User Manual

Allen-Bradley

1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules

Catalog Numbers 1732E-IB8M8SOER, 1732E-OB8M8SR

Important User Information

Solid-state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid-state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid-state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

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

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

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

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

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

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

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

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

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

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

Allen-Bradley, Rockwell Software, Rockwell Automation, ArmorBlock, RSLogix, RSLinx, and TechConnect are trademarks of Rockwell Automation, Inc.

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

Read this preface to familiarize yourself with the rest of the manual. It provides information concerning:

- who should use this manual
- the purpose of this manual
- related documentation
- conventions used in this manual

Who Should Use this Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules.

You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

Purpose of this Manual

This manual is a reference guide for the 1732E-IB8M8SOER, 1732E-OB8M8SR modules. It describes the procedures you use to install, wire, troubleshoot, and use your module.

Related Documentation

The following documents contain additional information concerning Rockwell Automation products. To obtain a copy, contact your local Rockwell Automation office or distributor.

Resource	Description
ArmorBlock Dual-Port EtherNet/IP 8-Point Digital Modules 1732E-WD002	Information on wiring the ArmorBlock Dual-Port EtherNet/IP 8-Point Digital Modules.
1732E ArmorBlock 2 Port Ethernet Module Installation Instructions, publication 1732E-IN007	Information on installing the ArmorBlock EtherNet/IP module.
1732E ArmorBlock 2 Port Ethernet Module Release Notes, publication 1732E-RN001	Release notes to supplement the existing documentation supplied with the ArmorBlock EtherNet/IP module.
ControlLogix Sequence of Events Module User Manual, publication 1756-UM528	A manual on how to install, configure and troubleshoot the ControlLogix Sequence of Events module in your ControlLogix application.
EtherNet/IP Embedded Switch Technology Application Guide, publication ENET-AP005	A manual on how to install, configure and maintain linear and Device-level Ring (DLR) networks using Rockwell Automation EtherNet/IP devices with embedded switch technology.
EtherNet/IP Modules in Logix5000 Control Systems User Manual, publication ENET-UM001	A manual on how to use EtherNet/IP modules with Logix5000 controllers and communicate with various devices on the Ethernet network.
Integrated Architecture and CIP Sync Configuration Application Techniques, publication IA-AT003	A manual on how to configure CIP Sync with Intergrated Architecture products. and applications.
Getting Results with RSLogix 5000, publication 9399-RLD300GR	Information on how to install and navigate RSLogix 5000. The guide includes troubleshooting information and tips on how to use RSLogix 5000 effectively.
Allen-Bradley Industrial Automation Glossary, AG-7.1	A glossary of industrial automation terms and abbreviations.

Common Techniques Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.

Chapter 5

Configure the Module for Your EtherNet/IP Network

Chapter 6

Configure the Module Using RSLogix 5000 Software

Chapter 7

Common Features of the 1732E-IB8M8SOER and 1732E-OB8M8SR Modules

Troubleshoot the Module

Chapter 12

Interpret Status Indicators

Appendix A

Specifications

Appendix B

Module Tags

Appendix C

Data Tables

Appendix D

Connect to Networks via Ethernet Interface

Appendix E

1732E ArmorBlock Embedded Web Server

Index

About ArmorBlock Modules

Overview

This chapter is an overview of the ArmorBlock family of modules. You will need to understand the concepts discussed in this chapter to configure your module and use it in an EtherNet/IP control system. The following table guides you where to find specific information in this chapter.

Module Features

The module features include:

- use of EtherNet/IP messages encapsulated within standard TCP/UDP/IP protocol
- common application layer with ControlNet and DeviceNet
- interfacing via Category 5 rated twisted pair cable
- half/full duplex 10 Mbit or 100 Mbit operation
- mounting on a wall or panel
- communication supported by RSLinx software
- IP address assigned via standard DHCP tools
- I/O configuration via RSLogix 5000 software
- no network scheduling required
- no routing tables required
- supports connections from multiple controllers simultaneously

Hardware/Software Compatibility

The module and the applications described in this manual are compatible with the following firmware versions and software releases.

Contact Rockwell Automation if you need software or firmware upgrades to use this equipment.

Product	Firmware Version / Software Release
1732E-IB8M8SOER and 1732E-OB8M8SR	Firmware rev. 1.001 or later

Product	Firmware Version / Software Release
1756-EN2T, 1756-EN2TR, 1756-EN3TR	3.x version when using RSLogix 5000 v18 or later
Studio 5000 Logix Designer[1]	18 or later
Studio 5000 Add-on Profile	3.01.01 or later
RSLinx software	2.56 or later

(1) Studio 5000 Logix Designer is the replacement for RSLogix 5000 (since v21). It provides one software package for discrete, process, batch, motion, safety and drive-based applications.

For a complete ControlLogix compatibility matrix, see publication IA-AT003.

Use of the Common Industrial Protocol (CIP)

The 1732E-IB8M8SOER and 1732E-OB8M8SR modules use the Common Industrial Protocol (CIP). CIP is the application layer protocol specified for EtherNet/IP, the Ethernet Industrial Protocol. It is a message-based protocol that implements a relative path to send a message from the "producing" device in a system to the "consuming" devices.

The producing device contains the path information that steers the message along the proper route to reach its consumers. Because the producing device holds this information, other devices along the path simply pass this information; they do not need to store it.

This has two significant benefits:

- You do not need to configure routing tables in the bridging modules, which greatly simplifies maintenance and module replacement.
- You maintain full control over the route taken by each message, which enables you to select alternative paths for the same end device.

Understand the Producer/ Consumer Model

The CIP "producer/consumer" networking model replaces the old source/ destination ("master/slave") model. The producer/consumer model reduces network traffic and increases speed of transmission. In traditional I/O systems, controllers poll input modules to obtain their input status. In the CIP system, input modules are not polled by a controller. Instead, they produce their data either upon a change of state (CoS) or periodically. The frequency of update depends upon the options chosen during configuration and where on the network the input module resides. The input module, therefore, is a producer of input data and the controller is a consumer of the data.

The controller can also produce data for other controllers to consume. The produced and consumed data is accessible by multiple controllers and other devices over the EtherNet/IP network. This data exchange conforms to the producer/consumer model.

Specify the Requested Packet Interval (RPI)

The Requested Packet Interval (RPI) is the update rate specified for a particular piece of data on the network. This value specifies how often to produce the data for that device. For example, if you specify an RPI of 50 ms, it means that every 50 ms the device sends its data to the controller or the controller sends its data to the device.

RPIs are only used for devices that exchange data. For example, a ControlLogix EtherNet/IP bridge module in the same chassis as the controller does not require an RPI because it is not a data-producing member of the system; it is used only as a bridge to remote modules.

Chapter Summary and What's Next

In this chapter you were given an overview of the 1732E ArmorBlock family of modules.

Notes:

Module Overview and Features

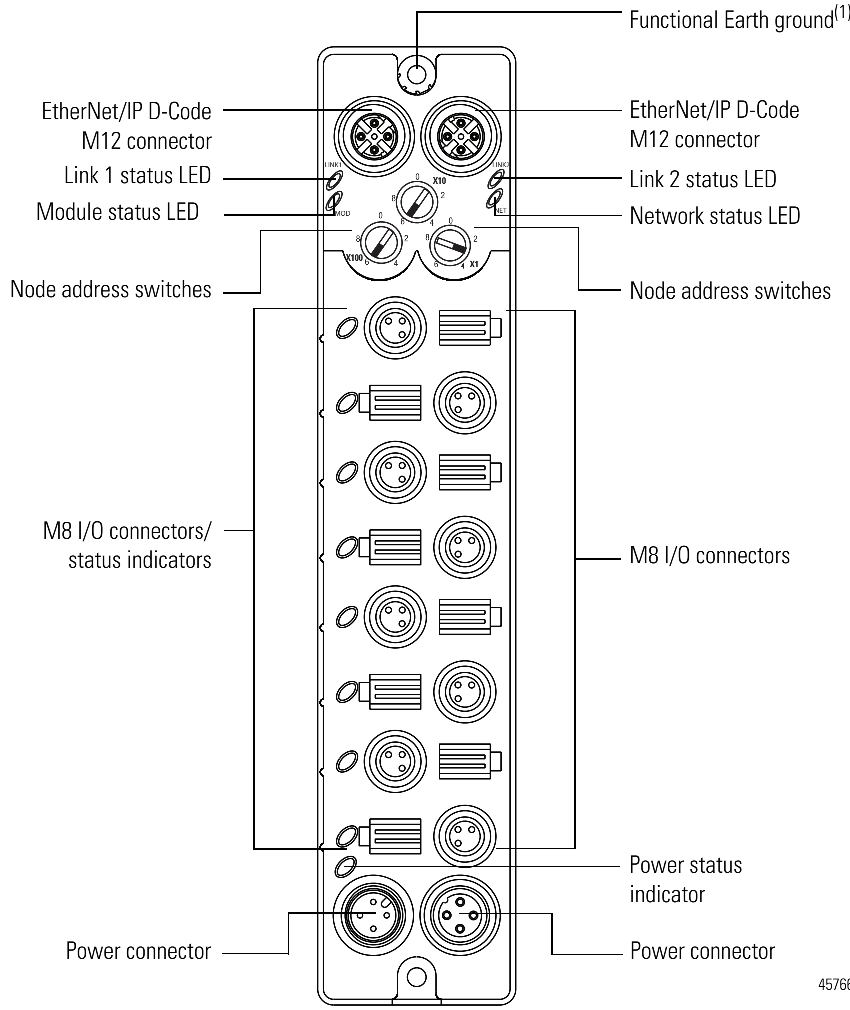
Overview

This chapter provides an overview of the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules, 1732E-IB8M8SOER and 1732E-OB8M8SR. The modules provide timestamping functionality when an input event occurs and allow for scheduling of outputs.

Although primarily described in this manual as having CIP Sync functionality, both modules can be configured to function as standard I/O modules.

The following table indicates where you can information on this chapter:

EtherNet/IP Network Overview



(1) Functional Earth grounds the I/O block's EtherNet/IP communication circuitry which is designed to mitigate the effect of noise on the network. It requires a solid earth ground connection, either through a metal screw to a grounded metal panel or through a wire.

The modules incorporate embedded switch technology. They support Star, Tree, Daisychain or Linear, and Ring network topologies.

- Star or Tree topologies can connect to either Port 1 or Port 2.
- Daisy Chain/Linear topologies will pass communications from Port 1 to 2, or Port 2 to 1.
- Ring topology will pass communications from Port 1 to 2, or Port 2 to 1.

The 1732E-IB8M8SOER and 1732E-OB8M8SR modules support the management of network traffic to ensure timely delivery of critical data, Quality of Service (QoS) and Internet Group Management Protocol (IGMP) protocols are supported.

If the ring topology is used, the *Ring Master* (not the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input or Scheduled Output) must be designated in the system, and determines the beacon rate and the timeout period. For more information on topologies, refer to publication ENET-AP005. The 1732E-IB8M8SOER and 1732E-OB8M8SR modules are CIP Sync slave only devices. There must be another module on the network that functions as a master clock.

Introduction to CIP Sync

CIP is the Common Industrial Protocol that we use to let all Rockwell Automation products communicate with each other whether it be on a DeviceNet, ControlNet, and/or an EtherNet network. Since it is an ODVA standard, other industrial product manufacturers develop products to communicate via the CIP protocol.

CIP Sync is a CIP implementation of the IEEE 1588 PTP (Precision Time Protocol) in which devices can bridge the PTP time across backplanes and on to other networks via EtherNet/IP ports.

What is IEEE 1588 PTP (Precision Time Protocol)?

The IEEE 1588 standard specifies a protocol to synchronize independent clocks running on separate nodes of a distributed measurement and control system to a high degree of accuracy and precision. The clocks communicate with each other over a communication network. In its basic form, the protocol is intended to be administration free. The protocol generates a master slave relationship among the clocks in the system. Within a given subnet of a network there will be a single master clock. All clocks ultimately derive their time from a clock known as the grandmaster clock. This is called Precision Time Protocol (PTP).

The PTP is a time-transfer protocol defined in the IEEE 1588-2008 standard that allows precise synchronization of networks, for example, Ethernet. Accuracy within the nanosecond range can be achieved with this protocol when using hardware generated synchronization.

IEEE 1588 is designed for local systems requiring very high accuracies beyond those attainable using Network Time Protocol (NTP). NTP is used to synchronize the time of a computer client or server to another server or reference time source, such as a GPS.

CIP Sync Support

CIP Sync supports the IEEE 1588-2008 synchronization standard. In this architecture, a grandmaster clock provides a master time reference for the system time. The 1732E-IB8M8SOER, 1732E-OB8M8SR modules are CIP Sync slave only devices. There must be another module on the network that will function as a master clock. The grandmaster could be:

- a 1756 ControlLogix L6 or L7 controller when using RSLogix 5000 software v18 or later.
- an Ethernet bridge that supports IEEE 1588 V2, or
- a Symmetricom Grand Master GPS or equivalent.

What is CIP Sync?

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

What is Time Stamping?

Each input has its own individual timestamp recorded for both ON and OFF transitions. The offset from the timestamp to the local clock is also recorded so that steps in time can be detected and resolved.

Timestamping uses the 64-bit system time whose time base is determined by the modules master clock resolved in microseconds. Each timestamp is updated as soon as an input transition is detected, before input filtering occurs. When filtering is enabled, the transition is only recorded if the transition passes the filter.

The module starts timestamping as soon as it powers up, even if it is not synchronized to a master clock. If it is synchronized to a master clock and then becomes unsynchronized it continues to time stamp. All time stamps and offsets have a value of zero at power-up.

For more information on how to use CIP Sync technology, see the Integrated Architecture and CIP Sync Configuration Application Technique publication IA-AT003.

Introduction to the Sequence of Events Input Module

The 1732E-IB8M8SOER is an input module that offers sub-millisecond timestamping on a per point basis in addition to providing the basic ON/OFF detection. It supports two modes of operation: Per Point Mode and FIFO (First In First Out) Mode. To learn more about using the modules in these modes of operation, see Operational Modes on page 68.

All input point event times are recorded and returned in a single buffer. The module returns two 64-bit timestamps for each input, thus allowing:

- ON and OFF events for each point to be displayed simultaneously in the input data.
- ladder logic not being explicitly required to see events, although needed to archive events.
- events to be kept in the controller memory during remote power loss thus eliminating data loss.

All inputs on the module can be filtered for both ON to OFF and OFF to ON transitions. The timestamp for a filtered input will be the time of the initial transition to the new state and not the time that the filter validates the event as real.

Selective Event Capturing allows particular events to be disabled per input and per transition, ON to OFF or OFF to ON.

Event latching ensures that events are not overwritten. A single transition in each direction is recorded per point. Any new event, which occurs after the point has captured a timestamp, is dropped until the stored events have been acknowledged.

If latching is not enabled in point mode, new events will overwrite old events when they are received. In FIFO mode, up to 256 events per input will be buffered before events are overwritten. Thus, if inputs are changing rapidly it may be possible that events will be lost either in the module or the controller prior to an event being operated on by ladder logic.

When events are lost, either old ones being overwritten or new ones being ignored due to latching, an EventOverflow bit will be set for each point that loses an event. The EventOverflow bit will clear when the blocking events for that point are acknowledged.

Timestamping is a feature that registers a time reference to a change in input data. For the 1732E-IB8M8SOER, the time mechanism used for timestamping is (PTP) system time. The 1732E-IB8M8SOER module is a PTP slave-only device. There must be another module on the network that functions as a master clock.

High Performance Sequence of Events Applications in the Logix Architecture

Sequence of Events (SOE) applications span a wide range of industry applications. Typically any event that needs to be compared against a second event can be classified as SOE.

- Used on discrete machines to identify failure points
- Used in Power Substations or power plants to indicate first fault conditions

- Used in SCADA applications to indicate pump failures or other discrete events
- Used in motion control applications to increase control coordination.
- Used in high speed applications
- Used in Global Position Registration

In today's environment, specifications for SOE applications typically require 1 ms or better resolution on timestamps. There are two types of SOE applications.

- First Fault – measures the time between events with no correlation to events outside of that system.
- Real Time – captures the time of an event occurrence as it relates to some master clock. Typically this is a GPS, NTP server or some other very accurate clock source. This method allows distributed systems to capture events and build a history of these events. These events are almost always digital, however some are analog for which lower performance requirements can be configured.

First Fault Detection

An example of first fault detection would be intermittent failure from a sensor on a safety system faults a machine and halts production cascading a flood of other interrelated machine faults. Traditional fault detection or alarms may not appear in the correct timed order of actual failure making root cause of the down time difficult or impossible.

Time Stamped I/O

High precision timestamps on I/O allows very accurate first fault detection making it easy to identify the initial fault that caused machine down time.

Common Time base for Alarming System logs user interaction as well as alarm events using common time reference.

The power industry requires sub 1 ms accuracy on first fault across geographically dispersed architecture.

High Speed Applications

Packaging machines or sorters that have fast part cycles are often bottlenecked by controller scan times. By switching to a time-based solution, you can remove many scan time critical components of the system. This programming technique allows you to do predictive events and schedule outputs to run things like diverters without having a scan time to match the part cycle time.

Motion Control

CIP Sync also provides a common time reference for distributed VFD drives, servos, and controllers throughout the system. This allows controllers to request axes, reach a pre-defined position at a known time reference, or run at a set speed using the same reference. Since all drives and controllers in the system have the same reference to time, the controller can issue simple requests for axes to reach target positions in a synchronized fashion.

Global Position Registration

Registration refers to a function usually performed by the drive where a physical input is triggered causing the drive to precisely capture the actual axis position when the input event occurred. Rather than wiring inputs to the registration input on all of the drives, this time-based system lets you wire an input to only one time based SOE input module. The timestamp returned for that input, can be used by the motion planner to calculate the actual axis position at the time the input triggered. This simplifies system installation, reduces wiring costs, and provides a global machine registration for all the axes in the system thru one SOE input.

Introduction to Scheduled Output Module

The 1732E-OB8M8SR Scheduled Output module is designed to work in conjunction with the MAOC motion instruction to provide position-based output control (also known as PLS). The MAOC instruction by itself allows position-based output control using the position of any motion axis in ControlLogix as the position reference and any output or boolean as the output. The MAOC updates the outputs based on motion axis position at the motion group coarse update rate (typically 2…10 ms). While this is adequate for some applications, it is too slow for many high speed applications typically found in converting and packaging segments. **The 1732E-OB8M8SR module improves performance by supporting the ability to schedule the output turn-on/turn-off time of its 8 outputs (outputs 0…7) in 1 µs increments**. Outputs are scheduled by entering data into one or more of the 16 schedules provided by the output connection data store.

IMPORTANT	When using the 1732E-OB8M8SR module with the MAOC instruction, make sure that you are using Studio 5000, version 21 or later. You must also select Yes for MAOC support and Per Point under Time Stamping.

Operation

This scheduled output implementation schedules outputs on a per point basis and each individual output point is controlled by its own timestamp.

Individual schedules are created in the controller, stored in the output image table for the module, and sent over the backplane to the Scheduled Output module. The schedule specifies a sequence count, the output point to be associated with the schedule, the time at which an output value should be applied to the physical output point, and the value to be applied at the scheduled time. The I/O module receives and stores the schedule. The CIPSync time of each schedule is monitored by the module. When a schedule has expired, that is the current time, matches the scheduled timestamp, the output value is then applied to the corresponding output bit. Timer hardware in the ASIC is used to optimize the scheduling algorithm. This hardware also reduces the latency and jitter performance. Status of each schedule is reported in the output echo connection and reflected in the input image for the module.

The scheduled output functionality relies on CIPSync time. Unused outputs may be used as normal outputs and are applied immediately rather than waiting for the CIPSync time to expire. A mask is sent to the module to indicate which outputs are to function as normal outputs. The scheduled output module supports up to 8 outputs that can be individually scheduled. The scheduled outputs must be between output points 0 and 7. The 1732E-OB8M8SR module supports up to 16 schedules with two schedules per output. Outputs that are not “scheduled” are used as normal output points. A mask is used to indicate which points are scheduled and which points are unscheduled. Jitter performance is less than 25 μs. All of the scheduling configuration is done through the MAOC instruction.

If a new schedule as indicated by a change in the sequence count is received by the I/O module before the current schedule has expired, the current schedule is overwritten. This mechanism can be used to cancel currently active schedule. Status bits returned in the output echo connection may be used to determine the current state of each schedule and to trigger corresponding event tasks.

If a new schedule is sent by the controller and the CIPSync time has already past, the output is asserted until the CIPSync time has completely wrapped around. The module does not check for an expired CIPSync time.

WARNING: If the time between two schedules is less than the minimum schedule interval (for example, 100 μs), then deviation occurs. This means that even though two outputs are scheduled at different times (for example, time 90 and time 110), they both activate at the same time (for example, time 90). The minimum schedule interval should not be set to faster than 100 μs.

High Speed Product Reject

In a control system you can program a scheduled output module, which can trigger multiple outputs simultaneously or trigger a reject at the precise point a product is at the reject station.

By using time to schedule the output in advance,and identifying when the product will be at a known position, hitting the exact point when a part is in front of a reject station on a high speed packaging machine, can be controlled.

Chapter Summary and What's Next

In this chapter, you were given an overview of the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules. The next chapter describes how the modules operate in an ArmorBlock system.

Notes:

Use the Modules in an ArmorBlock System

Introduction

This chapter describes how the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules operate in an ArmorBlock system.

Differences Between Module and Standard I/O

In many aspects, the modules behave the same as any other ArmorBlock digital module. However, the modules offer several significant differences from other EtherNet/IP ArmorBlock digital input modules, including those described in the following table.

Difference	Description
Additional data produced for controller	The modules produce significantly more data for its owner-controller than standard ArmorBlock digital input modules. While other input modules only produce ON/OFF and fault status, the modules produce data such as ON/OFF and fault status, timestamp data, indication of whether new data was produced for specific input points or if transitions were not timestamped.
CIP Sync	These modules have an internal clock that is synchronized with a master clock using CIP Sync. This clock is used for time stamping inputs and outputs.
Position Based Control using MAOC	The Motion Axis Output Cam (MAOC) instruction provides position-based control of outputs by using position and velocity information of any motion axis. The 1732E-OB8M8SR module can be specified as the output source for the MAOC instruction, then the MAOC instruction automatically handles the time-based output scheduling and enables it on the eight outputs on the 1732E-OB8M8SR module. The benefit of using output scheduling in this manner is that the resolution of the output control is improved from the motion coarse update rate (typically 1…32 ms) to 100 μs.
Only one owner-controller per module	While multiple controllers can simultaneously own other digital input modules, the module only supports a single owner-controller.

Similar Functionality to Standard ArmorBlock

This chapter focuses on how the module behavior differs from that of other ArmorBlock modules. However, you should be aware of aspects in which the module is similar to standard EtherNet/IP ArmorBlock modules. The following table describes the similarities.

Concept	Description
Ownership	Every module in an ArmorBlock system must be owned by a Logix5000 controller. This owner-controller: • stores configuration data for every module that it owns. • sends the module configuration data to define the module behavior and begin operation with the control system. **This module does not support multiple owner-controllers**.
Using RSLogix 5000 software	The I/O configuration portion of RSLogix 5000 software, v18 or greater, generates the configuration data for each module. Configuration data is transferred to the controller during the program download and subsequently transferred to the appropriate modules. Modules are ready to run as soon as the configuration data has been downloaded. Configure all modules for a given controller using RSLogix 5000 software and download that information to the controller.

Chapter Summary and What's Next

In this chapter, you learned about the differences between this module and other EtherNet/IP ArmorBlock I/O modules. The next chapter describes how to install and wire your module.

Install Your Module

Overview

This chapter shows you how to install and wire the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules. The only tools you require are a flat or Phillips head screwdriver and drill. This chapter includes the following topics:

Install the Module

To install the module:
- Set the network address
- Mount the module
- Connect the I/O, Network, and Auxiliary cables to the module.

Set the Network Address

The I/O block ships with the rotary switches set to 999 and DHCP enabled. To change the network address, you can do one of the following:
- adjust the node address switches on the front of the module.
- use a Dynamic Host Configuration Protocol (DHCP) server, such as Rockwell Automation BootP/DHCP.
- retrieve the IP address from nonvolatile memory.

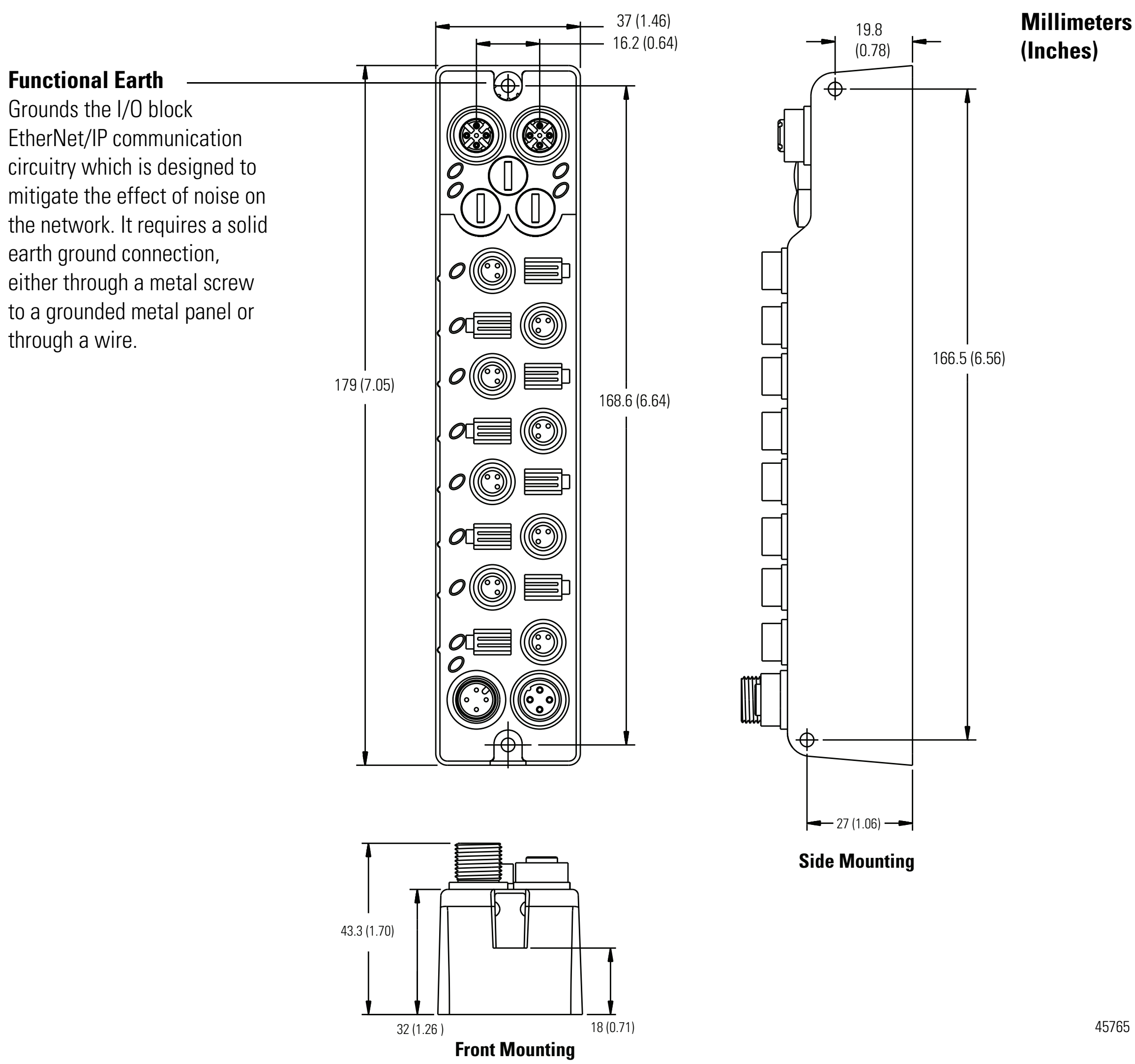
The I/O block reads the switches first to determine if the switches are set to a valid number. To set the network address:

1. Remove power.
2. Remove the switch dust caps.
3. Rotate the three (3) switches on the front of the module using a small blade screwdriver.

4. Line up the small notch on the switch with the number setting you wish to use.
   Valid settings range from 001…254.
5. Replace switch dust caps. Make sure not to over tighten.
6. Reapply power.

Mount the Module

To mount the module on a wall or panel, use the screw holes provided in the module. Refer to the drilling dimensions illustration to guide you in mounting the module.



Install the mounting base as follows:

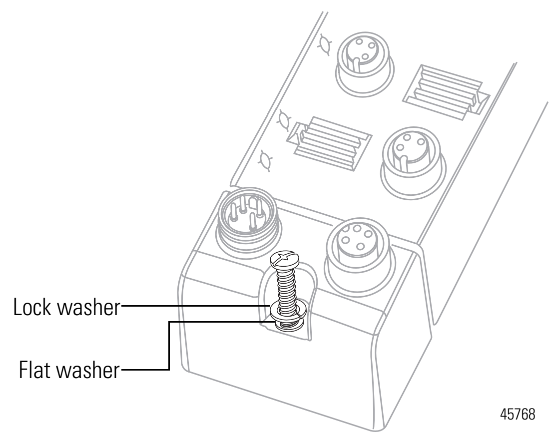
1. Lay out the required points as shown above in the drilling dimension drawing.

2. Drill the necessary holes for #6 (M3) pan head screws.
3. Mount the module using #6 (M3) screws.

Mount the Module in High Vibration Areas

If you mount the module in an area that is subject to shock or vibration, we recommend you use a flat and a lock washer to mount the module. Mount the flat and the lock washer as shown in the mounting illustration. Torque the mounting screws to 0.68 Nm (6 lb-in.).

