Model 2600 Family Instruction Manual

August 21, 2008

Sensoray Co., Inc.
7313 SW Tech Center Dr., Tigard, Oregon 97223
voice: 503.684.8005, fax: 503.684.8164, e-mail: sales@sensoray.com
www.sensoray.com

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Chapter 1: Introduction

1.1 Overview

The Sensoray Model 2600 Real-Time Industrial I/O Network is a low-cost, flexible system of DIN-rail mountable modules that are interconnected by standard Category-5 cables. The module system employs an Ethernet-based, client-server architecture, in which a Main Module (MM) behaves as both a four-channel serial communication server and as a gateway to a cluster of up to sixteen intelligent I/O Modules (IOMs). A virtually unlimited number of MMs may be connected to a network.

The module system connects to an Ethernet client by means of a low cost, 10BaseT/100BaseTX Ethernet interface. In addition to providing a platform-independent interface, the transformer-isolated Ethernet circuitry provides excellent noise immunity by eliminating client-server ground loops.

A typical system (see Figure 1) consists of a MM and as many as sixteen IOMs. Although the diagram depicts a specific combination of IOMs, virtually any combination of IOMs may be utilized in a system.

The MM employs a star topology to ensure that IOM faults will not take down the network. Each of the MM's sixteen IOM ports distributes communications and fuse-protected power to one IOM through a standard Category-5 cable. The IOM port's communication circuitry uses optical isolation to eliminate ground loops, leading to superior reliability and performance. IOMs may be safely connected to or disconnected from the MM while power is applied to the system.

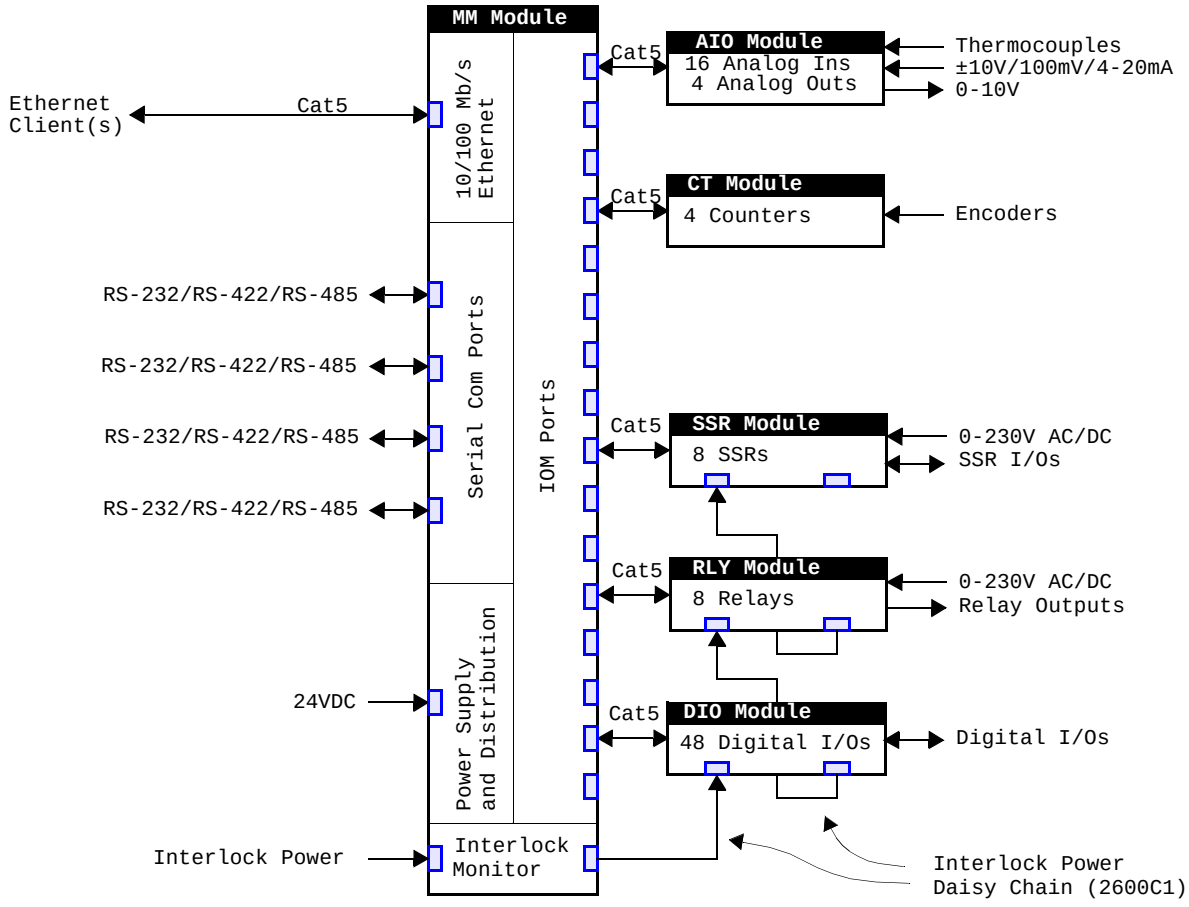
In addition to its sixteen IOM ports, the MM provides four asynchronous serial communication ports with RS-232, RS-422 and RS-485 physical layer capabilities. Each port is allocated 1KB transmit and 1KB receive buffers and operates at standard baud rates up to 115.2 Kb/s.

Multiple power supply voltages and application interlock contacts are easily accommodated by means of an interlock power daisy-chain. Every module that requires auxiliary power includes connectors for daisy-chaining the power to other modules via cable assembly 2600C1. The MM includes circuitry that enables the Ethernet client to monitor auxiliary power status.

