value is invalid, i.e. less than one.
X461	010	0 0110 0001	BadScalar_1	The Ctr1Scalar value is invalid, i.e. less than one when NumberofCounters = 01, 10 or 11.
X462	010	0 0110 0010	BadScalar_2	The Ctr2Scalar value is invalid, i.e. less than one when NumberofCounters = 10 or 11.
X463	010	0 0110 0011	BadScalar_3	The Ctr3Scalar value is invalid, i.e. less than one when NumberofCounters = 11.
X470	010	0 0111 0000	BadScale_0	The Ctr0CyclicRateUpdateTime is invalid, i.e. less than one.
X471	010	0 0111 0001	BadScale_1	The Ctr1CyclicRateUpdateTime is invalid, i.e. less than one when NumberofCounters = 01, 10 or 11.
X472	010	0 0111 0010	BadScale_2	The Ctr2CyclicRateUpdateTime is invalid, i.e. less than one when NumberofCounters = 10 or 11.
X473	010	0 0111 0011	BadScale_3	The Ctr3CyclicRateUpdateTime is invalid, i.e. less than one when NumberofCounters = 11.
X480	010	0 1000 0000	BadRangeLimit_0	The Range0To11[0].LowLimit is greater than or equal to the Range0To11[0].HighLimit.
X481	010	0 1000 0001	BadRangeLimit_1	The Range0To11[1].LowLimit is greater than or equal to the Range0To11[1].HighLimit.
X482	010	0 1000 0010	BadRangeLimit_2	The Range0To11[2].LowLimit is greater than or equal to the Range0To11[2].HighLimit.
X483	010	0 1000 0011	BadRangeLimit_3	The Range0To11[3].LowLimit is greater than or equal to the Range0To11[3].HighLimit.
X484	010	0 1000 0100	BadRangeLimit_4	The Range0To11[4].LowLimit is greater than or equal to the Range0To11[4].HighLimit.
X485	010	0 1000 0101	BadRangeLimit_5	The Range0To11[5].LowLimit is greater than or equal to the Range0To11[5].HighLimit.
X486	010	0 1000 0110	BadRangeLimit_6	The Range0To11[6].LowLimit is greater than or equal to the Range0To11[6].HighLimit.
X487	010	0 1000 0111	BadRangeLimit_7	The Range0To11[7].LowLimit is greater than or equal to the Range0To11[7].HighLimit.
X488	010	0 1000 1000	BadRangeLimit_8	The Range0To11[8].LowLimit is greater than or equal to the Range0To11[8].HighLimit.
X489	010	0 1000 1001	BadRangeLimit_9	The Range0To11[9].LowLimit is greater than or equal to the Range0To11[9].HighLimit.
X48A	010	0 1000 1010	BadRangeLimit_10	The Range0To11[10].LowLimit is greater than or equal to the Range0To11[10].HighLimit.
X48B	010	0 1000 1011	BadRangeLimit_11	The Range0To11[11].LowLimit is greater than or equal to the Range0To11[11].HighLimit.

Table 5.6 Configuration Error Codes

Hex Equivalent[1]	Module Error Code Binary	Extended Error Information Code Binary	Error	Description
X490	010	0 1001 0000	BadCtrAssignToRange_0	This error occurs if you try to set Range0To11[0].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X491	010	0 1001 0001	BadCtrAssignToRange_1	This error occurs if you try to set Range0To11[1].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X492	010	0 1001 0010	BadCtrAssignToRange_2	This error occurs if you try to set Range0To11[2].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X493	010	0 1001 0011	BadCtrAssignToRange_3	This error occurs if you try to set Range0To11[3].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X494	010	0 1001 0100	BadCtrAssignToRange_4	This error occurs if you try to set Range0To11[4].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X495	010	0 1001 0101	BadCtrAssignToRange_5	This error occurs if you try to set Range0To11[5].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X496	010	0 1001 0110	BadCtrAssignToRange_6	This error occurs if you try to set Range0To11[6].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X497	010	0 1001 0111	BadCtrAssignToRange_7	This error occurs if you try to set Range0To11[7].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X498	010	0 1001 1000	BadCtrAssignToRange_8	This error occurs if you try to set Range0To11[8].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).

Table 5.6 Configuration Error Codes

Hex Equivalent[1]	Module Error Code Binary	Extended Error Information Code Binary	Error	Description
X499	010	0 1001 1001	BadCtrAssignToRange_9	This error occurs if you try to set Range0To11[9].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X49A	010	0 1001 1010	BadCtrAssignToRange_10	This error occurs if you try to set Range0To11[10].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).
X49B	010	0 1001 1011	BadCtrAssignToRange_11	This error occurs if you try to set Range0To11[11].ToThisCounter to an invalid value (i.e. to a counter that is not available due to the number of counters selected).

(1) X represents the "Don't Care" digit.

The BadModConfigUpdate error conditions are shown in the following table. They occur when you attempt to change a forbidden module configuration parameter while a counter or range is still enabled. To recover from this situation:

- correct the configuration problem
- reconfigure the module[1]

(1) Refer to your controller's documentation for available reconfiguration methods.

Table 5.7 "BadModConfigUpdate" Error Prohibited Configuration Settings - *Do not set while counter or range is enabled.*

Configuration Parameter	Array Position		Prohibited from changing when indicated (•) bits are set:				
	Word	**Bit**	**Ctr0En**	**Ctr1En**	**Ctr2En**	**Ctr3En**	**RangeEn**
OverCurrentLatchOff	0	0	•	•	•	•	•
ProgToFaultEn	0	4					
NumberOfCounters	0	8 and 9	•	•	•	•	•
Filter_A0	1	0 and 1	•			•	
Filter_B0	1	3 and 4	•			•	
Filter_Z0	1	6 and 7	•			•	
Filter_A1	1	8 and 9		•	•		
Filter_B1	1	11 and 12		•	•		
Filter_Z1	1	14 and 15		•	•		
Out*n*ProgramMode	2	0 to 3					
Out*n*ProgramStateRun	2	4 to 7					
Out*n*ProgramValue	3	0 to 3					
Out*n*0FaultMode	4	0 to 3					
Out*n*FaultStateRun	4	4 to 7					
Out*n*FaultValue	5	0 to 3					
Ctr0MaxCount	6 and 7	--	•				
Ctr0MinCount	8 and 9	--	•				
Ctr0Preset(1)	10 and 11	--	(1)				
Ctr0Hysteresis	12	--	•				
Ctr0Scalar	13	--	•				
Ctr0CyclicRateUpdateTime	14	--	•				
Ctr0Config.OperationalMode	15	0 to 3	•				
Ctr0Config.StorageMode	15	8 to 10	•				
Ctr0Config.Linear	15	12	•				
Ctr1MaxCount	16 and 17	--		•			
Ctr1MinCount	18 and 19	--		•			
Ctr1Preset(1)	20 and 21	--		(1)			
Ctr1Hysteresis	22	--		•			
Ctr1Scalar	23	--		•			
Ctr1CyclicRateUpdateTime	24	--		•			
Ctr1Config.OperationalMode	25	0 to 3		•			
Ctr1Config.StorageMode	25	8 to 10		•			
Ctr1Config.Linear	25	12		•			
Ctr2MaxCount	26 and 27	--			•		
Ctr2MinCount	28 and 29	--			•		
Ctr2Preset(1)	30 and 31	--			(1)		
Ctr2Hysteresis	32	--			•		
Ctr2Scalar	33	--			•		
Ctr2CyclicRateUpdateTime	34	--			•		
Ctr2Config.Linear	35	12			•		
Ctr3MaxCount	36 and 37	--				•	
Ctr3MinCount	38 and 39	--				•	
Ctr3Preset(1)	40 and 41	--				(1)	
Ctr3Hysteresis	42	--				•	
Ctr3Scalar	43	--				•	
Ctr3CyclicRateUpdateTime	44	--				•	
Ctr3Config.Linear	45	12				•	
Ranges	46 to 117	--	can be changed while counters and ranges are enabled				

(1) Ctr*n*Preset can be changed while Ctr*n*En = 1.

Contacting Rockwell Automation

If you need to contact Rockwell Automation for assistance, please have the following information available when you call:

- a clear statement of the problem, including a description of what the system is actually doing. Note the LED state; also note input and output image words for the module.
- a list of remedies you have already tried
- processor type and firmware number (See the label on the processor.)
- hardware types in the system, including all I/O modules
- fault code if the processor is faulted

Then contact your local Allen-Bradley distributor or Rockwell Automation Technical Support.

Technical Support contact information:

- phone - **440-646-5800**
- internet - **http://support.rockwellautomation.com**

Specifications

General Specifications

Specification	Value
Dimensions	118 mm (height) x 87 mm (depth) x 35 mm (width), height including mounting tabs is 138 mm 4.65 in. (height) x 3.43 in (depth) x 1.38 in (width), height including mounting tabs is 5.43 in.
Approximate Shipping Weight (with carton)	309g (0.681 lbs.)
Bus Current Draw (max.)	425 mA at 5V dc 0 mA at 24V dc
Heat Dissipation	6.21 Total Watts (The Watts per point, plus the minimum Watts, with all points energized.)
Storage Temperature	-40°C to +85°C (-40°F to +185°F)
Operating Temperature	0°C to +60°C (32°F to +140°F)
Operating Humidity	5% to 95% non-condensing
Operating Altitude	2000 meters (6561 feet)
Vibration	Operating: 10 to 500 Hz, 5G, 0.030 in. peak-to-peak Relay Operation: 2G(1)
Shock	Operating: 30G, 11 ms panel mounted (20G, 11 ms DIN rail mounted) Non-Operating: 40G panel mounted (30G DIN rail mounted)
System Power Supply Distance Rating	4 (The module may not be more than 4 modules away from a system power supply.)
Recommended Cable	individually shielded, twisted-pair cable (or the type recommended by the encoder or sensor manufacturer)
Agency Certification	• C-UL certified (under CSA C22.2 No. 142) • UL 508 listed • CE compliant for all applicable directives
Hazardous Environment Class	Class I, Division 2, Hazardous Location, Groups A, B, C, D (UL 1604, C-UL under CSA C22.2 No. 213)
Radiated and Conducted Emissions	EN50081-2 Class A
Vendor I.D. Code	1
Product Type Code	109
Product Code	19
Electrical /EMC:	*The module has passed testing at the following levels:*
ESD Immunity (IEC61000-4-2)	• 4 kV contact, 8 kV air, 4 kV indirect
Radiated Immunity (IEC61000-4-3)	• 10 V/m , 80 to 1000 MHz, 80% amplitude modulation, +900 MHz keyed carrier
Fast Transient Burst (IEC61000-4-4)	• 2 kV, 5kHz
Surge Immunity (IEC61000-4-5)	• 1kV galvanic gun
Conducted Immunity (IEC61000-4-6)	• 10V, 0.15 to 80MHz(2)

(1) This rating applies for your system if a relay module such as the 1769-OW8 is used. If no relays are used, use the "Operating" vibration specification.

(2) Conducted Immunity frequency range may be 150 kHz to 30 MHz if the Radiated Immunity frequency range is 30 MHz to 1000 MHz.

Input Specifications

Specification	Value
Input Voltage Range	-30 to 30V dc(1)
On-State Voltage (max.)	30V dc(1)
On-State Voltage (min.)	2.6V dc
On-State Current (min.)	6.8 mA
Off-State Voltage (max.)	1.0V dc
Off-State Current (max.)	1.5 mA
Off-State Leakage Current (max.)	1.5 mA
Input Current (max.)	15 mA
Input Current (min.)	6.8 mA
Input Impedance (nominal)	1950 Ω
Pulse Width (min.)	250 nsec
Phase Separation (min.)	131 nsec
Input Frequency (max.)	1 MHz
Isolation (Inputs to the Bus and Input to Input)	Verified by one of the following dielectric tests: • 1200V ac or 1697V dc for 1 second • 75V dc working voltage (IEC Class 2 reinforced insulation)

(1) See Maximum Input Voltage - 24V dc Operation temperature derating on page A-4.

Output Specifications

Specification	Value
Output Voltage Range	5 to 30V dc(1)
On-State Voltage (max.)	User Power - 0.1V dc
On-State Output Current (max.)	1A per point(2)
	4A per module(3)
On-State Output Current (min.)	1 mA
On-State Voltage Drop (max.)	0.5V dc
Off-State Leakage Current (max.)	5 μA
Turn On Time (max.)	400 μs(4)
Turn Off Time (max.)	200 μs
Reverse Polarity Protection	30V dc
Isolation (Outputs to Bus)	Verified by one of the following dielectric tests: • 1200V ac or 1697V dc for 1 second • 75V dc working voltage (IEC Class 2 reinforced insulation)

(1) See Maximum Output Voltage - 24V dc Operation temperature derating on page A-4.

(2) See Maximum Output Current per Point - 5V dc Operation temperature derating on page A-5 and Maximum Output Current per Point - 24V dc Operation temperature derating on page A-6.