High Vibration Area Mounting



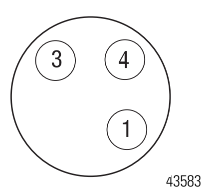
Wire the Module

The 1732E-OB8M8SR and 1732E-IB8M8SOER ArmorBlock EtherNet/IP modules have 3-pin pico-style I/O connectors. We provide caps to cover the unused connectors on your module. Connect the quick-disconnect cord sets you selected for your module to the appropriate ports.

I/O Connectors

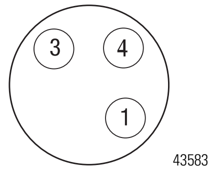
Refer to the pinout diagrams for the I/O connectors.

Pico-style 3-Pin Input Female Connector



(View into connector)
Pin 1 Sensor source voltage
Pin 3 Return
Pin 4 Input

Pico-style 3-Pin Output Female Connector

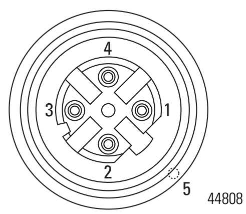


(View into connector)
Pin 1 Sensor Source Voltage
Pin 3 Return
Pin 4 Output

ATTENTION: Sensors/actuators power should not be supplied externally.

Ethernet Connectors

Refer to the pinout diagrams for the network connectors.



(View into connector)
Pin 1Tx+
Pin 2 Rx+
Pin 3 Tx-
Pin 4 Rx-
Pin 5 Shell

IMPORTANT Use the 1585D–M4DC–H: Polyamide small body unshielded mating connectors for the D-Code M12 female network connector.

Note that the distance between the center of each Ethernet connector is 16.2 mm (see dimensions on page 18).
Rockwell Automation recommends the use of suitable cable based on this measurement. Some of the recommended cables are 1585D-M4TBJM-x and 1585D-M4TBDM-x for daisychains.

IMPORTANT Use two twisted pair CAT5E UTP or STP cable.

D-Code M12 Pin	Wire Color	Signal	8-way Modular RJ45 Pin
1	White-Orange	TX+	1
2	White-Green	RX+	3
3	Orange	TX-	2
4	Green	RX-	6

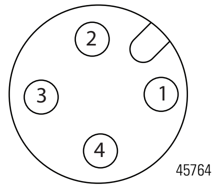
ATTENTION: Make sure all connectors and caps are securely tightened to properly seal the connections against leaks and maintain IP enclosure type requirements.

Power Connectors

Attach the mini-style 4-pin connector to the mini-style 4-pin receptacle as shown below.

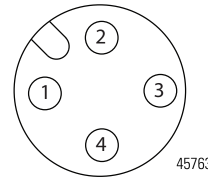
Micro-style 4-Pin Input Male Receptacle

Male Input



(View into receptacle)
Pin 1 Auxiliary power+
Pin 2 Module power+
Pin 3 Module power-
Pin 4 Auxiliary power-

Female Output



IMPORTANT	The maximum current that any pin on the power connectors can carry is 4 A.

The power required by the module is based on a 4-pin micro-style connector system. The module receives its required power through the male connector on the left. A female connector on the right is also provided so that power can be daisy-chained from module to module.

Both modules require two 24V DC (nominal) supplies. These supplies are called the Module Power and the Auxiliary Power. The Module Power powers the microprocessor and Ethernet portions of the module. The Auxiliary Power provides power for the Digital Outputs, the Digital Inputs, and the Sensor Voltage.

Internally, the Module Power and Auxiliary Power are isolated from each other.

The Module Power current required for a module can be estimated as 2.4W/(Module Power Voltage). For example, if the Module Power Voltage is 24V DC, then the Module Power current (Imp) would be,

$$Imp \sim 2.4W/24VDC = 100\ mA\ DC$$

If the power for four modules were daisy-chained together and the voltage is 24V DC, then the Module Power current through the first connector in the daisy-chain would be 4 x Im ~ 400 mA which is less than 4 A, so Module Power current is within acceptable limits.

The Auxiliary Power current is more complicated. The equation is below:

$$Iap \sim Iapm + Isp0 + Isp1 + Isp2 + Isp3 + Isp5 + Isp5 + Isp6 + Isp7 + IDO0 + IDO1 + IDO2 + IDO3 + IDO4 + IDO5 + IDO6 + IDO7 + IAPDC$$

Where:
Iap is the Auxiliary Power current through the first connector in the daisy-chain.
Iapm is the Auxiliary Power current required by the module itself.
IspN is the Sensor Power current for Digital Input N (0…7).
IDON is the Digital Output current for Digital Output N (0…7).
IAPDC is the Auxiliary Power current requirement for the remaining modules in the daisy-chain.
Iapm can be approximated by 0.5W/(Auxiliary Power Voltage).

The table Auxiliary Power Calculation shows the resulting Auxiliary Power current calculation for a system of four modules. The Auxiliary Power voltage is 24V DC in this example. As can be seen in the cell with value set in bold, the Auxiliary Power current through the first connector in the daisy-chain is 3.898A which is less than 4A, so this system is adequate.

Auxiliary Power Calculation

	Module 1	Module 2	Module 3	Module 4
IAPDC	3.108A	2.772A	1.301A	0.000A
Iapm	0.021A	0.021A	0.021A	0.021A
Isp0	0.000A	0.000A	0.300A	0.050A
Isp1	0.000A	0.000A	0.000A	0.000A
Isp2	0.000A	0.000A	0.000A	0.250A
Isp3	0.000A	0.000A	0.000A	0.000A
Isp4	0.000A	0.000A	0.000A	0.000A
Isp5	0.000A	0.000A	0.000A	0.000A
Isp6	0.000A	0.000A	0.000A	0.000A
Isp7	0.000A	0.000A	0.000A	0.000A
IDO0	0.270A	0.025A	0.500A	0.025A
IDO1	0.200A	0.290A	0.300A	0.500A
IDO2	0.300A	0.000A	0.250A	0.300A
IDO3	0.000A	0.000A	0.100A	0.125A
IDO4	0.000A	0.000A	0.000A	0.030A
IDO5	0.000A	0.000A	0.000A	0.000A
IDO6	0.000A	0.000A	0.000A	0.000A
IDO7	0.000A	0.000A	0.000A	0.000A
Iapm	**3.898A**	3.108A	2.772A	1.301A

ATTENTION: To comply with the CE Low Voltage Directive (LVD), this equipment and all connected I/O must be powered from a source compliant with the following:
Safety Extra Low Voltage (SELV) or Protected Extra Low Voltage (PELV).

ATTENTION: To comply with UL restrictions, this equipment must be powered from a source compliant with the following: Limited Voltage.

ATTENTION: The device meets UL Type 1 Enclosure rating.

Chapter Summary and What's Next

In this chapter, you learned how to install and wire your module. The following chapter describes how to configure your module to communicate on the EtherNet/IP network by providing an IP address, gateway address, and Subnet mask.

Configure the Module for Your EtherNet/IP Network

Introduction

Before using the modules in an EtherNet/IP network, you need to configure them with an IP address, subnet mask, and optional Gateway address. This chapter describes these configuration requirements and the procedures for providing them. Here are the ways you can do this:

- Use the Rockwell Automation BootP/DHCP utility, version 2.3 or greater, that ships with RSLogix 5000 or RSLinx software. You can also use this utility to reconfigure a device whose IP address must be changed.
- Use a third party DHCP (Dynamic Host Configuration Protocol) server.
- Use the Network Address switches.
- Have your network administrator configure the module via the network server.

See the table for a list of where to find specific information in this chapter.

Configuration Requirements

Before you can use your module, you must configure its IP address, its subnet mask, and optionally, gateway address. You have the option to use the Rockwell Automation BootP/DHCP utility, version 2.3 or greater, to perform the configuration. You also have the option to use a DHCP server or the network address switches to configure these parameters.

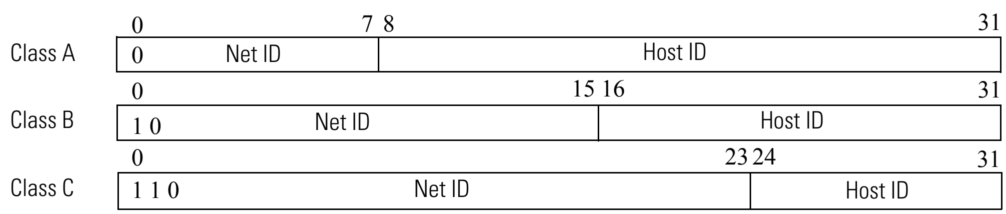
If the module needs to be reset to factory defaults, set the switches on the module to the value 888 and then cycle power to the module.

IMPORTANT	If using the BootP/DHCP utility, you will need to know the Ethernet hardware address of your module. Rockwell Automation assigns each module a unique 48-bit hardware address at the factory. The address is printed on a label on the side of your module. It consists of six hexadecimal digits separated by colons. This address is fixed by the hardware and cannot be changed. If you change or replace the module, you must enter the new Ethernet hardware address of the module when you configure the new module.

IP Address

The IP address identifies each node on the IP network (or system of connected networks). Each TCP/IP node on a network (including your module) must have a unique IP address.

The IP address is 32 bits long and has a net ID part and a Host ID part. Networks are classified A, B, C, (or other). The class of the network determines how an IP address is formatted.



You can distinguish the class of the IP address from the first integer in its dotted-decimal IP address as follows:

Classes of IP Addresses

Range of first integer	Class	Range of first integer	Class
0…127	A	192…223	C
128...191	B	224…255	other

Each node on the same logical network must have an IP address of the same class and must have the same net ID. Each node on the same network must have a different Host ID thus giving it a unique IP address.

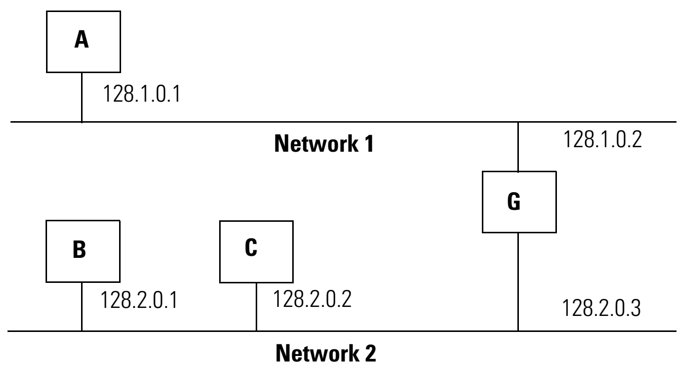
IP addresses are written as four decimal integers (0…255) separated by periods where each integer gives the value of one byte of the IP address.

EXAMPLE	For example, the 32-bit IP address: 10000000 00000001 00000000 00000001 is written as 128.1.0.1.

Gateway Address

This section applies to multi-network systems. If you have a single network system, go to the next section.

The gateway address is the default address of a network. It provides a single domain name and point of entry to the site. Gateways connect individual networks into a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks. The following figure shows gateway G connecting Network 1 with Network 2.



When host B with IP address 128.2.0.1 communicates with host C, it knows from C's IP address that C is on the same network. In an Ethernet environment, B then resolves C's IP address into a hardware address (MAC address) and communicates with C directly.

When host B communicates with host A, it knows from A's IP address that A is on another network (the net IDs are different). In order to send data to A, B must have the IP address of the gateway connecting the two networks. In this example, the gateway's IP address on Network 2 is 128.2.0.3.

The gateway has two IP addresses (128.1.0.2 and 128.2.0.3). The first must be used by hosts on Network 1 and the second must be used by hosts on Network 2. To be usable, a host's gateway must be addressed using a net ID matching its own.

Subnet Mask

The subnet mask is used for splitting IP networks into a series of subgroups, or subnets. The mask is a binary pattern that is matched up with the IP address to turn part of the Host ID address field into a field for subnets.

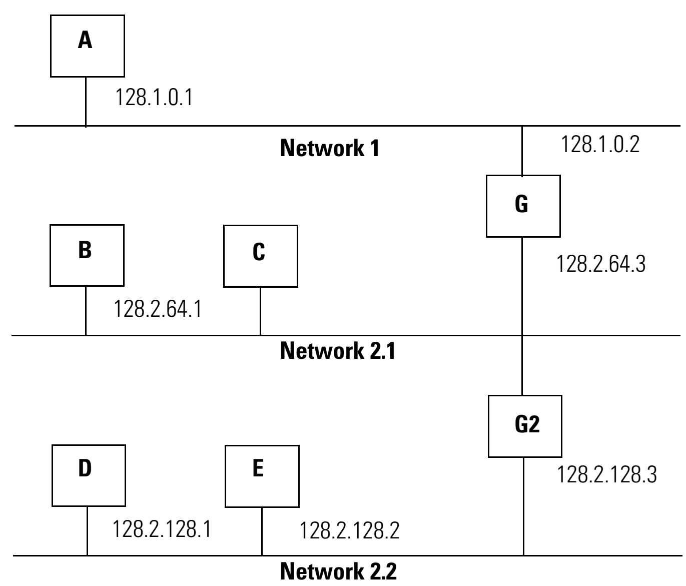
EXAMPLE Take Network 2 (a Class B network) in the previous example and add another network. Selecting the following subnet mask would add two additional net ID bits, allowing for four logical networks:

1111111 11111111 **11**000000 00000001 = 255.255.192.0

These two bits of the host ID used to extend the net ID

Two bits of the Class B host ID have been used to extend the net ID. Each unique combination of bits in the part of the Host ID where subnet mask bits are 1 specifies a different logical network.

The new configuration is:



A second network with Hosts D and E was added. Gateway G2 connects Network 2.1 with Network 2.2.
Hosts D and E use Gateway G2 to communicate with hosts not on Network 2.2.
Hosts B and C use Gateway G to communicate with hosts not on Network 2.1.
When B is communicating with D, G (the configured gateway for B) routes the data from B to D through G2.

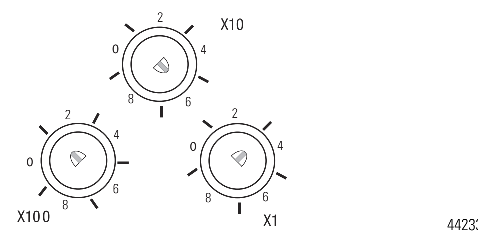
Set the Network Address

The I/O block ships with the rotary switches set to 999 and DHCP enabled. To change the network address, you can do one of the following:

1. Adjust the switches on the front of the module.
2. Use a Dynamic Host Configuration Protocol (DHCP) server, such as Rockwell Automation BootP/DHCP.
3. Retrieve the IP address from nonvolatile memory.

The I/O block reads the switches first to determine if the switches are set to a valid number. Set the network address by adjusting the 3 switches on the front of the module. Use a small blade screwdriver to rotate the switches. Line up the small notch on the switch with the number setting you wish to use. Valid settings range from 001…254.

Network Address Example



When the address switches are set to a value of 1, the default gateway address is 0.0.0.0. When the address switches are set from 002…254, the default gateway address is 192.168.1.1.

When the I/O block uses the network address set on the switches, the I/O block does not have a host name assigned to it or use any Domain Name Server.

If the switches are set to an invalid number (for example, 000 or a value greater than 254, excluding 888), the I/O block checks to see if DHCP is enabled. If DHCP is enabled, the I/O block asks for an address from a DHCP server. The DHCP server also assigns other Transport Control Protocol (TCP) parameters.

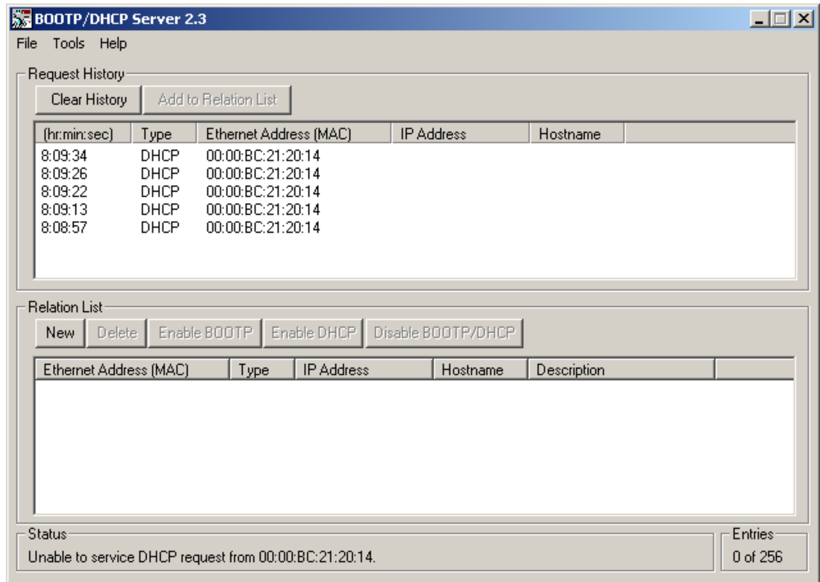
If DHCP is not enabled, and the switches are set to an invalid number, the I/O block uses the IP address (along with other TCP configurable parameters) stored in nonvolatile memory.

Use the Rockwell Automation BootP/DHCP Utility

The Rockwell Automation BootP/DHCP utility is a standalone program that incorporates the functionality of standard BootP/DHCP software with a user-friendly graphical interface. It is located in the Utils directory on the RSLogix 5000 installation CD. The module must have DHCP enabled (factory default and the network address switches set to an illegal value) to use the utility.

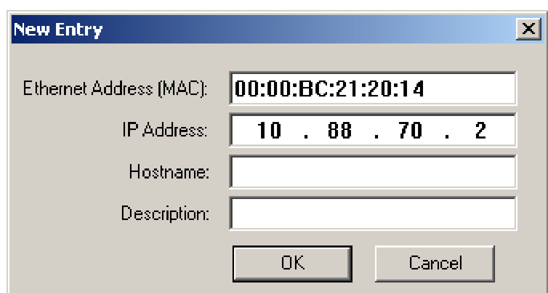
To configure your module using the BootP/DHCP utility, perform the following steps:

1. Run the BootP/DHCP software.
   The BOOTP/DHCP Request History dialog appears showing the hardware addresses of devices issuing BootP/DHCP requests.



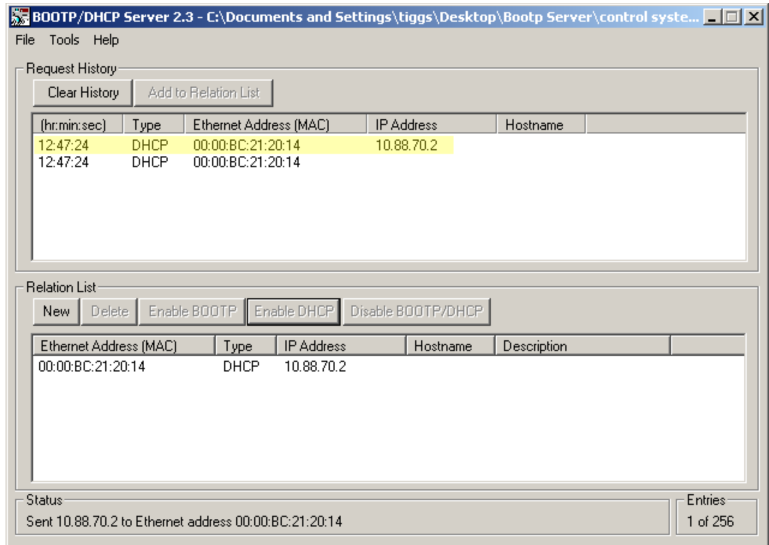
2. Double-click the hardware address of the device you want to configure.

   The New Entry dialog appears showing the device's Ethernet Address (MAC).



3. Enter the IP Address you want to assign to the device and click OK.

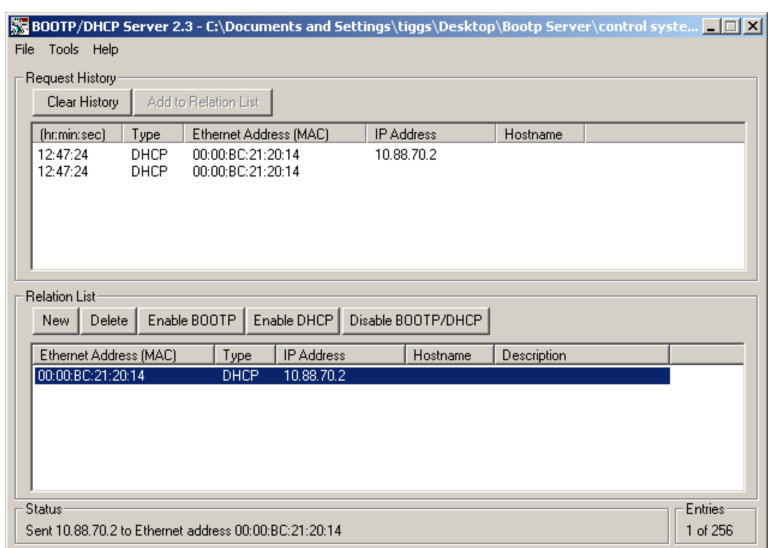
The device is added to the Relation List, displaying the Ethernet Address (MAC) and corresponding IP Address, Hostname and Description (if applicable).



When the IP address assignment is made, the address displays in the IP Address column in the Request History section.

4. To assign this configuration to the device, highlight the device in the Relation List panel and click Disable BOOTP/DHCP. When power is cycled to the device, it uses the configuration you assigned and does not issue a DHCP request.

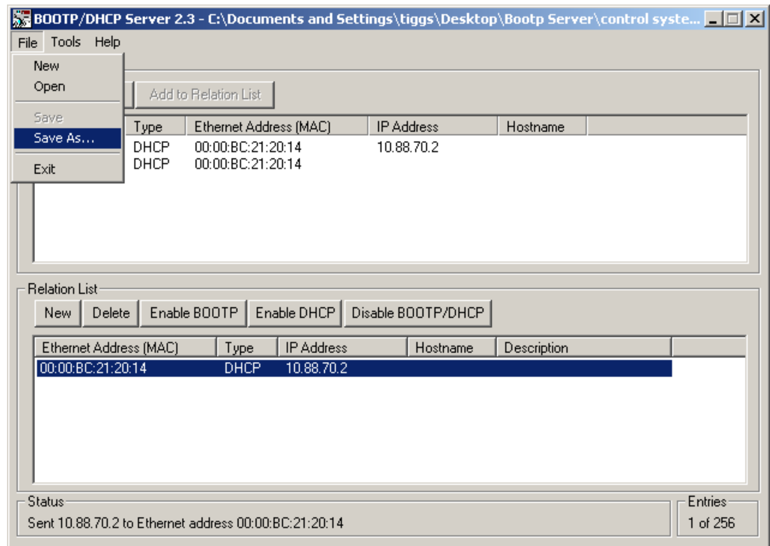
TIP	To enable DHCP for a device that has had DHCP disabled, highlight the device in the Relation List and click Enable DHCP. You must have an entry for the device in the Relation List panel to re-enable DHCP.



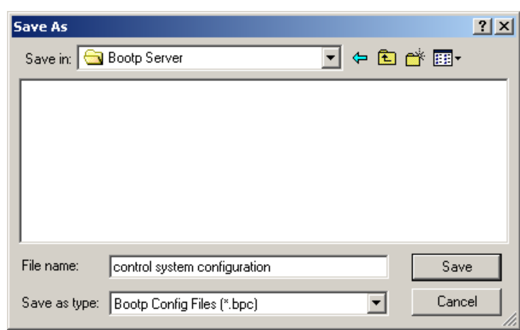
Save the Relation List

You can save the Relation List to use later. To save the Relation List do the following:

1. Select Save As... from the File menu.



The Save As dialog box appears.



2. Select the folder you want to save the list to.
3. Enter a file name for the Relation List (for example, control system configuration) and click Save.

   If you want to see your saved file names in the Open dialog box, save your files using the default file type (*.bpc).

Use DHCP Software to Configure Your Module

Dynamic Host Configuration Protocol (DHCP) software automatically assigns IP addresses to client stations logging onto a TCP/IP network. DHCP is based on BootP and maintains some backward compatibility. The main difference is that BootP was designed for manual configuration, while DHCP allows for dynamic allocation of network addresses and configurations to newly attached devices.

Be aware that a DHCP server typically assigns a finite lease time to the offered IP address. When 50 percent of the leased time has expired, the module will attempt to renew its IP address with the DHCP server. The module could be assigned a different IP address, which would cause communicating with the ControlLogix controller to cease.

ATTENTION: To avoid unintentional control, the module must be assigned a fixed IP address. The IP address of this module should not be dynamically provided. If a DHCP server is used, it must be configured to assign a fixed IP address for your module.

ATTENTION: Failure to observe this precaution may result in unintended machine motion or loss of process control.

Chapter Summary and What's Next

In this chapter, you learned how to configure the module to communicate on your EtherNet/IP network by providing an IP address, gateway address, and Subnet mask. The next chapter describes an example application in which you configure discrete I/O.

Notes:

Configure the Module Using RSLogix 5000 Software

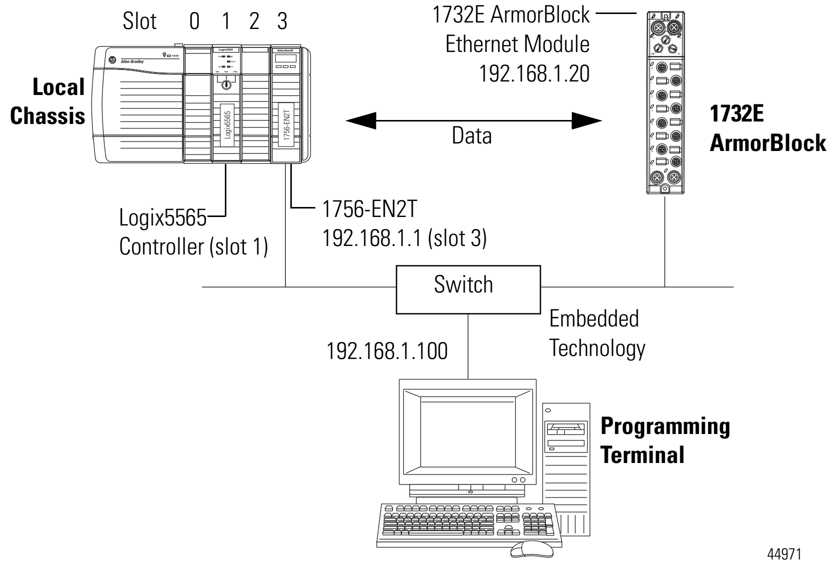
Introduction

This chapter guides you through the steps required to configure your 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules using the RSLogix 5000 software. Note that the modules presented in this chapter are configured using RSLogix 5000 software, version 18 or later.

The configuration of the two modules through the RSLogix 5000 software involve the same procedure. Note, however, that the two modules have different Module Definition properties and Configuration tabs. Both are distinctly covered in this chapter.

Set Up the Hardware

In this example, a ControlLogix chassis contains the Logix5565 processor in slot 1 and a 1756-EN2T bridge module in slot 3. The 1732E ArmorBlock module is mounted remotely.



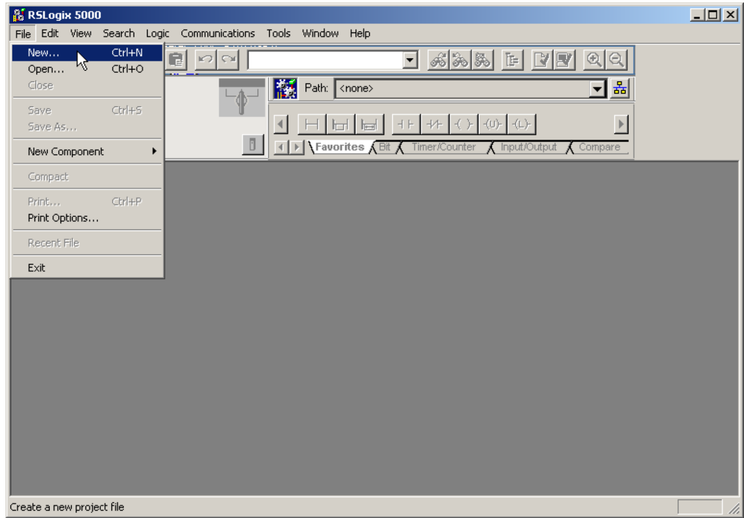
To work along with this example set up your system as shown.

- Note that in the example application, the Logix5565 controller and 1756-EN2T module (firmware version 2.3 or higher) are assumed to be in the slots shown.
- Verify the IP addresses for your programming terminal, 1756-EN2T module and 1732E ArmorBlock Ethernet module.
- Verify that you connected all wiring and cabling properly.
- Be sure you configured your communication driver (for example, AB_ETH-1 or AB-ETHIP-1) in RSLinx software.

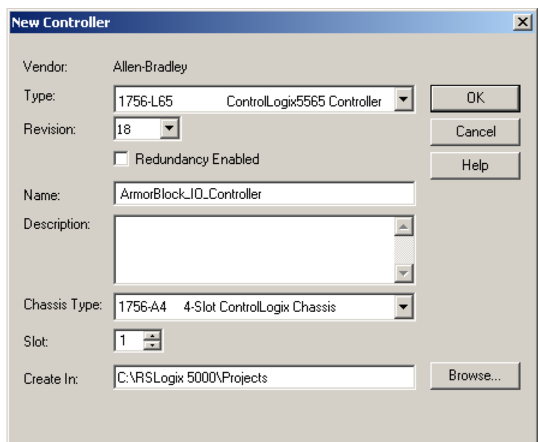
Create the Example Application

Perform the following steps to create the example application:

1. From the File menu, select New.



The New Controller dialog opens.



2. Enter an appropriate name for the Controller, for example, ArmorBlock_IO_Controller.
3. Select the correct version, chassis type, and slot number of the controller, and the folder where you want to save the RSLogix 5000 software file (Create In). The Description is optional.