All modules incorporate robust, two-piece mating connectors for quick, direct connection to field wiring. No external termination boards are required.

Application software development is simplified with the aid of distribution media that is supplied with module hardware at no additional cost.

Figure 1: System Block Diagram



1.2 Network Protocols

The MM implements a high-performance 16-bit UDP/IP stack in firmware. This stack includes standard network protocols that are supported by virtually all operating systems:

- IP - foundation for other supported network protocols.
- UDP - used to convey all gateway and server traffic between the MM and its Ethernet client. UDP is employed because it enables significantly higher performance than TCP and because embedded networks are more reliable than general-purpose LANs and WANs.
- ICMP - provides support for network diagnostics such as "ping" utilities.
- ARP - provides address resolution service for the Ethernet client.

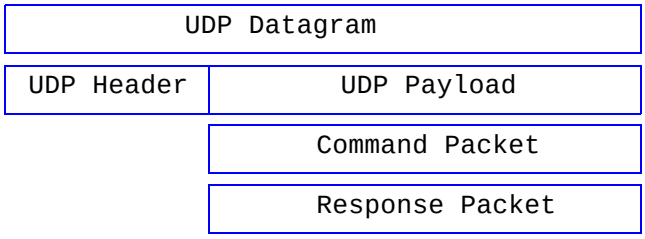
In addition, the MM firmware includes a multi-threaded server process that runs on top of UDP to enable it to provide I/O and communications services to one or more Ethernet clients.

1.2.1 UDP Encapsulation

The User Datagram Protocol (UDP) is used to transport commands and replies between the Ethernet client and the MM. A *command packet* is the payload portion of a UDP datagram that is sent from the client to the MM. A *response packet* is the payload portion of a UDP datagram that is sent from the MM to the client.

Every command/response packet (see Figure 2) is encapsulated in a single UDP payload. The UDP payload is completely populated by the encapsulated command/response packet. The MM determines the total number of bytes in a command packet by examining the UDP payload length in the command packets's UDP header. In a similar fashion, the Ethernet client may determine the size of the response packet by means of the response packet's UDP header.

Figure 2: UDP Datagram Structure



Since Ethernet is the physical layer employed by the MM, the maximum size of a framed UDP/IP packet is limited to 1500 bytes. In a particular application, however, the practical upper limit of packet size may be less than 1500 bytes due to real-time constraints dictated by the application system.

1.2.2 UDP Port Numbers

The MM listens at five different UDP ports. Each of these ports is dedicated to a specific communication resource on the MM as shown in Table 1.

Table 1: UDP Port Numbers Used by the MM

UDP Port	MM Communication Resource
10000	IOM gateway
10001	MM serial communication port 1
10002	MM serial communication port 2
10003	MM serial communication port 3
10004	MM serial communication port 4

The IOM gateway, and therefore all connected IOMs, are accessed through a single UDP port (port 10000). Each serial communication port is independently accessible by means of its own dedicated UDP port.

The Ethernet client specifies the UDP port that will be used, and hence the resource that will be accessed, when it sends a UDP datagram to the MM.

When the MM receives a UDP datagram at any of its five supported UDP ports, the MM assumes the datagram contains a command packet and processes it accordingly. For example, any UDP datagrams addressed to UDP port 10001 will be directed to the first serial communication port on the MM. The MM directs each response packet to the command originator's IP address and UDP port number.

Any UDP datagram that is directed to the MM's IP address, but has an unsupported UDP port number, will be dropped.

1.2.3 Resource Threads

Each of the MM's five communication resources is allocated a dedicated thread in the MM's local execution environment. These threads may (but are not required to) correspond to separate execution threads running on the Ethernet client.

Up to five resource threads can be made to execute concurrently, one per resource. For example, one thread might be communicating with a serial I/O device while at the same time another thread is interacting with the IOM gateway.

A resource thread is launched upon receipt of a command packet. The thread to be launched is determined by the UDP port number associated with the command packet. A resource thread is terminated upon transmission of its affiliated response packet or in response to a hardware reset on the MM.

Each resource is limited to a single thread in the MM's environment. The client may issue a command packet to a resource that is currently executing its thread, but the command packet will be queued and not acted upon until the original thread terminates. This feature makes it possible for multiple Ethernet clients to access the module network.

1.3 Module Network Topology

The MM has sixteen RJ-45 jacks—referred to as *IOM ports*—which may be connected to IOMs by means of standard Category-5 patch cables. These jacks supply power and communication signals to the connected IOM.

Every IOM has an RJ-45 jack which may be connected to any of the sixteen IOM ports on a MM. The connected MM will then act as a client of the IOM.

A high degree of flexibility is permitted with respect to the module network topology:

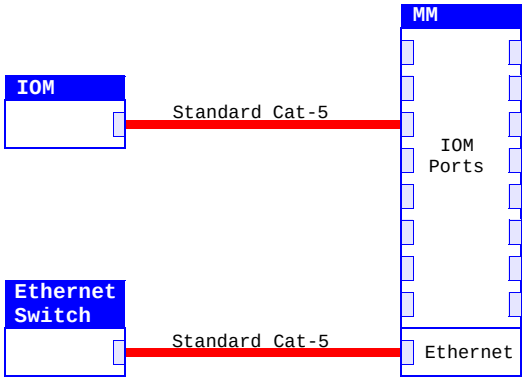
- Each IOM port may be connected to an IOM or left disconnected.
- Any IOM type may be connected to any IOM port.
- All of the IOM ports may be left disconnected, in which case the MM's gateway functions are simply not utilized.

1.3.1 Module Network

The MM serves as a client of all connected IOMs, and also as an IOM gateway for one or more Ethernet clients.