(3) See Maximum Output Current per Module - 5V dc Operation temperature derating on page A-5 and Maximum Output Current per Module - 24V dc Operation temperature derating on page A-6.

(4) Maximum turn-on time applies to output voltage range of 5 to 7V dc. For output voltages greater than 7V dc, the maximum turn-on time is 200 μs.

Throughput and Timing

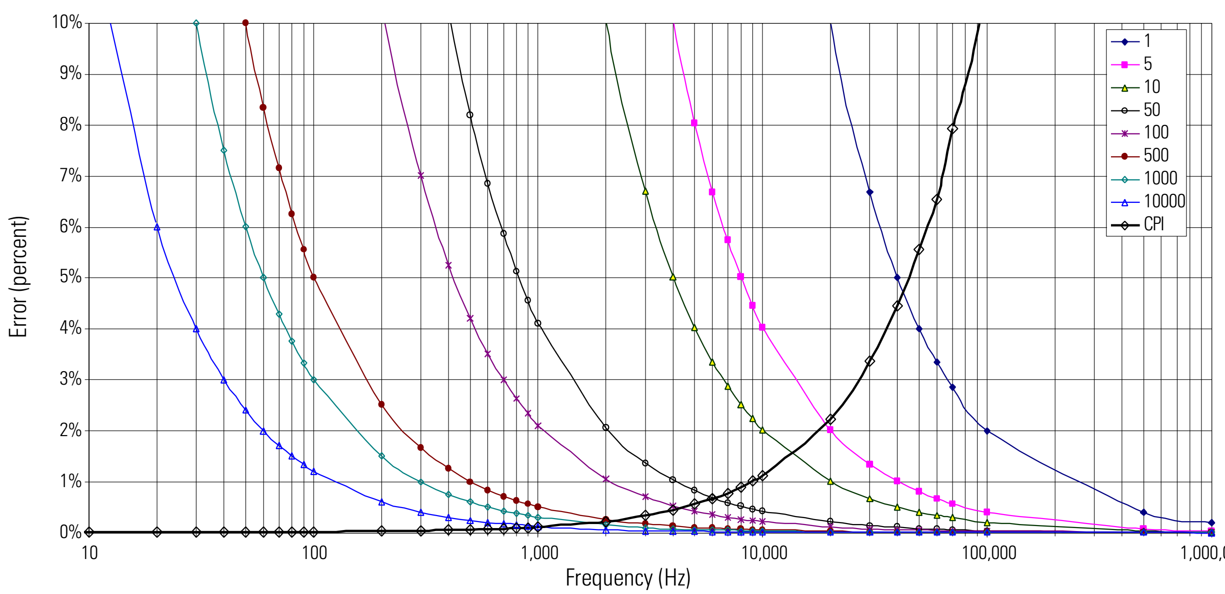
Operation	Description	Timing
Input File Update Time	The delay between the time the module receives a pulse and when the Compact bus count value is updated.	1 ms (maximum)
Output Turn-on Time	The time it takes for the real output to reach 90% output voltage after commanded by the module, not including processor scan time.	400 µs (maximum)
Output Turn-off Time	The time it takes for the real output to reach 10% output voltage after commanded by the module, not including the processor scan time.	200 µs (maximum)
Rate Accuracy	The accuracy of the reported rate as compared to actual input rate in the equation: reported rate/actual input rate.	Depends on frequency. See graph below.

Rate Accuracy

The following graph shows rate error at various frequencies. Pointing out a few trends may assist you in reading the graph:

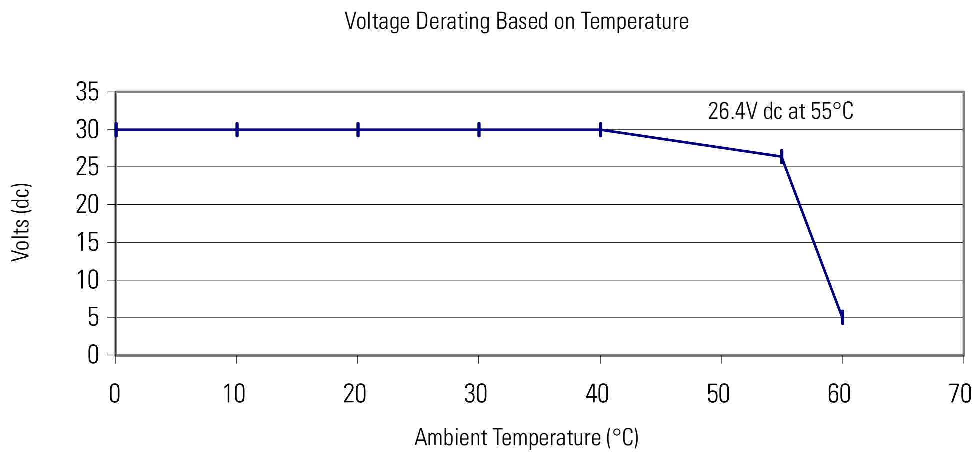
- Of the lines that rise at low frequencies, the leftmost is a 10 second update time (Ctr*n*CyclicRateUpdateTime = 10000).
- The rightmost of these lines is a 1 ms update time (Ctr*n*CyclicRateUpdateTime = 1).
- The line that rises at high frequencies illustrates Ctr[*n*].PulseInterval.

Figure A.1 Rate Errors Comparison



Temperature Derating

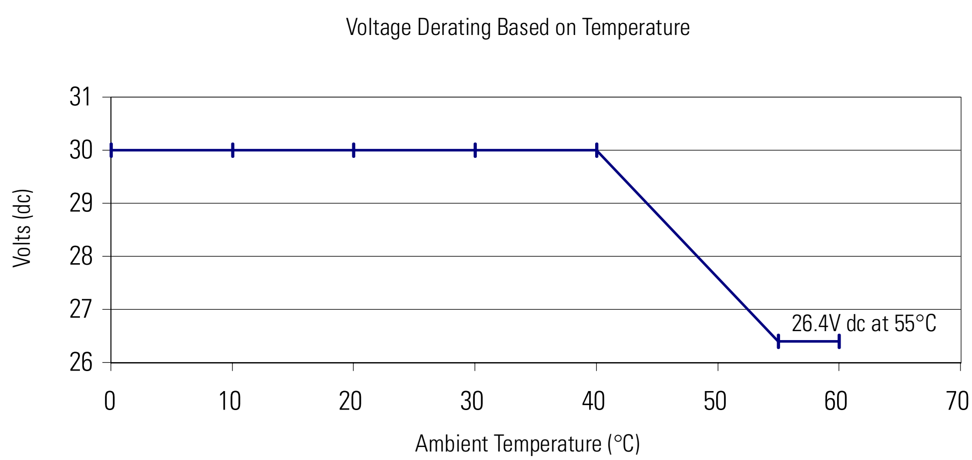
Maximum Input Voltage - 24V dc Operation



Temperature	Derated Voltage(1)
0°C to 40°C (-32°F to 104°F)	30V dc
55°C (131°F)	26.4V dc
60°C (140°F)	5V dc

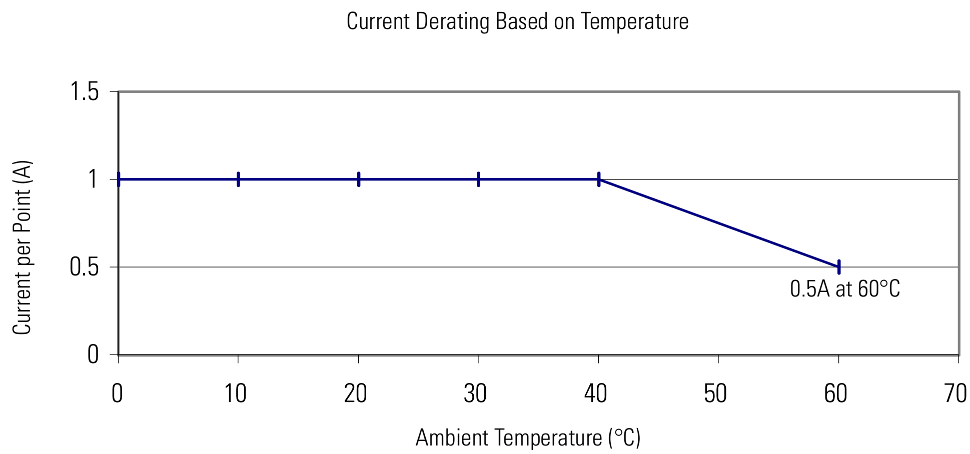
(1) Input voltage derating between 55°C and 60°C is achieved by using a dropping resistor.
For 24V dc input voltage, use a 2.4 kΩ, ½ Watt resistor.
For input voltages greater than 24V dc, use a ½ Watt resistor with value: 125 x (V_{in} - 5V).

Maximum Output Voltage - 24V dc Operation



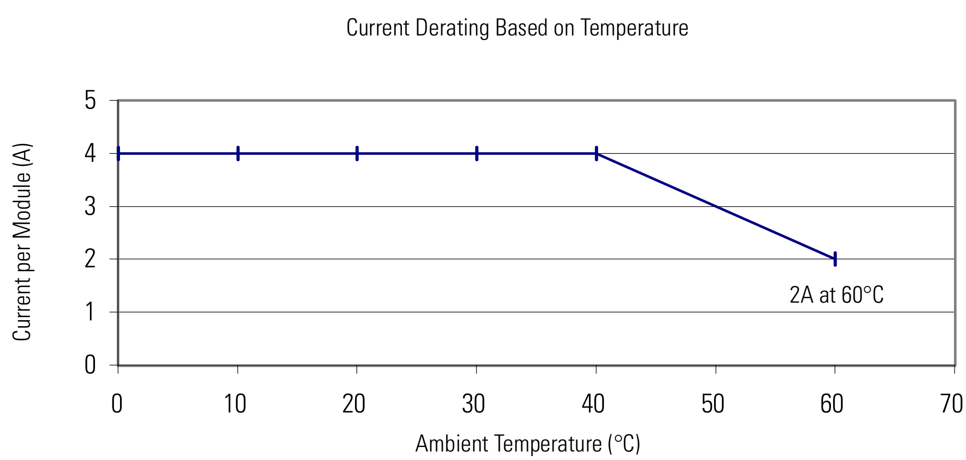
Temperature	Derated Voltage
0°C to 40°C (-32°F to 104°F)	30V dc
55°C to 60°C (131°F to 140°F)	26.4V dc

Maximum Output Current per Point - 5V dc Operation



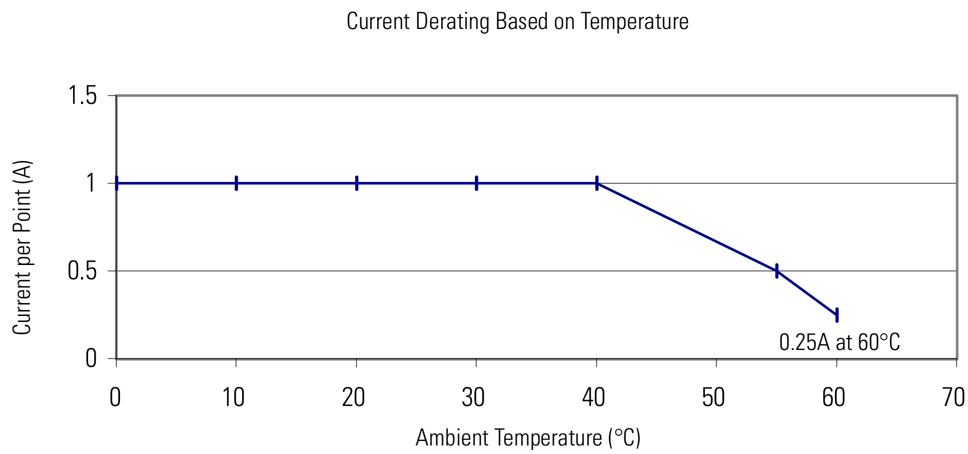
Temperature	Derated Current
0°C to 40°C (-32°F to 104°F)	1A
60°C (140°F)	0.5A

Maximum Output Current per Module - 5V dc Operation



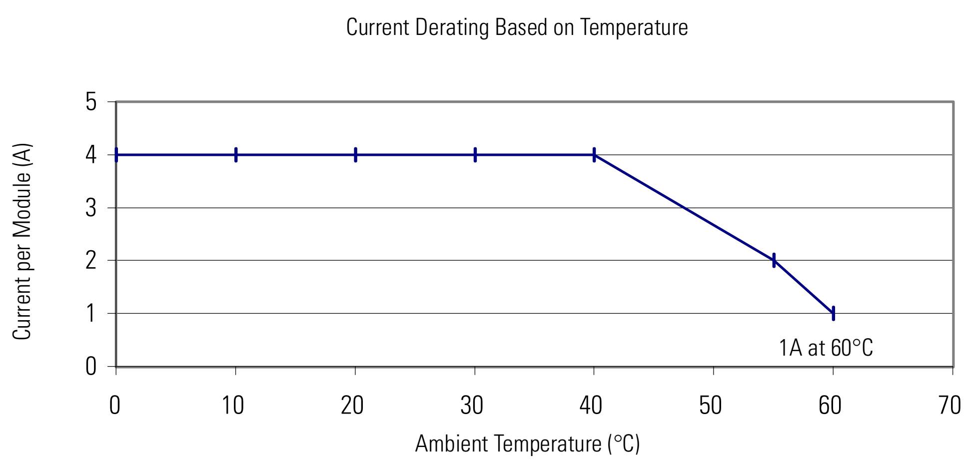
Temperature	Derated Current
0°C to 40°C (-32°F to 104°F)	4A
60°C (140°F)	2.0A

Maximum Output Current per Point - 24V dc Operation



Temperature	Derated Current
0°C to 40°C (-32°F to 104°F)	1A
55°C (131°F)	0.5A
60°C (140°F)	0.25A

Maximum Output Current per Module - 24V dc Operation



Temperature	Derated Current
0°C to 40°C (-32°F to 104°F)	4A
55°C (131°F)	2A
60°C (140°F)	1A

Dimensions

NOTE: All dimensions are in mm (inches). Hole spacing tolerance: ±0.04 mm (0.016 in.).