   To use redundancy in your system, select the Redundancy Enabled checkbox.
4. Click OK.

Configure Your I/O Module

You must configure your module upon installation. The module will not work until it has been configured with at least the default configuration.

RSLogix 5000 Configuration Software

You must use **RSLogix 5000**, **version 18 or later**, to configure your module. You have the option of accepting default configuration for your module or writing point-level configuration specific to your application.

Both options are explained in detail, including views of software screens, in this chapter.

Overview of the Configuration Process through RSLogix 5000

When you use the RSLogix 5000 software to configure a module, you must perform the following steps:

1. Add the Local EtherNet/IP Bridge (1756-EN2T, 1756-EN2TR, or 1756-EN3TR) to your project's I/O Configuration.
2. Add the 1732E-IB8M8SOER or 1732E-OB8M8SR module as a child of the 1756-EN2T module.
3. Accept the default configuration or change it to specific configuration for the module.
4. Edit configuration for a module when changes are needed.

Add a New Bridge and Module to Your RSLogix 5000 Project

After you have started RSLogix 5000 software and created a controller, you must add a new bridge and a new module to your project. The bridge allows your module to communicate with the controller.

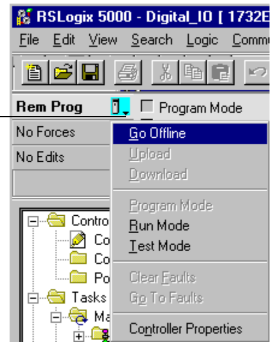
The wizard allows you to create a new module and write configuration. You can use default configuration or write specific configuration for your application.

IMPORTANT	Click Help on the configuration dialogs shown in this section if you need assistance in selecting and setting the parameters.

Add the Local EtherNet/IP Bridge to the I/O Configuration

1. If necessary, go offline.

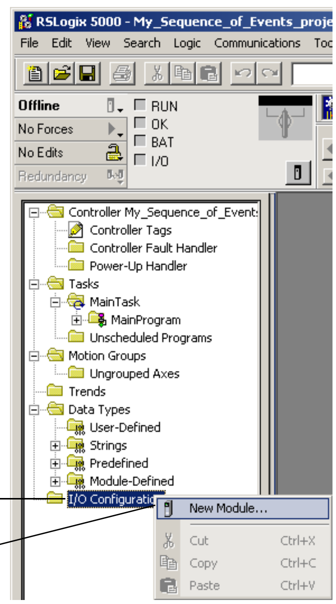
If you are not offline, use this pull-down menu to go offline.



2. Add the EtherNet/IP Bridge to your RSLogix 5000 project.

A. Right-click I/O Configuration.

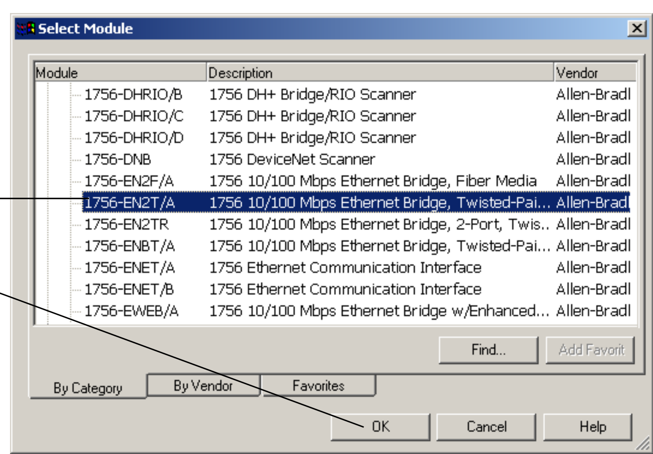
B. Select New Module.



3. Expand Communications and select the new module in the Select Module dialog that appears. Select the 1756-EN2T EtherNet/IP Bridge.

A. Select the 1756-EN2T EtherNet/IP Bridge.

B. Click OK.



4. The New Module dialog opens.
Configure the bridge module as illustrated below.

A. Name the bridge.

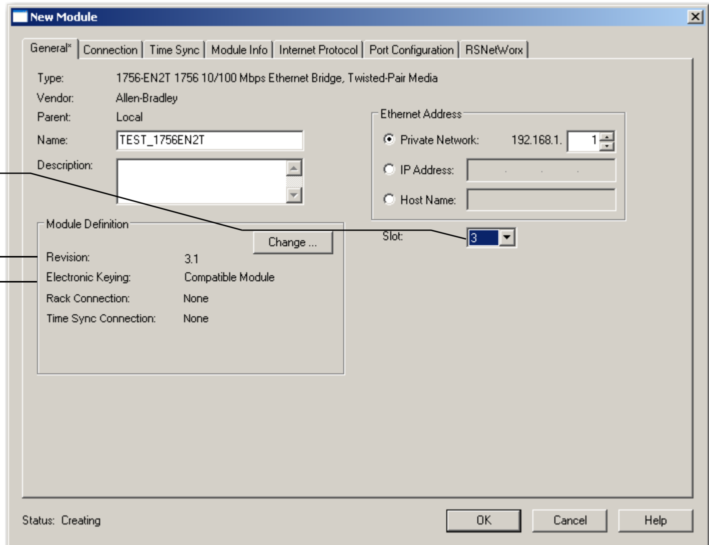
B. Enter the IP address.

C. Select slot 3 for the EtherNet/IP bridge.

D. Make sure the Minor Revision number matches your module revision number.

E. Choose an Electronic Keying method. For more information, see page 48.

F. Click OK.



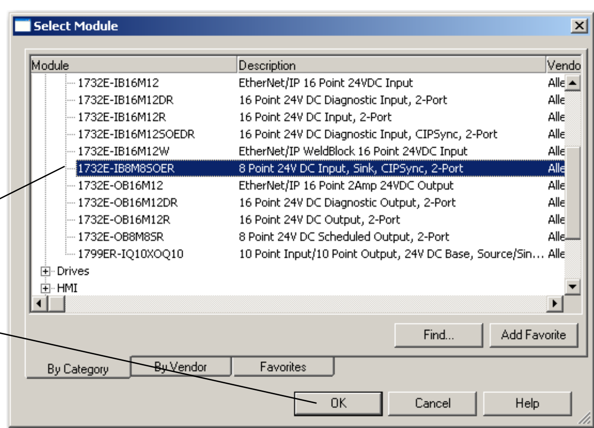
The local 1756-EN2T communication module will communicate with the 1732E ArmorBlock module on EtherNet. Before you can communicate with your module, you need to add it as a *child* of the 1756-EN2T communication module. For more information about using 1756 controller and EtherNet/IP products, see publication ENET-UM001.

Add the I/O module as a child of the 1756-EN2T module

1. Right-click the Ethernet folder that appears below the 1756-EN2T bridge you added to the I/O Configuration tree and select New Module.
2. Expand Digital in the Select Module dialog that appears.
Select the 1732E-IB8M8SOER or the 1732E-OB8M8SR module.

A. Select the I/O module.

B. Click OK.



TIP	If the 1732E-IB8M8SOER, 1732E-OB8M8SR modules are not listed in the digital section of the Select Module dialog, you may need to download the Add-On Profile (AOP) and install it as an add-on to RSLogix 5000. The AOP file can be downloaded from: support.rockwellautomation.com/controlflash/LogixProfiler.asp

3. The New Module dialog appears.
Fill in the Module Properties information as shown, and then click OK.

Module Properties Values

Field Name	Value
Name	TEST_1732EIB8M8SOER or TEST_1732EOB8M8SR
IP address	192.168.1.20
Electronic keying	Compatible Module
Connection Format	Data.This field does not exist for the 1732E-OB8M8SR module.
Revision	1.1
Timestamp	Per Point

You can either accept or change the default configuration as shown.

1732E-IB8M8SOER

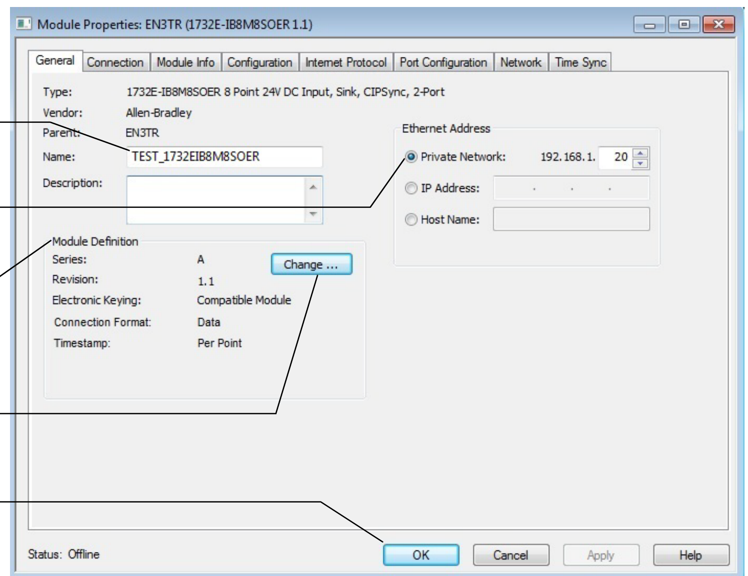
A. Name the module.

B. Enter theIP address of the module as shown.

C. Make sure the Module Definition information matches this example.

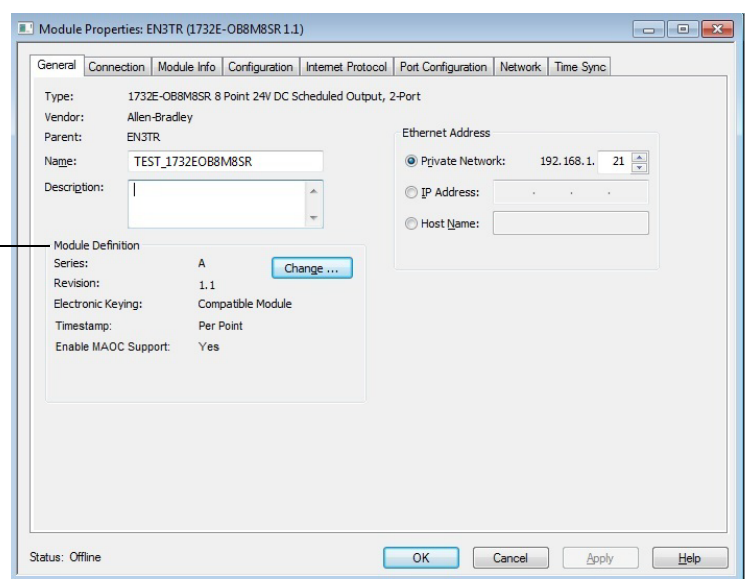
D. Click Change... to edit the Module Definition for your module before downloading the program to the controller.

E. Click OK to accept the default configuration.



1732E-OB8M8SR

On the General tab, the 1732E-OB8M8SR module does not have a Connection Format field and has an Enable MAOC Support field under the Module Definition section.



Use the Default Configuration

If you use the default configuration and click OK, you are done.
You can skip to Download Your Configuration on page 43 for instructions on how to download your default configuration to the controller.

Change the Default Configuration

If you click Change... in step D on page 40, you can change the Module Definition information. Note that the 1732E-IB8M8SOER and 1732E-OB8M8SR modules have slightly different Module Definition dialogs. The1732E-OB8M8SR module does not have the Connection field. The screenshots below will guide you through the dialog.

Select tabs on the Module Properties dialog to edit specific configuration for your module in RSLogix 5000 software, for example the Configuration tab.

Some of the screens that appear during this initial module configuration process are blank (such as Module Info, Network, and Time Sync) and are not shown here. These screens mostly provide information and status and can be important during online monitoring. To see these screens in use, see Chapter 10, Troubleshoot the Module on page 87.

When you click Change..., the Module Definition dialog is shown. Through the Module Definition dialog, you can:

A. Select the module series.

B. Make sure the Major and Minor Revision numbers match your module's revision.

C. Choose an Electronic Keying method.

D. On the **1732E-IB8M8SOER** module, select the Connection type. Available options are Data and Listen Only. (This field is not available for 1732E-OB8M8SR module.)

E. Select the Timestamp Format. On the **1732E-IB8M8SOER module,** the options available are Per Point, First In First Out, and None. On the **1732E-OB8M8SR module,** the options available are Per Point, and None. For more information about timestamping format, see Use the Sequence of Events Input and Scheduled Output Modules on page 63.

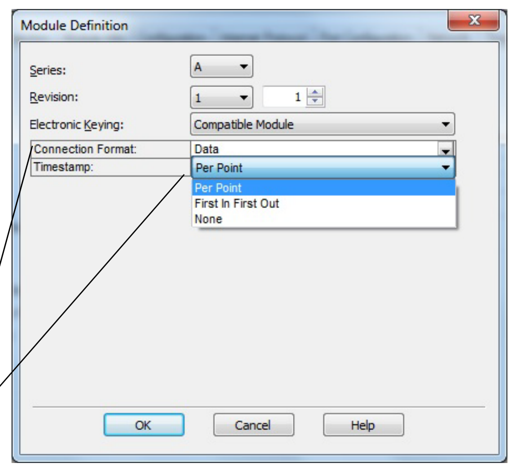
F. On the **1732E-OB8M8SR module,** select whether MAOC support is required (Yes) or not (No).

G. Click OK to return to the General tab of the Module Properties dialog.

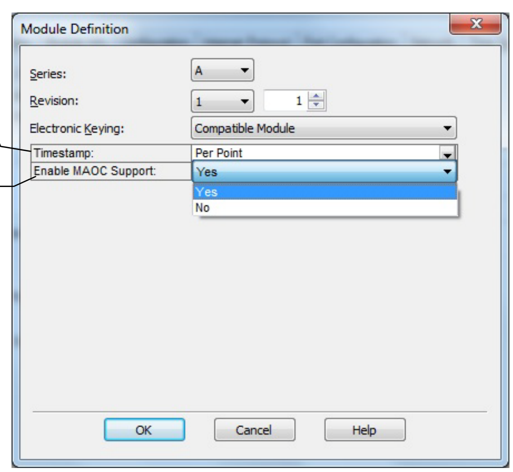
H. On the General tab, you can click OK to close the Module Properties dialog and download your configuration, or

I. Click the Connection tab to configure connection properties.

On 1732E-IB8M8SOER



On 1732E-OB8M8SR



From the Connection tab, you can:

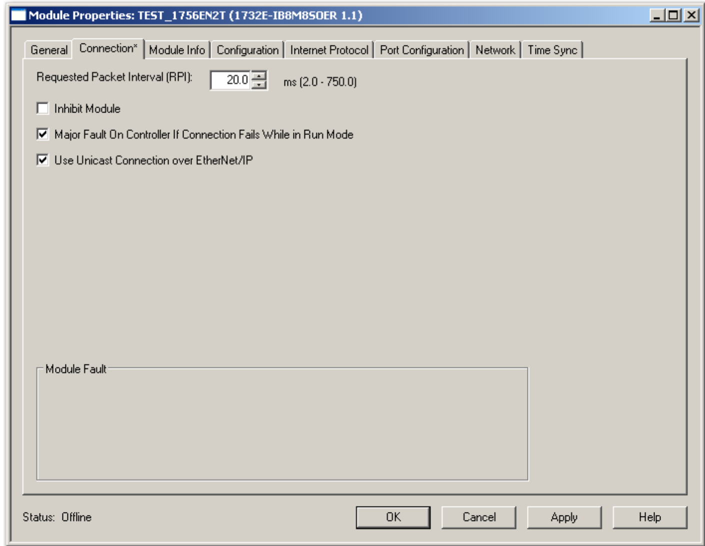
A. Change the RPI.

B. Inhibit the module. For more information on Module Inhibiting, see page 49.

C. Make sure a Major Fault occurs on the module's owner-controller if there is a connection failure between the module and the controller.

D. Click the Configuration tab to go to the next screen, or

E. Click OK to close the Module Properties dialog and download your configuration.



The next tab available in the Module Properties dialog is the Configuration tab. Note that the 1732E-IB8M8SOER and 1732E-OB8M8SR modules do not have the same Configuration tabs. The following screenshots will guide you through the Configuration tabs of each.

For the **1732E-IB8M8SOER** module, you can do the following through the Configuration tab:

A. Set the Input Filter Times. For more information on Input Filters, see page 55

B. Enable Timestamp Capture for all input points or for specific points. For more information on Timestamp Capture, see page 54.

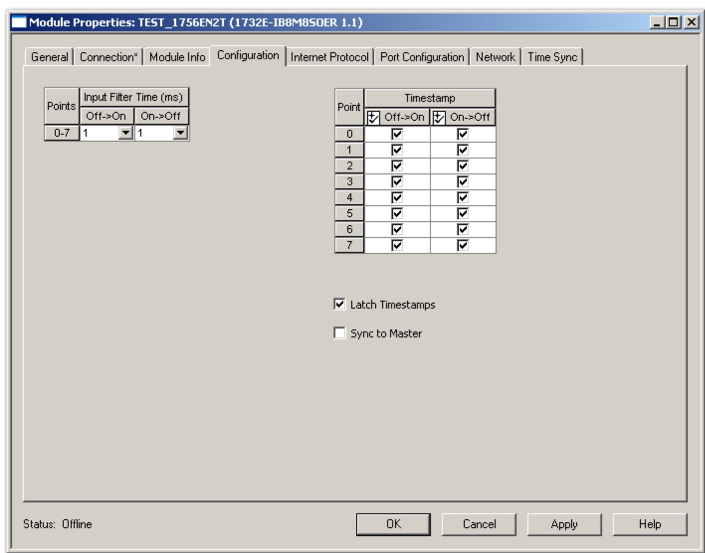
C. Click the box Latch Timestamps to enable Timestamp Latching. For more information, see page 54.

D. Click the box to Sync to Master. The feature is not enabled by default.

E. Click OK to close the Module Properties dialog and download your configuration.

F. Click Help to access the RSLogix 5000 software Add-On Profile help for descriptions of tabs that are not required for setting up your module.

1732E-IB8M8SOER Configuration Tab



For the **1732E-OB8M8SR**, you can do the following through the Configuration tab:

A. Set the Output State during Program Mode and Fault Mode for Points 0…7.

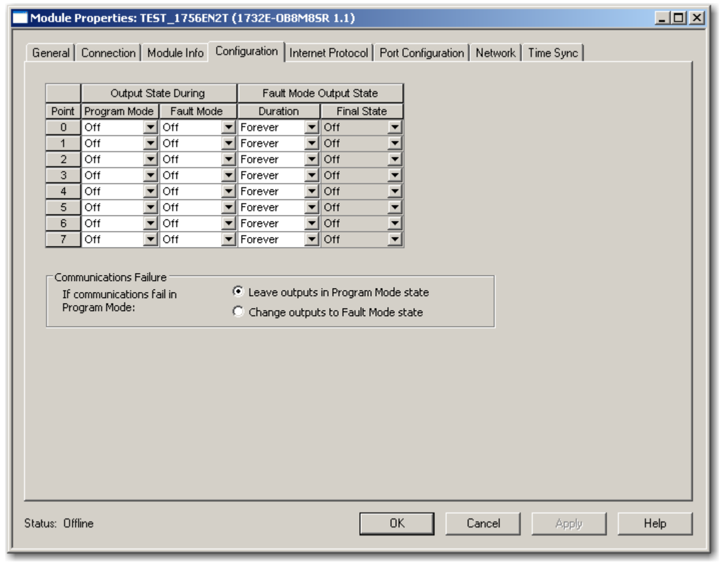
B. Set the Fault Duration and Fault Final State for Points 0…7.
Grayed out unless Hold Duration is something other than “Forever”.

C. Click OK to close the Module Properties dialog and download your configuration, or

D. Click Help to access the RSLogix 5000 software Add-On Profile help for descriptions of tabs that are not required for setting up your module.

For more information on Output States, see Configurable Point-Level Output Fault States on page 61.

1732E-OB8M8SR Configuration Tab



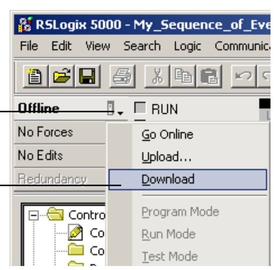
Download Your Configuration

After you write configuration for your module, the module does not use this configuration until you download it to the owner-controller. The download transfers the entire program to the controller, overwriting any existing program.

Download module configuration as shown below:

A. Click here to see the pull-down menu.

B. Click download.



Depending on your application, a variety of RSLogix 5000 software screens may appear to choose a path to your ControlLogix controller and to verify the download. Navigate those screens as best fits your application.

This completes the download process.

Edit Your Configuration

After you have set configuration for a module, you can review and change your choices. You can change configuration data and download it to the controller while online. This is called **dynamic reconfiguration**.

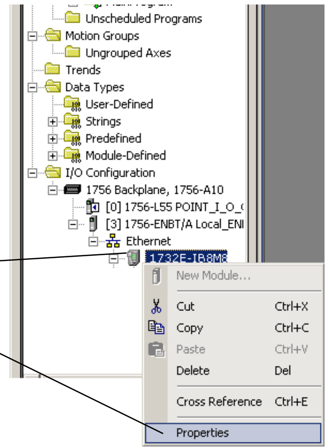
Your freedom to change some configurable features, though, depends on whether the controller is in Remote Run Mode or Program Mode.

IMPORTANT	Although you can change configuration while online, you must go offline to add or delete modules from the project.

The editing process begins on the main page of RSLogix 5000 software:

A. Right-click the module.

B. Select Properties.



The General tab of the Module Properties dialog appears.

Click the tab of the page that you want to view or reconfigure and make any appropriate changes, as shown in the example.

The Internet Protocol tab is grayed out if you are offline. Through this tab, you can do the following:

A. Specify Internet Protocol Settings. This allows you to set manual or automatic configuration for your IP settings.

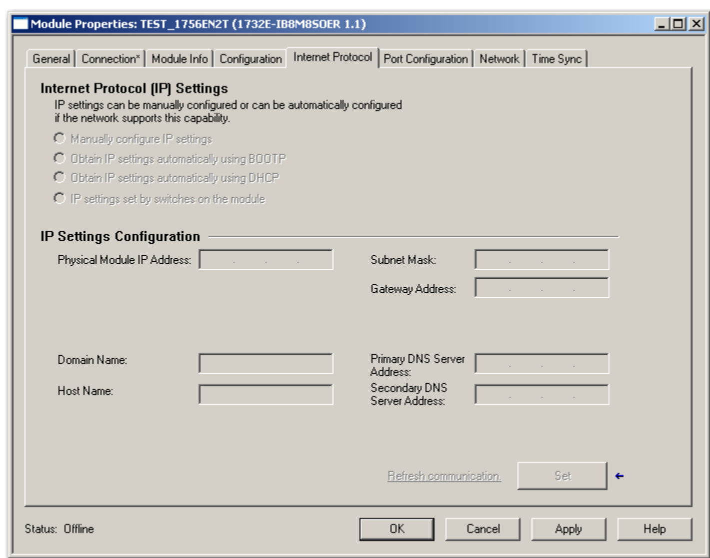
B. To manually configure, specify IP Settings Configuration by providing the following information:

- Physical Module IP address
- Subnet Mask
- Gateway Address
- Domain Name
- Host Name
- Primary and Secondary DNS Server

C. If you make changes in Step A or Step B, then click Set. Changes will not take effect until you reset the module or cycle the power to the module.

D. Click OK to close the Module Properties dialog and download your configuration, or

E. Click Port Configuration tab to go to the next screen.



The Port Configuration screen is grayed out unless you are online with the controller and module. On this screen, you can:

A. Enable or disable external ports.

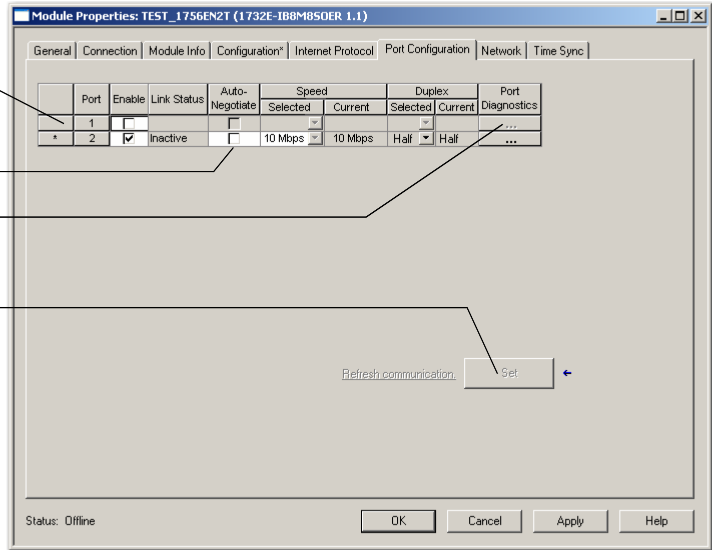
B. Select Auto-negotiate on enabled ports. If Auto-negotiate is disabled then select the correct speed and duplex.

C. Click Port Diagnostics to display the Port Diagnostics dialog.

D. If you make changes in Step A or Step B then click Set. Changes will not take effect until you reset the module or cycle the power to the module.

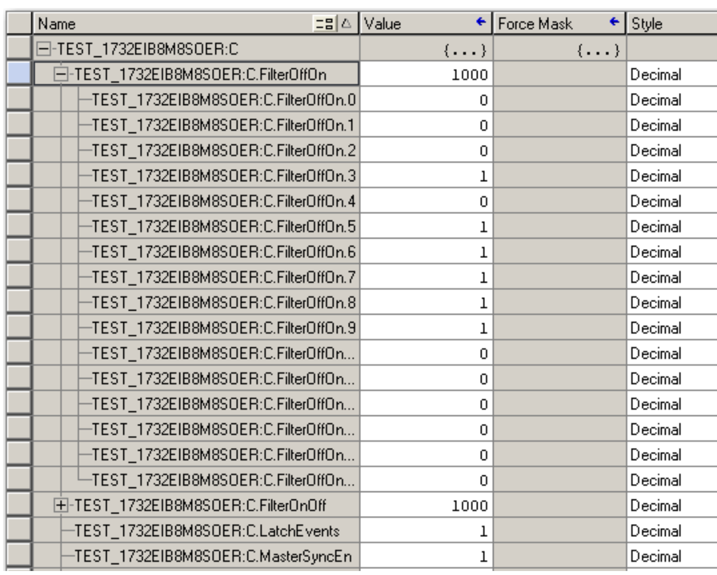
E. Click the Network tab to see the next screen, or

F. Click OK to close the Module Properties dialog and download your configuration.



Access Module Data in RSLogix 5000 Software

Use the following information to use the 1732E-IB8M8SOER, 1732E-OB8M8SR data in the ladder logic program.



Name	Value	Force Mask	Style
-TEST_1732EIB8M8SOER:C	{...}	{...}	
-TEST_1732EIB8M8SOER:C.FilterOffOn	1000		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.0	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.1	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.2	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.3	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.4	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.5	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.6	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.7	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.8	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn.9	1		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
TEST_1732EIB8M8SOER:C.FilterOffOn...	0		Decimal
+TEST_1732EIB8M8SOER:C.FilterOnOff	1000		Decimal
TEST_1732EIB8M8SOER:C.LatchEvents	1		Decimal
TEST_1732EIB8M8SOER:C.MasterSyncEn	1		Decimal

Use the controller tags in your ladder program to read input data or write output data.

- For RSLogix 5000 programming instructions, refer to RSLogix 5000 Getting Results, publication no. 9399-RLD300GR.
- For ControlLogix controller information, refer to ControlLogix System User Manual, publication no. 1756-UM001.

Configure RSLogix 5000 and the 1756-EN2T Communication Module for CIP Sync

If you are using RSLogix 5000 software version 17, follow these steps to configure the 1756-EN2T communication module to be the PTP (CIP Sync) master clock.

1. In your web browser, go to the Rockwell Automation Sample Code Library at http://samplecode.rockwellautomation.com/idc/groups/public/documents/webassets/sc_home_page.hcst.
   The Search Our Sample Code Library page appears.
2. In the Filename/ID field enter **MMS_048132**.
3. Click Search.
   The 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modulesmodule synchronizes to the grandmaster clock as a child module as described in the document.

If you are using RSLogix 5000 version 18 or greater, refer to publication IA-AT003 for instructions on configuring the 1756-EN2T communication module and the ContolLogix processor so that the processor can function as the PTP (CIP Sync) master clock.

Chapter Summary and What's Next

In this chapter, you read about configuring your module in RSLogix 5000. The next chapter describes the module features.

Common Features of the 1732E-IB8M8SOER and 1732E-OB8M8SR Modules

Introduction

This chapter describes the features **common** to both the 1732E Sequence of Events Input and Scheduled Output modules.

To learn more about the features specific to the two modules, see:

- Specific Features of the 1732E-IB8M8SOER Sequence of Events Input Module on page 53
- Specific Features of the 1732E-OB8M8SR Scheduled Output Module on page 59

Communications Format

The communications format determines what operational mode your module uses and, consequently, what tags RSLogix 5000 generates when configuration is complete. Once a module is created, you cannot change the communications format unless you delete and recreate the module.

The communication format determines:

- what type of configuration options are made available.
- what type of data is transferred between the module and its owner-controller.
- what tags are generated when the configuration is complete.

Once a module is created, you cannot change the communication format unless you delete and recreate the module. The communication format also defines the connection between the controller writing the configuration and the module itself. The number and type of choices varies depending on what module you are using and whether it is in a local or remote chassis.

The table describes the communication formats used with the modules.

Data Return	Communication Format	Module
Module returns input data with the value of the system clock (from its local chassis) when the input data changes.	CIPSync/PTP time input data	1732E-IB8M8SOER
The owner-controller sends the module output data and a CIPSync (PTP) time value	Scheduled output data	1732E-OB8M8SR

Electronic Keying

Electronic keying allows the ControlLogix system to control what modules belong in the configured system.