The IOM may be connected to any of the MM's IOM ports by means of a standard Category-5 patch cable as shown in Figure 3. Although not shown in Figure 3, as many as 15 additional IOMs may be connected to unused IOM ports as required.

Figure 3: Connection for a Module Network



In Figure 3, the Ethernet client is not connected directly to the 2601, but is instead connected to a switch. The switch, in turn, is connected to the MM.

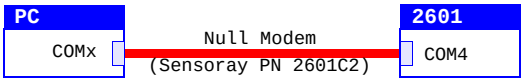
If the MM will be connected directly to an Ethernet client, a crossover cable must be used in place of the standard Category-5 patch cable. This would be the case, for example, if the MM is connected directly to an Ethernet interface card on a computer.

1.3.2 Setting the IP Address

Network configuration settings and firmware upgrades can be uploaded into the MM's flash memory when it is connected as described below.

During the firmware upgrade process, a local computer will act as a setup manager for the target MM. The computer must have an available RS-232 port, which connects to the MM's COM4 port by means of an RS-232 null modem cable as shown in Figure 4.

Figure 4: Connection for Network/Firmware Reconfiguration



A utility program, Cfg2601.exe, is included with the 2600 distribution media. When executed in conjunction with the hardware shown in Figure 4, this program will enable you to examine and modify network settings and upload new firmware onto a 2601.

The MM's default IP address is 10.10.10.1 as shipped from the factory. To change the IP address:

- Connect your hardware as shown in Figure 4.
- Run Cfg2601.exe. For example, invoke with "Cfg2601 2" if the MM is connected to COM2 on the PC.
- Cycle power on the MM or wait for it to self-reset in response to a network communication timeout.
- Select the "Change network configuration" option.
- Enter the new IP address.
- Select the "Run application firmware" option.
- Select the "Exit program" option.

Chapter 2: System Wiring

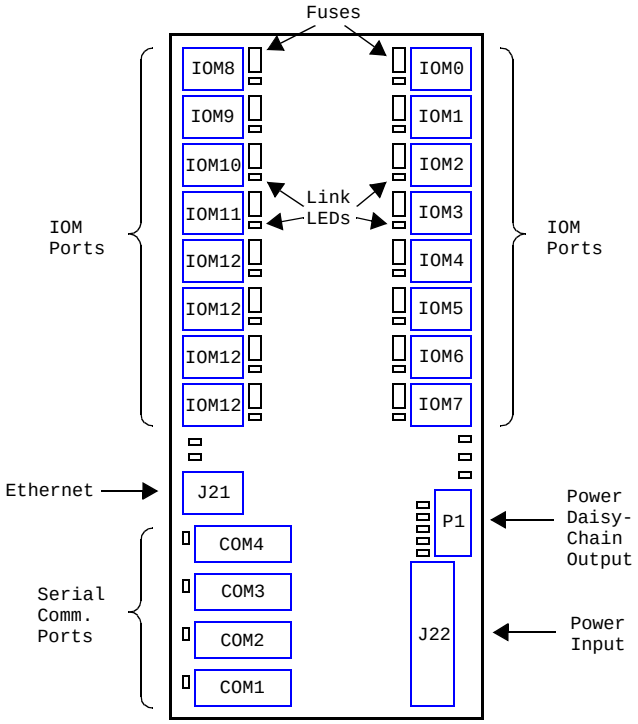
2.1 Overview

This chapter discusses the electrical connectors on the MM and how to interconnect the MM, IOMs and power supplies to form a complete system.

2.2 MM Connectors

The locations of all of the connectors on the MM are shown in Figure 5. Each of the connectors is discussed in detail in the following sections.

Figure 5: MM Connector Locations



2.2.1 Terminal Block

The MM employs a pluggable terminal block (TB) to connect one or more power supplies to the module system. The TB is installed onto Header J22.

At least one 24VDC power supply must be connected to the TB to render the module system functional. This power source supplies operating power to the MM and to all IOMs that are connected to the MM's IOM ports.

In addition to the required 24VDC supply, as many as five auxiliary DC power supplies may be connected to the TB on circuits PWR0 to PWR4 (see Table 2). These circuits may be used to supply power to any IOMs that require external power supplies for I/O operation, such as the Model 2610 digital I/O module and the Model 2650 relay module. Power supplies connected to these circuits may be routed through interlock contacts, and they may have output amplitudes less than 24V.

Table 2: Pinout of Pluggable Terminal Block at J22

Pin	Name	Function
1, 2	+24V	+24VDC power for the MM and all connected IOMs. One of these pins must be connected to the +24V supply; connections to the other +24V pin may be used to minimize external TBs.
3, 4, 5	GND	Return for the +24VDC power supply that is connected to pins 1 and 2. One of these must be connected to the 24VDC power supply return; connections to the other GND pins may be used to minimize external TBs.
6	PWR0	Optional positive DC power #0.
7	PWR1	Optional positive DC power #1.
8	PWR2	Optional positive DC power #2.
9	PWR3	Optional positive DC power #3.
10	PWR4	Optional positive DC power #4.

2.2.2 Power Daisy-Chain

Connector P1 may be used to daisy-chain the +24V and PWR0 to PWR4 power supply voltages out to any IOMs that require access to these voltages.

Table 3: Pinout of Connector P1

Pin	Name	Function
1	+24V	+24V power, always on.
2	0	Optional positive DC power #0.
3	1	Optional positive DC power #1.
4	2	Optional positive DC power #2.
5	3	Optional positive DC power #3.
6	4	Optional positive DC power #4.

Five LED indicators are located adjacent to P1. Each of these indicators lights when power is applied to its associated PWR input signal, as shown on the silkscreen legend next to each indicator.