Figure A.2 Compact I/O with CompactLogix Controller and Power Supply

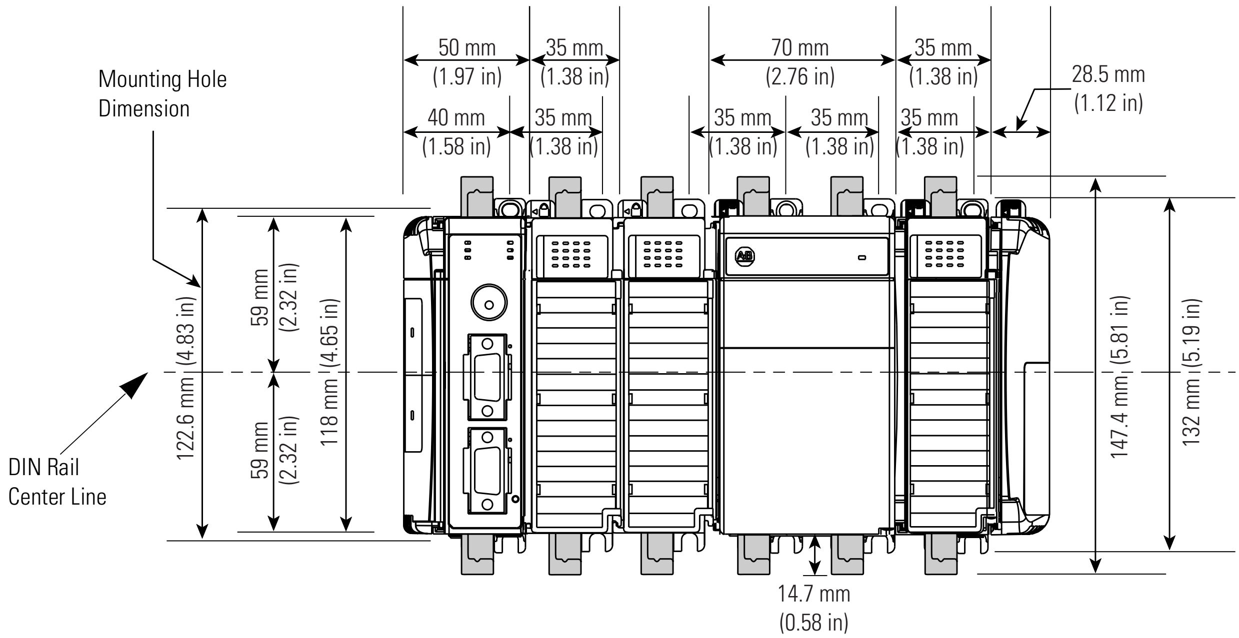
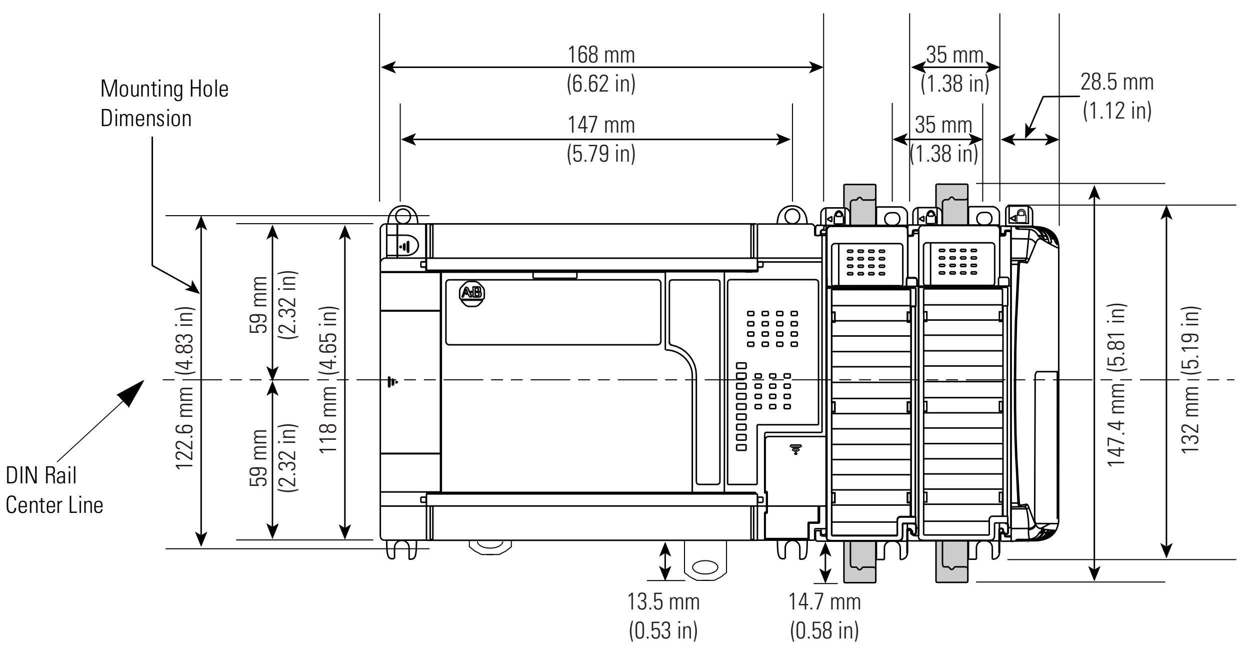
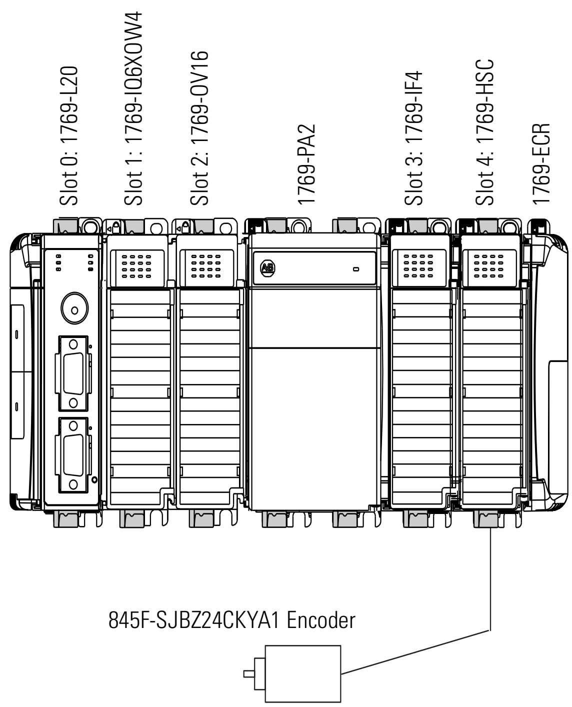


Figure A.3 Compact I/O with MicroLogix 1500 Base Unit and Processor



1769-HSC Module with CompactLogix Controllers and an Allen-Bradley 845F Encoder

System Diagram



845F Encoder Wiring to the 1769-HSC

Table B.1 Encoder Wiring

845F Encoder Wire	Color	1769-HSC Terminal
Blue/Black Wire Pair	Blue	A0+
	Black	A0-
White/Black Wire Pair	White	B0+
	Black	B0-
Green/Black Wire Pair	Green	Z0+
	Black	Z0-
Red/Black Wire Pair	Red	+24V dc Power Supply
	Black	24V dc Common

Purpose

The purpose of this application example is to demonstrate how to wire an Allen-Bradley 845F optical incremental encoder to a 1769-HSC module and ultimately monitor the Current Count value in the CompactLogix controller. We will also control 2 onboard outputs with 2 Ranges.

Scope

This example will cover the following steps:

1. Add the 1769-HSC High Speed Counter module into a CompactLogix system using the Generic profile in RSLogix 5000 programming software.

2. Configure the 1769-HSC by entering configuration information into Configuration and Output tags created in RSLogix 5000 for the 1769-HSC module.

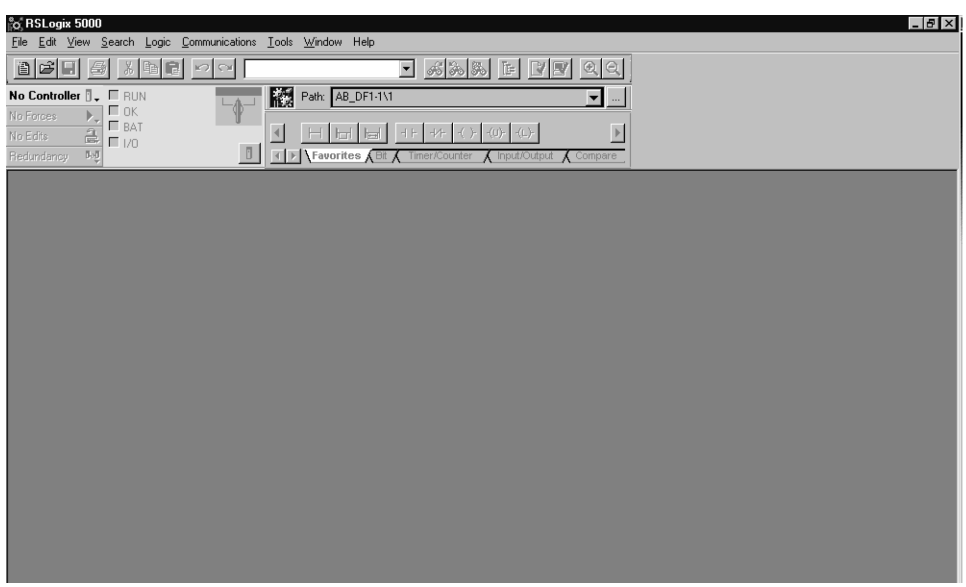
3. Monitor the Current Count value from the 1769-HSC module in the Input Tag created for the module.

4. Verify that module outputs 0 and 1 turn on when the Current Counts value is in the specified Ranges.

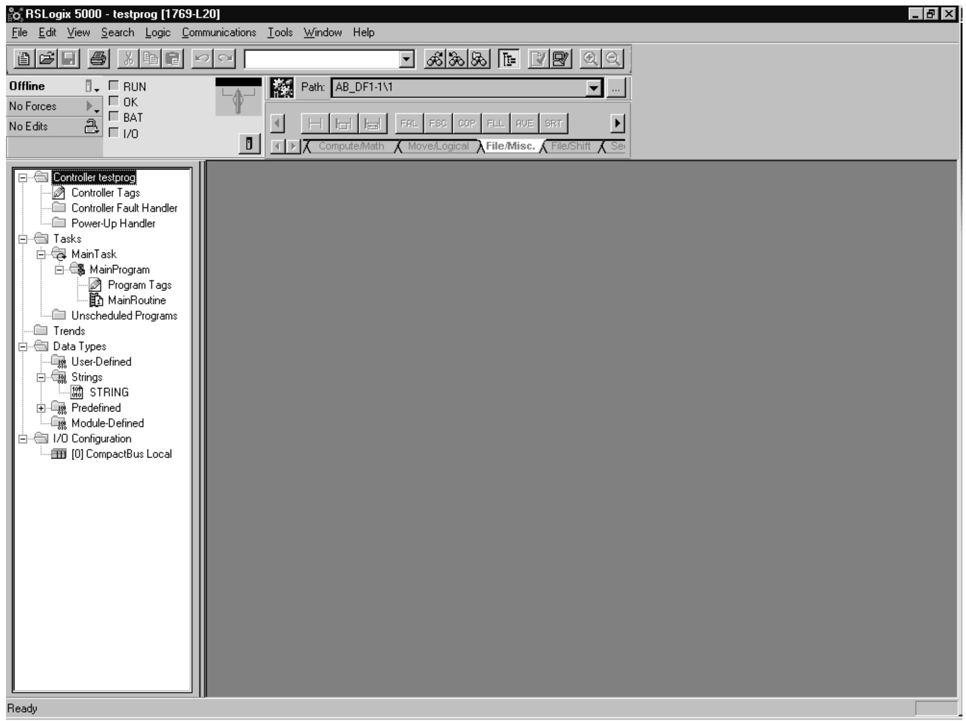
Adding a 1769-HSC High Speed Counter Module into a CompactLogix System

This example uses a 1769-L20 CompactLogix controller. The 1769-L30 controller will operate the same with respect to the 1769-HSC module.