During module configuration, you must choose one of the following keying options for your module:

- Exact Match
- Compatible Module
- Disable Keying

When the controller attempts to connect to and configure a module (for example, after program download), the module compares the following parameters before allowing the connection and configuration to be accepted:

- Vendor
- Product Type
- Product Code
- Major Revision – Change that affects the module function or RSLogix 5000 interface
- Minor Revision – Change that does not affect the module's intended function or RSLogix 5000 interface

The comparison is made between the keying information present in the module and the keying information in the controller program, preventing the inadvertent operation of a system with the wrong module. For example, if you select Exact Match and a module with revision 1.2 is placed in a location configured for a module with revision 1.4, the controller does not make a connection to the new module because of the mismatched revisions.

The following table describes the keying options available with your module.

Keying option	Definition
Exact Match	All of the parameters listed above must match or the inserted module rejects a connection to the controller.
Compatible Module	The Compatible Module mode allows the module to determine whether it can emulate the module defined in the configuration sent from the controller. Some modules can emulate older revisions. The module will accept the configuration if the configuration's major.minor revision is less than or equal to the physical module's revision. For example, if the configuration contains a major.minor revision of 1.7, the module must have a firmware revision of 1.7 or higher for a connection to be made. When a module is inserted with a major.minor revision that is less than the revision configured (that is, the module has a revision of 1.6 and the slot is configured for a module with revision 1.8), no connection is made between the controller and the I/O module. **TIP** We recommend using Compatible Module whenever possible. Remember, though, with major revision changes, the module only works to the level of the configuration. At the time of this printing, the module uses a major.minor revision of 1.6(1). However, if a new major revision for the module is released, consider this example. If a module is configured for major.minor revision of 1.7 and you insert a module with a major.minor revision of 2.3, the module works at the 1.7 level, with respect to module functions that are related to RSLogix 5000 software such as interface changes. Anomaly updates that are affected by the module's firmware, though, would work at the 2.3 revision level. If possible, we recommend that you make sure configuration is updated to match the revision levels of all I/O modules, including your module. Failure to do so may not prevent the application from working but may defeat the purpose of upgrading your modules revision levels.
Disable Keying	The inserted module attempts to accept a connection to the controller regardless of its type.
	Be extremely cautious when using the disable keying option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss. If keying is disabled, a controller makes a connection with most modules of the same type as that used in the configuration. A controller will NOT establish a connection if any of the following conditions exist, even if keying is disabled: • The module is configured for one module type (for example, input module) and a module of another type (for example, output module) is used. • The module cannot accept some portion of the configuration. For example, if a non-diagnostic input module is configured for a diagnostic input module, the controller cannot make a connection because the module will not accept/process the diagnostic configuration.

(1) Minor revisions are incremented by single counts such that minor level 10 (major.minor revision level = 1.10) follows minor revision level 9 (1.9).

Module Inhibiting

With module inhibiting, you can indefinitely suspend a connection between an owner-controller and a module. This process can occur in the following way:

- You write configuration for a module but inhibit the module to prevent it from communicating with the owner-controller. In this case, the owner-controller does not establish a connection and configuration is not sent to the module until the connection is uninhibited.

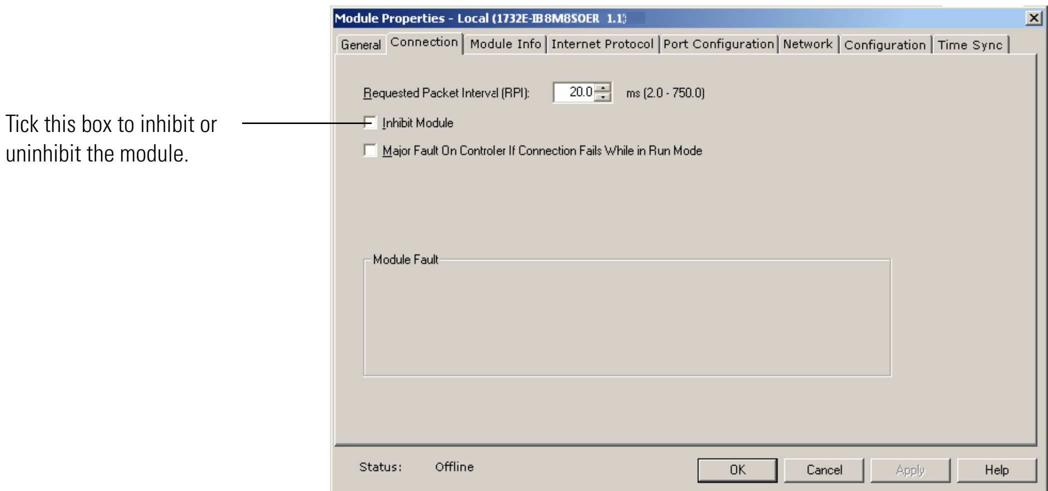
The following examples are instances where you may need to use module inhibiting:

- You want to FLASH upgrade your module. We recommend you:

a. Inhibit the module.
b. Perform the upgrade.
c. Uninhibit the module.

- You are using a program that includes a module that you do not physically possess yet, but you do not want the controller to continually look for a module that does not exist yet. In this case, you can inhibit the module in your program until it physically resides on the network.

You can inhibit your module on the Connection tab in RSLogix 5000, as shown in the example.



Module Fault Reporting

Your module provides both a hardware and software indication when a module fault occurs. The module's status indicators and RSLogix 5000 display each fault and include a fault message describing the nature of the fault.

This feature allows you to determine how the fault affects your module and what action you should take to resume normal operation. For more information on how to use hardware and software indicators when a module fault occurs, see Interpret Status Indicators on page 91 and Troubleshoot the Module on page 87.

Fully Configurable via Software

RSLogix 5000 software uses a custom, easily understood interface to write configuration. All module features are enabled or disabled through the I/O configuration portion of the software.

You can also use the software to interrogate your module to retrieve:

- serial number
- revision information
- product code
- vendor identification
- error/fault information

- diagnostic counters.

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

Producer/Consumer Model

By using the Producer/Consumer model, modules can produce data without having been polled by a controller first. The module produces the data and the owner-controller device consumes it.

Status Indicator Information

Each module has Status Indicators on the front of the module that allows you to check the module health and operational status.

For more information on how to use the module status indicators, and RSLogix 5000, when troubleshooting your application, see Interpret Status Indicators on page 91 and Troubleshoot the Module on page 87.

Agency Certifications

The module is marked for any agency certifications (for example, c-UL-us, CE, C-Tick and EtherNet/IP) it has obtained. See the module label for all agency certifications. For more information on full certification specifications, see Appendix A on page 93.

Chapter Summary and What's Next

In this chapter, you read about the features common to both Sequence of Events Input and Scheduled Output modules. The next chapter describes the features specific to the Sequence of Events Input module.

Notes:

Specific Features of the 1732E-IB8M8SOER Sequence of Events Input Module

Introduction

This chapter describes the features specific to the 1732E Sequence of Events Input module.

These features are configurable through the RSLogix 5000 software.

Determine Module Compatibility

Primarily, the Sequence of Events Input module 1732E-IB8M8SOER is used to interface to sensing devices and detect whether they are ON or OFF and to timestamp ON and OFF transitions. The module converts ON/OFF signals from user devices to appropriate logic level for use in the processor. Typical input devices include:

- auxiliary contacts
- limit switches

When designing a system using this module, you must consider:

- the voltage necessary for your application
- whether you need a solid state device
- current leakage
- if your application should use sinking or sourcing wiring.

For more information on compatibility of other Rockwell Automation products to modules, see the I/O Systems Overview, publication CIG-SO001.

Operational Modes

The 1732E-IB8M8SOER input module operates in FIFO and Per Point modes:

- FIFO – Each channel provides buffering of the timestamped input data for every input transition. A minimum of ten buffers is provided for every input channel (total twenty data buffers for every input to contain both OFF to ON and ON to OFF transition). The stored data is utilized on a first-in-first-out (FIFO) basis.
- Per Point Mode – The module produces timestamps for up to 2 input transitions per input, one for OFF to ON transitions and another for ON to OFF transitions; these timestamps can occur simultaneously on separate inputs.

For detailed information about operational modes, see Use the Sequence of Events Input and Scheduled Output Modules on page 63.

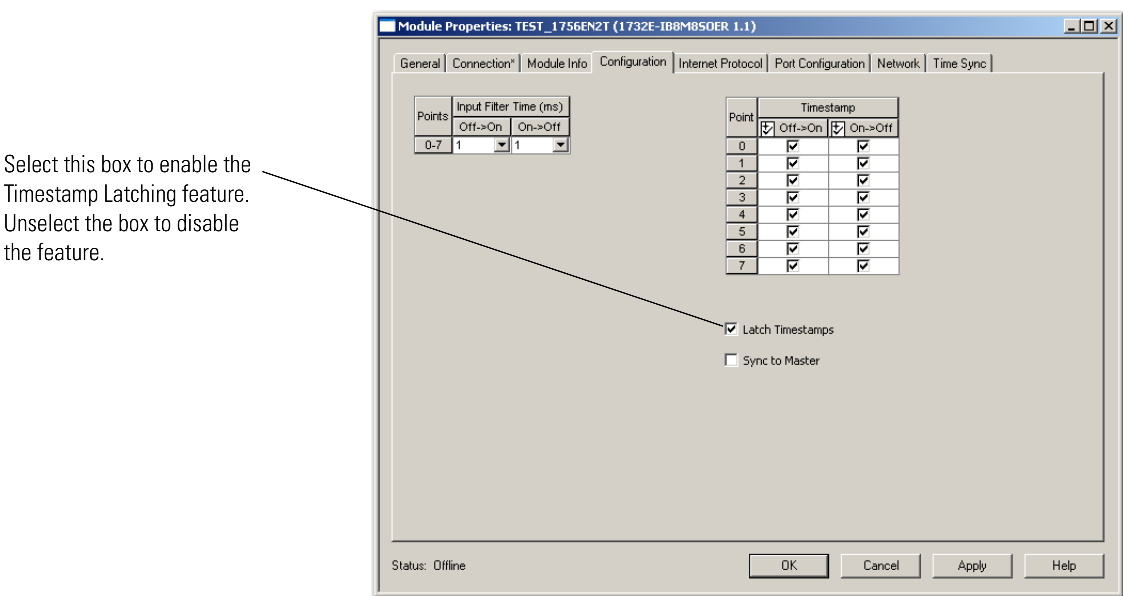
Timestamp Latching

Timestamp Latching can be used to prevent the module from overwriting input data once it is timestamped. This feature is available on the 1732E-IB8M8SOER input module.

- If Timestamp Latching is **enabled**, the module timestamps an input in a given direction and ignores future input transitions in that direction until the controller acknowledges the timestamp data already received.
- If Timestamp Latching is **disabled**, the module timestamps every input transition and may overwrite previously recorded timestamp data if the controller does not acknowledge the data quickly enough.

This feature is set on a modulewide basis and is enabled by default. It works in both FIFO and Per Point modes.

Use the Configuration tab in RSLogix 5000 software to enable Timestamp Latching, as shown in the example.



Software Configurable Input Filters

To account for hard contact "bounce", you can configure ON to OFF and OFF to ON input filter times in RSLogix 5000 software for your module. These filters define how long an input transition must remain in the new state before the module considers the transition valid.

IMPORTANT	Input filters are applied to all inputs on the module. You cannot apply input filters to individual inputs on the module.

When an input transition occurs, the module timestamps the transition on the initial edge of the transition and stores data for the transition on-board; the module then scans the input where the transition occurred every millisecond for the length of the filter time setting to verify that the input remains in the new state (remained OFF or ON).

- If the input remains in the new state for a time period equal to the filter time setting, the module sends data for the transition to the controller.

  When an input transition is detected, the module counts the number of 1 ms intervals the input is in the new state until the count reaches the filter value.
- If the input changes state again (returns to the original state) before the length of time of the filter setting has elapsed, the module starts decrementing the number of 1 ms intervals counted until it reaches zero. At this point the module stops filtering the input and discards the timestamp. During this continued scan period, one of the following events occurs:
  - At some point while still filtering the input, the input returns to the transitioned state and remains there until the module counts the number of 1 ms intervals equal to the filter setting. In this case, the module sends data from the transition to the controller.
  - The input does not remain in the transitioned state for a time period equal to the filter setting and the 1 ms counter decrements to zero. In this case, the module does not consider the original transition valid and drops the timestamp.

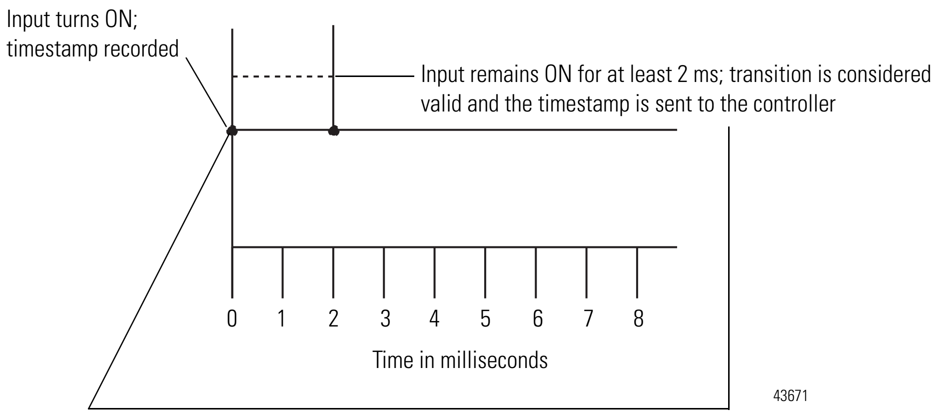
The following example illustrates how the module's input filters operate.

In the example, a module:

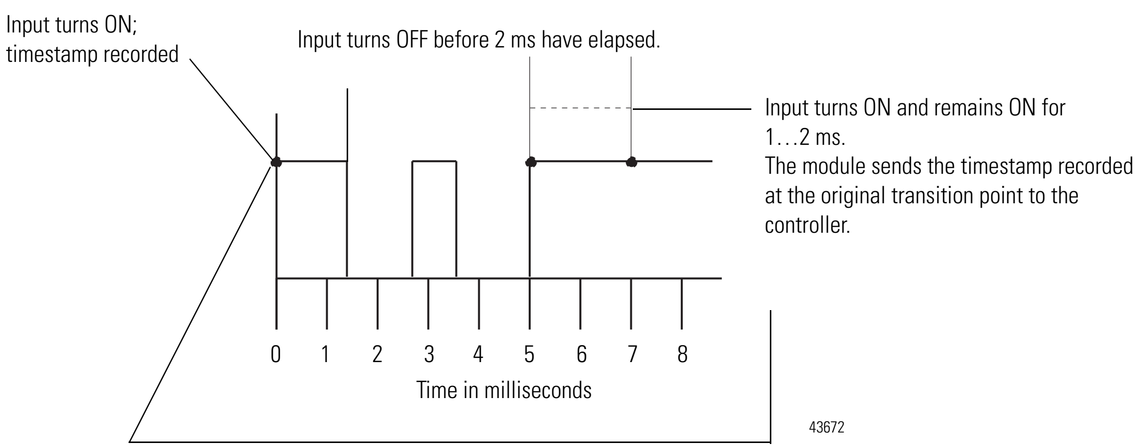
- is Timestamp Capture-enabled for all of its points
- uses a 2 ms input filter setting for OFF to ON transitions

Three possible scenarios can result after an input transitioning from OFF to ON in the given circumstances.

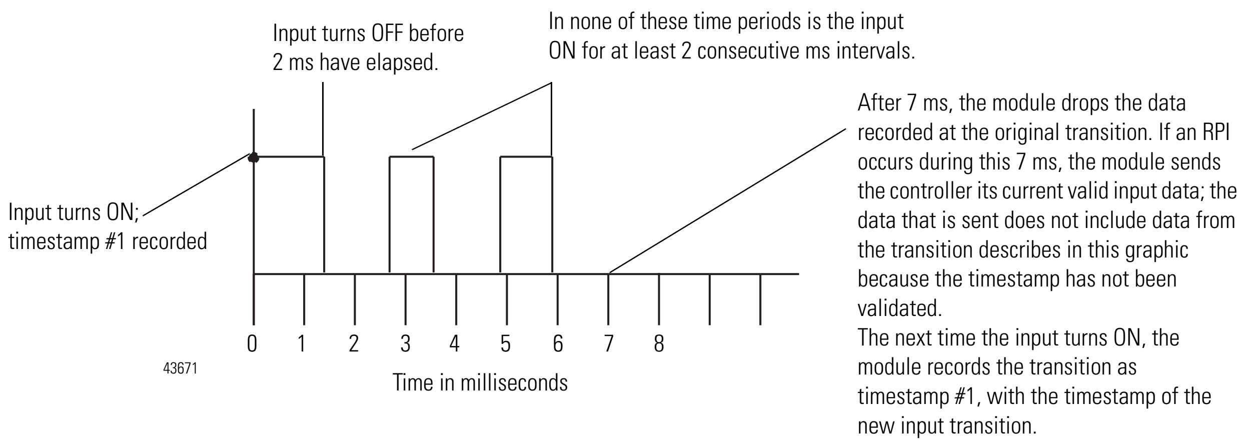
- Scenario #1 (no bounce) – The input turns ON and remains for the full 2 ms. In this case, the module considers the transition valid and sends the data recorded at the transition to the controller.
  Note the input was sampled as being on three different times: 0 ms, 1 ms and 2 ms.



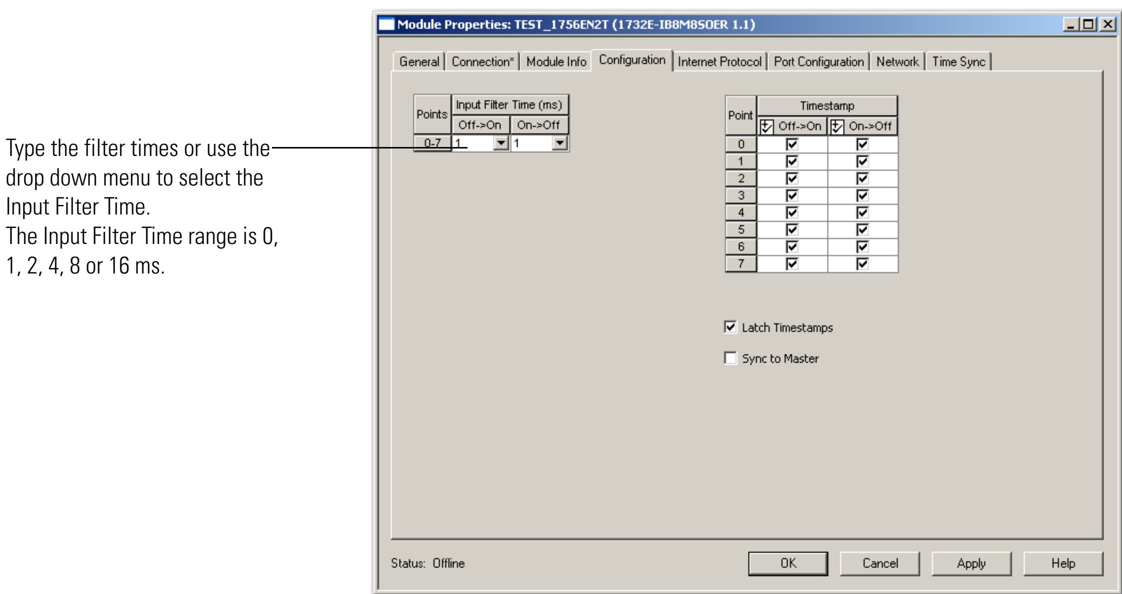
- Scenario #2 – The input turns ON but turns OFF before 2 ms (length of the input filter setting) elapses. In this case, the module continues to scan the input every millisecond. At some point, less than 2 ms later, the input turns ON again and remains for 1 to 2 ms, the third ON sampled 1 ms interval (in this case at 6 ms). In this case, the module considers the transition valid and sends the data timestamped at the original transition to the controller.



- Scenario #3 – The input turns ON but turns OFF before 2 ms (length of the input filter setting) elapses. In this case, the module continues to scan the input every millisecond until the 1 ms counter decrements to zero. The input never remains ON for at least 2 consecutive ms intervals, the third ON sampled 1 ms interval. In this case, the module considers the transition invalid and drops the data timestamped at the original transition.



Use the Configuration tab in RSLogix 5000 software to configure Input Filters, as shown in the example below.



Sync to Master

The Sync to Master feature in the 1732E-IB8M8SOER module indicates whether the module should have synchronization with a master clock. When enabled, the module remains in, or transitions to, the configuring state until synchronized with a master clock. When disabled, the module operates normally whether it is synchronized with a master clock or not.

The Sync to Master attribute is a read/writeable Boolean with a default value of 0 (master synchronization disabled).

Chapter Summary and What's Next

In this chapter, you learned about the features of the Sequence of Events Input module. The next chapter describes the features specific to the Scheduled Output module.

Specific Features of the 1732E-OB8M8SR Scheduled Output Module

Introduction

This chapter describes the features specific to the 1732E Scheduled Output module.

These features can be configured through the RSLogix 5000 software.

Determine Module Compatibility

The Scheduled Output module 1732E-OB8M8SR serves to provide high speed scheduling of every output signal based on time scheduling. Time scheduling is obtained via CIP synchronization (CIP Sync).

All outputs can be individually scheduled. Users shall be able to define the number of outputs to be used for scheduling purposes, leaving the other outputs for use as “normal” outputs.

When designing a system using this module, you must consider:
- the voltage necessary for your application
- whether you need a solid state device
- current leakage
- if your application should use sinking or sourcing wiring.

For more information on compatibility of other Rockwell Automation products to modules, refer to the Product Compatibility and Download Center page at: http://www.rockwellautomation.com/global/support/pcdc.page

Operational Mode	Required version of Studio 5000	Configuration to use
Normal Output	Version 18 or later	Timestamp = none MAOC Support = No

Operational Mode	Required version of Studio 5000	Configuration to use
Scheduled Output without use of MAOC	Version 18 or later	Timestamp = none MAOC Support = No
Scheduled Output with use of MAOC	Version 21 or later	Timestamp = Per Point MAOC Support = Yes

Operational Modes

The 1732E-OB8M8SR output module has two modes of handling the individual output points:

- Normal Output – The output behaves like a normal output point such that the module updates the output point upon receiving new I/O data from the client controller.
- Scheduled Mode – The output module is updated at a specific scheduled time. The client controller sends the output data value along with the associated time information to the output point. When the system time as kept by the module reaches the scheduled time, the output value is written. If the module system time is already past the scheduled time, the output value is written immediately.

Each individual output point can provide high-speed scheduling of every output signal based on time scheduling. The user can configure which output points are to be used for scheduling purposes, leaving the other output points for use as "normal" outputs.

Time-Scheduled Output Control

Time-scheduled output control is a feature available on the eight outputs of the 1732E-OB8M8SR module.

By using the time-scheduled output control feature, the module can turn the outputs On or Off at a specific CIPSync time. You can set the time setpoint (in 1 µs increments) for the output to turn On or Off in the application program. The 1732E-OB8M8SR module manages the time locally as such that the output is turned On or Off at the time specified.

MAOC Instructions with Time-Scheduled Output Control

The Motion Axis Output Cam (MAOC) instruction provides position-based control of outputs, by using position and velocity information of any motion axis. When the 1732E-OB8M8SR module is specified as the output source for the MAOC instruction, then the MAOC instruction automatically handles the time-based output scheduling and enables it on the eight outputs on the 1732E-OB8M8SR module. The benefit of using output scheduling in this manner is that the resolution of the output control is improved from the motion coarse update rate (typically 1…32 ms), to 100 µs.

To learn more about this feature and usage of the module, see the chapter entitled,Use the Sequence of Events Input and Scheduled Output Modules on page 63.

Configurable Point-Level Output Fault States

Individual outputs can be independently configured to unique fault states, either On, Off, or Hold in case of a communication failure or Program mode.

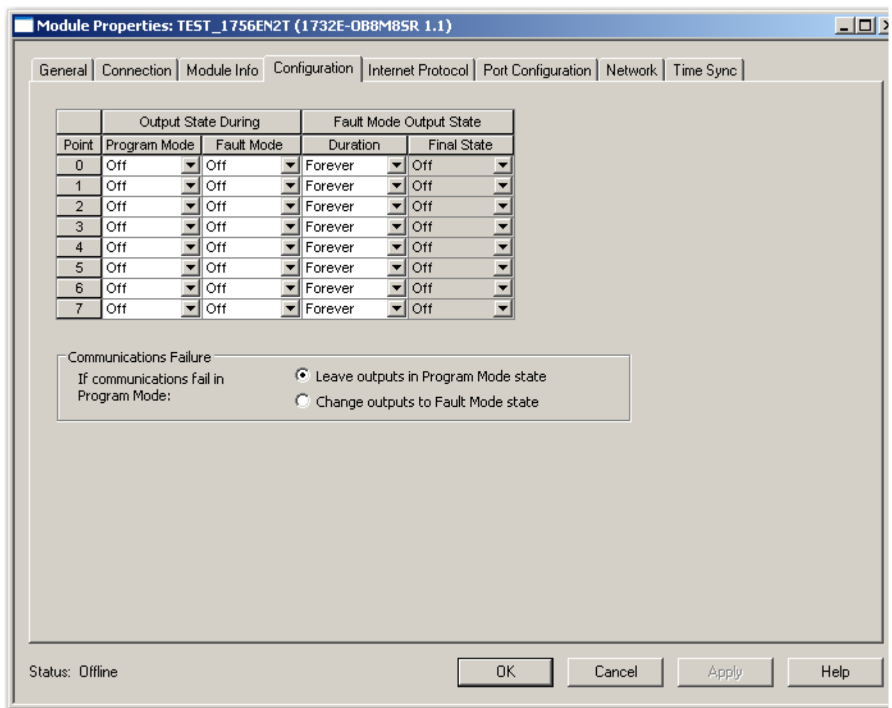
Through the RSLogix 5000 software, the user can set output state during Program Mode and Fault Mode for channels 0…7. Valid values are On, Off, and Hold.

Fault Duration can be set with 1, 2, 5, 10 seconds and Forever or 0.

Fault Final State can be set as On or Off. On the Configuration tab, it is grayed out unless Fault Duration is something other than "Forever".

Follow these steps to enable a fault state.

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



2. Click the pull-down arrow to choose the Program Mode for each channel. Options available are Off (default), On and Hold.
3. Click the pull-down arrow to choose the Fault Mode for each channel. Options available are Off (default), On and Hold.
4. Specify Fault Duration.
   When active, Fault Duration has options of "Forever", "1 Second", "2 Seconds", "5 Seconds" or "10 Seconds". Default is "Forever" or 0.

5. Specify the Fault Final State.
   Note that this is grayed out unless Hold Duration is something other than "Forever". When active, Fault Final State has the options, "Off" and "On". Default value is Off.
6. If communications fail in program mode, specify whether to "Leave outputs in program mode" or "Change outputs to fault mode state".
7. Click OK.

Output State

The Scheduled Output module allows the user to define output state when in Program Mode and Fault Mode.

Program Mode refers to the state where the following events occur:
- Controller program is not executing.
- Inputs are still actively producing data.
- Outputs are not actively controlled and go to their configured Program mode.

Fault Mode selects the behavior the output channel takes if a communication fault occurs. FaultValue defines the value to go to on fault if the bit is set.

Fault Mode provides individual fault mode selection for output channels. When this selection is disabled [the bit is reset (0)] and the system enters the fault mode, the module holds the last output state value. This means that the output remains at the last converted value prior to the condition that caused the system to enter the fault mode.

Chapter Summary and What's Next

In this chapter, you read about the features specific to the Scheduled Output module. The next chapter describes using the modules.

Use the Sequence of Events Input and Scheduled Output Modules

Introduction

This chapter describes how to use the Sequence of Events Input and Scheduled Output modules (1732E-IB8M8SOER, 1732E-OB8M8SR).

This chapter has two main sections:
- Use the 1732E-IB8M8SOER Sequence of Events Input Module on page 65-83
- Use the 1732E-OB8M8SR Scheduled Output Module on page 81-88

The following table includes the list of topics available in this chapter.

Overview

The 1732E-IB8M8SOER module can be configured to timestamp two transitions per input, one in each direction (OFF to ON and ON to OFF).

When specific points that are Timestamp Capture-enabled transition (for example, input 1 is configured so that Timestamp Capture is enabled for OFF to ON transitions and the input turns ON), the module timestamps the transition with the current system time value on the network. The module produces data for the owner-controller the RPI after the input filter criteria have been met and at subsequent RPIs.

For the 1732E-OB8M8SR module, timestamping can be used in conjunction with the scheduled outputs feature so that after input data changes state and a timestamp occurs, an output point will actuate at some configured time in the future. You can schedule outputs up to 16 seconds into the future.

When you use timestamping of inputs and scheduled outputs, you must:

- choose a Communication Format for each input and output module that allows timestamping. See Communication Format for more information.
- disable Change of State for all input points on the input module except the point being timestamped.

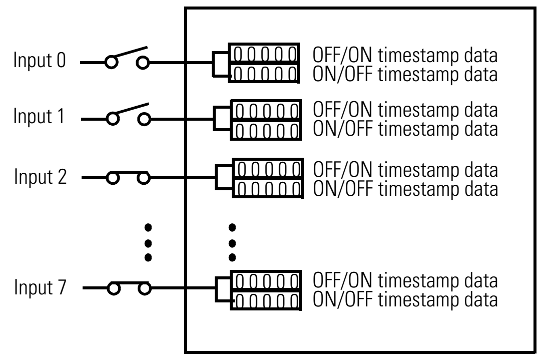
Use the Sequence of Events Input Module

The following section describes how to use the Sequence of Events Input module.