2.2.3 Ethernet Connector

An RJ-45 jack at J21 is used for interfacing the MM to a 10BaseT/100BaseTX Ethernet network. A standard Category-5 shielded or unshielded twisted pair (STP or UTP) patch cable should be plugged into this jack, and the other end of the patch cable should be plugged into an Ethernet switch. If the MM will be connected directly to a dedicated Ethernet client, a Category-5 UTP crossover cable should be used instead.

Two LED indicators, located near to J21, light to indicate link detection and receive activity.

2.2.4 COM Port Connectors

See Section 4.2.1 for information about the asynchronous serial communication (COM) port connectors.

2.2.5 IOM Port Connectors

Sixteen RJ-45 jacks comprise the IOM ports on the MM, as shown in Figure 5. Each of these jacks has an identical pinout (see Table 4) consisting of a data transmit pair, data receive pair, and two pairs for +24V power and power return.

Table 4: IOM Port Connector Pinout

Pin	Name	Function
1	Rx+	Communication stream input, true.
2	Rx-	Communication stream input, complement.
3	Tx+	Communication stream output, true.
6	Tx-	Communication stream output, complement.
4, 5	+24V	+24V power, fused.
7, 8	GND	Return for +24V power.

These jacks connect the MM to IOMs by means of standard Category-5 patch cables.

2.2.5.1 IOM Fuses

IOM ports are individually fused so that an IOM fault or wiring error will not damage the MM or cause a hard fault on the module network. Fuses are socketed so that they may be easily replaced without removing the MM from the system. Each IOM port fuse is located adjacent to its affiliated RJ-45 connector as shown in Figure 5.

Fuses may be replaced with power applied to the MM, although it is recommended that no IOM be connected to an IOM port that is undergoing fuse replacement.

2.3 System Wiring Example

A typical module network (see Figure 6) is described in this section. This example system serves as a reference for discussion of various issues that are common to most systems.

The system consists of four different modules: model 2601 MM module, 2610 digital I/O module, 2652 solid-state relay module, and 2608 analog I/O module.

2.3.1 Power Supplies

The example system uses two power supplies. One supply provides the 24VDC that is required for all basic MM and IOM operation. A second, optional 12VDC power supply is connected to the PWR4 circuit; this supplies power to any I/O devices connected to the 2610 that require 12VDC.

Note that the 12VDC power is daisy-chained out of the 2610 to the 2652 even though it is not used by other modules. For simplicity and to allow for future expansion, it is good practice to employ standardized daisy-chain cables with all six conductors installed and aligned by pin number (e.g., daisy-chain power cable, Sensoray part number 2600C1).

2.3.1.1 Ground Connections

All power supply grounds must be connected together, as the IOM circuitry depends on a single power supply ground reference for proper operation. In addition, the power supply grounds must be connected to chassis ground to prevent excessive common-mode voltages from developing on the power supply grounds.

If possible, all power supply grounds should be connected together at a single point to minimize the potential for ground loops.

2.3.2 IOM Power and Communications

Each IOM obtains its operating power and communication path from its connection to one of the MM's IOM ports. A single, Category-5 cable between the MM and IOM supplies both power and communications to the IOM.

2.3.3 Interlocks

Two interlock circuits are included to implement fail-safe shutdown for critical circuitry:

- Two emergency stop switches are connected in a wired-or configuration so that power will be cut on PWR0 if either of these switches is opened. Because PWR0 is daisy-chained to both the 2610 and 2652 modules, any channels on these modules can be configured to use PWR0 as a power source; such channels will be automatically disabled if either ESTOP switch is opened.
- A hood interlock switch cuts power to PWR1 when it is opened. This could be used by a heater control on the 2610 or 2652 module. When the associated heating chamber is manually opened, the hood interlock switch opens and automatically cuts power to the heater control.

Note that the 2608 module does not require access to any of the auxiliary power sources. This is because all power required by the 2608 is derived from the +24V provided by the IOM port.

2.3.4 Ethernet

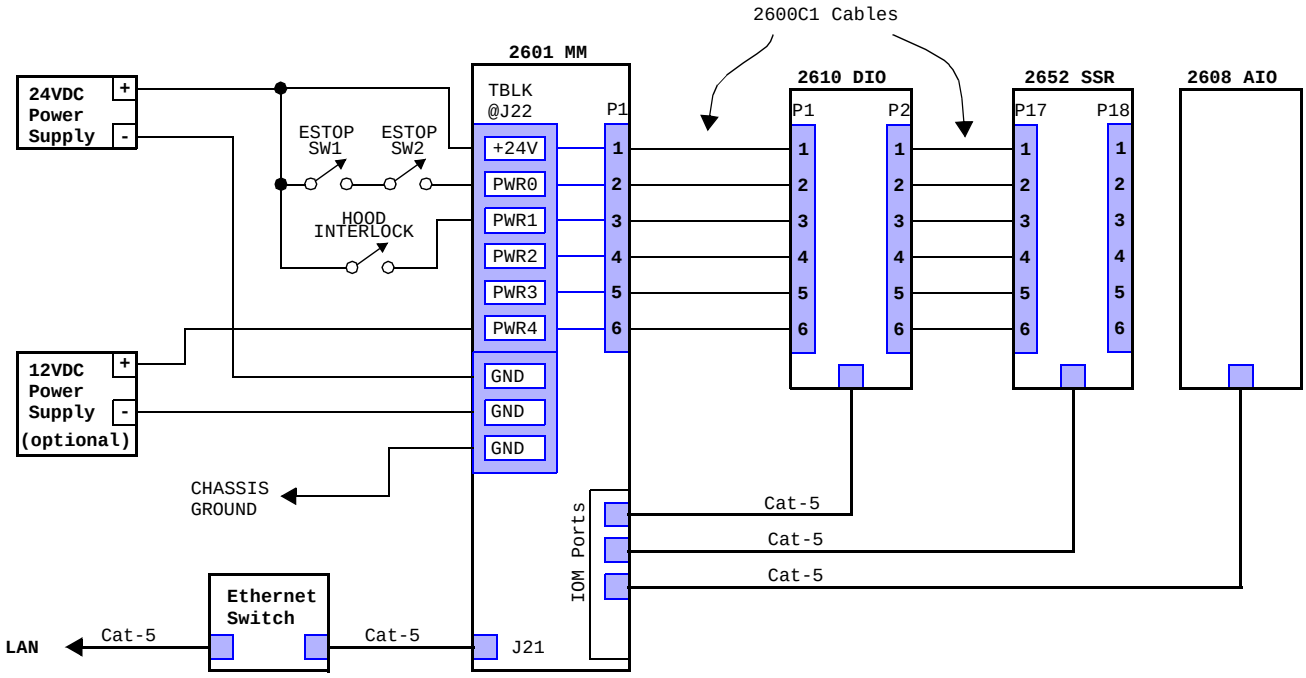
In the example system, a standard, Category-5 patch cable is used to connect the MM to an Ethernet switch, which in turn enables access to any host on the LAN.