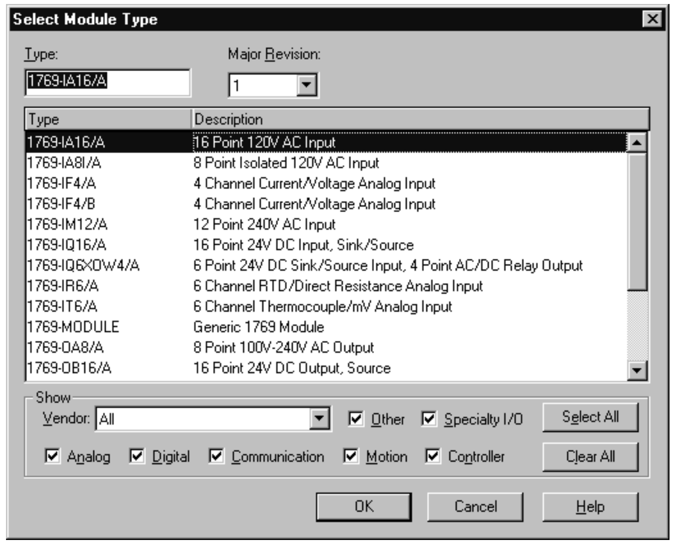
1. Start the RSLogix 5000 programming software by double clicking its icon on your desktop. The following screen appears:



2. Click the “New” icon or the File pull-down menu and select New. In the box that appears, choose the correct controller type (1769-L20 CompactLogix 5320 controller for this example) and give your project a name. Then click OK and the following screen will appear:



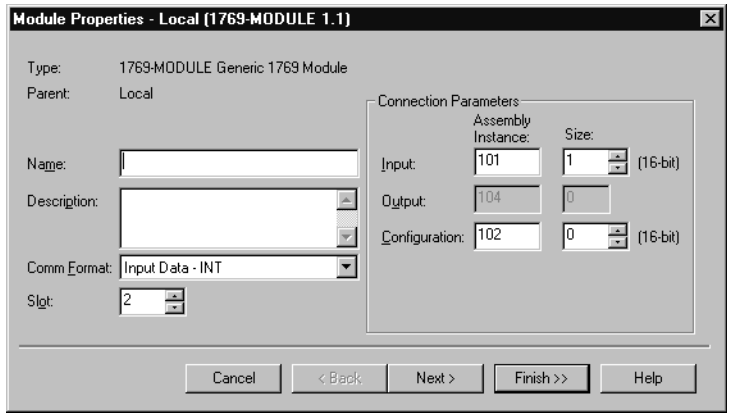
3. The area on the left of this screen is called the Controller Organizer. To add I/O modules to your CompactLogix Project, right-click on the last parameter listed in the Controller Organizer, called “[0] CompactBus Local” and choose “New Module”. The following screen will appear:



4. This screen displays all 1769 I/O modules that have a Thin Profile. Any module listed by its catalog number has such a profile. To add these modules to your CompactLogix system, click on the module, then click OK. The screen that appears allows you to name the module. All other parameters should be left at their defaults.

5. Click “Finish” and your module will be displayed below the “[0] CompactBus Local” in slot 1. The next I/O module you configure, by default, will be placed in slot 2 and so on. Configure the remaining I/O modules that are listed by name in the same manner.

6. The 1769-HSC module does not yet have a Thin Profile. This means that until this module appears in the I/O list by name, the “Generic 1769 Module” profile will be used to add the 1769-HSC module to your CompactLogix system. To add it to your system, click on the “Generic 1769 Module”, then click OK and the following screen appears:



7. Fill in this Generic profile screen as follows:

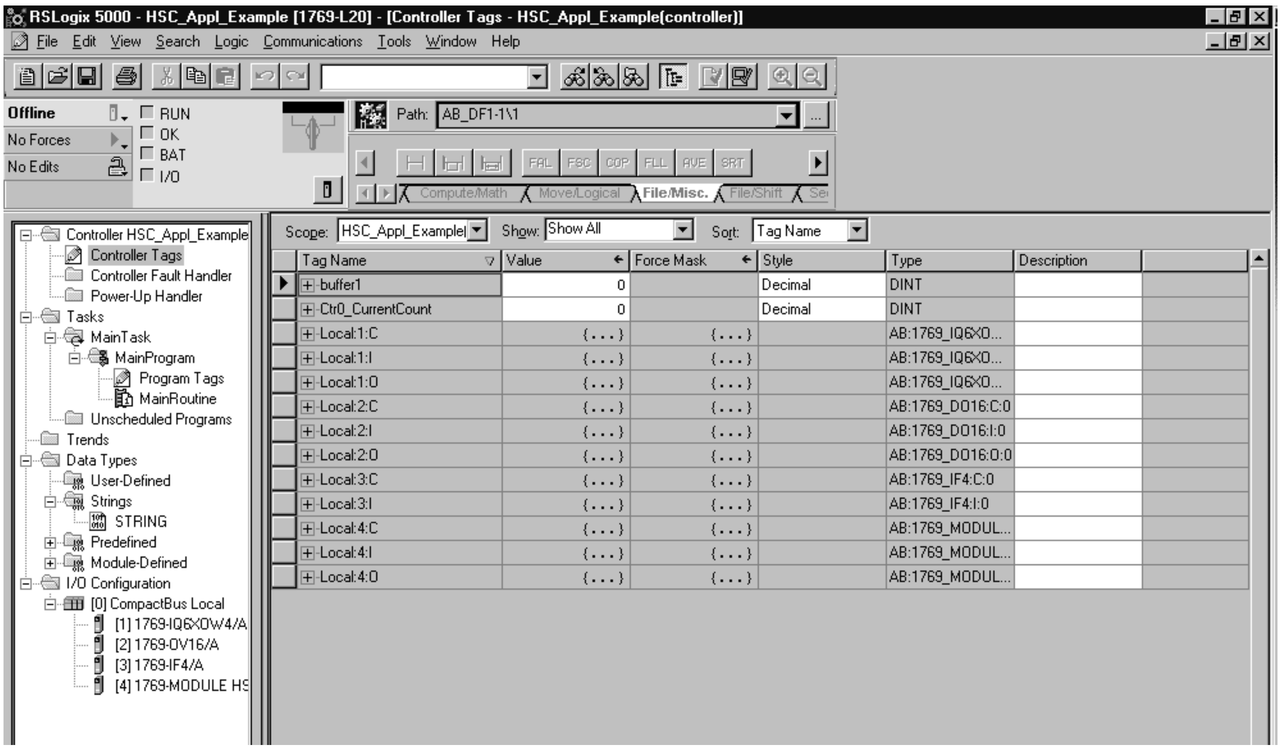
Name	**Give your HSC module a name**
Comm Format:	Data-INT
Input:	Assembly Instance = 101, size = 35
Output:	Assembly Instance = 100, size = 34
Configuration:	Assembly Instance = 102, size = 118
Slot:	For this example, the HSC module is in slot 4

8. When you have entered the data into your Generic profile screen, click Finish. Your module will be added to your CompactLogix system and it will be displayed under the [0] CompactBus Local in the Controller Organizer.

Configuring Your 1769-HSC Module

When the 1769-HSC module is added to the CompactLogix project, Input, Output and Configuration tags are automatically created in the Controller Tags area.

1. Double click on Controller Tags. The following screen appears:



This screen displays all the tags created for all the I/O modules added to the system. Discrete I/O modules are not configurable at this time, but all other types of I/O modules must be configured. In this example, the 1769-IF4 and the 1769-HSC must be configured. Refer to your *Compact I/O Analog Modules User Manual*, publication 1769-UM002A-EN-P, for information on configuring the 1769-IF4 module.

The tags for I/O modules are displayed in the following format, where s is the slot number of the module:

Tag	Description
Local:s:I	Input Image
Local:s:O	Output Image
Local:s:C	Configuration Data

Each of these tags has a plus sign to its left. Click on the plus sign to the left of any tag to open it. For the 1769-HSC in slot 4, click on the plus sign to the left of Local:4:C. A Reserved tag along with a Data tag are displayed. We need only be concerned with the Data tag. This is where we enter our configuration parameters for the 1769-HSC module.

2. Expand Local:4:C.Data by clicking its plus sign. A configuration tag with a length of 198 words is displayed, but only the first 118 words are needed to configure the 1769-HSC module. This 118 word configuration file is shown in Table 4.3. Word 0 in Table 4.3 corresponds to Local:4:C.Data[0], Word 1 corresponds to Local:4:C.Data[1] and so on.

It is best to configure the module in your offline project, then download the project to your CompactLogix controller. This is due to the fact that configuration files are downloaded to the I/O modules only at download, when an inhibited module is uninhibited and at power up. For this example, we will configure the module to operate in the following way:

<table>
<tr><th>Configuration Parameter</th><th colspan="2">Value</th></tr>
<tr><td>Number of Counters</td><td colspan="2">1</td></tr>
<tr><td>Maximum Count Value</td><td colspan="2">1,200,000</td></tr>
<tr><td>Minimum Count Value</td><td colspan="2">0</td></tr>
<tr><td>Scalar</td><td colspan="2">1</td></tr>
<tr><td>Update Time Value</td><td colspan="2">1</td></tr>
<tr><td>Operational Mode</td><td colspan="2">Quadrature encoder X 4</td></tr>
<tr><td rowspan="10">Ring Counter Two Ranges</td><td rowspan="5">Range0</td><td>Maximum Value: 600,000</td></tr>
<tr><td>Minimum Value: 500,000</td></tr>
<tr><td>Control Output 0 with this range</td></tr>
<tr><td>Range Type: Count Value</td></tr>
<tr><td>ToThisCounter: 0</td></tr>
<tr><td rowspan="5">Range1</td><td>Maximum Value: 1,200,000</td></tr>
<tr><td>Minimum Value: 1,000,000</td></tr>
<tr><td>Control Output 1 with this range</td></tr>
<tr><td>Range Type: Count Value</td></tr>
<tr><td>ToThisCounter: 0</td></tr>
</table>

Translate the configuration parameters above into the 1769-HSC Configuration file, per Chapter 4 as follows:

Configuration Tag	Hex Value	Description
Local:4:C.Data[0]	16#0000	Number of Counters =1
Local:4:C.Data[1]	16#0000	No filters used in this example
Local:4:C.Data[2]	16#0000	PSO and PSR not used
Local:4:C.Data[3]	16#0000	PVO not used
Local:4:C.Data[4]	16#0000	FSO and FSR not used
Local:4:C.Data[5]	16#0000	FVO not used
Local:4:C.Data[6]	16#4F80	Low word for Ctr0MaxCount
Local:4:C.Data[7]	16#0012	High word for Ctr0MaxCount
Local:4:C.Data[8]	16#0000	Low word for Ctr0MinCount
Local:4:C.Data[9]	16#0000	High word for Ctr0MinCount
Local:4:C.Data[10]	16#0000	Low word for Ctr0Preset
Local:4:C.Data[11]	16#0000	High word for Ctr0Preset
Local:4:C.Data[12]	16#0000	Hysteresis not used
Local:4:C.Data[13]	16#0001	Not used, must set to 1
Local:4:C.Data[14]	16#0001	Not used, valid range:1-32767
Local:4:C.Data[15]	16#0006	Operational Mode: Quadrature Encoder X4
Local:4:C.Data[16] through Local:4:C.Data[45] are for configuring counters 1 through 3. Since we are only using counter 0 in this example, these words should not be modified.		
Local:4:C.Data[46]	16#27C0	Low word for Range0HighLimit
Local:4:C.Data[47]	16#0009	High word for Range0HighLimit
Local:4:C.Data[48]	16#A120	Low word for Range0LowLimit
Local:4:C.Data[49]	16#0007	High word for Range0LowLimit
Local:4:C.Data[50]	16#0001	Enable Output 0 for Range0
Local:4:C.Data[51]	16#0000	For Counter0, Counter Value
Local:4:C.Data[52]	16#4F80	Low word for Range1HighLimit
Local:4:C.Data[53]	16#0012	High word for Range1HighLimit
Local:4:C.Data[54]	16#4240	Low word for Range1LowLimit
Local:4:C.Data[55]	16#000F	High word for Range1LowLimit
Local:4:C.Data[56]	16#0002	Enable Output 1 for Range1
Local:4:C.Data[57]	16#0000	For Counter0, Counter Value
Local:4:C.Data[58] through Local:4:C.Data[117] are for configuring ranges 2 through 11. Since we are only using ranges 0 and 1 in this example, these words should not be modified.		