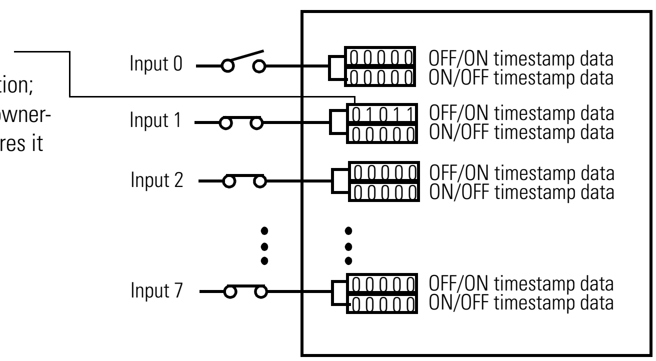
How Does 1732E-IB8M8SOER Store Timestamp Data?

With each timestamped transition, 1732E-IB8M8SOER stores data for that point. An overview of how the module stores timestamp data is shown in the following figure.

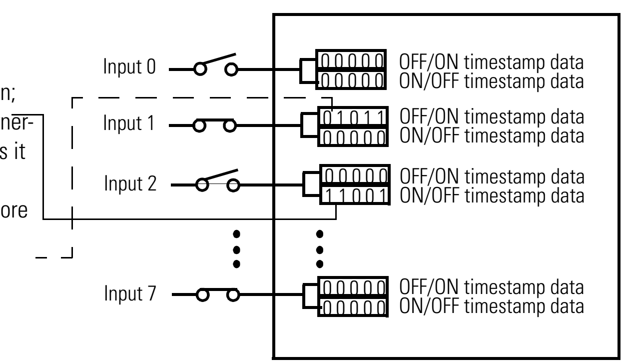
The module is installed, wired to input devices and ready to begin operation. All inputs are configured to timestamp any transition that occurs.
At this point, timestamp data for each input is 0 because no input transitions have occurred.
Note that only 8 bits of the 64-bit timestamp are shown.



Input 1 transitions from OFF to ON.
The module timestamps the transition; the module sends the data to the owner-controller (not shown) and also stores it locally.



Input 2 transitions from ON to OFF.
The module timestamps the transition; the module sends the data to the owner-controller (not shown) and also stores it locally.
Note that the module continues to store the timestamp for the OFF to ON transition on input 1.



Generally the following occurs:

1. The module timestamps each transition for inputs that are Timestamp Capture-enabled. The module can timestamp each transition with a unique system time.

2. The module sends all of its input data, including the new data from the most recent transition, to the controller immediately after timestamping the transition and passing the input filter to make sure the transition was valid.
3. You copy new data from the controller tags to a separate data structure for later sorting.
4. Acknowledge the timestamp, using output tags, so that the module can capture another timestamp on that input without losing any data.
5. Once the data is copied to a separate data structure, you may sort the data in the controller to determine the order of events.

Some of these typical events are described in greater detail in the rest of this chapter. For typical applications for Sequence of Events modules, refer to High Performance Sequence of Events Applications in the Logix Architecture on page 9.

Use Timestamp Latching

When enabled, timestamp latching prevents the module from overwriting recorded timestamp data once a transition occurs. This feature is set on a modulewide basis and is enabled by default. The following table describes how Timestamp Latching affects the module.

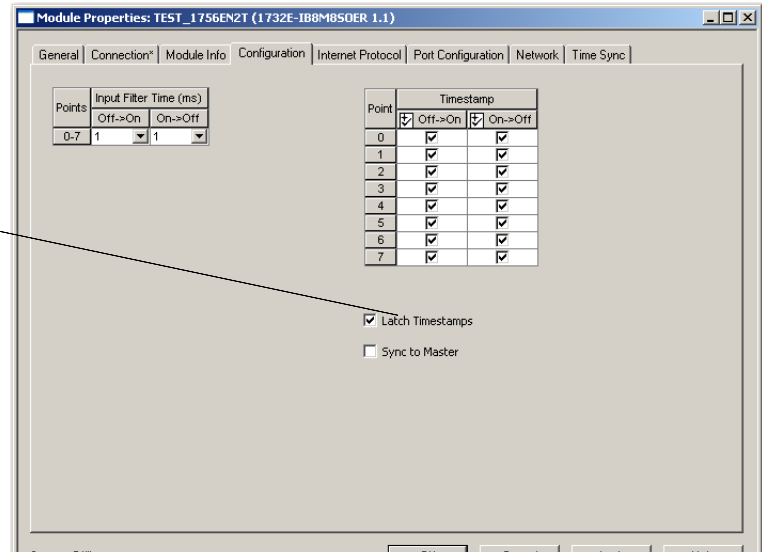
If Timestamp Latching is:	the following occurs(1)
Enabled	The module timestamps two transitions for each input–one for OFF to ON and one for ON to OFF. If similar transitions occur on inputs where a transition has already been timestamped and the data was not yet acknowledged (for more information on Acknowledge Timestamp Latching Timestamp Data, see page 76), the module does not timestamp the new transition. When a transition is not timestamped, the module sets the I.EventOverflow tag for that point to inform the controller that an input transitioned but a timestamp was not produced for the transition. **By default, Timestamp Latching is enabled**.
Disabled	The module timestamps each transition for each input as it occurs. In this case, when multiple transitions occur in the same direction on the same input, the module records the new timestamp data, overwriting any previously-recorded data which had yet to be acknowledged (for more information on Acknowledge Timestamp Latching Timestamp Data, see page 76). When the module overwrites data, it sets the I.EventOverflow tag for that point to inform the controller that events have been overwritten.

(1) This table assumes the transition occurs on inputs that have Timestamp Capture enabled. If Timestamp Capture is disabled, the module does not timestamp transitions on that input and, therefore, Timestamp Latching does not affect module behavior.

IMPORTANT	We suggest you monitor the I.EventOverflow bits to make sure you are aware of transitions that were either not timestamped or when timestamp data was overwritten.

Use the Configuration tab in RSLogix 5000 software to enable Timestamp Latching on the 1732E-IB8M8SOER, as shown in the example.

Select this box to enable the Timestamp Latching feature. De-select the box to disable the feature.



Use Timestamp Capture

Timestamp Capture causes the module to timestamp specific input transitions (Off to On and On to Off). However, keep the following in mind when using this feature:

Typically, Timestamp Latching is enabled. The configuration of this feature determines whether the module timestamps only the first transition on an input until the timestamp is acknowledged, or every transition on an input while overwriting timestamps that have not yet been acknowledged.

If Timestamp Capture is enabled, the module timestamps only the enabled transitions (OFF to ON and ON to OFF) for each input.

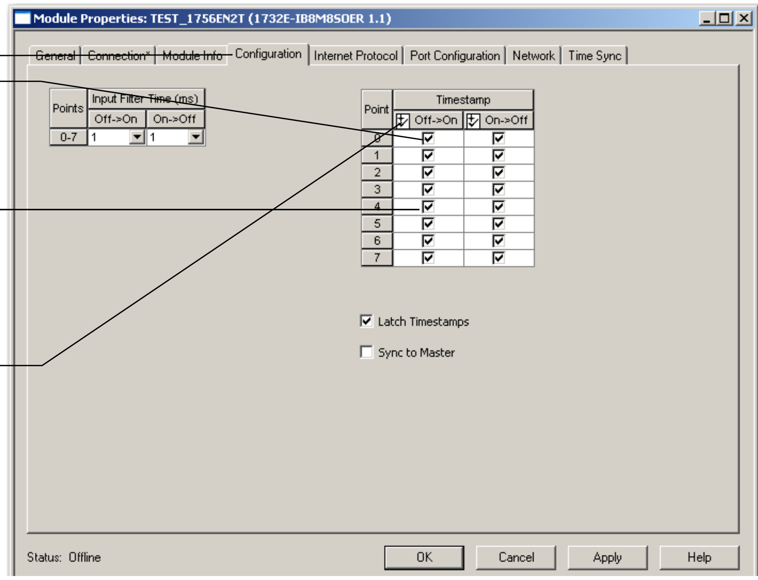
Whenever an input transition is timestamped as a valid transition, the module sends updated input data for all inputs to the controller at the next RPI and at every subsequent RPI.

Use the Configuration tab in RSLogix 5000 software to set Timestamp Capture, as shown in the example below.

Click the Configuration tab.

- Select the individual boxes for each input point to enable Timestamp Capture for that point.
- Unselect the individual boxes for each input point to disable Timestamp Capture for that point.

You can also use these boxes to enable or disable all points simultaneously.



Operational Modes

The 1732E-IB8M8SOER module operates in FIFO and Per Point modes:

- FIFO – Each channel provides buffering of the timestamped input data for every input transition. There are 256 timestamp buffers for each of the 8 input channels. Each buffer can contain a single Off to On or On to Off event.
- Per Point Mode – The module produces timestamps for up to 2 input transitions per input, one for OFF to ON transitions and another for ON to OFF transitions; these timestamps can occur simultaneously on separate inputs.

Use the Sequence of Events Module in FIFO Mode

In First In First Out (FIFO) mode, the Sequence of Events module timestamps multiple input transitions on any CIPSync/PTP Capture-enabled inputs. The module stores the timestamp data in on-board buffers that hold 256 timestamp events per channel. Each of the 8 inputs has its own buffer.

When an input transitions, the module timestamps the event and records specific input data related to the transition. The owner-controller must retrieve the data from the Sequence of Events module using one of the two methods described later in this chapter.

IMPORTANT	Keep in mind that, although the Sequence of Events module can store up to 256 events per input, if you manage the buffer effectively (that is, retrieve data in a timely fashion), the module can timestamp an infinite number of input transitions and the controller will be able to retrieve and use the data.

How Does the On-Board Buffer Work in FIFO Mode?

The module stores up to 256 events per input. Once data is stored on the module, the controller must retrieve it. Typically, the controller retrieves data from the first slot in the on-board buffer; the data in the first slot is also known as the current event.

IMPORTANT	The current event is the event for which the Sequence of Events module is currently producing data. The current event is NOT the most recently-timestamped input transition.

After the controller retrieves the current event data, it acknowledges the data and clears it from the Sequence of Events module's on-board buffers, and the data from the next slot in the buffer becomes the current event (that is, the module produces this data for the controller).

In FIFO mode, generally the following occurs:

1. You configure the Sequence of Events module to operate in FIFO mode via the Communication Format selection.
2. The Sequence of Events module timestamps each transition and stores the data in its on-board buffer. The module can timestamp each transition with a unique CIPSync/PTP as long as the transitions occur 25 μs apart.
3. The controller retrieves data from the Sequence of Events module as described in the following steps:
   a. Immediately after the Sequence of Events module timestamps an input transition, it records data in the first slot of its on-board buffer and produces the data for the controller; the data is the current event. The module produces the data from the current event at every subsequent RPI until the controller clears it (as described in step C).
   b. The controller copies the data from the controller tags to a separate data structure for later use.
   c. The controller acknowledges the current event in the Sequence of Events module's buffer by I.EventNumber to O.EventNumber.
   d. Once the current event is cleared from the Sequence of Events module's buffer, data for the next transition stored in the buffer becomes the current event, and the module begins producing this data for the controller as described in step a.

4. The Sequence of Events module timestamps input transitions and records the data in its on-board buffer as long as the buffer is not full. (The module stores up to 256 events per input.)

Typical Applications of FIFO Mode

FIFO mode is intended for use in applications where multiple transitions occur on multiple inputs in relatively rapid succession (that is, faster than the controller can acknowledge the data as the transitions occur). Because of this intention, the Sequence of Events module uses an on-board buffer to store the data for up to 256 events per input.

The following are example typical applications for FIFO mode:

- Sequence monitoring
- Process and machine optimization

Configure the Module for FIFO Mode

You configure the same general set of configurable features whether you are using the Sequence of Events module in CIPSync/PTP Per Point mode or FIFO mode. However, for some features, the module behavior as dictated by the feature, varies according to operational mode. For example, Latch Timestamp impacts the Sequence of Events module behavior slightly differently in CIPSync/PTP Per Point mode than in FIFO mode.

You should be aware of the impact the following configurable features have on module behavior in FIFO mode:

- Communications Format
- Latch Timestamp
- Enable CIPSync/PTP Timestamp Capture

Choose a Communications Format

During initial module configuration, you must choose a communication format for the module. The communications format determines what operational mode your Sequence of Events module uses and, consequently, what tags RSLogix 5000 generates when configuration is complete.

To operate the Sequence of Events module in FIFO mode, you must choose the FIFO communication format, as shown below.

Manage the Data in FIFO Mode

In FIFO Mode, the Sequence of Events module sends input data for the current event to the controller immediately after the first input transition has been timestamped and at each RPI. You must manage the data coming from the Sequence of Events module.

The following occurs in the process of the managing data coming from the Sequence of Events module in CIPSync/PTP Per Point mode:

1. The controller retrieves current event data from the Sequence of Events module in one of two retrieval methods.
2. The controller copies the relevant portions of the current event data to a separate array.
3. At the user's discretion, controller clears current data from the Sequence of Events module by copying the current event number (I.EventNumber) to the O.EventAck tag, preparing the module send data from the next current event.

This process is described in the rest of this section.

Retrieve Data in FIFO Mode

In FIFO Mode, the Sequence of Events module automatically sends the controller the data from the first timestamped transition in its buffer. The controller must retrieve the data for the remaining timestamped transitions in the Sequence of Events module's buffers.

The controller can retrieve data in one of the following ways:
- Standard Retrieval – By default, the module uses this retrieval method which returns events in order of occurrence.
- Retrieval by Point

Standard Retrieval

In Standard Retrieval, the controller retrieves the data for each event in the order in which the events occurred. This retrieval method includes the following:

1. At each RPI, the Sequence of Events produces the current event in its on-board buffer.
2. The controller copies relevant input data from the current event to a separate data structure for later use.
3. The controller clears the current event from the Sequence of Events module's on-board buffer by acknowledging the data via the module's output word.
4. When the current event is cleared, the next event in the module's on-board buffer becomes the new current event. If no other events are present, the event data will be 0.
5. The Sequence of Events module produces the new current event as described in step 1.

This process continues as long as the Sequence of Events module timestamps input transitions and the controller continues to retrieve the data for each transition.

Retrieval by Point

Retrieval by Point is similar to Standard Retrieval by time except that with this method, the controller only retrieves timestamp data for input transitions that occurred on a specific point.

The Sequence of Events module still timestamps input transitions for any events that occur on Enable Timetamp Capture-enabled inputs. The module stores up to 256 events per input.

IMPORTANT	The module stores up to 256 events per input. Each input's buffer is independent of the others. You must manage the module buffer effectively to make sure that the module timestamps all transitions on a specific input. Consider the following example: You are using Retrieve by Point to retrieve timestamp data from input 4, and 10 of the first 34 input transitions that the Sequence of Events module timestamps occur on input 4. If Latch Timestamp is enabled for the module and you fail to clear any of the timestamp data for input 4 before the input transitions again, the Sequence of Events module will not timestamp the next transition for input 4, even though there are 126 slots still available in the module's on-board buffer.

Additional Module Settings Required with Retrieval by Point Method

By default, the Sequence of Events module operates as if the controller will use Standard Retrieval to retrieve data. To use the Retrieval by Point method, you must change the following two tags in the module's output word:

- O.RetrieveByPoint = 1 (default value is 0)
- O.PointtoRetrieve = input point for which you want the controller to retrieve data – for example, if you want to retrieve the data for input 8, you must change this tag to 8.

Retrieval Process Similar to Standard Retrieval

After you make the output tag changes listed above, the controller retrieves the data for each transition on the specified input in the order in which the transitions occur.

The only exception to the process is that in Retrieval by Point, the current event is not necessarily the data in the first slot of of the on-board buffer. Instead, the current event is the first slot (up to 256 events per input) that contains data for a transition timestamped at the specified input.

For example, if you are using Retrieval by Point to retrieve data for input 7 and the first transition that occurs on input 7 is the 5th transition the module timestamped, the current event is located in slot 5 of the module's on-board buffer.

Change Between Retrieval Methods

When using the Sequence of Events module in FIFO mode, you may determine that you need to change retrieval methods. You can change retrieval methods, but keep the following in mind before doing so:

- The change will NOT take effect until all events are acknowledged/cleared from the module's buffers.
- When you change retrieval methods dynamically, the ideal way is to reset events in the module buffers (as described above) and immediately switch FIFO retrieval modes. Make sure you do not need the data being cleared from the module buffer prior to resetting events.

To change retrieval methods, change the O.RetrieveByPoint tag to the new method.

- To use the standard retrieval by time method, O.RetrieveByPoint = 0
- To use the retrieval by point method, O.RetrieveByPoint = 1

Manage the Data

The module sends all of its input data to the controller the next RPI after an input transition has been timestamped and at each subsequent RPI. You must manage the data coming from the module.

The following occurs in the process of the managing data coming from the module:

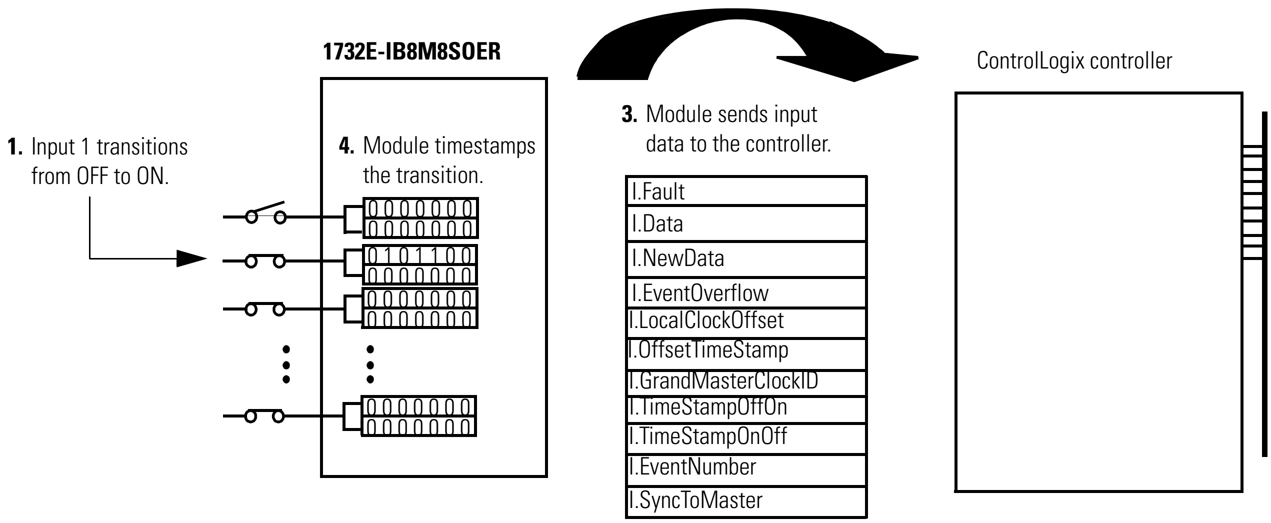
1. The module sends data to the controller.
2. The controller copies the relevant portions of the input data to separate array.
3. At the user's discretion, the controller clears latched timestamp data from the module via the O.EventAck and O.NewData tags, preparing the module to timestamp the next transition.

This process is described in the rest of this section.

Module Sends Data to the Controller

The following figure shows an example of the module sending data to the controller. In the example, the following occurs:

1. Input 1 transitions from OFF to ON. (The input has Timestamp Capture enabled).
2. The module timestamps the transition.
3. The module sends its input data, including the transition timestamp from input 1, to the controller.



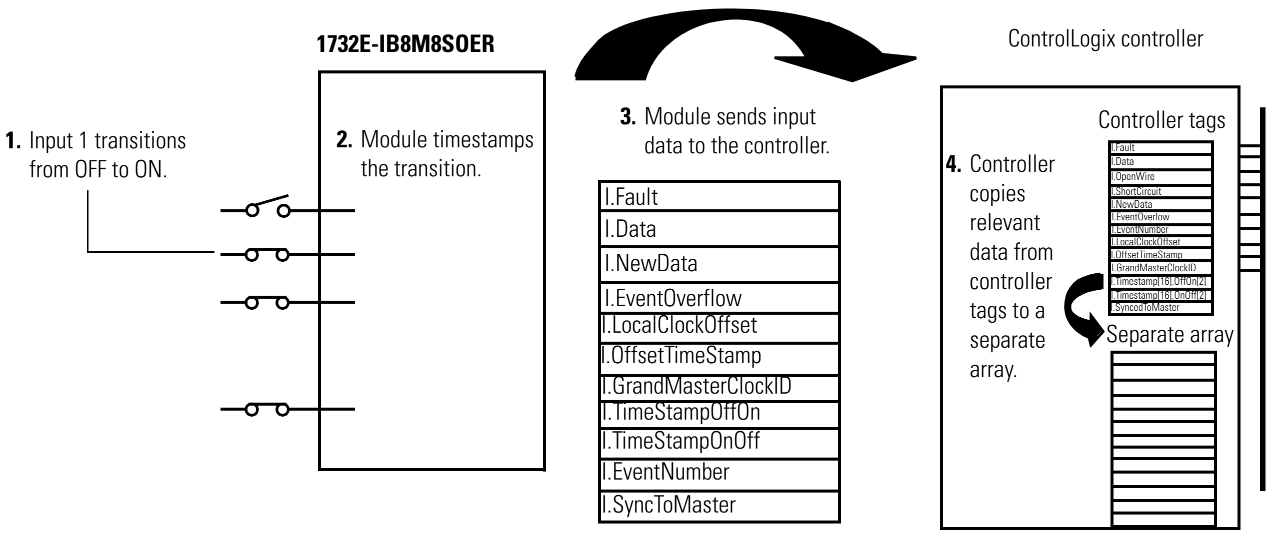
The Module Tags describe the data that is sent for each input. These tags are sent to the controller the next RPI after the module timestamps a transition on any input as well as all other RPIs. For detailed descriptions of the tags, refer to Appendix B.

Copy Relevant Input Data to a Separate Data Structure

When the module sends input data to the controller, the data is stored in the controller tags. We recommend you use a COP or CPS instruction to programmatically copy new timestamp data from the controller tags to a separate array in the controller memory. Later, you can combine timestamp data from multiple modules and use a Sort routine to determine the order of events, with relative time reference, that occurred in a specific time period.

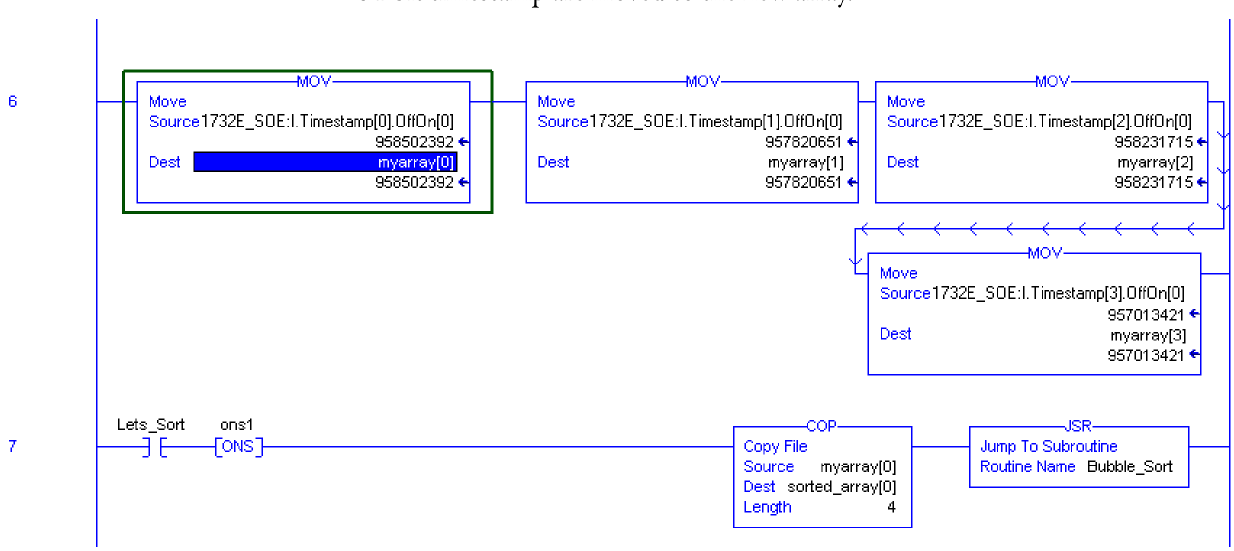
IMPORTANT	When you copy relevant timestamp data from the controller tags to a separate data structure, make sure you copy enough information for each timestamp that you can differentiate between timestamps for different inputs.

The following figure shows when to use the COP instruction. In this example, the module timestamped a transition on input 1 and is sending input data to the controller at each RPI. The controller copies input data from the controller tags to a separate data structure.



Your application determines what input data should be copied from the controller tags to a separate data structure. Although you can copy all the input data to another array, typically, only the data from specific tags is copied.

The following figure shows an example of ladder logic in which the controller only moves OFF to ON timestamp data for inputs 0…3 from the controller tags to a separate data structure named myarray. The data in the myarray structure is then moved to another array used to sort the data. In this example, 32 bits of each 64-bit timestamp are moved to the new array.



Acknowledge Timestamp Latching Timestamp Data

In most cases, Timestamp Latching is enabled. This means that once the module timestamps an input transition, the module will not timestamp another transition in the same direction on the same input until you acknowledge the data from the first timestamped transition; when you **acknowledge data**, you **clear it from the module**.

To clear data from the module, you must acknowledge them via the module output tags. You can clear data in the following ways:

- Clear latched timestamp data for specific inputs – As data is acknowledged, it is cleared from the module, and the module will once again timestamp the first new transition for the input in the cleared direction(s).

  To clear timestamp data for specific inputs, you must complete the following steps:

  a. Write to the EventAck output tag (*O.EventAck*). This tag determines which edge you will clear (acknowledge).
  - 0 = clear only the falling edge timestamp (I.Timestamp[x].OnOff)
  - 1 = clear only the rising edge timestamp (I.Timestamp[x].OffOn)
  - 2 = clear both the falling and rising edge timestamps

  b. Change the NewDataAck output tag (*O.NewDataAck.x*) to a rising edge (set the tag =1). This tag determines which inputs will be cleared (acknowledged). There are 8 bits (x = 0…7) that can be transitioned; each corresponding to an input. More than one bit can be transitioned at the same time.
  - If the bit = 0, change the bit to 1.
  - If the bit = 1, change the bit to 0, wait for at least one RPI, and change the bit to 1.

  The corresponding I.EventOverflow and I.NewData tags are also cleared.

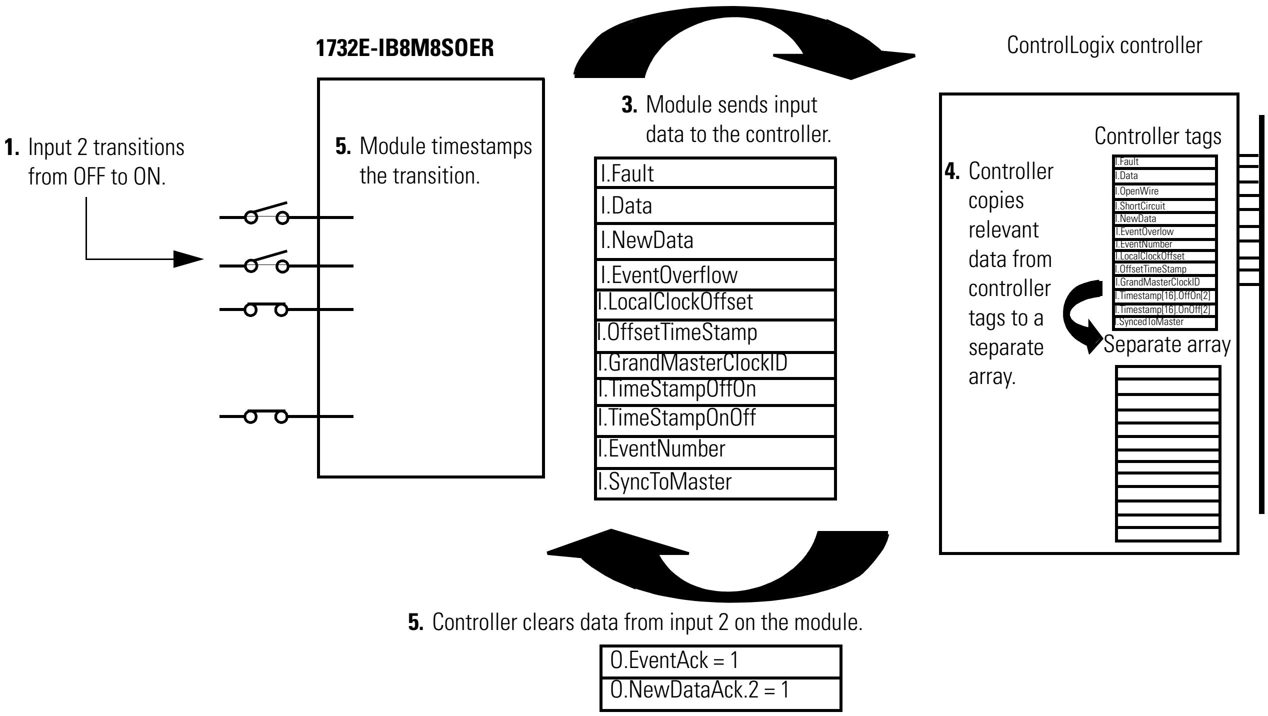
- **Clear all latched data for the module** – This transition erases all timestamp data from the module, clearing data from all inputs simultaneously. Once the data is cleared, the module timestamps the first transition in each direction for each input and sends the data to the controller (assuming those inputs are configured with Timestamp Capture enabled in each direction).