Although the MM may be connected directly to a single client, there are some situations that require inclusion of a switch. A switch is required if:

- A portable diagnostics host (e.g., laptop computer with Ethernet) must be allowed access to the module network.
- Two or more clients must access the module network.

If the MM is directly connected to an Ethernet client, a Category-5 crossover cable must be used instead of a standard patch cable.

Figure 6: System Wiring Example



2.4 Model 2601 Specifications

Table 5: Model 2601 Specifications

Ethernet	Type	10BaseT, 100BaseTx
	Bit rate	10/100 Mbps, auto-detecting
	Isolation	1500Vrms
	CMR	-40dB typical
Serial Comm. Interface	Channels	4
	Type	RS-232, RS-422, RS-485. Software selectable per channel.
	Bit rate	All standard rates from 300 bps to 115.2 Kbps. Software selectable per channel.
	Memory	1KB transmit buffer and 1KB receive buffer, per channel.
	Termination (RS-422/485)	120 ohms, enabled via mating connector pin wiring.
	Receiver turnaround (RS-422/485)	250 microseconds
IOM Ports	Ports	16
	COM type	Optically isolated asynchronous serial stream, 115.2 Kbps
	Fuse	Socketed 500mA fuse per IOM port. Wickman 396-0500-044 or equivalent.
Interlock Power Daisy Chain Output	Voltage range	+5VDC to +24VDC, as required by daisy chain loads.
	Current	10A maximum, per circuit.
Mating Connectors (*) = included with module	Ethernet (qty 1)	RJ-45 plug, AMP 554169 or equivalent.
	Serial COM ports (qty 4)	DB-9 female, Norcomp 171-009-202-001 or equivalent.
	IOM ports (qty 16)	RJ-45 plug, AMP 554169 or equivalent.
	Terminal block (qty 1)	(*) Pluggable TB, Phoenix Contact 17.92.83.8 or equivalent.
	Power daisy chain (qty 1)	AMP 770849-6 with 640706-1 pin contacts, or equivalent. Accepts Sensoray cable assembly, part number 2600C1.
Temperature	Operating range	0 to 70°C.
Power	Input power	24VDC ±5% @130mA, with no I/O modules connected.
Dimensions	Outer dimensions	5.0H x 10.2W x 2.0D inches, including DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel; additional clearance required for field wiring.

Chapter 3: IOM Gateway

3.1 Overview

This chapter describes the behavior of the MM in its role as communication gateway to all connected IOMs.

In addition to its primary function, which is to route communication traffic between Ethernet clients and IOMs, the gateway also provides clients with access to the following utility resources:

- **Local reset log.** This enables a client to determine whether the MM has been reset.
- **Reset control.** This enables a client to initiate soft and hard resets on the MM.
- **Interlock status.** A client may use this to examine the states of external power supplies and application interlock contacts.
- **Link status.** These status indicators, which indicate the real time link status for all IOM ports, are dynamically updated when IOM port connections are changed.
- **MM attributes.** This exposes local MM attributes such as the MM firmware version number.

3.1.1 Gateway Packets

An Ethernet client communicates with the gateway, and all IOMs connected to the gateway, by means of gateway packets. There are two types of gateway packets: *command* and *response*, both of which constitute the payload portion of a UDP datagram.

Gateway command packets are transmitted from the client to the MM, which in turn forwards packet components to their appropriate target modules for execution. The MM gathers responses from the target modules, assembles the responses into a single gateway response packet, and then transmits the response packet to the originating client.

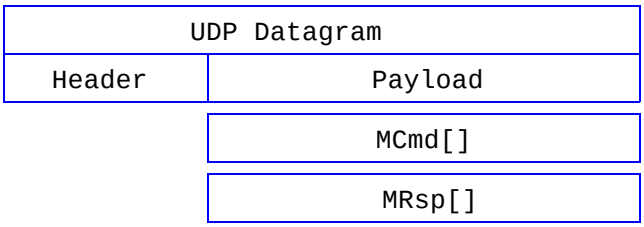
A *gateway command packet* consists of a list of one or more *module commands* (MCmd) that are packed together in the command packet without delimiters. Each MCmd is directed to a specific module for execution. The target module for a particular MCmd may be any connected IOM or the gateway itself. MCmds are executed in the order in which they appear in the command packet.