TIP

To enter double integer (DINT) values into 2 integer words, create a single DINT in your Controller Tags area and call it “Buffer” or something similar.

Enter any DINT value into this tag in the decimal radix, then change the radix to Hex. The DINT value will be displayed in two 4-digit hex values. The 4-digit hex value on the left is the high word and the one on the right is the low word.

Enter these values into the Configuration file or Output file where appropriate, in the hex radix. For example, the Ctr0MaxCount value is a DINT, represented in the Configuration tag for the 1769-HSC as 2 integer words, Local:4:C.Data[6] and Local:4:C.Data[7]. The value we want to enter here is 1,200,000.

Enter this value into our DINT “Buffer” in decimal, then change to the hex radix. The result is 16#0012_4f80. The low word is 4f80 hex and must be entered into tag Local:4:C.Data[6]. The high word is 0012 hex and must be entered into tag Local:4:C.Data[7]. Be sure to be in the hex radix before entering the hex values into these words.

3. To fully configure the 1769-HSC module, we must now modify parameters in the Output tag as well. Click the plus sign to the left of Local:4:O, then click the plus sign to the left of Local:4:O.Data. 34 words of output image appear. Addresses for these 34 words are:

Local:4:O.Data[0] through Local:4:O.Data[33].

For this example, only the first 6 words are modified. Words Local:4:O.Data[6] through Local:4:O.Data[33] are for Counters 1 to 3 and Ranges 12 to 15, which we are not using in this example.

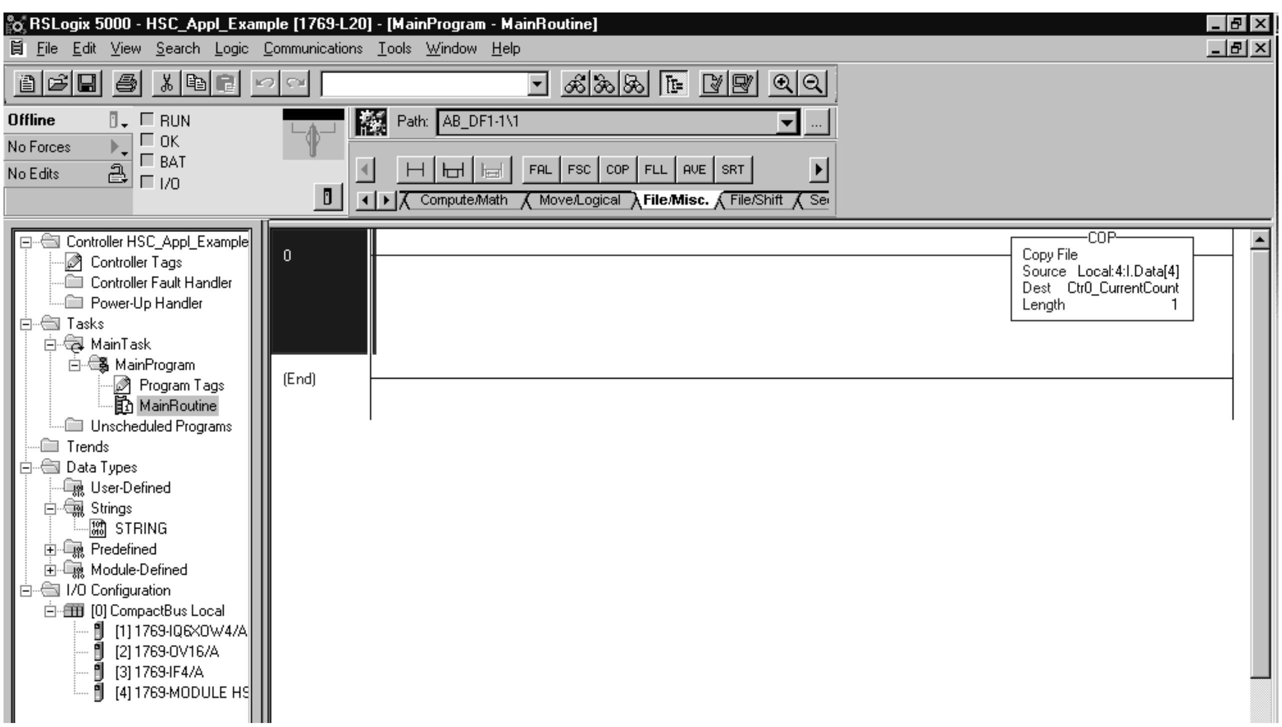
The 6 Output words are as follows:

Output Tag	Hex Value	Description
Local:4:O.Data[0]	16#0000	
Local:4:O.Data[1]	16#0003	Enables Outputs 0 and 1 to be controlled by Ranges 0 and 1.
Local:4:O.Data[2]	16#0003	Enable Ranges 0 and 1
Local:4:O.Data[3]	16#0000	Not using Interrupts
Local:4:O.Data[4]	16#0000	Not using Interrupts
Local:4:O.Data[5]	16#0001	Enable Counter 0
Local:4:O.Data[6] through Local:4:O.Data[33] are not used by this example and should not be modified		

Monitoring the Current Count Value and Verifying Output Operation

The Current Count value for Counter0 is represented in the Input tag for the module with 2 integer words. Since this value is a DINT value, we must copy the two integer words to a DINT tag to properly view the Current Count of Counter0.

1. In the Controller tags screen, enter the edit mode and create a tag called "Ctr0CurrentCount". Be sure this tag is a DINT. Then enter the following ladder rung:



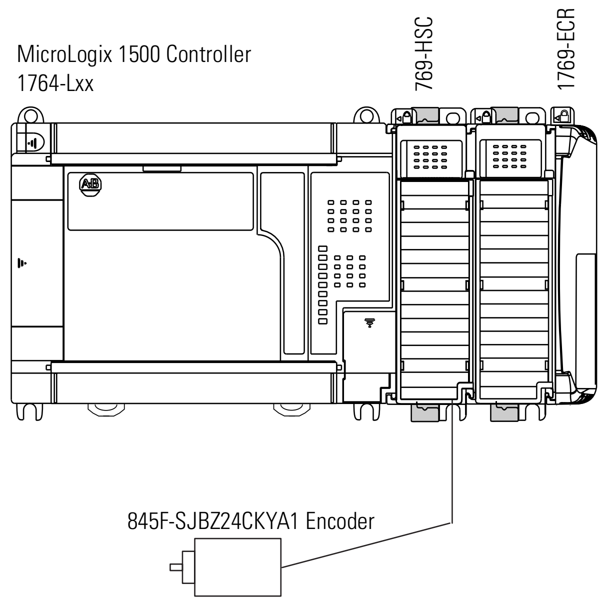
2. Notice that the Source of the COP instruction is the first of the two integer tags that represent the Current Count for Counter0. The destination of the COP instruction is the DINT you just created. The length of a COP instruction is always determined by the Destination tag, in this case a single DINT. If this were reversed and the Source of the COP were the DINT and the Destination was the address of the first of two integers, then the length would be 2.

3. Save the program and download it to your controller. Place the controller into the RUN mode and spin the shaft on your 845F encoder. Tag “Ctr0CurrentCount” will display the current count data for Counter0 of the 1769-HSC. This count, for this example is the number of pulses received from the encoder times 4 (we chose the operating mode to be “Quadrature Encoder X4”).

4. Continue to spin the encoder shaft until the current count value is within the limits set for Range0 (500,000 to 600,000). Output0 should turn on only when the current count value is equal to or within the Range0 limits. Output1 should turn on only when the Current Counts value is equal to or within the Range1 limits (1,000,000 to 1,200,000). These two outputs will be off for all other values of the Current Count for Counter0.

1769-HSC Module with MicroLogix 1500 Controllers and an Allen-Bradley 845F Encoder

System Diagram



845F Encoder Wiring to the 1769-HSC

Table C.1 Encoder Wiring

845F Encoder Wire	Color	1769-HSC Terminal
Blue/Black Wire Pair	Blue	A0+
	Black	A0-
White/Black Wire Pair	White	B0+
	Black	B0-
Green/Black Wire Pair	Green	Z0+
	Black	Z0-
Red/Black Wire Pair	Red	+24V dc Power Supply
	Black	24V dc Common

Purpose

The purpose of this application example is to demonstrate how to wire an Allen-Bradley 845F optical incremental encoder to a 1769-HSC module and ultimately monitor the Current Count value in the MicroLogix 1500 controller. We will also control 2 onboard outputs with 2 Ranges.

Scope

This example will cover the following steps:

1. Add the 1769-HSC High Speed Counter module into a MicroLogix 1500 system using the RSLogix 500 programming software.

2. Configure the 1769-HSC by entering configuration information into I/O Configuration created in RSLogix 500 for the 1769-HSC module.

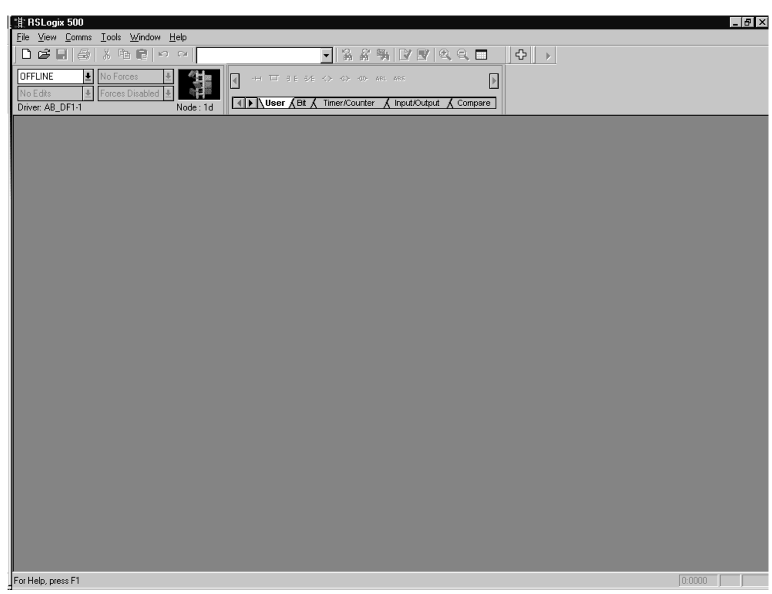
3. Monitor the Current Count value from the 1769-HSC module

4. Verify that module outputs 0 and 1 turn on when the Current Count value is within the specified Ranges.

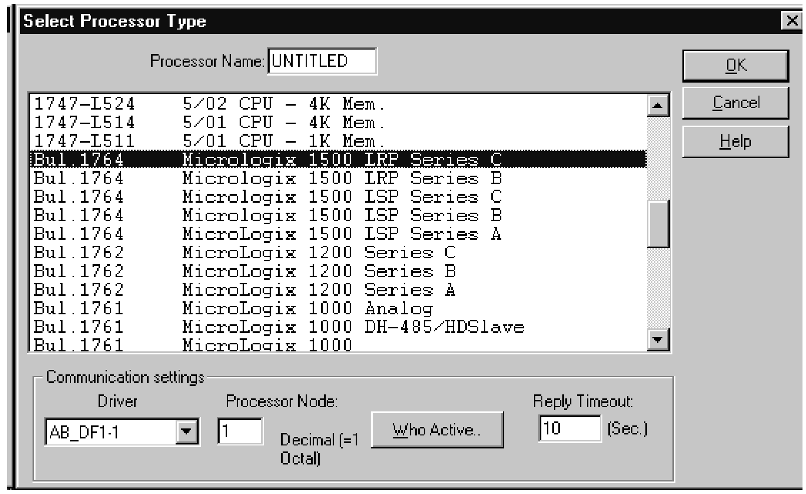
Adding a 1769-HSC High Speed Counter Module into a MicroLogix 1500 System

This example uses a MicroLogix 1500 controller.