  To clear all data for the module, transition the O.ResetEvents tag to 1.
  – If the bit = 0, change the bit to 1.
  – If the bit = 1, change the bit to 0, wait for at least one RPI, and change the bit to 1.

The following figure shows when to clear data from the module. In this example, the module sent input data to the controller, and the controller copied the relevant input data to a separate structure. Now, the controller must clear the data from the module.

In this example, to clear data from the module, the controller writes the following to the Sequence of Events output word:

- O.EventAck = 1
- O.NewDataAck.2 = 1



If **Timestamp Latch is disabled**, the module sends new data, from subsequent transitions, to the controller as soon as they occur. The controller overwrites timestamp data from the last transition, regardless of whether it saved the data or not.

If the controller does not acknowledge the timestamp data then the NewData bits in the input tags remains set and the EventOverflow bit is set as well.

Sort the Data

If you need to determine the order of events that occurred in a cascade, you must use a Sort routine to determine the order of events. Rockwell Automation offers a sample sort routine that you can use to determine the order of events in an event cascade.

Visit the Rockwell Automation Sample Code Library at
http://samplecode.rockwellautomation.com/idc/groups/public/documents/webassets/sc_home_page.hcst.

Clear All Data From the Module Buffer At Once

If necessary, you can reset the events in the module, in effect clearing all data from previously timestamped transitions. In other words, when all data is cleared from the module buffers, all of the module input tags return to 0.

To reset events in the module buffer, transition the O.ResetEvents tag to 1 as described below:

- If the bit = 0, change the bit to 1.
- If the bit = 1, change the bit to 0, wait for at least one RPI, and change the bit to 1.

Once the data is cleared, the module begins timestamping input transitions again and storing them in its on-board buffer.

Propagate a Signal From Input Pin to EtherNet

The module receives a signal at its input pin and processes it internally before sending the input and timestamp data to the controller at the Requested Packet Interval (RPI) via EtherNet.

When you operate the module, you must account for signal propagation delays that exist during internal processing. Some of these delays are inherent to the module and others are controlled by temperature and input voltage.

During processing, the following delays exist:

- hardware delay – The time it takes an input signal to propagate from the module input pin to its microprocessor. This time varies according to input transition type (OFF to ON/ON to OFF), input voltage and temperature.
- firmware delay time – The time it takes the module to acquire a timestamp once its microprocessor receives the input signal.
- input filter delay – user-configurable number from 0…16 ms. The input filter does not affect when the timestamp is acquired. It acquires the "firmware delay time" after the input changes state at the module microprocessor. The input filter simply delays the amount of time the input must be in a certain state before input is considered valid and the timestamp data will be sent to the controller.
- RPI – Once the timestamp is acquired by the microprocessor and the input is filtered, the input and timestamp data is sent to the controller at the next RPI.

Timestamp Accuracy = +/- 12.5 μs.(1)

Module Input Pin OFF->ON to Timestamp (Hardware + Firmware) Delay (μs)

Ambient Temp °C Voltage	-20	25	60
10V DC	23	24	25
24V DC	18	19	19
30V DC	18	19	19

Module Input Pin ON->OFF to Timestamp (Hardware + Firmware) Delay (μs)

Ambient Temp °C Voltage	-20	25	60
10V DC	59	75	84
24V DC	70	84	93
30V DC	71	85	94

Maximum input frequency (for each input) = 250 Hz 50% duty cycle. The module can provide unique timestamps for input transitions on separate inputs as long as they occur 25 μs apart. An input that changes state less than 25 μs after another input may receive the timestamp of the first input.

EXAMPLE

For example, if you are **turning ON** a 1732E-IB8M8SOER module input at 24V DC in 25 °C conditions, the signal propagation delay is 19 μs. If you want to calculate the actual time the signal reaches the module input pin, subtract 19 μs from the timestamp.

If you are **turning OFF** an input at 30V DC in 60 °C conditions, the signal propagation delay is 94 μs. If you want to calculate the actual time the signal reaches the module's input pin, subtract 94 μs from the timestamp.

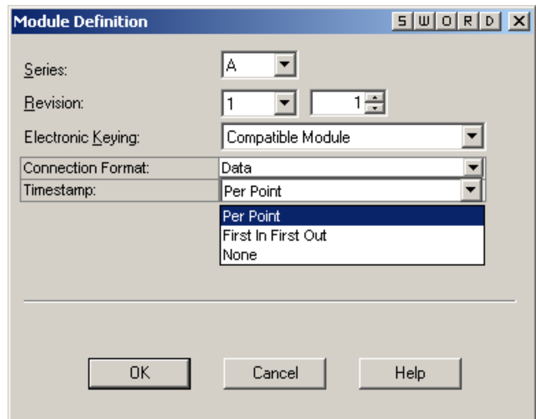
The timestamps acquired are accurate to +/- 40 μs as noted earlier.

The timestamp data being produced on EtherNet is also delayed by the input filter setting and the RPI setting.

(1) The timestamp accuracy of +/- 40 μs does not include errors introduced by the module clock being tuned using CIP Sync. This error can be less than one microsecond on a properly configured network.

Per Point Mode of Operation

The Per Point mode of operation provides a single On and Off timestamp for each input point on the module. The 1732E-IB8M8SOER module employs employs CIP Sync Per Point.

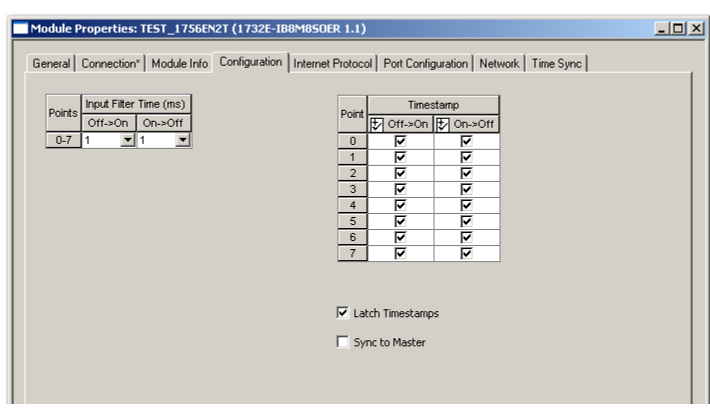


Per Point operation begins with the selection of the appropriate Timestamp Format in RSLogix 5000 software.

Choosing the Per Point Timestamp Format results in an input tag structure with a single On and Off timestamp value per input point, as well as some additional general-purpose CIP Sync status tags. It results in the creation of two additional input tags to assist in timestamp corrections in a CIP Sync architecture: I:LocalClockOffset and I:OffsetTimestamp.

For more information about Per Point mode, see the ControlLogix Sequence of Events Module User Manual, publication 1756-UM528.

The Latch checkbox latches CIP Sync timestamp so that recorded events are not discarded until you acknowledge them. As a result, if latching is selected and new events occur, they will be ignored until the existing event is acknowledged.



Use the Scheduled Output Module

The Motion Axis Output Cam (MAOC) instruction offers the functionality to set and reset output bits based on an axis position.

IMPORTANT	When using the 1732E-OB8M8SR module with the MAOC instruction, make sure that you are using Studio 5000, version 21 or later. You must also select Yes for MAOC support and Per Point under Time Stamping.

Usage with MAOC Instruction

When used with motion and the MAOC instruction values in the output image are controlled by the Motion Planner firmware in the controller, the Motion Planner triggers the data to be sent to the module. Although, the normal program/task scan also triggers data to be sent to the module. Data integrity is maintained by the firmware always setting the sequence count for a given schedule last.

When a programmed on or off event is detected, a schedule is sent to the output module to turn the output on/off at the appropriate time within the next coarse update period.

The Output Cam instruction divides the coarse update period into sixteen time slots. For example, a coarse update period of 2 ms will yield sixteen 125 µs time slots. Cam on/off events will be assigned to time slots based on their position within the coarse update period. If both latch and unlatch events for a cam element are assigned to the same time slot, they will cancel each other out. This implies that the minimum pulse width of a cam element is greater than one time slot.

The minimum pulse width of a cam element should be greater than the 100 µs 1732E-OB8M8SR minimum pulse width, or the 1/16 coarse update minimum pulse width, whichever is larger.

IMPORTANT	The 1732E-OB8M8SR Scheduled Output Module can be associated with one (1) MAOC axis/execution target only.

The MAOC instruction detects latch and unlatch events one coarse update ahead and schedules the event to occur within the next coarse update. This is accomplished by applying a one coarse update internal delay to each scheduled output latch and unlatch position. When the latch or unlatch event is detected, the delta time from the start of the coarse update to the event is calculated, and the output is scheduled to occur at the CIPSync corresponding to the next coarse update period. To facilitate this, Output Cam functionality has access to the CIPSync captured when the current coarse update period occurred.

The MAOC instruction is able to process scheduled output bits for the 1732E-OB8M8SR. The MAOC instruction sets the schedule mask bits that are defined for use by the application.

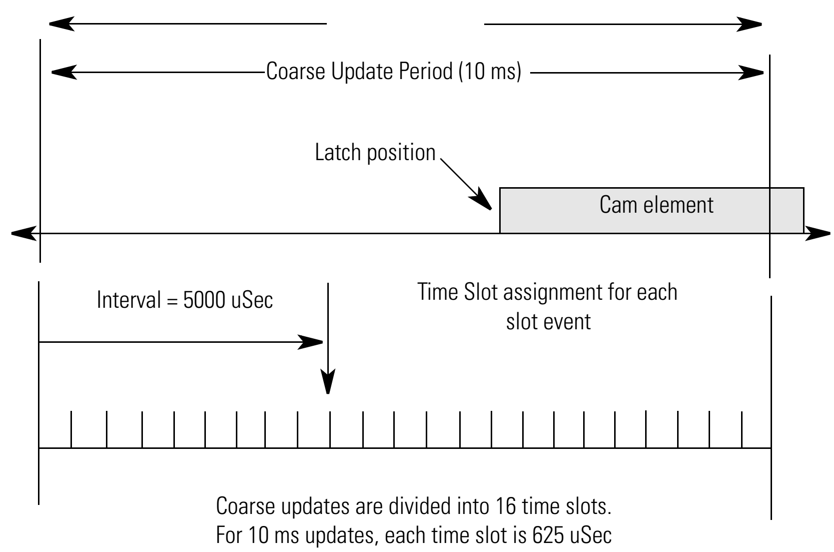
IMPORTANT	The outputs 0…7 can be forced by forcing the Data Bit to 0 or 1 and its corresponding bit in the ScheduleMask to 0.

Due to the limit of 16 schedules supported by the 1732E-OB8M8SR module, some constraints are applied to the number of events that can be processed every coarse update period.

Only eight schedules are available each coarse update. This allows for two consecutive coarse updates in which each update contains eight output events.

The following diagram illustrates the relationship between the coarse update period, a cam latch event and the time slots.

Inter-relationship of Coarse Update Period, Cam Latch, and Time Slots



Each Time Slot stores the information described in the following table.

Time Slot Information

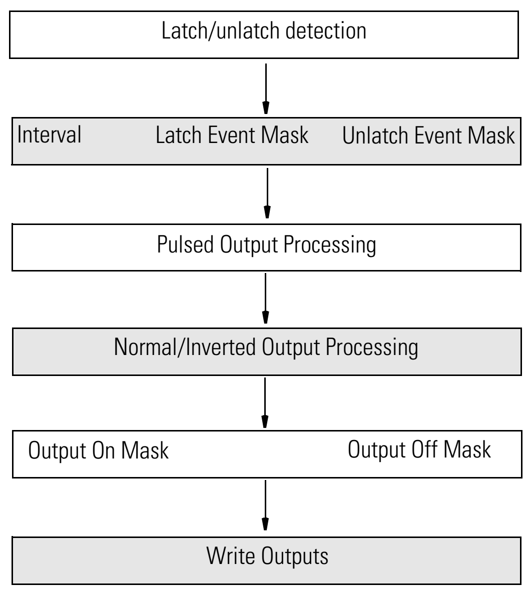
Topic	Description
Latch Event Mask	When a latch event is detected, the time slot in which it belongs is calculated and the bit in the Latch Event Mask corresponding to the output bit of the latch is set.
Unlatch Event Mask	When an unlatch event is detected, the time slot in which it belongs is calculated and the bit in the Unlatch Event Mask corresponding to the output bit of the unlatch is set.
Interval	The time in micro-seconds from the start of the coarse update in which the Latch or Unlatch event occurs.

Time Slot Information

Topic	Description
Pulse Off Mask	For pulsed outputs, the time slot in which a pulse off event is calculated and the bit in the Pulse Off Mask corresponding to the output bit of the pulse event is set.
Output On Mask	For normal outputs, the bit corresponding to the output bit of the Latch or Pulse On event is set indicating that the output is to be turned on for these events. For inverted outputs, the bit corresponding to the output bit of the Unlatch or Pulse Off event is set indicating that the output is to be turned on for these events.
Output Off Mask	For normal outputs, the bit corresponding to the output bit of the Unlatch or Pulse Off event is set indicating that the output is to be turned off for these events. For inverted outputs, the bit corresponding to the output bit of the Latch or Pulse On event is set indicating that the output is to be turned off for these events.

The following is a simplified overview of how Time Slot data is utilized.

Overview of How Time Slot Data Utilization



Time slots are also used to process overlapping cam elements. A semaphore is maintained to indicate the currently active state of each output bit. In addition, if a programmed cam element Latch and Unlatch event occurs in the same time slot, they cancel each other out.

I/O Subsystem

The user can specify the Output parameter of an MAOC instruction as either a memory tag or an Output Module's data tag. A pointer to the tag is passed into the MAOC instruction. Also passed into the MAOC instruction is an internal parameter of type IO_MAP. If the Output parameter references controller memory, the IO_MAP parameter is NULL. If the Output parameter references an output module, the IO_MAP parameter points to the map structure for the module. The MAOC instruction can then determine if the Output parameter is associated with a 1732E-OB8M8SR module by checking the module type stored in the driver table.

Output Data Structure

Field	Size	Description
Value	4 bytes	Data values for un-scheduled output bits. 0 = Off 1 = On
Mask	4 bytes	Selects which output bits are to be scheduled. The eight bits (0…7) can be scheduled. 0 = Not scheduled 1 = Scheduled

Array of 16 Schedule Structures

Field	Size	Description
Schedule ID	1 byte	Valid ID's are 1…16. Any other value indicates that the schedule is not to be considered.
Sequence Number	1 byte	The 1732E-OB8M8SR maintains a copy of the schedule. A change in sequence number tells the 1732E-OB8M8SR to process the data in this schedule.
Point ID	1 byte	Indicates the output bit associated with this schedule. Entered as a value 00…07.
Point Value	1 byte	Next state of output bit specified in Point ID. 0 = Off 1 = On
Timestamp	4 bytes	The lower 32 bits of CIPSync. Indicates when to change the state of the specified output bit.

Schedule Processing

The Value and Mask fields are processed and all unscheduled data bits are moved to the module output data store. This data is written to the output terminals after all schedules have been processed.

Each schedule is processed. The schedule is not considered if:

- the Schedule ID is not in the range of 1…16.
- the Point ID is not in the range of 0…7.
- the Sequence Number has not changed.

If the schedule is to be considered, it is marked “active”. All “active” schedules are examined every 200 ms. The schedule timestamp is compared to the current CIPSync time. If the current CIPSync time is greater than or equal to the scheduled Time Stamp, the Point Value in the schedule is moved to the module output data store for the specified output bit.

Chapter Summary and What’s Next

In this chapter, you learned how to use the modules. The next chapter describes interpreting the status indicators.

Notes:

Troubleshoot the Module

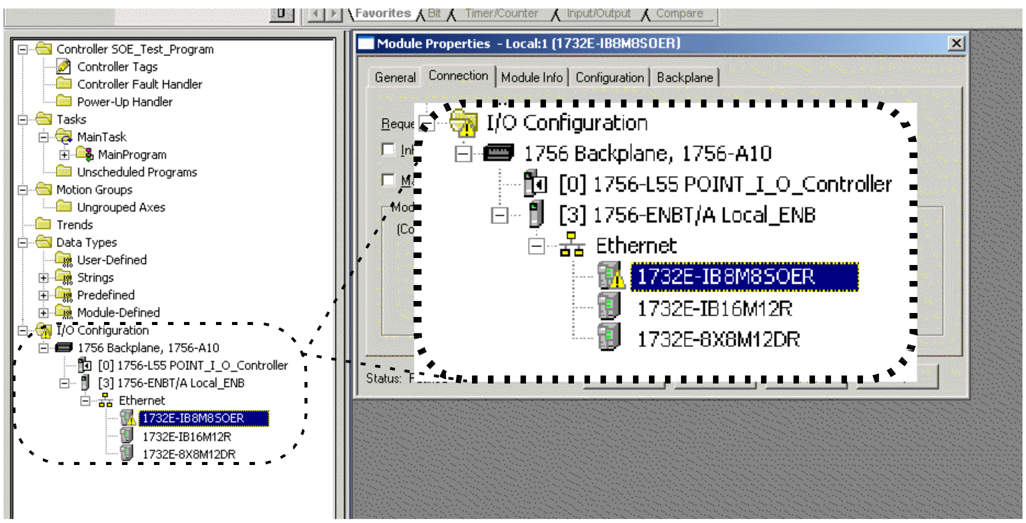
This chapter describes how to troubleshoot the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules using RSLogix 5000 software.

Troubleshoot the Module

In addition to the status indicators on the module, RSLogix 5000 software alerts you to fault and other conditions in one of three ways:

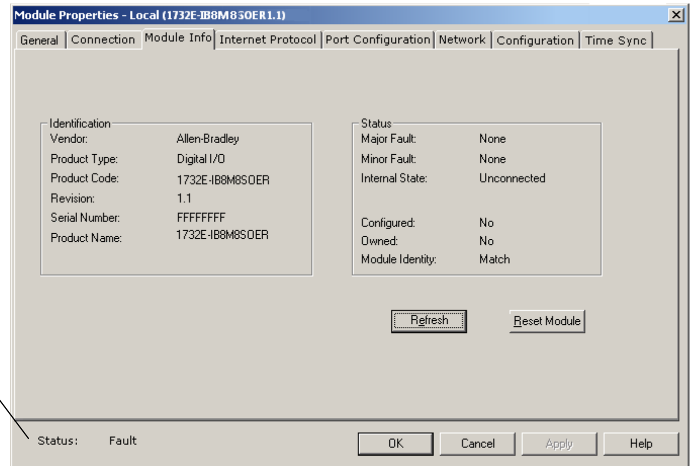
- Warning signal on the main screen next to the module – This occurs when the connection to the module is broken.

Warning icon appears when a communications fault occurs or if the module is inhibited



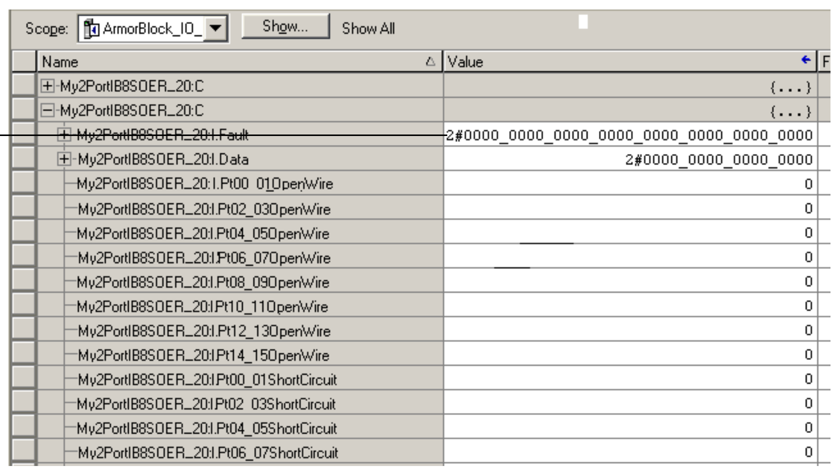
Warning signal – The module has a communications fault

- Message in a screen's status line.



Status line provides information on the module fault and on the connection to the module

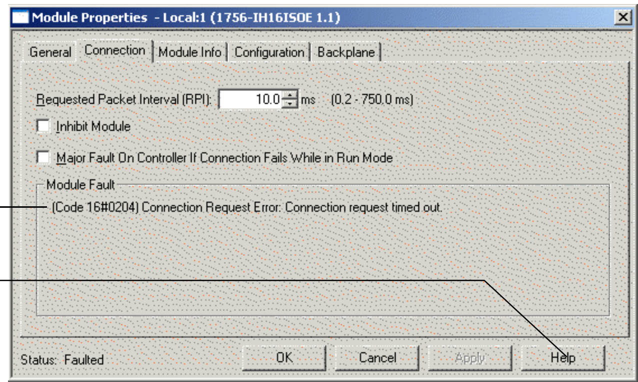
- Notification in the Tag Monitor – General module faults are also reported in the Tag Monitor. Communication faults are reported in the input tags.



RSLogix 5000 software generates 1 s in response to a module communication fault.
In this example, a communication fault occurred between the controller and the module, so the controller automatically writes 1 s for all bits in the word.

Determine Fault Type

When you are monitoring a module's configuration properties in RSLogix 5000 software and receive a Communications fault message, the Connection page lists the type of fault.



The fault type is listed here.

Click Help for a detailed listing of the possible faults, their causes and suggested solutions.

For a detailed listing of the possible faults, their causes and suggested solutions, see Module Faults in the RSLogix 5000 Software Online Help.

Refer to the RSLogix 5000 AOP help to troubleshoot using the Module Info tab, Internet Protocol tab, Port Diagnostics dialog, Time Sync tab, or Network tab. Access the AOP help by clicking Help on any of these tabs.

Chapter Summary

This chapter provided the user with an explanation of how to troubleshoot the Sequence of Events Input and Scheduled Sourcing Output modules using RSLogix 5000 software.

Notes:

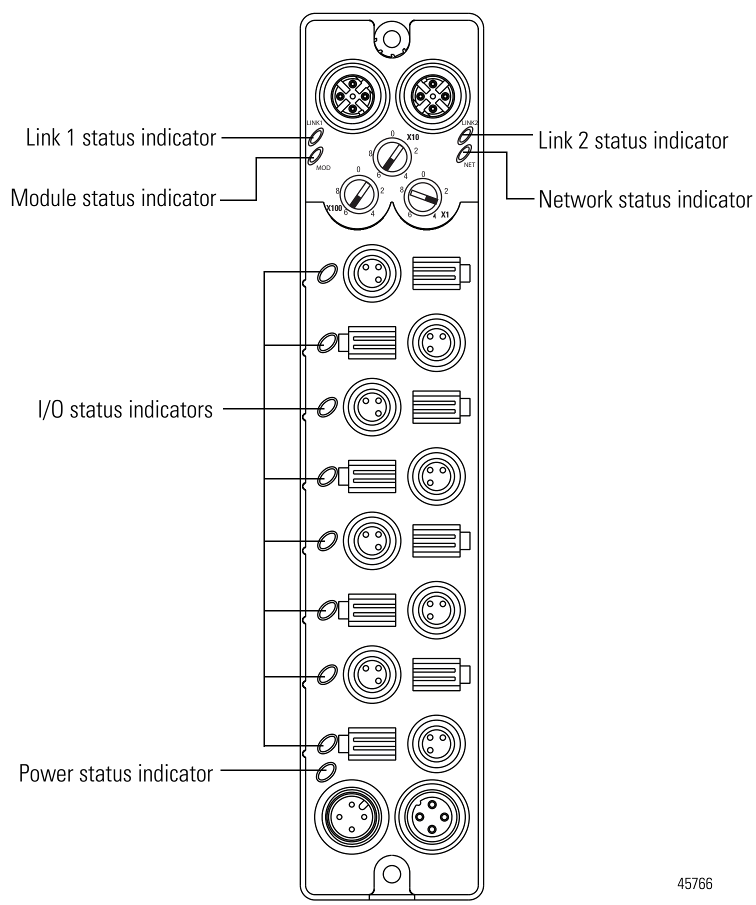
Interpret Status Indicators

Introduction

This chapter contains information about status indicators.

This module has the following indicators:

- Network, Module, and Link status indicators for EtherNet/IP
- Auxiliary power indicator
- Individual I/O status indicators for inputs.



Indicator Status for Modules

Indicator	Status	Description
Module status	Off	No power applied to the device.
	Flashing red/green	The module is performing POST (Power-On Self Test), which completes within 30 s.
	Green	Device operating normally.
	Flashing red	Module has experienced a recoverable fault.
	Red	Unrecoverable fault – may require device replacement.
	Flashing green	On **1732E-IB8M8SOER** and **1732E-OB8M8SR**: Device is not synchronized to master clock.
Network status	Off	The device is not initialized or the module does not have an IP address.
	Flashing green	The device has no CIP connections. The device has an IP address, but no CIP connections are established.
	Green	The device is online, has an IP address, and CIP connections are established.
	Flashing red	One or more connections have timed out.
	Red	The module has detected that its IP address is already in use.
	Flashing red/green	The module is performing a power-on self test (POST).
Network link status (Link 1/Link 2)	Off	No link established.
	Green	Link established on indicated port at 100 Mbps.
	Flashing green	Link activity present on indicated port at 100 Mbps.
	Yellow	Link established on indicated port at 10 Mbps.
	Flashing yellow	Link activity present on indicated port at 10 Mbps.
Power status	Off	No power to device or input not valid.
	Green	Power applied to device.
I/O status	Off	Output/input not energized.
	Yellow	Output/input energized.

IMPORTANT	The Module Status Indicator will flash red and green for a maximum of 30 s while the module completes its POST (Power-On Self Test).

Chapter Summary and What's Next

In this chapter, you learned how to interpret the Status Indicators on the module. The next chapter describes how to troubleshoot the module using RSLogix 5000 software.

Specifications

Specifications

The ArmorBlock Sequence of Events Input and Scheduled Sourcing Output modules (1732E-IB8M8SOER, 1732E-OB8M8SR) have the following specifications.

General Specifications

Attributes	Value
Voltage, power, max	30V DC
Voltage, power, min	12V DC
Current, Module Power, max per module	300 mA @ 24V DC
Current, Auxiliary Power, module only (no Digital Output loads, no Sensor Voltage Loads, and no power daisy-chain loads)	25 mA @ 24V DC
Current, Auxiliary Power, max per module (module plus Digital Output Loads, plus Sensor Voltage Loads, plus power daisy-chain loads)	4A @ 24V DC
Isolation voltage	Type tested @ 707V DC for 60s
Communication rate	EtherNet/IP 10/100 Mbps Full or half-duplex 100 meter per segment
Status indicators	Module status – red/green Network status – red/green Link status – green/yellow Power status – green I/O LED – yellow
Dimensions, approx., HxWxD	179 x 37 x 27 mm (7.05 x 1.46 x 1.06 in.)
Pilot Duty Rating	DC-14
Weight, approx.	0.34 kg (0.75 lb)
Wiring category(1)	1 – on signal ports 1 – on power ports 1 – on communication ports

(1) Use this Conductor Category information for planning conductor routing. Refer to publication 1770-4.1, Industrial Automation Wiring and Grounding Guidelines.

Input Specifications – 1732E-IB8M8SOER

Attributes	Value
Number of inputs	8 Sink Type
On-state voltage	11V DC, min 24V DC, nom 30V DC, max
Off-state voltage, max	5V DC

Input Specifications – 1732E-IB8M8SOER

Attributes	Value
On-state current, min	180 µA @ 11V DC
On-state current, max	5.0 mA @ 30V DC
Off-state current, max	90 µA @ 5V DC
Voltage sensor source, max	30V DC
Voltage sensor source, min	10V DC
Input filter	0 ms (default), 2 ms, 4 ms, 8 ms, and 16 ms

Output Specifications – 1732E-OB8M8SR

Attributes	Value
Number of outputs	8 sourcing type
On-state voltage	11V DC, min 24V DC, nom 30V DC, max
On-state current	0.5 A per output, up to 4 A per module
Leakage current, off-state output, max	50 µA
Pilot Duty Rating	DC-14
Surge current per output, max	1.2 A for 10 ms, repeatable every 2 s

Environmental Specifications

Attribute	Value
Temperature, operating	IEC 60068-2-1 (Test Ad, Operating Cold), IEC 60068-2-2 (Test Bd, Operating Dry Heat), IEC 60068-2-14 (Test Nb, Operating Thermal Shock): -20…60 °C (-4…140 °F)
Temperature, nonoperating	IEC 60068-2-1 (Test Ab, Unpackaged Nonoperating Cold), IEC 60068-2-2 (Test Bb, Unpackaged Nonoperating Dry Heat), IEC 60068-2-14 (Test Na, Unpackaged Nonoperating Thermal Shock): -40…85 °C (-40…185 °F)
Temperature, ambient, max	60 °C (140 °F)
Relative humidity	IEC 60068-2-30 (Test Db, Unpackaged Damp Heat): 5…95% noncondensing
Vibration	IEC60068-2-6 (Test Fc, Operating): 5 g @ 10…500 Hz
Shock, operating	IEC60068-2-27 (Test Ea, Unpackaged Shock): 30 g
Shock, nonoperating	IEC60068-2-27 (Test Ea, Unpackaged Shock): 50 g
Emissions	CISPR 11: Group 1, Class A
ESD immunity	IEC 61000-4-2: 6 kV contact discharges 8 kV air discharges
Radiated RF immunity	IEC 61000-4-3: 10V/m with 1 kHz sine-wave 80% AM from 80…2000 MHz 10V/m with 200 Hz 50% Pulse 100% AM @ 900 MHz 10V/m with 200 Hz 50% Pulse 100% AM @ 1890 MHz 1V/m with 1 kHz sine-wave 80% AM from 2000…2700 MHz

Environmental Specifications

Attribute	Value
EFT/B immunity	IEC 61000-4-4: ±3 kV at 5 kHz on power ports ±3 kV at 5 kHz on signal ports ±3 kV at 5 kHz on communication ports
Surge transient immunity	IEC 61000-4-5: ±2 kV line-line(DM) and ±2 kV line-earth(CM) on power ports ±1 kV line-line(DM) and ±2 kV line-earth(CM) on signal ports ±2 kV line-earth(CM) on shielded ports ±2 kV line-earth(CM) on communication ports
Conducted RF immunity	IEC 61000-4-6: 10V rms with 1 kHz sine-wave 80% AM from 150 kHz…80 MHz
Enclosure type rating	Meets IP65/66/67/69K (when marked)

Certifications

Certification (when product is marked)[1]	Value
c-UR-us	UL Listed Industrial Control Equipment, certified for US and Canada. See UL File E322657.
CE	European Union 2004/108/EC EMC Directive, compliant with: EN 61326-1; Meas./Control/Lab., Industrial Requirements EN 61000-6-2; Industrial Immunity EN 61000-6-4; Industrial Emissions EN 61131-2; Programmable Controllers (Clause 8, Zone A & B)
C-Tick	Australian Radiocommunications Act, compliant with: AS/NZS CISPR 11; Industrial Emissions
EtherNet/IP	ODVA conformance tested to EtherNet/IP specifications.