A *gateway module response* (MRsp) is produced by a target module in response to every fully executed MCmd. After issuing an MCmd to a target module, the MM always waits to receive the associated MRsp before issuing the next MCmd. As each MRsp is received, it is inserted into a *gateway response packet*. The MRsps are sequenced in the response packet in the order they were produced, thereby matching the order of execution of the associated MCmds. When all of the MCmds belonging to a command packet have executed, the response packet is sent to the client.

3.1.1.1 UDP Encapsulation

Every gateway packet is encapsulated in a single UDP payload as shown in Figure 7.

Figure 7: Gateway Packet Encapsulation

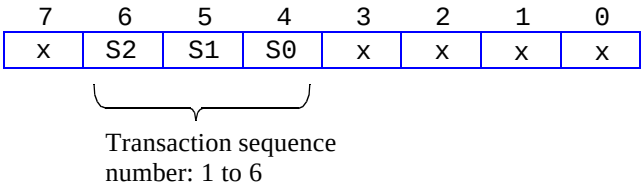


Gateway command packets must always be sent to UDP port 10000, and gateway response packets always originate from UDP port 10000.

3.1.1.2 Retry Management

The first byte of each gateway (and comport) command packet may be modified by the client so as to enable the MM's retry control mechanism.

Figure 8: Modified First Byte



Bits 4 to 6 are replaced with a transaction sequence number that has a numeric value from one to six. This sequence number is used by the MM to detect duplicate command packets resulting from transaction retries.

When the MM receives a first-attempt command packet (i.e., a command packet that is not recognized as being a duplicate of an earlier received command packet), all of the packet's commands are executed and the response packet is both sent to the client and cached locally. When the MM receives a duplicate (i.e., a "retry") of an earlier command packet, the packet's commands will not be executed nor will a response packet will be generated; instead, the previously cached response packet will be sent to the client as a reply to the duplicate command packet.

If the transaction sequence number equals zero or seven, the command packet is always regarded as a first attempt and the response packet is not cached. In this case, retries are not possible.

3.1.2 Gateway Thread

All of the gateway's IOM ports share a single, multiplexed communication interface on the MM. As a result, all communications between the MM and its connected IOMs must take place in a serial fashion, and only one IOM may communicate with the MM at a time. Since the MM multiplexes/demultiplexes communications between the client and the IOMs, the gateway employs a single thread to communicate with all IOMs.

Upon receipt of a gateway command packet, the MM will launch its gateway thread. As the gateway thread parses the packet, each MCmd is extracted and forwarded to its target module for execution. The gateway waits for an MRsp, which is appended to the gateway response packet, and then the parser moves on to the next MCmd. When all MCmds have been processed in this way, the gateway response packet is sent to the Ethernet client and the gateway thread terminates.

In preparation for communicating with an IOM, the gateway connects its IOM communication interface to the target IOM. After sending the MCmd to and receiving the MRsp from the target IOM, the gateway disconnects its multiplexed interface from the IOM port.

3.1.2.1 Overlapped Transactions

As long as at least one of the MM's fifteen packet buffers are available, the MM will queue any gateway command packets that are received while the gateway thread is running. When a gateway response packet is sent, the MM will process the oldest gateway command packet pending in the queue.

An Ethernet client may perform *overlapped* gateway transactions. This means that after it transmits a gateway command packet, a client may send another command packet before the first response packet is received. Any client that implements overlapped gateway transactions must specify a different sender's UDP port number for each transaction so that it will be able to match each response packet to its associated command packet. In general, this implies that the client is running a separate thread for each overlapped gateway transaction.

Similarly, an Ethernet client may send a command packet to the MM while the MM is processing another command packet from a different client.

3.2 Action Lists

Every module command contains an *action list* (AL). The AL contains a list of one or more operations that are to be performed.

Each action list (AL) member, which is called an *action*, specifies a single operation that is to be performed. Actions are packed end-to-end in the AL without delimiters.

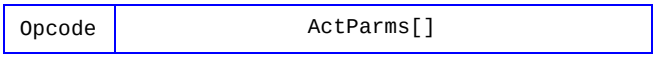
Actions are executed in the order they appear in the AL, until either all actions have been performed, or an unsupported or improperly structured action is encountered in the AL. In the latter case, the non-compliant action is not executed, further AL processing is immediately terminated, and the target module will not generate a response.

3.2.1 Action Structure

Every action has a one-byte header that contains an opcode, followed by any required command-specific parameters. Actions may vary in length depending on the number and types of any parameters that are needed to completely specify the action.

Any AL that targets a specific IOM (via the gateway) is limited to a maximum length of 252 bytes, including the AL header. If the size of a target IOM's actions exceeds this limit, the actions may be distributed among two or more ALs such that the resulting ALs do not exceed the length limit.

Figure 9: Structure of a Single Action



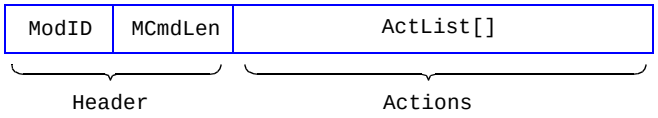
Field	Type	Function
`Opcode`	`BYTE`	Action opcode. This opcode instructs the target module as to what action is to be performed.
`ActParms[]`	`BYTE`	Action-specific parameters. This field contains zero or more bytes, depending on the number and sizes of the supplied parameters. Regardless of their actual types, these parameters are treated as a single byte array by the MM because in its capacity as a client-to-IOM gateway, the MM has no knowledge of the actions supported by IoModules.

3.3 Module Commands (MCmd)

Every MCmd begins with a two-byte header that consists of a target module identifier and a command byte count. A variable-sized field, which follows the header, contains the *action list* (AL), which is a list of actions that are to be performed by the target module.