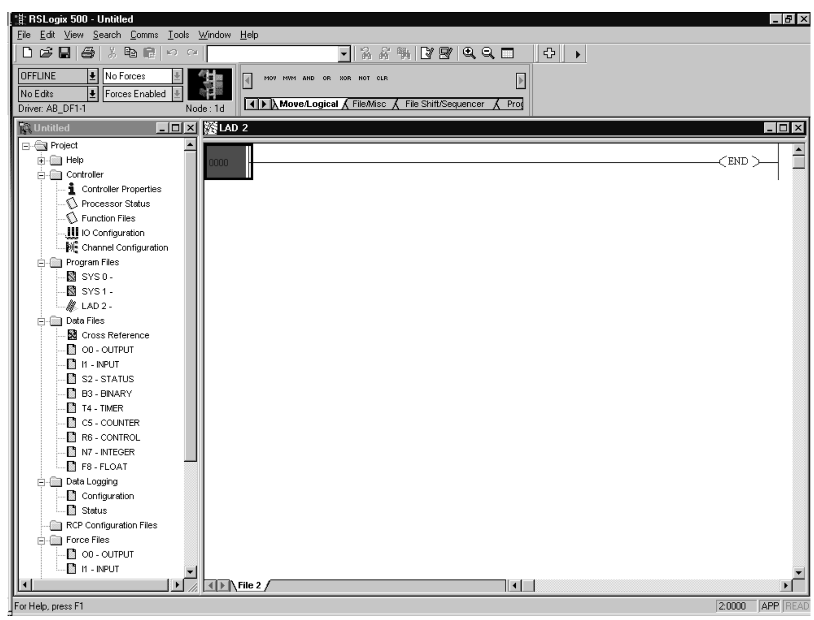
1. Start the RSLogix 500 programming software by double clicking its icon on your desktop or from the **Start>Programs>Rockwell Software>RSLogix 500 English>RSLogix 500 English**. The following screen appears:



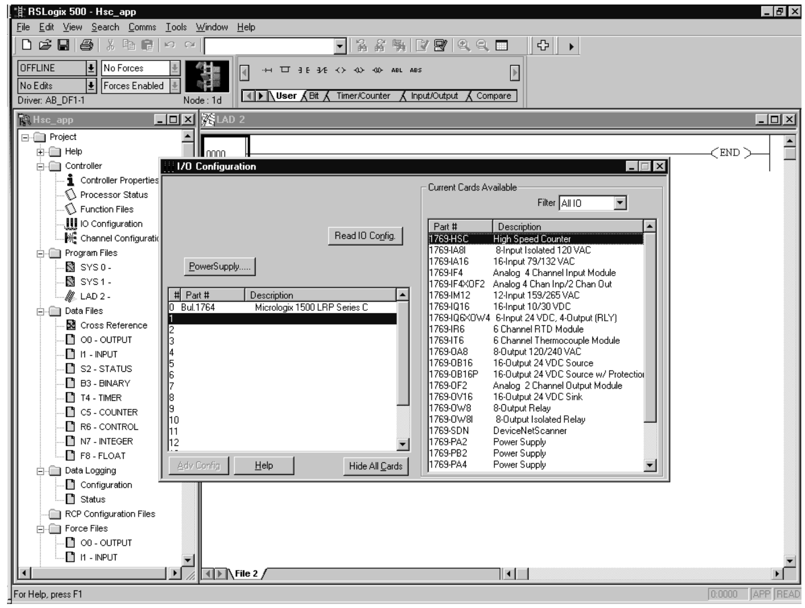
2. Click the “New” icon or the File pull down menu and select New. The following screen appears:



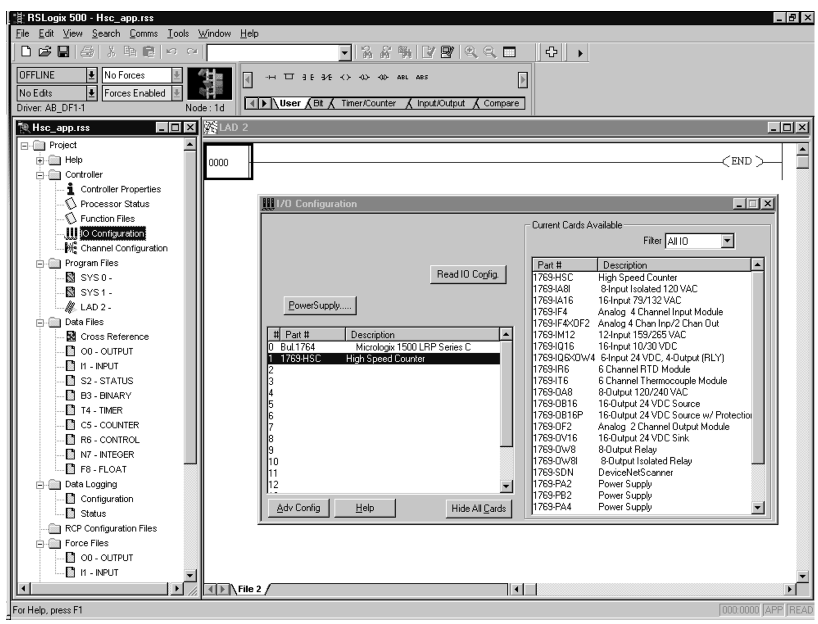
3. Choose the correct controller type (Bul.1764 MicroLogix 1500 LRP series C controller for this example) and give your processor a name. Then click OK and the following screen appears:



4. The area on the left of this screen is called the Project Menu. To add I/O modules to your MicroLogix 1500 Project, left click on the I/O Configuration parameter listed in the Project Menu. The following screen appears:



5. This screen displays all 1769 I/O modules supported by the MicroLogix 1500. To add the 1769-HSC module to your MicroLogix 1500 system, double left click on the module or click, hold and drag the module to its desired slot. In this case we will use slot 1.

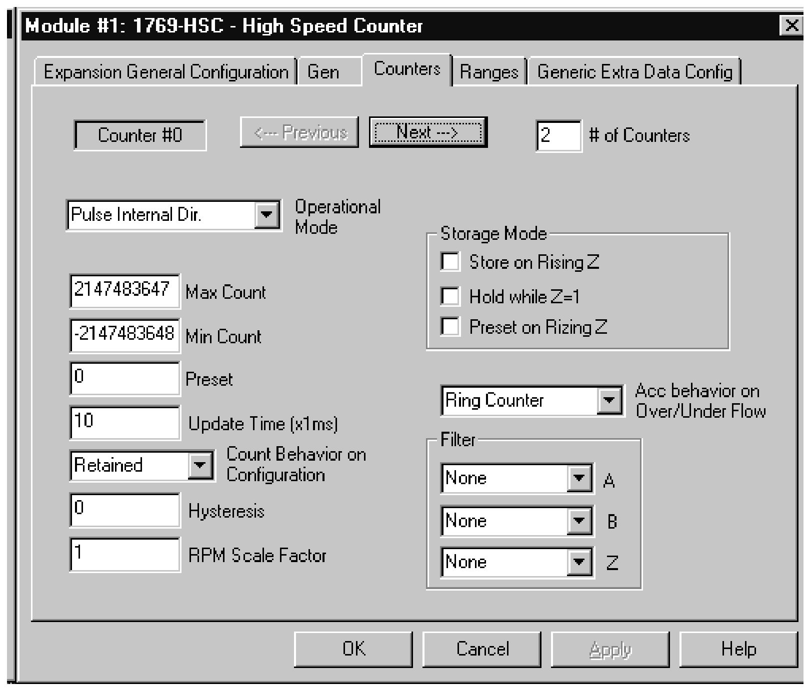


Configuring Your 1769-HSC Module

Configuration of the module is done in your offline project, and then downloaded to the MicroLogix 1500 controller. This is due to the fact that configuration files are downloaded to the I/O modules only at download.

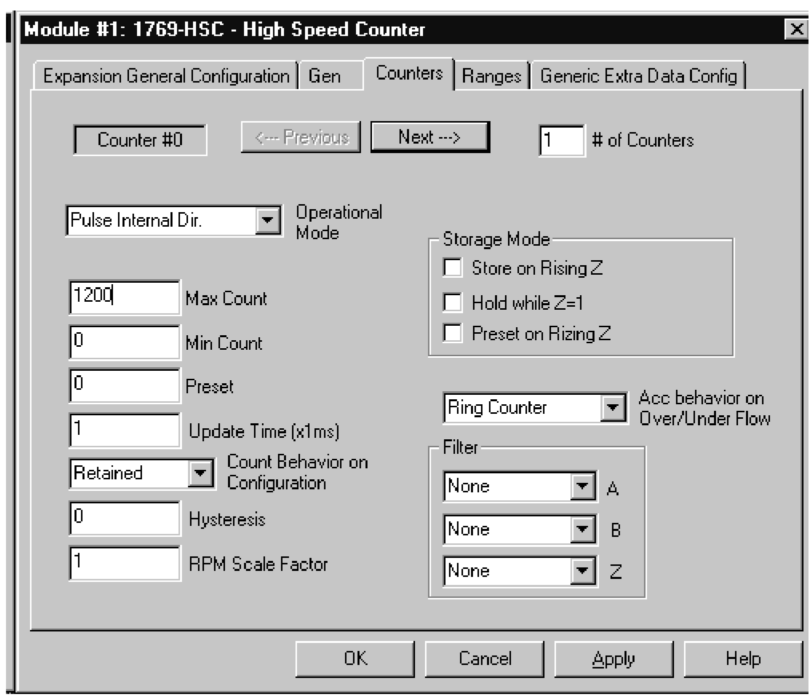
1. Click the Adv Config button to open the 1746-HSC-module configuration file.

2. Then select the Counter Tab to display the counter configuration screen with all its default values.

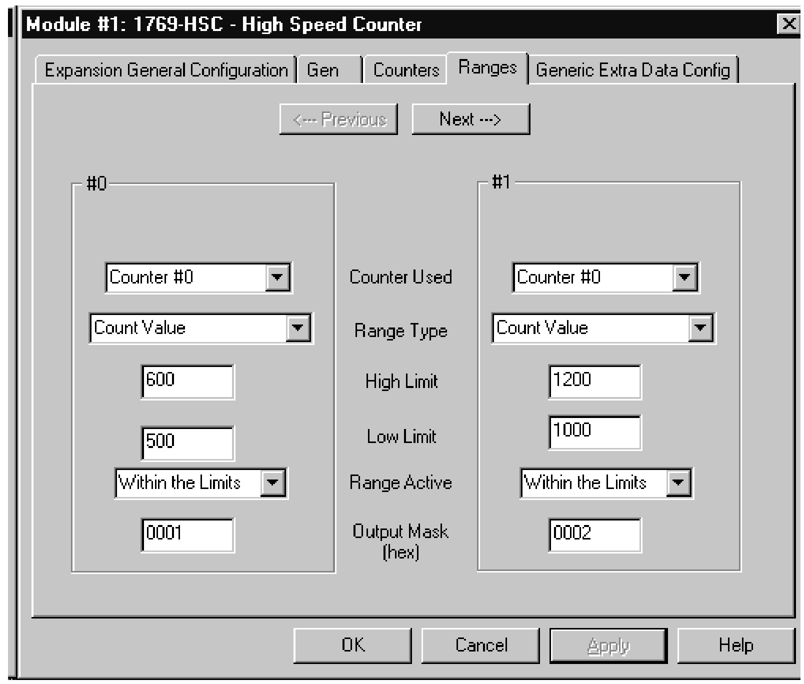


3. For this example, configure the module to operate in the following way:

Number of Counters :	1 (default = 2)
Maximum Count Value:	1,200,000 (default = 2147483647)
Minimum Count Value:	0 (default = –2147483648)
Preset:	1 (default = 0)
Update Time Value:	1 (default = 10)
Operational Mode:	(Quadrature) Encoder X 4
	(default = Pulse Internal Dir)
Count Behavior	
On Configuration:	Retained (default = Retained)
Hysteresis:	0 (default = 0)
RPM Scale Factor	1 (default = 0)
Number Of Counters	1 (default = 2)
Storage modes	All Unchecked (default = all unchecked)
Acc behavior on	
Over/Under flow	Ring Counter (default = Ring counter)
A, B, Z Filters	None (default = none)



4. Select the Range Tab to display the counter range configuration screen with all its default values. This configuration will use two of the 12 ranges available for the ring counter.



5. Configure the module to operate with the following values:

Range #0:	
Counter used:	Counter #0
Range Type:	Count Value
High limit	600,000
Low limit	500,000
Range Active	Within the limits
Output Mask	0001
Range #1:	
Counter used:	Counter #0
Range Type:	Count Value
High limit	1,200,000
Low limit	1,000,000
Range Active	Within the limits
Output Mask	0002

6. In order to fully configure the 1769-HSC module, we must now modify parameters in the Output Data file as well. Click on the Output Data file on the left side under the Data files. 34 words of output image will appear. Addresses for these 34 words are Output word [0] through Output word [33].

7. For this example, only the first 6 words are modified. Output Word [6] through Output Word [33] are for Counters 1 –3 and Ranges 12-15, which we are not using in this example.

   The 6 Output words are as follows:

Output Data File	Decimal Value	Description
Output Word [0]	0	Not used
Output Word [1]	3	Enables Outputs 0 and 1 to be controlled by Ranges 0 and 1.
Output Word [2]	3	Enable Ranges 0 and 1
Output Word [3]	0	Not using Interrupts
Output Word [4]	0	Not using Interrupts
Output Word [5]	1	Enable Counter 0

   Output Word [6] through Output Word [33] are not used by this example and should not be modified.