(1) See the Product Certification link at http://www.rockwellautomation.com/products/certification/ for Declarations of Conformity, Certificates, and other certification details.

16 self-configuring[1]

Notes:

Module Tags

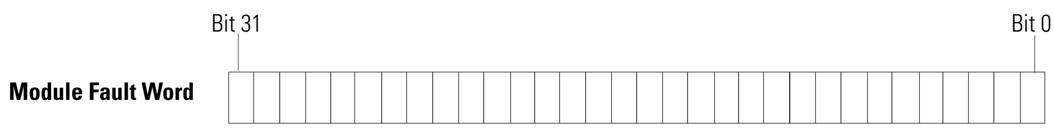
Fault and Status Reporting Between the Module and Controllers

The 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules modules send fault/status data to the owner-controller. The module maintains a Module Fault Word, the highest level of fault reporting.

The following table describes the tag that can be examined in ladder logic to indicate when a fault has occurred for your module:

Tag	Description
Module Fault Word	This word provides fault summary reporting. The tag name is Fault.

- If a communication fault occurs on the module, all 32 bits in the Module Fault Word are set to 1.



A communications fault sets **all bits** in the Module Fault Word.

42676

Module Tag Names and Definitions

The 1732E-IB8M8SOER and 1732E-OB8M8SR has three sets of tags:
- Configuration
- Input
- Output

Tags Used

Configuration, Input and Output Tags for 1732E-IB8M8SOER

The following table describes the configuration tags generated in RSLogix 5000 software when you use your 1732E-IB8M8SOER module.

Configuration Tags – 1732E-IB8M8SOER

Tag Name	Type	Description
C.FilterOffOn	INT	Sets the OFF to ON filter time for all 8 inputs. Times are set in µs increments of 0, 1000 (default), 2000, 4000, 8000 and 16000 µs. 0 = no filtering For more information on Software Configurable Input Filters, see page 55.

Configuration Tags – 1732E-IB8M8SOER

Tag Name	Type	Description
C.FilterOnOff	INT	Sets the ON to OFF filter time for all 8 inputs. Times are set in µs increments of 0, 1000 (default), 2000, 4000, 8000 and 16000 µs. 0 = no filtering For more information on Software Configurable Input Filters, see page 55.
C.LatchEvents	BOOL	Latches events so that an event will not be overwritten until acknowledged. 0 = SOE not latched 1 = SOE latched (default) Latched means that a sequence of events of LO to HI and HI to LO then LO to HI will cause the first LO to HI transition to be recorded and the final LO to HI to be ignored. All subsequent transitions on that point will be ignored until acknowledged/reset. If the bit is not set, the new LO to HI will overwrite the first LO to HI event immediately, even if the controller has yet to extract that data.
C.MasterSyncEn	BOOL	PTP enabled bit indicates if the module is expected to sync to a master clock. 0 = Synchronization indication disabled (default) 1 = Synchronization indication enabled If not enabled (0) then the Module Status Indicators will not flash green if we are not synched to a master clock. Disabling the bit does not prevent the module from synchronizing to a master clock.
C.Pt0CaptureOffOn…Pt7CaptureOffOn	BOOL	Enables capturing OFF to ON events on a per point basis. If disabled (0), that point will not record timestamp data for OFF to ON input transitions. 0 = Capture disabled for OFF to ON input transitions 1 = Capture enabled (default) for OFF to ON input transitions This option is useful if you want to avoid reporting data on the module for events in which you have no interest.
C.Pt0CaptureOnOff…Pt7CaptureOnOff	BOOL	Enables capturing ON to OFF events on a per point basis. If disabled (0), that point will not record timestamp data for ON to OFF input transitions. 0 = Capture disabled for ON to OFF input transitions 1 = Capture enabled (default) for ON to OFF input transitions This option is useful if you want to avoid reporting data on the module for events in which you have no interest.

Input Tags – 1732E-IB8M8SOER

Tag Name	Type	Set on Per Point or Modulewide basis	Description
I.Fault	DINT	Modulewide	Communication fault – The controller sets this tag to 1 for all 32 bits if a communication fault occurs on the module otherwise all bits are zero.
I.Data	SINT	Per point	Status of the input point. This data is filtered if the Input Filter feature is used on the module. Thus, an input change must pass through the filter before it is seen in this tag. 0 = input is OFF 1 = input is ON For example, if input 3 is ON, *I.Data.3* = 1.
I.NewData	SINT	Per point[1]	Flag indicating if new timestamp data was detected on the input. 0 = no new timestamp data on the input 1 = new timestamp data on the input (since last acknowledged) Because input data for all inputs is sent the next RPI after each timestamped transition, this tag is useful to quickly determine on which input the transition occurred. For example, if the module sends new input data to the owner-controller and *I.NewData.5* = 1, you know that at least one of the timestamps for input 5 (*I.Timestamp[5].OffOn* or *I.Timestamp[5].OnOff*) has new data. This tag only clears when the controller acknowledges the new data or all events on the module are reset. For more information, see page 78.

Input Tags – 1732E-IB8M8SOER

Tag Name	Type	Set on Per Point or Modulewide basis	Description
I.EventOverflow	SINT	Per point	Set for an input when the module either: • Does not timestamp a transition on the input – The module has Timestamp Latch enabled and a similar transition has already been timestamped on this input but has not been cleared via the O.EventAck and O.NewDataAck output tags (see page 76). or • Overwrites previously-recorded timestamp data for the input – The module has Timestamp Latch disabled and multiple transitions occur on the input. In this case, timestamp data from new transitions are recorded before previously-recorded transitions were cleared from the input via the O.EventAck and O.NewDataAck output tags (see page 76). This value is cleared if the module is reset.
I.LocalClockOffset	DINT[2]	Modulewide	The offset from the local clock to the system time. This value is useful for detecting steps in time. This value updates when a PTP update is received.
I.OffsetTimeStamp	DINT[2]	Modulewide	The time when the PTP message was received to cause the Local Clock Offset to update. This value is initially zero. The first timestamp occurs when the module synchronizes with the Grandmaster clock.
I.GrandMasterClockID	DINT[2]	Modulewide	The ID number of the Grandmaster clock that the module is synchronized to.
I.Timestamp[8].OffOn	DINT[2]	Per point	Timestamp value with an input's OFF to ON transition. This tag is a 8 x 1 32-bit array. This value is cleared after the data has been acknowledged via the *O.EventAck* and *O.NewData* tags. For more information on clearing timestamp data, see page 76.
I.Timestamp[8].OnOff[2]	DINT[2]	Per point	Timestamp value with an input ON to OFF transition. This tag is a 8 x 1 32-bit array. This value is cleared after the data has been acknowledged via the *O.EventAck* and *O.NewData* tags. For more information on clearing timestamp data, see page 76.
I.EventNumber.x	DINT	Modulewide	Running count of the timestamped transitions; this tag increments by one with each new transition that the module timestamps and rolls over to 1, not 0. This value is cleared if the module is reset.
I.SyncToMaster	BOOL	Modulewide	Indicates if the module is synchronized with a master clock. 1 = Synchronized 0 = Not synchronized

(1) With the Per point tags, there is one bit per input. For example, bit 0 represents input 0, bit 7 represents input 7 and so on.

Output Tags – 1732E-IB8M8SOER

Tag Name	Type	Description
O.EventAck	DINT	For the bits selected in the O.NewDataAck tag, this tag selects which edge to acknowledge, On to Off, Off to On or both. 0 = acknowledging an ON to OFF event 1 = acknowledging an OFF to ON event 2 = acknowledging both ON to OFF and OFF to ON events The O.NewDataAck tag must also be used to acknowledge the event(s).
O.NewDataAck	SINT	Allows I.NewData bits and I.Timestamp data updates in the Input tag to function as intended. I.NewData bits are set and I.Timestamp data updates when a transition occurs and clear only after they are acknowledged via the O.NewDataAck bit. Typically, the following events occur: • An event occurs on an input. • The module sets the I.NewData bit and I.Timestamp data for the input where the event occurred. • The controller records the new data. • The controller acknowledges the new data by causing a 0 to 1 transition on the corresponding O.NewDataAck bit. • The I.NewData bit and I.Timestamp data clears. • When another event occurs on the input, the sequence begins at the top bullet in this list. The controller must cause a 0 to 1 transition in this bit to acknowledge new data for an input; in other words, if the NewDataAck bit is 0 when new data is received, the controller must change this bit to 1 to acknowledge the data. If NewDataAck bit is 1 when new data is received, the controller must change this bit to 0 and then at least one RPI later to 1 to acknowledge the new data.
O.PtToRetrieve	SINT	When RetrieveByPoint is set, determines point requested for events to be returned. Allows a user to query events in the sequence they happened by point instead of the overall sequence the events occurred.
O.ResetEvents	BOOL	Erases all recorded events when transitioned from 0…1.
O.RetrieveByPt	BOOL	Changes retrieval mechanism from sequential (when 0) to retrieving events on a per point basis according to value set in PointToRetrieve field.

Configuration, Input, Output Tags for 1732E-OB8M8SR

The following table describes the configuration tags generated in RSLogix 5000 software when you use your 1732E-OB8M8SR module.

Configuration Tags – 1732E-OB8M8SR

Tag Name	Type	Description
C.ModuleCfgBits	SINT	Configuration Revision Number.
C.ProgToFaultEn	BOOL	Selects whether (enabled) or not (disabled) to apply the fault value when the output is already being set to the program value when a Fault (connection timeout) occurs in Program Mode.

Configuration Tags – 1732E-OB8M8SR

C.Pt00FaultMode…Pt07FaultMode	BOOL	The Pt0xFaultMode is used in conjunction with FaultValue to configure the state of output x (that is, Pt00FaultMode for output 0, Pt01FaultMode for output 1, Pt02FaultMode for output 2, and so on up to Pt07FaultMode for output 7) when a communications fault occurs. A value of 0 means that, in the case of a communications fault, the value in FaultValue will be used (Off or On). A value of 1 means that the last state will be held. By default this value is 0.
C.Pt00FaultValue…Pt07FaultValue	BOOL	The Pt0xFaultValue is used in conjunction with FaultMode to configure the state of the output x (that is, Pt00FaultValue for output 0, Pt01FaultValue for output 1, Pt02FaultValue for output 2 and so on up to Pt07FaultValue for output 7) when a communications fault occurs. 0 = Off (default) 1 = On
C.Pt00FaultFinalState…Pt07FaultFinalState	BOOL	If FaultMode is set (Hold Last State) and HoldLastStateDuration[8] is non-zero, this tag determines the final Output state after the configured time out occurs.
C.Pt00ProgMode…Pt07ProgMode	BOOL	The Pt0xProgMode is used in conjunction with ProgValue to configure the state of output 0 when the controller is in Program mode. A value of 0 means that the ProgValue (Off or On) will be used when the controller is in Program mode. A value of 1 means that the last state will be held. By default this value is 0.
C.Pt00ProgValue…Pt07ProgValue	BOOL	The Pt0xProgValue is used in conjunction with ProgMode to configure the state of output x when the controller is in Program mode. A value of 0 is Off, and a value of 1 is On. By default this value is 0.
C.Pt00HoldLastStateDuration…Pt07HoldLastStateDuration	BOOL	If FaultMode is set (Hold Last State), this value determines the length of time the last state is to be held prior to the FaultFinalState being applied. Valid values are 0 = Hold Forever, and either 1, 2, 5, or 10 (indicating hold time in seconds). All other values reserved.

Input Tags – 1732E-OB8M8SR

Tag Name	Type	Description
I.Fault	DINT	Communication fault – The controller sets this tag to 1 for all 32 bits if a communication fault occurs on the module otherwise all bits are zero.
I.Data	SINT	Status of the input point. This data is filtered if the Input Filter feature is used on the module. Thus, an input change must pass through the filter before it is seen in this tag. 0 = input is OFF 1 = input is ON For example, if input 3 is ON, I.Data.3 = 1.
I.SyncToMaster	BOOL	When set, indicates the module has synced to a Valid Time Master. 1 = Synchronized 0 = Not synchronized
I.SyncTimeout	BOOL	Indicates a Valid local CIP Sync Timemaster has since timed out.
I.LateScheduleCount	INT	Indicates that a schedule request arrived at the module after the schedule time. The counter rolls over to 1 every 65,535 late updates.
I.LostScheduleCount	INT	Indicates that a schedule sequence number has been skipped, thus a schedule request has been lost. The counter rolls over to 1 every 65535 lost updates.
I.LocalClockOffset	DINT[2]	The offset from the local clock to the system time. This value is useful for detecting steps in time. This value updates when a PTP update is received.

Input Tags – 1732E-OB8M8SR

Tag Name	Type	Description
I.OffsetTimeStamp	DINT[2]	The time when the PTP message was received to cause the Local Clock Offset to update. This value is initially zero. The first timestamp occurs when the module synchronizes with the Grandmaster clock.
I.GrandmasterClockID	DINT[2]	The ID number of the Grandmaster clock that the module is synchronized to.
I.TimeStamp	DINT[2]	Timestamp to be used with scheduled outputs and CIPSync/PTP. Used to synchronize outputs across the system by indicating the time which the output module is to apply its outputs.
I.Schedule.State	SINT	Current state of the schedule at index xx: 0 = Inactive 1 = Active, (that is, schedule is next to be applied but not within next scheduling) 2 = Current, (that is, schedule is next to be applied and within next scheduling period 3 = expired, (that is, schedule has been applied) 4 = Schedule discarded – request in error 5 = Late, but applied – schedule arrived after scheduled time but was applied since no more current schedule received
I.Schedule.SequenceNumber	SINT	Echo of SequenceNumber from the output image. Valid values 0…3.

Output Tags – 1732E-OB8M8SR

Tag Name	Type	Description
O.Data	SINT	Output Data to apply to unscheduled channels (those with a value of zero configured in ScheduleMask).
O.ScheduleMask	SINT	Mask indicating which channels are scheduled. Per bit 0 = use Data value (normal output) 1 = use scheduled output.
O.TimeStampOffset	DINT	System Time to Local Time Offset. Output should monitor for a delta between send value and module's value and run a Step Compensation Algorithm if the difference is >10 ms.
O.ScheduleTimeStamp		This is the 64-bit system time the schedule TimeStampOffset will use to apply the output. The ouput will transition at ScheduleTimeStamp + TimeStampOffset.
O.Schedule.ID	DINT	Indicates which schedule is to be loaded with attached data. Valid schedules are 1…16. 0 = No schedule.
O.Schedule.SequenceNumber	SINT	Echo of sequence number in output data. In order to acknowledge receipt of an event the user must transition the corresponding NewDataAck bit from 0 to 1 and set the EventAck to indicate whether to acknowledge the Off-On or On-Off transition for the input. the NewDataAck bits and EventAck are in consumed assembly 139. Timestamps are zero at power-up and after a timestamp is acknowledged. The time base and epoch of the timestamps are determined by the grandmaster clock of the system. All data listed in this assembly is in Little Endian format, LSB first, in increasing byte order to MSByte last. Value must be changed for a new schedule to be recognized. Valid values 0…3.
O.Schedule.OutputPtSelect	SINT	The output point the schedule is applied to. Valid values 0…7.
O.Schedule.Data	SINT	Output Data to apply to unscheduled channels (those with a value of zero configured in ScheduleMask).
O.Schedule.TimeStampOffset	DINT	Offset to add with TimestampOffset to determine absolute Time to apply Data to physical Output. Allows for range of approx. ±35 minutes from base TimestampOffset.

Data Tables

Communicate with Your Module

Read this section for information about how to communicate with your module.

I/O messages are sent to (consumed) and received from (produced) the ArmorBlock I/O modules. These messages are mapped into the processor or scanner memory. The following table lists the assembly instances and connection points for the 1732E EtherNet/IP ArmorBlock Sequence of Events Input and Scheduled Sourcing Output Modules.

Connection Points

In the following tables, Input pertains to the produced data from the module to the controller, and Output refers to consumed data by the module from the controller.

Catalog Number	Connection Format	Timestamp	Module-Defined Data Type	Assembly Instance / Connection Point (decimal)
1732E-IB8M8SOER	Data (default)	Per Point (default)	Configuration	111
			Input	157
			Output	159
	Data	First In First Out	Configuration	111
			Input	158
			Output	159
	Data	None	Configuration	111
			Input	104
			Output	191
	Listen Only	None	Configuration	189
			Input	104
			Output	191
1732E-OB8M8SR(1)		Per Point	Configuration	171
			Input	177
			Output	174
		None	Configuration	171
			Input	191
			Output	34

(1) There is no Listen Only connection for the 1732E-OB8M8SR because consumed assembly data is dependent on data in the produced assembly.

Configuration Assembly Instance 124 Data Structure (Configuration Header)

Configuration Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved (Ignored)							CRN
1	Reserved (Ignored)							
2	Reserved (Ignored)							
3	Reserved (Ignored)							
CRN	Configuration Revision Number							

Assembly Instance 4 Data Structure

Produced Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0

Assembly Instance 34 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0

Assembly Instance 111 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Group 0 Input OFF_ON Delay Filter (Low Byte)							
5	Group 0 Input OFF_ON Delay Filter (High Byte)							
6	Group 0 Input ON_OFF Delay Filter (Low Byte)							
7	Group 0 Input ON_OFF Delay Filter (High Byte)							
8	Reserved (Ignored)						Master Sync Enable	Latch Events
9	Capture OffOn7	Capture OffOn6	Capture OffOn5	Capture OffOn4	Capture OffOn3	Capture OffOn2	Capture OffOn1	Capture OffOn0
10	Capture OnOff7	Capture OnOff7	Capture OnOff7	Capture OnOff7	Capture OnOff7	Capture OnOff7	Capture OnOff7	Capture OnOff7
11	Reserved (Ignored)							
Where:	***MasterSyncEnable*** – This is a PTP enable bit which indicates if the module is expected to synch to a master clock. If not enabled (0), then the Module LED does not flash green if not synched to a master clock. Disabling the bit does not prevent the module from synchronizing to a master clock. ***LatchEvents*** – When set, latches events means that an event is not overwritten until acknowledged. In Per Point mode, a sequence of events of LO, HI, LO will cause the first LO and HI transitions to be recorded and the final LO to be ignored. All subsequent transitions on that point will be ignored until acknowledged/reset. If the bit is not set, the new LO will overwrite the first LO event immediately, even if the controller has yet to extract that data. In FIFO mode, when set events will not be erased from the FIFO until read. ***CaptureOffOn*** – Enables capturing Off to On events on a per point basis. If cleared, that point will not record Off to On events. Useful to not use up buffer space on events user does not care about. ***CaptureOnOff*** – Enables capturing On to Off events on a per point basis. If cleared, that point will not record On to Off events. Useful to not use up buffer space on events user does not care about.							

Produced Assembly Instance 157 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved (Must be 0)							
1	Reserved (Must be 0)							
2	Reserved (Must be 0)							
3	Reserved (Must be 0)							
4	In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0
5	NewData7	NewData6	NewData5	NewData4	NewData3	NewData2	NewData1	NewData0
6	EventOv7	EventOv6	EventOv5	EventOv4	EventOv3	EventOv2	EventOv1	EventOv0
7	Pad							
8…15	Local Clock Offset (64 bit)							
16…23	Offset Time Stamp (64 bit)							
24…31	Grandmaster Clock ID (64 bit) 8 bytes SINT array							
32…39	In0 Off-On Time Stamp (64 bit)							
40…47	In0 On-Off Time Stamp (64 bit)							
48…55	In1 Off-On Time Stamp (64 bit)							
56…63	In2 Off-On Time Stamp (64 bit)							
64…71	In2 Off-On Time Stamp (64 bit)							
72…79	In2 On-Off Time Stamp (64 bit)							
80…87	In3 Off-On Time Stamp (64 bit)							
88…95	In3 On-Off Time Stamp (64 bit)							
96…103	In4 Off-On Time Stamp (64 bit)							
104 …111	In4 On-Off Time Stamp (64 bit)							
112…119	In5 Off-On Time Stamp (64 bit)							
120…127	In5 On-Off Time Stamp (64 bit)							
128…135	In6 Off-On Time Stamp (64 bit)							
136…143	In6 On-Off Time Stamp (64 bit)							
144…151	In7 Off-On Time Stamp (64 bit)							
152…159	In7 On-Off Time Stamp (64 bit)							
160…163	Event Number (32 Bits)							
164	Reserved (Must be 0)							Synched to Master

Produced Assembly Instance 157 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Where:	***NewData*** – New data, has been detected upon that input and an unread event is queued for that point. ***EventOv*** – Set whenever the module begins to lose events for that input point. Events may be lost when new events are either ignored or overwriting existing events which have yet to be acknowledged. ***EventNumber*** – Running count of events which increments by one each new event. Allows the controller to check for a new events easily by comparing this number to the last retrieved event. If the EventNumber reaches its maximum value and rolls over it will roll-over to 1, not 0. ***In Time Stamp*** – Timestamp corresponding to when an event was recorded at one of the modules inputs. ***Local Clock Offset*** – The offset from the local clock to the system time. This value is useful for detecting steps in time. This value will update when a PTP update is received. ***Offset Time Stamp*** – The time when the PTP message was received that caused the Local Clock Offset to update. This value will initially be zero and the first timestamp will occur when the module synchronizes with the master clock. ***Grandmaster Clock ID*** – The ID number of the Grandmaster clock the module is synchronized to. ***Synced to Master*** – 1 indicates the module is synchronized with a master clock. 0 indicates it is not. In order to acknowledge receipt of an event the user must transition the corresponding NewDataAck bit from 0 to 1 and set the EventAck to indicate whether to acknowledge the Off-On or On-Off transistion for the input. The NewDataAck bits and EventAck are in consumed assembly instance159. Time stamps will be zero at power-up and after a time stamp is acknowledged. The time base and epoch of the timestamps are determined by the grand-master clock of the system. All data listed in this assembly is in Little Endian format, LSB first, in increasing byte order to MSByte last.							

Produced Assembly Instance 158 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved (Must be 0)							
1	Reserved (Must be 0)							
2	Reserved (Must be 0)							
3	Reserved (Must be 0)							
4	In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0
5	NewData7	NewData6	NewData5	NewData4	NewData3	NewData2	NewData1	NewData0
6	EventOv7	EventOv6	EventOv5	EventOv4	EventOv3	EventOv2	EventOv1	EventOv0
7	Pad							
8…15	Local Clock Offset (64 bit)							
16…23	Offset Time Stamp (64 bit)							
24…31	Grandmaster Clock ID (64 bit) 8 bytes SINT array							
32…39	Input Time Stamp 0 (64 bit)							
40…43	Event Number 0 (32 bit)							
44	Event Point 0							
45	Reserved (Must be 0)							EventData0
46…47	Pad (16 Bits)							
48…55	Input Time Stamp 1 (64 bit)							
56…59	Event Number 1 (32 bit)							
60	Event Point 1							
61	Reserved (Must be 0)							EventData1
62…63	Pad (16 Bits)							
64…71	Input Time Stamp 2 (64 bit)							

Produced Assembly Instance 158 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
72…75	Event Number 2 (32 bit)							
76	Event Point 2							
77	Reserved (Must be 0)							EventData2
78…79	Pad (16 Bits)							
80…87	Input Time Stamp 3 (64 bit)							
88…91	Event Number 3 (32 bit)							
92	Event Point 3							
93	Reserved (Must be 0)							EventData3
94…95	Pad (16 Bits)							
96…103	Input Time Stamp 4 (64 bit)							
104…107	Event Number 4 (32 bit)							
108	Event Point 4							
109	Reserved (Must be 0)							EventData4
110…111	Pad (16 Bits)							
112…119	Input Time Stamp 6 (64 bit)							
120…123	Event Number 6 (32 bit)							
124	Event Point 6							
125	Reserved (Must be 0)							
126…127	Pad (16 Bits)							
128…135	Input Time Stamp 6 (64 bit)							
136…139	Event Number 6 (32 bit)							
140	Event Point 6							
141	Reserved (Must be 0)							EventData6
142…143	Pad (16 Bits)							
144…151	Input Time Stamp 7 (64 bit)							
152…155	Event Number 7 (32 bit)							
156	Event Point 7							
157	Reserved (Must be 0)							
158…159	Pad (16 Bits)							
160…167	Input Time Stamp 8 (64 bit)							
168…171	Event Number 8 (32 bit)							
172	Event Point 8							
173	Reserved (Must be 0)							EventData8
174…175	Pad (16 Bits)							
176…183	Input Time Stamp 9 (64 bit)							
184…187	Event Number 9 (32 bit)							
188	Event Point 9							

Produced Assembly Instance 158 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
189	Reserved (Must be 0)							
190…191	Pad (16 Bits)							
192…195	Events Queued Count (32 Bits)							
196	Reserved (Must be 0)							Synched to Master
Where:	***NewData*** – Set when new data has been detected upon that input and an unread event is queued for that point. Cleared when all the time stamps for that point have been acknowledged or if the corresponding NewDataAck bit transitions from 0…1. ***EventOv*** – Set when the module begins to lose events for that input point. Events may be lost when new events are either ignored or overwriting existing events which have yet to be acknowledged. Cleared when the corresponding NewDataAck bit transitions from 0…1. ***EventNumber*** – Running count of events which increments by one each new event. Allows controller to check for a new event by comparing this number to the last retrieved event. Acknowledge of receipt of event which causes next event to display is to write this EventNumber back into the output data EventAck field. If the EventNumber reaches it maximum value and rolls over it should roll-over to 1 not 0. ***Input Time Stamp*** – Timestamp corresponding to when an event was recorded at one of the modules inputs.							
	Local Clock Offset – The offset from the local clock to the system time. This value is useful for detecting steps in time. This value will update when a PTP update is received. ***Offset Time Stamp*** – The time when the PTP message was received that caused the Local Clock Offset to update. This value will initially be zero and the first timestamp will occur when the module synchronizes with the master clock. ***Grandmaster Clock ID*** – The ID number of the Grandmaster clock the module is synchronized to. ***EventPoint*** – Which of the 8 channels the event was recorded on (values of 0…7). ***EventData*** – Bit indicating if event was a change of state to a 1 or 0. ***EventsQueuedCount*** – How many events are currently queued up which have not been read. ***Synced to Master*** – 1 indicates the module is synchronized with a master clock. 0 indicates it is not. In order to acknowledge receipt of an event the user must write this EventNumber back into the output data EventAck field. The EventAck is in consumed assembly 159. When the users returns the EventAck, then ack all time stamps in this assembly that are less than or equal to the EventAck. Time stamps will be zero at power-up and after a time stamp is acknowledged. The time base and epoch of the timestamps are determined by the grand-master clock of the system. All data listed in this assembly is in Little Endian format, LSB first, in increasing byte order to MSByte last.							

Consumed Assembly Instance 159 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0…3	Event Ack (32 Bits)							
4	NewData Ack7	NewData Ack7	NewData Ack7	NewData Ack7	NewData Ack7	NewData Ack7	NewData Ack7	NewData Ack7
5	Point to Retrieve							
6	Reserved (Ignore)						Retrieve by Point	Reset Events

Consumed Assembly Instance 159 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Where:	**FIFO Mode** ***EventAck*** – The controller writes back the EventNumber read to transition buffers. All events with EventNumbers less than or equal to the EventAck will be acknowledged. If the RetreiveByPoint bit is set the PointToRetrieve attribute must also be used to specify the point to retrieve next. ***NewDataAck*** – Forces NewData bits clear even if all the timestamps for that point have not been acknowledged. Also clears the EventOv bits. ***PointToRetrieve*** – When RetrieveByPoint is set, determines point requested for events to be returned. Allows a user to query events in the sequence they happened by point instead of the overall sequence the events occurred. ***RetrieveByPoint*** – Changes retrieval mechanism from sequential (when 0) to retrieving events on a per point basis according to value set in PointToRetrieve field. **Per Point Mode** ***EventAck*** – Is a 0 or 1 to indicate acknowledging an OnOff or OffOn event respectively, or a 2 to acknowledge both. ***NewDataAck*** – Acknowledges the corresponding inputs timestamp and clears its NewData and EventOv bits. ***PointToRetrieve*** – Not used. ***RetrieveByPoint*** – Not used. ***Reset Events*** – When transitioned from 0 to 1, erases all recorded time stamped events.							