MCmds may vary in length, depending on the number and character of the actions to be performed. The total length of an MCmd, including its header, may not exceed 254 bytes. This maximum length limit may be further constrained by the target IOM; refer to the documentation for individual IOM types for details.

Figure 10: MCmd Structure



Field	Type	Function
ModID	BYTE	Target module identifier. This specifies the module to which the MCmd is to be directed. In the case of IOMs, ModID is the MM's IOM port number to which the target module is attached, in the range 0x00 to 0x0F. To address the gateway, use the value 0xFF. Any value other than 0x00-0x0F and 0xFF is illegal.
MCmdLen	BYTE	Length of the MCmd, including the two header bytes. Since MCmds are limited to a maximum length of 254 bytes, MCmdLen may have any value in the range 0x02 to 0xFE.
ActList[]	BYTE	List of actions to be performed by the target module. When the target module is an IOM (vs. the gateway), the gateway regards ActList[] to be a byte string because the gateway has no knowledge of the number or types of parameters required by IOMs. See Section 3.2 for details.

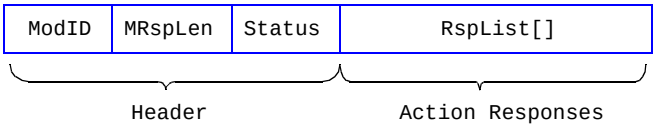
3.4 Module Responses (MRsp)

A target module will produce one MRsp in response to every received MCmd that is determined to be valid and error-free, even if none of the actions in the corresponding MCmd produce responses. If the received MCmd contains errors, such as improperly formatted or unrecognized actions, the target module will not transmit an MRsp.

Every MRsp begins with a three-byte header consisting of the module identifier, response byte count and module status. A variable-sized field, which follows the header, comprises a list of zero or more action responses that were produced by the module as a result of executing the associated MCmd.

MRsps may vary in length, depending on the number and character of the action responses that are produced by the source module. The total length of an MRsp, including its header, may not exceed 254 bytes; this limit is imposed by the MRsp format. Most IOM types limit the maximum MRsp size to less than 254 bytes due to internal buffer size limitations. Refer to the IOM model documentation to find the specification for an the IOM's maximum MRsp length.

Figure 11: MRsp Structure



Field	Type	Function
ModID	BYTE	Module identifier. ModID indicates the module that produced the MRsp. In the case of an IOM, ModID is simply the MM's IOM port number to which the module is attached, in the range 0x00 to 0x0F. The special value 0xFF indicates that the MM gateway produced the MRsp. Any value other than 0x00-0x0F or 0xFF is illegal.
MRspLen	BYTE	Length of the MRsp, including the three header bytes. Since MRsps are limited to a maximum length of 254 bytes, MRspLen may have any value in the range 0x02 to 0xFE.
Status	BYTE	Module status. See Section 3.4.1 for details.
RspList[]	BYTE	List of action responses produced by the module. This list is treated as a byte string by the gateway because the gateway has no knowledge of the number or types of response parameters produced by IOMs. The number of bytes in RspList[] depends on the actions executed; all actions have a fixed response size.

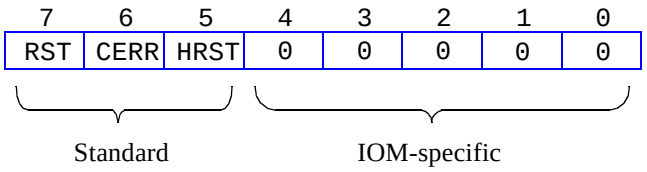
If several actions are to be executed by an IOM, it is possible that the total length of a single MRsp will exceed the maximum length supported by the IOM. In such cases it is best to break up the IOM's MCmd into multiple MCmd's and distribute the actions among the MCmd's so that none of the associated MRsp's will exceed the IOM's maximum MRsp length.

3.4.1 Status Byte

A status byte is included in the header of every MRsp. This byte contains boolean bit flags that indicate various conditions on the source module that are in effect *before* any actions execute.

With the exception of one bit, the format of the status byte is standardized for all IOM types as well as the gateway itself. The most-significant three flag bits are common to the gateway and all IOMs, while all other bytes are available for custom use by individual module types.

Figure 12: Status Byte



RST indicates whether the module has undergone a hardware or software reset. The RST flag is automatically set in response to a power-up reset or a watchdog reset or by invoking a HardReset or SoftReset action. The RST flag may be cleared by issuing a ResetFlags action.

CERR indicates that the module detected an error in a received ModCmd. This flag will be set in response to a bad checksum, improper command length, unrecognized command opcode or invalid command parameter. The CERR flag may be cleared by issuing a ResetFlags action.

HRST indicates that the module has undergone a hard reset. The HRST flag is automatically set in response to a power-up or watchdog reset or by invoking a HardReset action. The HRST may be cleared by issuing a ResetFlags action. This bit is meaningful only on IOMs; the gateway always sets this bit to zero.

The Ethernet client must never reset the HRST flag because control of this flag is reserved for gateway-to-IOM link control. Any attempts by the client to reset HRST may result in unpredictable behavior.

IOM-specific bits are reserved for custom use by the various IOM types. Each IOM type defines the meaning of these bits as appropriate for the functions performed by the module.

Refer to the specific IOM documentation for a description of the definitions of these bits. Unless otherwise stated in the IOM documentation, any of the IOM-specific status bits may be cleared by issuing a ResetFlags action.

3.5 Format Translation

IOMs employ message formats that differ slightly from those used in the Ethernet client's MCmds and MRsps. When forwarding communications between the client and target IOMs, the gateway transforms messages to accommodate these format differences. There are two key format differences:

- IOMs utilize a checksum byte that is not required by the client. The checksum byte is computed and added to messages that are forwarded to IOMs, and stripped from messages that are forwarded to the client.
- Ethernet clients utilize a target module identifier byte that is not required by target IOMs. This byte is stripped from messages that are forwarded to IOMs, and added to messages that are forwarded to the client.