Monitoring the Current Count and Verifying Output Operation

No program logic is needed for this example. Save the program and download it to your controller. Place the controller into the RUN mode and spin the shaft on your 845F encoder. Input words 4 and 5 "Current Count" will display the current count data for Counter #0 of the 1769-HSC. This count, for this example, is the number of pulses received from the encoder times 4 (we chose the operating mode to be "Quadrature Encoder X4").

Continue to spin the encoder shaft until the current count value is within the limits set for Range0 (500,000 to 600,000). Output 0 should turn on only when the current count value is equal to or within the Range0 limits. Output1 should turn on only when the Current Count value is equal to or within the Range1 limits (1,000,000 to 1,200,000). These two outputs will be off for all other values of the Current Count for Counter 0.

You could also use a CPW instruction to monitor 32-bit values via ladder logic.

Programming Quick Reference

This appendix contains at-a-glance listings of the:

- Configuration Array
- Output Array
- Input Array

These sheets are also available electronically. They can be downloaded from **www.theautomationbookstore.com**.

Search for Item Number 1769-QR002A-EN-E. You can print out the PDF file for your reference.

Configuration Array

The default value for the Configuration Array is all zeros except where noted.

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description		
0							NumberOfCtrs					PFE			CtrRst	OCLO	GeneralConfigBits	→	OvercurrentLatchOff CtrReset ProgToFaultEn NumberOfCounters_0[1] NumberOfCounters_1
1	Filter_Z1			Filter_B1			Filter_A1		Filter_Z0			Filter_B0			Filter_A0		FilterA0_0, FilterA0_1 -- ...Z1_1		
2									Out3 PSR	Out2 PSR	Out1 PSR	Out0 PSR	Out3 PM	Out2 PM	Out1 PM	Out0 PM	Out0ProgramStateRun --Out3... and Out0ProgramMode -- Out3...		
3													Out3 PV	Out2 PV	Out1 PV	Out0 PV	Out0ProgramValue --- Out3...		
4									Out3 FSR	Out2 FSR	Out1 FSR	Out0 FSR	Out3 FM	Out2 FM	Out1 FM	Out0 FM	Out0FaultStateRun --Out3FaultStateRun and Out0FaultMode -- Out3FaultMode		
5													Out3 FV	Out2 FV	Out1 FV	Out0 FV	Out0FaultValue -- Out3FaultValue		
6 7	Ctr0MaxCount[2]																Ctr0MaxCount		
8 9	Ctr0MinCount[3]																Ctr0MinCount		
10 11	Ctr0Preset																Ctr0Preset		
12	Ctr0Hysteresis																Ctr0Hysteresis		
13	Ctr0Scalar[4]																Ctr0Scalar		
14	Ctr0CyclicRateUpdateTime[5]																Ctr0CyclicRateUpdateTime		
15				Linear		Storage Mode								Operational Mode			Ctr0ConfigFlags	→	Ctr0Config.OperationalMode_0 Ctr0Config.OperationalMode_1 Ctr0Config.OperationalMode_2 Ctr0Config.StorageMode_0 Ctr0Config.StorageMode_1 Ctr0Config.StorageMode_2 Ctr0Config.Linear
16 17	Ctr1MaxCount[2]																Ctr1MaxCount		
18 19	Ctr1MinCount[3]																Ctr1MinCount		
20 21	Ctr1Preset																Ctr1Preset		
22	Ctr1Hysteresis																Ctr1Hysteresis		
23	Ctr1Scalar[4]																Ctr1Scalar		
24	Ctr1CyclicRateUpdateTime[5]																Ctr1CyclicRateUpdateTime		
25				Linear		Storage Mode								Operational Mode			Ctr1ConfigFlags	→	Ctr1Config.OperationalMode_0 Ctr1Config.OperationalMode_1 Ctr1Config.OperationalMode_2 Ctr1Config.StorageMode_0 Ctr1Config.StorageMode_1 Ctr1Config.StorageMode_2 Ctr1Config.Linear
26 27	Ctr2MaxCount[2]																Ctr2MaxCount		
28 29	Ctr2MinCount[3]																Ctr2MinCount		
30 31	Ctr2Preset																Ctr2Preset		
32	Ctr2Hysteresis																Ctr2Hysteresis		
33	Ctr2Scalar[4]																Ctr2Scalar		
34	Ctr2CyclicRateUpdateTime[5]																Ctr2CyclicRateUpdateTime		
35				Linear													Ctr2ConfigFlags	→	Ctr2Config.Linear
36 37	Ctr3MaxCount[2]																Ctr3MaxCount		
38 39	Ctr3MinCount[3]																Ctr3MinCount		
40 41	Ctr3Preset																Ctr3Preset		
42	Ctr3Hysteresis																Ctr3Hysteresis		
43	Ctr3Scalar[4]																Ctr3Scalar		
44	Ctr3CyclicRateUpdateTime[5]																Ctr3CyclicRateUpdateTime		
45				Linear													Ctr3ConfigFlags	→	Ctr3Config.Linear
46 47	Range0to11[0].HighLimit																Range0to11[0].HighLimit		
48 49	Range0to11[0].LowLimit																Range0to11[0].LowLimit		
50	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[0].OutputControl		
51								Inv				Type			ToThisCtr		Range0to11[0].ConfigFlags	→	Range0To11[0].ToThisCounter_0 Range0To11[0].ToThisCounter_1 Range0To11[0].Type Range0To11[0].Invert
52 53	Range0to11[1].HighLimit																Range0to11[1].HighLimit		
54 55	Range0to11[1].LowLimit																Range0to11[1].LowLimit		
56	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[1].OutputControl		
57								Inv				Type			ToThisCtr		Range0to11[1].ConfigFlags	→	Range0To11[1].ToThisCounter_0 Range0To11[1].ToThisCounter_1 Range0To11[1].Type Range0To11[1].Invert
58 59	Range0to11[2].HighLimit																Range0to11[2].HighLimit		
60 61	Range0to11[2].LowLimit																Range0to11[2].LowLimit		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description		
62	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[2].OutputControl		
63								Inv				Type			ToThisCtr		Range0to11[2].ConfigFlags	→	Range0To11[2].ToThisCounter_0 Range0To11[2].ToThisCounter_1 Range0To11[2].Type Range0To11[2].Invert
64 65	Range0to11[3].HighLimit																Range0to11[3].HighLimit		
66 67	Range0to11[3].LowLimit																Range0to11[3].LowLimit		
68	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[3].OutputControl		
69								Inv				Type			ToThisCtr		Range0to11[3].ConfigFlags	→	Range0To11[3].ToThisCounter_0 Range0To11[3].ToThisCounter_1 Range0To11[3].Type Range0To11[3].Invert
70 71	Range0to11[4].HighLimit																Range0to11[4].HighLimit		
72 73	Range0to11[4].LowLimit																Range0to11[4].LowLimit		
74	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[4].OutputControl		
75								Inv				Type			ToThisCtr		Range0to11[4].ConfigFlags	→	Range0To11[4].ToThisCounter_0 Range0To11[4].ToThisCounter_1 Range0To11[4].Type Range0To11[4].Invert
76 77	Range0to11[5].HighLimit																Range0to11[5].HighLimit		
78 79	Range0to11[5].LowLimit																Range0to11[5].LowLimit		
80	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[5].OutputControl		
81								Inv				Type			ToThisCtr		Range0to11[5].ConfigFlags	→	Range0To11[5].ToThisCounter_0 Range0To11[5].ToThisCounter_1 Range0To11[5].Type Range0To11[5].Invert
82 83	Range0to11[6].HighLimit																Range0to11[6].HighLimit		
84 85	Range0to11[6].LowLimit																Range0to11[6].LowLimit		
86	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[6].OutputControl		
87								Inv				Type			ToThisCtr		Range0to11[6].ConfigFlags	→	Range0To11[6].ToThisCounter_0 Range0To11[6].ToThisCounter_1 Range0To11[6].Type Range0To11[6].Invert
88 89	Range0to11[7].HighLimit																Range0to11[7].HighLimit		
90 91	Range0to11[7].LowLimit																Range0to11[7].LowLimit		
92	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[7].OutputControl		
93								Inv				Type			ToThisCtr		Range0to11[7].ConfigFlags	→	Range0To11[7].ToThisCounter_0 Range0To11[7].ToThisCounter_1 Range0To11[7].Type Range0To11[7].Invert
94 95	Range0to11[8].HighLimit																Range0to11[8].HighLimit		
96 97	Range0to11[8].LowLimit																Range0to11[8].LowLimit		
98	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[8].OutputControl		
99								Inv				Type			ToThisCtr		Range0to11[8].ConfigFlags	→	Range0To11[8].ToThisCounter_0 Range0To11[8].ToThisCounter_1 Range0To11[8].Type Range0To11[8].Invert
100 101	Range0to11[9].HighLimit																Range0to11[9].HighLimit		
102 103	Range0to11[9].LowLimit																Range0to11[9].LowLimit		
104	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[9].OutputControl		
105								Inv				Type			ToThisCtr		Range0to11[9].ConfigFlags	→	Range0To11[9].ToThisCounter_0 Range0To11[9].ToThisCounter_1 Range0To11[9].Type Range0To11[9].Invert
106 107	Range0to11[10].HighLimit																Range0to11[10].HighLimit		
108 109	Range0to11[10].LowLimit																Range0to11[10].LowLimit		
110	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[10].OutputControl		
111								Inv				Type			ToThisCtr		Range0to11[10].ConfigFlags	→	Range0To11[10].ToThisCounter_0 Range0To11[10].ToThisCounter_1 Range0To11[10].Type Range0To11[10].Invert
112 113	Range0to11[11].HighLimit																Range0to11[11].HighLimit		
114 115	Range0to11[11].LowLimit																Range0to11[11].LowLimit		
116	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range0to11[11].OutputControl		
117								Inv				Type			ToThisCtr		Range0to11[11].ConfigFlags	→	Range0To11[11].ToThisCounter_0 Range0To11[11].ToThisCounter_1 Range0To11[11].Type Range0To11[11].Invert

(1) The default value for NumberOfCounters is 01 (two counters declared).

(2) The default value for Ctr*n*MaxCount is +2,147,483,647 decimal for counters 0 and 1. The default value is 0 for counters 2 and 3.

(3) The default value for Ctr*n*MinCount is -2,147,483,648 decimal for counters 0 and 1. The default value is 0 for counters 2 and 3.

(4) The default value for Ctr*n*Scalar is 1 for counters 0 and 1. The default value is 0 for counters 2 and 3.

(5) The default value for Ctr*n*CyclicRateUpdateTime is 10 for counters 0 and 1. The default value is 0 for counters 2 and 3.

Output Array

The default value for the Output Array is all zeros.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description	
0	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	OutputOnMask.0 -- OutputOnMask.15	
1	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	OutputOffMask.0 -- OutputOffMask.15	
2	R15	R14	R13	R12	R11	R10	R09	R08	R07	R06	R05	R04	R03	R02	R01	R00	RangeEn.0 -- RangeEn.15	
3																	Reserved	
4									RBF								ResetBlownFuse	
5							RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RCU	RCO	SP	En	Ctr0ControlBits →	Ctr0En Ctr0SoftPreset Ctr0ResetCountOverflow Ctr0ResetCountUnderflow Ctr0DirectionInvert Ctr0DirectionInhibit Ctr0ZInvert Ctr0ZInhibit Ctr0ResetRisingEdgeZ Ctr0ResetCtrPresetWarning
6							RPW	RREZ	Z Inh	Z Inv	D Inh	D Inv	RCU	RCO	SP	En	Ctr1ControlBits	
7							RPW					D Inv	RCU	RCO	SP	En	Ctr2ControlBits	
8							RPW					D Inv	RCU	RCO	SP	En	Ctr3ControlBits	
9																	Reserved	
10 11	Range12To15[0].HiLimOrDirWr																Range12To15[0].HiLimOrDirWr	
12 13	Range12To15[0].LowLimit																Range12To15[0].LowLimit	
14	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range12To15[0].OutputControl.0 ... .15	
15								Inv			LDW	Type			ToThisCtr		Range12To15[0].Config Flags →	Range12To15[0].ToThisCounter_0 Range12To15[0].ToThisCounter_1 Range12To15[0].Type Range12To15[0].LoadDirectWrite Range12To15[0].Invert
16 17	Range12To15[1].HiLimOrDirWr																Range12To15[1].HiLimOrDirWr	
18 19	Range12To15[1].LowLimit																Range12To15[1].LowLimit	
20	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range12To15[1].OutputControl.0 ... .15	
21								Inv			LDW	Type			ToThisCtr		Range12To15[1].Config Flags →	Range12To15[1].ToThisCounter_0 Range12To15[1].ToThisCounter_1 Range12To15[1].Type Range12To15[1].LoadDirectWrite Range12To15[1].Invert
22 23	Range12To15[2].HiLimOrDirWr																Range12To15[2].HiLimOrDirWr	
24 25	Range12To15[2].LowLimit																Range12To15[2].LowLimit	
26	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range12To15[2].OutputControl.0 ... .15	
27								Inv			LDW	Type			ToThisCtr		Range12To15[2].Config Flags →	Range12To15[2].ToThisCounter_0 Range12To15[2].ToThisCounter_1 Range12To15[2].Type Range12To15[2].LoadDirectWrite Range12To15[2].Invert
28 29	Range12To15[3].HiLimOrDirWr																Range12To15[3].HiLimOrDirWr	
30 31	Range12To15[3].LowLimit																Range12To15[3].LowLimit	
32	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Range12To15[3].OutputControl.0 ... .15	
33								Inv			LDW	Type			ToThisCtr		Range12To15[3].Config Flags →	Range12To15[3].ToThisCounter_0 Range12To15[3].ToThisCounter_1 Range12To15[3].Type Range12To15[3].LoadDirectWrite Range12To15[3].Invert