Configuration Assembly Instance 163 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Group 0 Input OFF_ON Delay Filter (Low Byte)							
5	Group 0 Input OFF_ON Delay Filter (High Byte)							
6	Group 0 Input ON_OFF Delay Filter (Low Byte)							
7	Group 0 Input ON_OFF Delay Filter (High Byte)							
8	Reserved (Ignored)				IV_G0	IA_G0	FV_G0	FA_G0
9	Reserved (Ignored)							
Where:	***FA_G0*** = Fault Action Group 0 (0 = Apply Fault Value; 1= Hold Last State) ***IA_G0*** = Idle Action Group 0 (0 = Apply Fault Value; 1= Hold Last State) ***FV_G0*** = Fault Value Group 0 (0 = OFF;1 = ON) ***IV_G0*** = Idle Value Group 0 (0 = OFF;1 = ON)							

Configuration Assembly Instance 171 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved (Ignored)							CRN
1	Reserved (Ignored)							ProgTo FaultEn
2	FaultMode Ch7	FaultMode Ch6	FaultMode Ch5	FaultMode Ch4	FaultMode Ch3	FaultMode Ch2	FaultMode Ch1	FaultMode Ch0
3	Reserved (Ignored)							
4	FaultVal Ch7	FaultVal Ch6	FaultVal Ch5	FaultVal Ch4	FaultVal Ch3	FaultVal Ch2	FaultVal Ch1	FaultVal Ch0
5	Reserved (Ignored)							
6	FaultFinal State Ch7	FaultFinal State Ch6	FaultFinal State Ch5	FaultFinal State Ch4	FaultFinal State Ch3	FaultFinal State Ch2	FaultFinal State Ch1	FaultFinal State Ch0
7	Reserved (Ignored)							

Configuration Assembly Instance 171 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8	ProgMode Ch7	ProgMode Ch6	ProgMode Ch5	ProgMode Ch4	ProgMode Ch3	ProgMode Ch2	ProgMode Ch1	ProgMode Ch0
9	Reserved (Ignored)							
10	ProgVal Ch7	ProgVal Ch6	ProgVal Ch5	ProgVal Ch4	ProgVal Ch3	ProgVal Ch2	ProgVal Ch1	ProgVal Ch0
11	Reserved (Ignored)							
12…15	Hold Last State Duration [0]							
16…19	Hold Last State Duration [1]							
20…23	Hold Last State Duration [2]							
24…27	Hold Last State Duration [3]							
28…31	Hold Last State Duration [4]							
32…35	Hold Last State Duration [5]							
36…39	Hold Last State Duration [6]							
40…43	Hold Last State Duration [7]							
Where:	***ProgToFaultEn*** – Selects whether (enabled) or not (disabled) to apply the fault value when the output is already being set to the program value when a Fault (connection timeout) occurs in Program Mode. ***FaultMode*** – If Fault occurs (connection loss): When clear use FaultValue for Output. When set, Output Holds Last State for HoldLastStateDuration. ***Fault Value*** – If corresponding FaultMode bit clear, defines Output Value on Fault (connection timeout). Not used if FaultMode bit is set. ***Hold Last State Duration*** – If FaultMode is set (Hold Last State), this value determines the length of time the last state is to be held prior to the FaultFinalState being applied. Valid values are 0 = Hold Forever, and either 1, 2, 5, or 10 (indicating hold time in seconds). All other values reserved. ***FaultFinalState*** – If FaultMode is set (Hold Last State) and HoldLastStateDuration[8] is non-zero, determines the final Output state after the configured time out occurs. ***ProgMode*** – If ProgramMode event occur: When clear use ProgValue for Output. When set, Output Holds Last State. ***ProgValue*** – If corresponding ProgMode bit clear, defines Output value on Program Mode. Not used if ProgMode bit set.							

Consumed Assembly Instance 174 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Data Ch7	Data Ch6	Data Ch5	Data Ch4	Data Ch3	Data Ch2	Data Ch1	Data Ch0
1	Reserved (Ignored)							
2	ScheduleMa sk Ch7	Schedule Mask Ch7	Schedule Mask Ch7	Schedule Mask Ch7	Schedule Mask Ch7	Schedule Mask Ch7	Schedule Mask Ch7	Schedule Mask Ch0
3	Reserved (Ignored)							
4	Timestamp Offset							
5	Schedule Timestamp							
6	Schedule[0].ID							
7	Schedule[0].SequenceNumber							
8	Schedule[0].OutputPointSelect							
9	Schedule[0].Data							

Consumed Assembly Instance 174 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
10	Schedule[0].TimestampOffset							
11	Schedule[1].ID							
12…15	Schedule[1].SequenceNumber							
16…19	Schedule[1].OutputPointSelect							
20…23	Schedule[1].Data							
24…27	Schedule[1]. TimestampOffset							
28…31	Schedule[2].ID							
32…35	Schedule[2].SequenceNumber							
36…39	Schedule[2].OutputPointSelect							
40…43	Schedule[2].Data							
44…47	Schedule[2]. TimestampOffset							
48	Schedule[3].ID							
49	Schedule[3].SequenceNumber							
50	Schedule[3].OutputPointSelect							
51	Schedule[3].Data							
52…55	Schedule[3]. TimestampOffset							
56	Schedule[4].ID							
57	Schedule[4].SequenceNumber							
58	Schedule[4].OutputPointSelect							
59	Schedule[4].Data							
60…63	Schedule[4]. TimestampOffset							
64	Schedule[5].ID							
65	Schedule[5].SequenceNumber							
66	Schedule[5].OutputPointSelect							
67	Schedule[5].Data							
68…71	Schedule[5]. TimestampOffset							
72	Schedule[6].ID							
73	Schedule[6].SequenceNumber							
74	Schedule[6].OutputPointSelect							
75	Schedule[6].Data							
76…79	Schedule[6]. TimestampOffset							
80	Schedule[7].ID							
81	Schedule[7].SequenceNumber							
82	Schedule[7].OutputPointSelect							
83	Schedule[7].Data							
84…87	Schedule[7]. TimestampOffset							
88	Schedule[8].ID							

Consumed Assembly Instance 174 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
89	Schedule[8].SequenceNumber							
90	Schedule[8].OutputPointSelect							
91	Schedule[8].Data							
92…95	Schedule[8]. TimestampOffset							
96	Schedule[9].ID							
97	Schedule[9].SequenceNumber							
98	Schedule[9].OutputPointSelect							
99	Schedule[9].Data							
100…103	Schedule[9]. TimestampOffset							
104	Schedule[10].ID							
105	Schedule[10].SequenceNumber							
106	Schedule[10].OutputPointSelect							
107	Schedule[10].Data							
108…111	Schedule[10]. TimestampOffset							
112	Schedule[11].ID							
113	Schedule[11].SequenceNumber							
114	Schedule[11].OutputPointSelect							
115	Schedule[11].Data							
116…119	Schedule[11]. TimestampOffset							
120	Schedule[12].ID							
121	Schedule[12].SequenceNumber							
122	Schedule[12].OutputPointSelect							
123	Schedule[12].Data							
124…127	Schedule[12]. TimestampOffset							
128	Schedule[13].ID							
129	Schedule[13].SequenceNumber							
130	Schedule[13].OutputPointSelect							
131	Schedule[13].Data							
132…135	Schedule[13].Data							
136	Schedule[14].ID							
137	Schedule[14].SequenceNumber							
138	Schedule[14].OutputPointSelect							
139	Schedule[14].Data							
140…143	Schedule[14]. TimestampOffset							
144	Schedule[15].ID							
145	Schedule[15].SequenceNumber							
146	Schedule[15].OutputPointSelect							

Consumed Assembly Instance 174 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
147	Schedule[15].Data							
148…151	Schedule[15]. TimestampOffset							
Where:	***Data*** – Output Data to apply to unscheduled channels (those with a value of zero configured in ScheduleMask). ***ScheduleMask*** – Mask indicating which channels are scheduled. Per bit 0 = use Data value (normal output); 1 = use scheduled output. ***TimestampOffset*** – System Time to Local Time Offset. Output should monitor for a delta between send value and module's value and run a Step Compensation Algorithm if the difference is >10us. ***Schedule Timestamp*** – "Master" time for which Schedule[xx].TimestampOffset modifies to determine the actual schedule time for each output to be applied. ***Schedule[xx].ID*** – Indicates which schedule is to be loaded with attached data. Valid schedules are 1…16. 0 = No schedule. ***Schedule[xx].SequenceNumber*** – Value must be changed for a new schedule to be recognized. Valid values 0…3. ***Schedule[xx].OutputPointSelect*** – Output point schedule is applied to. Valid values 0…7. ***Schedule[xx].Data*** – Valid value 0 = OFF; 1 = ON ***Schedule[xx].TimestampOffset*** – Offset to add with ScheduleTimestamp (output bytes 12…19) to determine absolute Time to apply Data to physical Output. Allows for range of ~+/-35 Minutes from base ScheduleTimestamp.							

Produced Assembly Instance 177 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0…3	Reserved (Must be 0)							
4	Data Ch7	Data Ch6	Data Ch5	Data Ch4	Data Ch6	Data Ch2	Data Ch1	Data Ch0
5…9	Reserved (Must be 0)							
10							Sync Timeout	SyncTo Master
11	Reserved (Must be 0)							
12…13	Late Schedule Count							
14…15	Lost Schedule Count							
16…23	LocalClockOffset (64 bit)							
24…31	OffsetTimestamp (64 bit)							
32…39	GrandMasterClockID (64 bit)							
40…47	Timestamp (64 bit)							
48	Schedule[0].State (8 bit)							
49	Schedule[0].SequenceNumber (8 bit)							
50…51	Reserved 16 bits (Must be zero)							
52	Schedule[1].State (8 bit)							
53	Schedule[1].SequenceNumber (8 bit)							
54…55	Reserved 16 bits (Must be zero)							
56	Schedule[2].State (8 bit)							
57	Schedule[2].SequenceNumber (8 bit)							
58…59	Reserved 16 bits (Must be zero)							
60	Schedule[3].State (8 bit)							
61	Schedule[3].SequenceNumber (8 bit)							

Produced Assembly Instance 177 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
62…63	Reserved 16 bits (Must be zero)							
64	Schedule[4].State (8 bit)							
65	Schedule[4].SequenceNumber (8 bit)							
66…67	Reserved 16 bits (Must be zero)							
68	Schedule[5].State (8 bit)							
69	Schedule[5].SequenceNumber (8 bit)							
70…71	Reserved 16 bits (Must be zero)							
72	Schedule[6].State (8 bit)							
73	Schedule[6].SequenceNumber (8 bit)							
74…75	Reserved 16 bits (Must be zero)							
76	Schedule[7].State (8 bit)							
77	Schedule[7].SequenceNumber (8 bit)							
78…79	Reserved 16 bits (Must be zero)							
80	Schedule[8].State (8 bit)							
81	Schedule[8].SequenceNumber (8 bit)							
82…83	Reserved 16 bits (Must be zero)							
84	Schedule[9].State (8 bit)							
85	Schedule[9].SequenceNumber (8 bit)							
86…87	Reserved 16 bits (Must be zero)							
88	Schedule[10].State (8 bit)							
89	Schedule[10].SequenceNumber (8 bit)							
90…91	Reserved 16 bits (Must be zero)							
92	Schedule[11].State (8 bit)							
93	Schedule[11].SequenceNumber (8 bit)							
94…95	Reserved 16 bits (Must be zero)							
96	Schedule[12].State (8 bit)							
97	Schedule[12].SequenceNumber (8 bit)							
98…99	Reserved 16 bits (Must be zero)							
100	Schedule[13].State (8 bit)							
101	Schedule[13].SequenceNumber (8 bit)							
102…103	Reserved 16 bits (Must be zero)							
104	Schedule[14].State (8 bit)							
105	Schedule[14].SequenceNumber (8 bit)							
106…107	Reserved 16 bits (Must be zero)							
108	Schedule[15].State (8 bit)							
109	Schedule[15].SequenceNumber (8 bit)							
110…111	Reserved 16 bits (Must be zero)							

Produced Assembly Instance 177 Data Structure

Consumed Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Where:	***Data*** – Output echo data. ***SyncToMaster*** – When set, indicates the module has synced to a Valid Time Master. ***Sync Timeout*** – We had a Valid local CIP Sync Timemaster which has since timed out. ***LateScheduleCount*** – Indicates that a schedule request arrived at the module after the schedule time. The counter rolls over to 1 every 65,535 late updates. ***LostScheduleCount*** – Indicates that a schedule sequence number has been skipped, thus a schedule request has been lost. The counter rolls over to 1 every 65535 lost updates. ***LocalClockOffset*** – CIP Sync Local Clock Offset. ***OffsetTimestamp*** – Timestamp of when last CIP Sync Time was updated. ***GrandMasterClockID*** – The ID of the CIP Sync Grand Master the module is synced to. ***Timestamp*** – 64 bit Timestamp of last scheduled output applied. ***Schedule[xx].State*** – Current state of the schedule at index xx: 0 = Inactive 1 = Active, that is, schedule is next to be applied but not within next scheduling 2 = Current, that is, schedule is next to be applied and within next scheduling period 3 = Expired, that is, schedule has been applied. 4 = Schedule discarded - request in error 5 = Late, but applied - schedule arrived after scheduled time but was applied since no more current schedule received ***Schedule[xx].SequenceNumber*** - Echo of sequence number in output data. In order to acknowledge receipt of an event the user must transition the corresponding NewDataAck bit from 0 to 1 and set the EventAck to indicate whether to acknowledge the Off-On or On-Off transition for the input. the NewDataAck bits and EventAck are in consumed assembly 139. Timestamps are zero at power-up and after a timestamp is acknowledged. The time base and epoch of the timestamps are determined by the grandmaster clock of the system. All data listed in this assembly is in Little Endian format, LSB first, in increasing byte order to MSByte last.							

Notes:

Connect to Networks via Ethernet Interface

This appendix:
- describes ArmorBlock module and Ethernet communication.
- describes Ethernet network connections and media.
- explains how to establish connections with the ArmorBlock module.
- lists Ethernet configuration parameters and procedures.
- describes configuration for subnet masks and gateways.

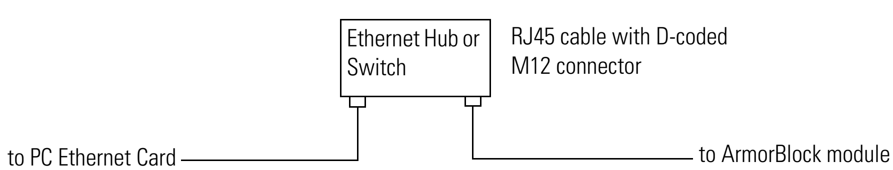
ArmorBlock Module and Ethernet Communication

Ethernet is a local area network that provides communication between various devices at 10 or 100 Mbps. The physical communication media options for the ArmorBlock modules are:
- built-in
  - twisted-pair (10/100Base-T)
- with media converters or hubs
  - fiber optic
  - broadband
  - thick-wire coaxial cable (10Base-5)
  - thin-wire coaxial cable (10Base-2)

ArmorBlock module and PC Connections to the Ethernet Network

The ArmorBlock module utilizes 10 Base-T or 100 Base-TX media. Connections are made directly from the ArmorBlock module to an Ethernet hub or switch. Since the ArmorBlock module incorporates embedded switch technology, it can also be connected to other modules in a Star, Tree, Daisy Chain or Linear, and Ring network topologies. The network setup is simple and cost effective. Typical network topology is pictured below.

Ethernet Network Topology



IMPORTANT	The ArmorBlock module contains two 10/100Base-T, M12-D (4-pin) Ethernet connectors which connect to standard Ethernet hubs or switches via RJ-45 (8-pin) twisted-pair straight-through cable. It can also connect to another ArmorBlock module via a four wire twisted pair straight-through or cross-over cable. To access other Ethernet mediums, use 10/100Base-T media converters or Ethernet hubs or switches that can be connected together via fiber, thin-wire, or thick-wire coaxial cables, or any other physical media commercially available with Ethernet hubs or switches.

Connecting to an Ethernet Network

The ArmorBlock module supports the following Ethernet settings:

- 10 Mbps half duplex or full duplex
- 100 Mbps half duplex or full duplex

Mode selection can be automatic, based on the IEEE 802.3 auto negotiation protocol. In most cases, using the auto negotiation function results in proper operation between a switch port and the ArmorBlock module.

With RSLogix 5000 programming software version 18 or later, you can manually set the communication rate and duplex mode of an Ethernet port you have connected to the switch port. The settings of the Ethernet port and the switch port must match.

Cables

Shielded and non-shielded twisted-pair 10/100Base-T cables with D-coded M12 connectors are supported. The maximum cable length (without repeaters or fiber) is 100 m (323 ft). However, in an industrial application, cable length should be kept to a minimum.

EtherNet/IP Connections

TCP/IP is the mechanism used to transport Ethernet messages. On top of TCP, the EtherNet/IP protocol is required to establish sessions and to send MSG commands. Connections can be initiated by either a client program (RSLinx application) or a processor.

The client program or processor must first establish a connection to the ArmorBlock module to enable the ArmorBlock module to receive solicited messages from a client program or processor.

In order to exchange I/O data with another device on Ethernet, that device must first originate a connection with the ArmorBlock via TCP/IP. Once an I/O connection is established via TCP/IP the I/O data is exchanged via UDP/IP.

Duplicate IP Address Detection

The ArmorBlock module firmware supports duplicate IP address detection.

When you change the IP address or connect one of the modules to an EtherNet/IP network, the module checks to make sure that the IP address assigned to this device does not match the address of any other network device. The module will periodically check for a duplicate IP address on the network. If the module determines that there is a conflict (another device on the network with a matching IP address), the Network Status Indicator becomes solid red.

To correct this conflict, the IP address of one of the modules will need to changed. If you decide to change the IP address of the ArmorBlock then, assign a unique IP address to the module then cycle power to the module.

If you decide to change the IP address of the other module, remove the device with the incorrect IP address or correct its conflict. To get the ArmorBlock out of conflict mode, cycle power to the module or disconnect its Ethernet cables and reconnect the cables. If you choose to disconnect the Ethernet cables to correct this conflict you will need to disconnect both Ethernet cables from two port Ethernet modules at the same time.

Configure Ethernet Communications on the ArmorBlock module

There are five ways to configure ArmorBlock module Ethernet communications.

- via a DHCP request at module powerup
- manually setting the configuration parameters using RSLogix 5000 software
- manually setting the configuration parameters using RSLinx software
- manually configuring the network settings using the embedded web server
- set the IP address of the module using the modules network address switches. See Connecting to an Ethernet Network on page 118.

The configuration parameters are shown in the Configuration Parameters table, and the configuration procedures follow.

Configuration Parameters

Parameter	Description	Default	Status
Hardware Address	The ArmorBlock module Ethernet hardware address.	Ethernet hardware address	read only
IP Address	The ArmorBlock module internet address (in network byte order). The internet address must be specified to connect to the TCP/IP network.	0 (undefined)	read/write

Configuration Parameters

Subnet Mask	The ArmorBlock module subnet mask (in network byte order). The Subnet Mask is used to interpret IP addresses when the internet is divided into subnets. A Subnet Mask of all zeros indicates that no subnet mask has been configured. In this case, the controller assumes a Subnet Mask of 255.255.255.0.	0 (undefined)	read/write
Gateway Address	The address of a gateway (in network byte order) that provides connection to another IP network. A Gateway Address of all zeros indicates that no gateway has been configured.	0 (undefined)	read/write
Host name	The Host Name is a unique name that identifies a device on a network. It must start with a letter, end with a letter or digit, and have as interior characters only letters, digits or hyphens. Maximum length is 64 characters. It must have an even number of characters.	NULL (undefined)	read/write
Default Domain Name	The default domain name can have the following formats: 'a.b.c', 'a.b' or 'a', where a, b, c must start with a letter, end with a letter or digit, and have as interior characters only letters, digits or hyphens. Maximum length is 48 characters.	NULL (undefined)	read/write
Primary Name Server	This is the IP address of the computer acting as the local Ethernet network Primary Domain Name System (DNS) server.	0 (undefined)	read/write
Secondary Name Server	This is the IP address of the computer acting as the local Ethernet network Secondary Domain Name System (DNS) server.	0 (undefined)	read/write
DHCP Enable	When DHCP is enabled, a DHCP server automatically assigns network related parameters to the ArmorBlock module when it logs on to a TCP/IP network. There must be a DHCP server on the network capable of allocating network addresses and configuring parameters to newly attached device. When DHCP is disabled, the ArmorBlock module uses the locally configured network related parameters (IP Address, Subnet Mask, Gateway Address, and so on).	1 (enabled)	read/write
Auto Negotiate and Port Setting	When Auto Negotiate is disabled (unchecked), the Ethernet speed/duplex is forced to either 10 Mbps/Half-duplex, 10 Mbps/Full-duplex, 100 Mbps/Half-duplex, or 100 Mbps/Full-duplex, as selected in the Port Setting field. When Auto Negotiate is enabled (checked), the ArmorBlock module will automatically negotiate the link speed and duplex with the module it is connected to.	Auto Negotiate enabled	read/write

Configure Using RSLogix 5000 Software

Refer to the online documentation provided with your programming software or see Configure the Module for Your EtherNet/IP Network on page 23 and Configure the Module Using RSLogix 5000 software on page 33.

Configure Using Web Server

The 1732E EtherNet/IP ArmorBlock Supporting Sequence of Events module includes an embedded web server which allows viewing of module information, TCP/IP configuration, and diagnostic information.

For more information on ArmorBlock module embedded web server capability, refer to Appendix E on page 121.

1732E ArmorBlock Embedded Web Server

Introduction

Rockwell Automation offers enhanced 1732E ArmorBlock for your EtherNet/IP control systems so you can monitor data remotely via web pages.

This chapter shows how you can use the 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules module's web server.

Typical Applications

The module provides access to internal and network diagnostics. This access opens up different, remote access applications to control systems. Use the ArmorBlock I/O web browser to remotely access module data. Use a web browser to monitor live module data and access diagnostic information.

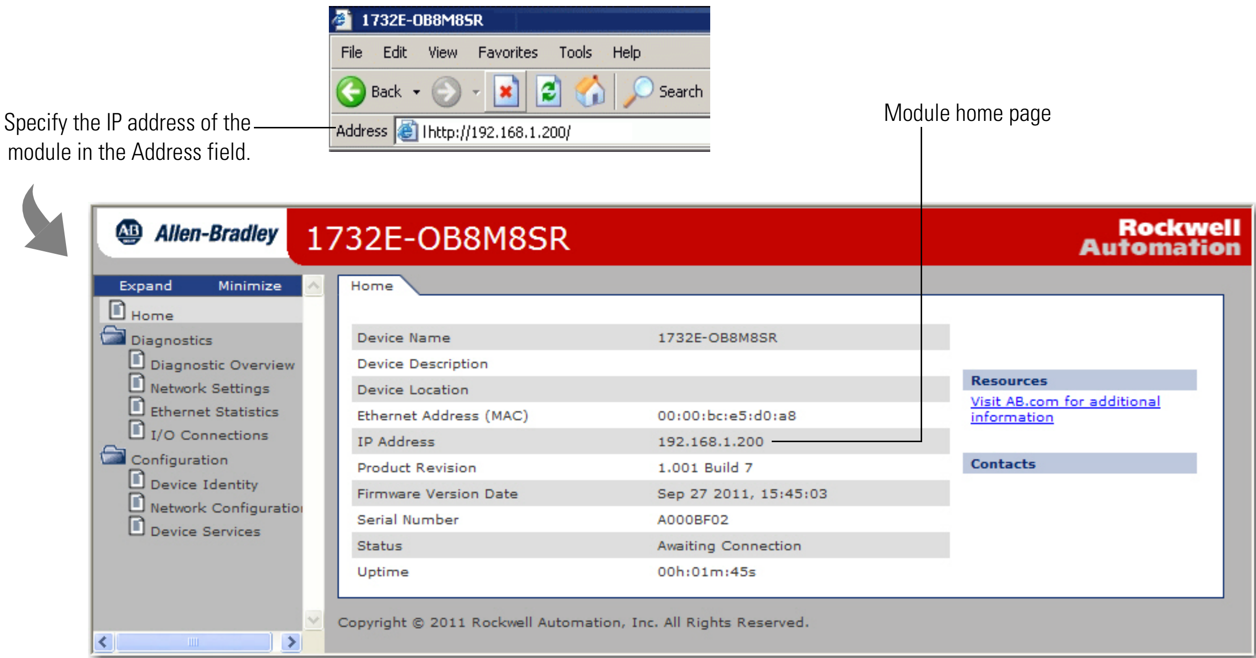
Browser Requirements

You can access the 1732E ArmorBlock I/O web pages only with Internet Explorer 6.0 or higher. To access data view pages, the browser requires Javascript support.

The supported display size is 640 x 480 or greater. Smaller display sizes work but might require extensive scrolling to view the information.

Access the Home Page of the Web Server

From your web browser, enter the IP address of the 1732E ArmorBlock module. The module displays its home page.



Log On to the Web Server

Many of the features of the 1732E ArmorBlock I/O require you to log on with appropriate access. If you select a feature, such as Configuration, the 1732E ArmorBlock I/O prompts you to enter your user name and password. The user name is Administrator. The default password is blank. Both are case sensitive.

Default Access
User Name: administrator
Password: <blank>

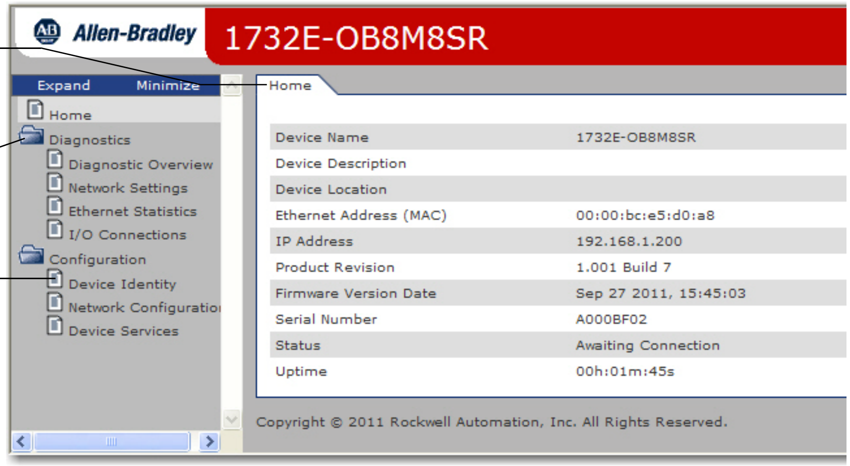
Navigate the 1732E ArmorBlock I/O

You navigate the web server pages by using the navigation panel on the left of the screen. There are also tabs across the top you can use to navigate the sections within folders

Tabs across the top match the documents within a folder, as shown in the left navigation panel.

Click folders to open and close additional levels of information.

Click a document to display a web page showing specific information.

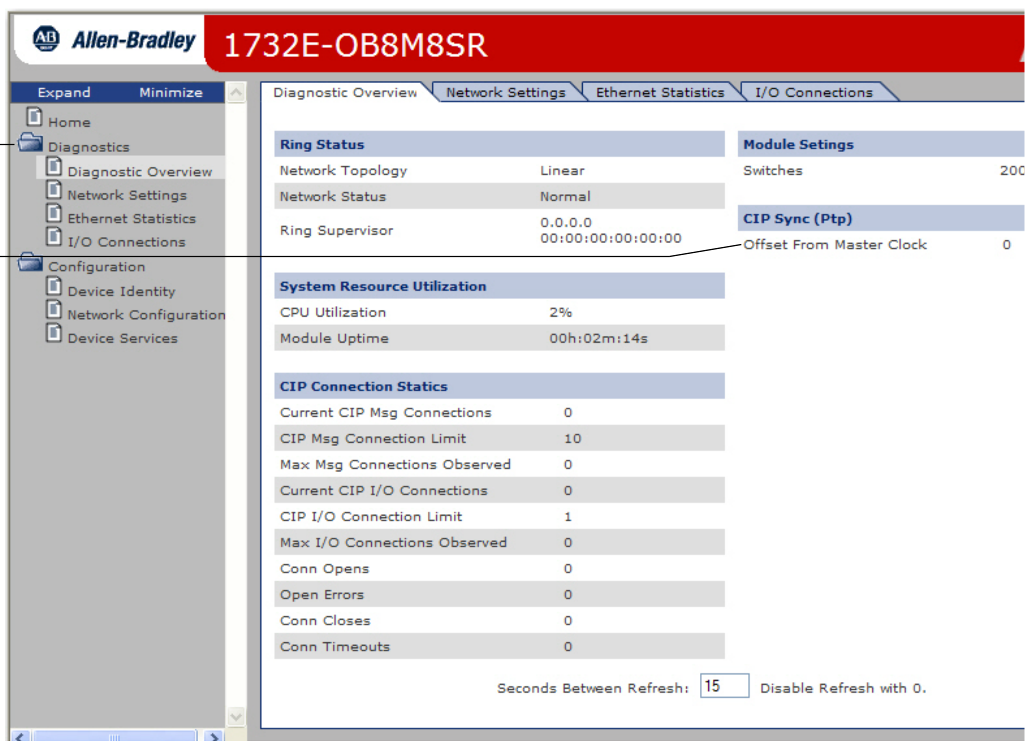


Access Diagnostic Information

You can view 1732E ArmorBlock EtherNet/IP Dual Port 8-Point Sequence of Events Input and Scheduled Output Modules specific diagnostic information, such as Offset From Master Clock by clicking Diagnostic Overview on the navigational panel on the left.

Click the Diagnostic folder to expand the navigation, then click the Diagnostic Overview page.

View the deviation between the local clock and its master clock in nanoseconds.



Notes:

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

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

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

New Product Satisfaction Return

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

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

Documentation Feedback

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

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

Rockwell Automation Publication 1732E-UM003C-EN-E - October 2015
Supersedes Publication 1732E-UM003B-EN-E - March 2014
Copyright © 2015 Rockwell Automation, Inc. All rights reserved.