Input Array

The default value for the Input Array is all zeros.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description	
0											Z1	B1	A1	Z0	B0	A0	InputStateA0 -- InputStateZ1	
1	Out15	Out14	Out13	Out12	Out11	Out10	Out09	Out08	Out07	Out06	Out05	Out04	Out03	Out02	Out01	Out00	Readback.0 -- Readback.15	
2	InvalidRangeLimit12…15				InvalidCtrAssignToRange12…15				GenErr	InvOut	MCfg		Out0Overcurrent -- Out3…				Status Flags →	InvalidRangeLimit12 ... 15 InvalidCtrAssignToRange12 ... 15 GenError InvalidOutput ModConfig Out0Overcurrent0 ... 3
3	R15	R14	R13	R12	R11	R10	R09	R08	R07	R06	R05	R04	R03	R02	R01	R00	RangeActive.0 -- RangeActive.15	
4 5	Ctr[0].CurrentCount																Ctr[0].CurrentCount	
6 7	Ctr[0].StoredCount																Ctr[0].StoredCount	
8 9	Ctr[0].CurrentRate																Ctr[0].CurrentRate	
10 11	Ctr[0].PulseInterval																Ctr[0].PulseInterval	
12										C0PW	RV		IDW	REZ	CUdf	COvf	Ctr[0].StatusFlags →	Ctr[0].Overflow Ctr[0].Underflow Ctr[0].RisingEdgeZ Ctr[0].InvalidDirectWrite ---------------- Ctr[0].RateValid Ctr[0].PresetWarning
13																	Reserved	
14 15	Ctr[1].CurrentCount																Ctr[1].CurrentCount	
16 17	Ctr[1].StoredCount																Ctr[1].StoredCount	
18 19	Ctr[1].CurrentRate																Ctr[1].CurrentRate	
20 21	Ctr[1].PulseInterval																Ctr[1].PulseInterval	
22										C1PW	RV	IC	IDW	REZ	CUdf	COvf	Ctr[1].StatusFlags →	Ctr[1].Overflow Ctr[1].Underflow Ctr[1].RisingEdgeZ Ctr[1].InvalidDirectWrite Ctr[1].InvalidCounter Ctr[1].RateValid Ctr[1].PresetWarning
23																	Reserved	
24 25	Ctr[2].CurrentCount																Ctr[2].CurrentCount	
26 27	Ctr[2].CurrentRate																Ctr[2].CurrentRate	
28										C2PW	RV	IC	IDW		CUdf	COvf	Ctr[2].StatusFlags →	Ctr[2].Overflow Ctr[2].Underflow ---------------- Ctr[2].InvalidDirectWrite Ctr[2].InvalidCounter Ctr[2].RateValid Ctr[2].PresetWarning
29																	Reserved	
30 31	Ctr[3].CurrentCount																Ctr[3].CurrentCount	
32 33	Ctr[3].CurrentRate																Ctr[3].CurrentRate	
34										C3PW	RV	IC	IDW		CUdf	COvf	Ctr[3].StatusFlags →	Ctr[3].Overflow Ctr[3].Underflow ---------------- Ctr[3].InvalidDirectWrite Ctr[3].InvalidCounter Ctr[3].RateValid Ctr[3].PresetWarning

A

Allen-Bradley contact information P-3, 5-14

C

cable 3-11, A-1
channel diagnostics 5-4
CompactLogix controller
  application example B-1
  compatibility 1-1
  user manual P-2
configuration 4-1
Configuration Array 4-2, D-2
configuration errors 4-2, 5-7
Configuration Flags 4-3, 4-4, 4-15
Control Bits 4-24
Count Overflow 4-38
Count Underflow 4-38
count value 2-14, 2-15
counter
  basic description 1-1
counter configuration
  summary 2-4
Counter Control Bits 4-21
Counter Preset Warning 4-39
counter types
  linear counter 2-14
  ring counter 2-15
Ctr0Config.Linear 5-13
Ctr0Config.OperationalMode 5-13
Ctr0Config.StorageMode 5-13
Ctr0CyclicRateUpdateTime 5-13
Ctr0Hysteresis 5-13
Ctr0MaxCount 5-13
Ctr0MinCount 5-13
Ctr0Preset 5-13
Ctr0Scalar 5-13
Current Count 4-32, 4-36
current draw A-1
Current Rate 4-37
Cyclic Rate Update Time 4-3, 4-14

D

Defaults
  Configuration Array 4-2, D-2
  Counter Defaults 2-1, 5-3
  Counter Maximum Count 4-12, D-3
  Counter Minimum Count 4-12, D-3
  Counter Preset 4-13
  Counter Reset 2-17, 4-7
  Counter Scalar 4-14, D-3
  Cyclic Rate Update Time 4-14, D-3
  Default Safe State 2-30
  Input Array D-4
  Number of Counters 4-8, D-3
  Output Array 4-21, D-4
DeviceNet adapter
  compatibility 1-1
  user manual P-2
dimensions A-4
DIN rail mounting 3-9
Direct Write 2-16
Direct Write Value 4-26
Direction Inhibit 2-7, 4-25
Direction Invert 2-7, 4-25

E

electrical noise 3-4
EMC Directive 3-1
Enable Counter 4-24
error
  cyclic rate error 2-23
  general error bit 4-32, 4-34
  invalid counter 4-24, 4-39
  invalid counter assigned to range 4-29, 4-35
  invalid direct write 4-38
  invalid output 4-34
  invalid output bit 4-21
  invalid range 4-26, 4-27
  invalid range limit 4-35
  per pulse error 2-22
error codes 5-8
error definitions 5-5
errors
  BadCounterMode 5-9
  BadCounterNum 5-9
  BadCtrAssignToRange 5-11
  BadHysteresis 5-10
  BadMin 5-9
  BadModConfigUpdate 5-9, 5-12
  BadPreset 5-9
  BadRangeLimit 5-10
  BadScalar 5-10
  BadScale 5-10
  configuration 5-7
  configuration errors 4-2

- critical 5-5
- extended error information field 5-6
- General Configuration Error 5-9
- hardware 5-6
- module error field 5-6
- non-critical 5-5
- UnusedConfigBitSet 5-9

European Union Directives 3-1
extended error codes 5-8
extended error information field 5-6

F

Fault State Run 2-29
Filter Selection 4-3, 4-8
FilterA0 5-13
FilterA1 5-13
FilterB0 5-13
FilterB1 5-13
FilterZ0 5-13
FilterZ1 5-13
finger-safe terminal block 3-13

G

gate/preset functions 2-4
gating 2-16
General Configuration Bits 4-3, 4-6
General Error Bit 4-34
grounding 3-11

H

hardware errors 5-6
hardware features 1-3
heat considerations 3-4
Hold Last State 2-27
Hysteresis 2-20, 4-3, 4-13

I

Input Array D-5
input array 4-31
input operational mode 2-7
- pulse/external direction 2-8
- pulse/internal direction 2-9
- up and down pulses 2-10
- X1 quadrature encoder 2-11
- X2 quadrature encoder 2-12
- X4 quadrature encoder 2-12

Input State 4-32, 4-33
inputs
- basic description 1-2
- block diagram 2-2
- differential encoder wiring 3-17
- discrete device wiring 3-19
- isolation 3-11
- reducing noise 3-4, 3-12
- single-ended encoder wiring 3-18
- terminal block wiring 3-16

installation
- grounding 3-11
- heat and noise considerations 3-4

Invalid Counter 4-39
Invalid Counter Assigned to Range 4-35
Invalid Direct Write 4-38
Invalid Output 4-34
Invalid Range Limit 4-35

L

LEDs
- overview 1-4
- power-up diagnostics 5-4
- safety considerations 5-1
- troubleshooting status 1-4, 5-4

linear counter 2-14, 4-16
Load Direct Write 4-30

M

Masks 2-23
Maximum Count 4-3, 4-11
MicroLogix 1500
- application example C-1, D-1
- compatibility 1-1
- user manual P-2

Minimum Count 4-3, 4-12
Module Configured 4-34
module error field 5-6
mounting 3-7

N

nameplate label 1-3
number of counters 2-4, 4-8, 5-13

O

operating
- block diagrams 2-2
- description 2-1

Operational Mode 4-15
Output Array D-4
output array 4-21
output control 2-23
Output Control Example 2-30
Output Fault Mode 4-3, 4-10
Output Fault State Run 4-3, 4-10
Output Fault Value 4-3, 4-11
Output Off Mask 2-24, 4-21, 4-23
Output On Mask 2-23, 4-21, 4-22
Output Program Mode 4-3
Output Program State Run 4-3
Output Program Value 4-3, 4-10
outputs
    basic description 1-2
    block diagram 2-3
    isolation 3-20
    overcurrent autoreset operation 3-20
    required power supply 3-20
    short-circuits 3-20
    terminal block wiring 3-16
    thermal protection 3-20
    transient pulse warning 3-21
    wiring diagram 3-22
Overcurrent 2-27
Overcurrent Feedback 4-33
OverCurrentLatchOff bit 3-20, 4-6, 5-13
overflow 2-14, 2-15, 2-21
    linear counter 2-14

P

panel mounting 3-7
power-up diagnostics 5-4
Preset 4-3, 4-13
preset/reset 2-17
program alteration 5-2
Program Mode 4-9
Program State Run 2-28, 4-9
Program to Fault Enable 2-29, 4-7
programming 4-1
Pulse Interval 4-32, 4-37
pulse/external direction 2-8
pulse/internal direction 2-9

R

Range Active 4-32, 4-35
Range Configuration Flags 4-19, 4-21, 4-22, 4-29
range control 2-24
Range Enable 4-21, 4-23
Range High Limit 4-4, 4-17, 4-26
Range High Limit Direct Write Value 4-21, 4-22
Range Invert 4-20, 4-30
Range Low Limit 4-4, 4-17, 4-21, 4-22, 4-27
Range Output Control 4-4, 4-18, 4-21, 4-22, 4-28
Range Type 4-19
RangeType 4-29
Rate 2-18
rate accuracy A-3
Rate Method 2-22
rate range 2-18
Rate Valid 2-21, 4-39
Readback 2-31, 4-32, 4-33
real outputs 2-23
removing terminal block 3-12
replacing a module 3-10
Reset 2-17, 4-7
Reset Blown Fuse 4-21, 4-23
Reset Counter Overflow 4-24
Reset Counter Preset Warning 4-25
Reset Counter Underflow 4-24
Reset Rising Edge Z 4-25
reset/preset 2-17
ring counter 2-15, 4-16
Rising Edge Z 4-38
Rockwell Automation contact information P-3, 5-14
Rockwell Software
    application example B-2, C-2

S

Safe State Control 2-27
safety circuits 5-2
Scalar 2-20, 4-3, 4-14
short-circuits 3-20
Soft Preset 4-24
spacing 3-7
specifications A-1
Status bits 4-31
Status Flags 4-32, 4-33, 4-38
Storage Mode 4-16
Stored Count 4-36

T

technical support 5-14
temperature derating A-4
terminal block
    identification drawing 1-3

removing 3-12
wiring 3-13
terminal door label 3-15
terminal screw torque 3-14
Throughput and Timing A-3
Timing A-3
ToThisCounter 4-19
troubleshooting
safety considerations 5-1

U

underflow 2-14, 2-15, 2-21
linear counter 2-14
up and down pulses 2-10
User-Defined Safe State 2-28
utput 4-21

V

virtual outputs 2-23

W

wire size 3-14
wiring 3-1
grounding and wiring guidelines manual P-2
module 3-14
routing considerations 3-4
terminal block 3-13

X

X1 quadrature encoder 2-11
X2 quadrature encoder 2-12
X4 quadrature encoder 2-12

Z

Z Inhibit 4-25
Z input
functions 2-16
gating 2-16
preset/reset 2-16
setting in output array 4-38
Z Invert 4-25

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733
Publication 1769-UM006A-EN-P - March 2002