3.5.1 Checksum

All checksums are expressed as a one-byte values. The checksum is computed as follows:

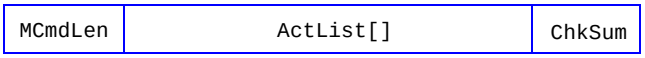
- Sum together all of the bytes that are to be included in the checksum. All of the included bytes are treated as unsigned integer values.
- Truncate the accumulated sum to its least significant byte.
- Compute the one's complement of the byte; this is the checksum value.

3.5.2 I/O Module Commands

Before forwarding an MCmd to its target module, the gateway translates the MCmd into the format required by the target module. The translated MCmd is called a *ModCmd*.

As shown in Figure 13, every ModCmd begins with a one-byte header that contains the ModCmd's byte count. A variable-sized field, which follows the header, comprises a list of actions that are to be performed by the target module. The action list is copied from the MCmd to the ModCmd without modification. A checksum byte is computed and appended to the end of the action list.

Figure 13: ModCmd Structure



Field	Type	Function
MCmdLen	BYTE	Length of the ModCmd, including the header and checksum bytes.
ActList[]	BYTE	List of actions to be performed by the target module.
ChkSum	BYTE	Checksum of all bytes preceding the ChkSum, including the header byte.

3.5.3 I/O Module Responses (ModRsp)

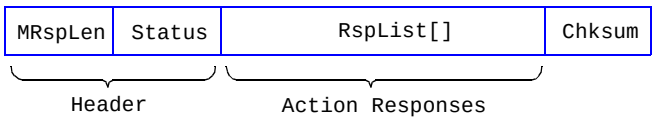
Each response message received from an IOM is called a *ModRsp*. Before forwarding a received ModRsp to the Ethernet client, the MM translates the ModRsp into a MRsp, which is the format required by the client.

Every ModRsp begins with a two-byte header. The first header byte, MRspLen, indicates the total length of the ModRsp including the header bytes.

A variable-sized field, RspList[], comprises a list of responses that were produced as result of executing the associated ModCmd's action list. The response list is copied from the ModRsp to the MRsp without modification.

The checksum is computed and validated against the received checksum at the end of the ModRsp, and the received checksum byte is stripped from the MRsp.

Figure 14: ModRsp Structure



Field	Type	Function
MRspLen	BYTE	Length of the ModRsp, including the header and checksum bytes.
Status	BYTE	Module status in effect before any actions execute. See Section 3.4.1 for details.
RspList[]	BYTE	List of responses generated by actions.
ChkSum	BYTE	Checksum of all bytes preceding the ChkSum, including the header byte.

3.6 Packet Routing

The gateway is responsible for routing MCmds to their target modules and routing the resulting MRsps to the Ethernet client. As described in the following sections, the routing process consists of multiplexing and demultiplexing the messages that pass between the client and IOMs.

3.6.1 Demultiplex Parser

Each received gateway command packet is demultiplexed into individual MCmds, which are then transformed into equivalent ModCmds and forwarded to the appropriate target modules for further processing.

Since MCmds may have variable length, the gateway must detect MCmd boundaries by parsing the gateway command packet. As the parser scans the command packet, it constructs a list of MCmd boundaries. When the end of the command packet is reached, all MCmd boundaries are known and the parser is finished. The entire command packet is parsed before any MCmds are forwarded to their target modules.

3.6.1.1 Dropped Packets

A gateway command packet will be dropped if its encapsulating UDP datagram has a UDP checksum error. In addition, the parser will drop the command packet in response to any of the following conditions:

- The parser reaches the end of the command packet, but the MCmd that is being parsed has not ended.
- The declared size of the MCmd that is being parsed exceeds the maximum valid MCmd size.

If the command packet is dropped for any reason, the gateway thread will terminate prematurely and no gateway response packet will be sent to the Ethernet client. A properly designed client will detect this by means a communication time-out while waiting for the gateway response packet to arrive.

3.6.2 Forwarding Sequence

When the command packet parser has finished parsing an error-free command packet, the gateway will begin to forward the command packet's MCmds to their target modules. The following sequence is repeated for each MCmd in the command packet:

- The MM connects its IOM communication interface to the IOM port associated with the MCmd.
- The MCmd is transformed into an equivalent ModCmd.
- The ModCmd is transmitted to the target IOM.
- The MM waits for a ModRsp, or a communication time-out, whichever occurs first.
- If a ModRsp was received, it is checked for errors by validating its size and checksum. If no errors are detected, the ModCmd is transformed into an equivalent MRsp and appended to the response packet. If a ModRsp was received but is found to have errors, no MCmd will be posted to the response packet. If a communication time-out occurred, the gateway assumes that no response is forthcoming and no MCmd will be posted to the response packet.

After executing the above sequence for all MCmds in the command packet, the gateway will transmit the response packet to the Ethernet client.

3.6.3 Missing Responses

The Ethernet client is responsible for detecting the absence of any expected MRsps in a response packet. See Section 3.8.2.2 for details.

3.7 Link Detection

3.7.1 Visual Indicators

A visual indicator LED, designated "LNK," is present on all IOMs. In addition, the MM has its own visual LNK indicators, one per IOM port (see Figure 5). When a cable connection is made from an IOM to one of the MM's IOM ports, the LNK indicators at both ends of the connection will activate to provide visual confirmation that the link is detected.

No communications are required between the Ethernet client and IOMs to light the LNK indicators. The MM automatically establishes and maintains communication links to all attached IOMs.

When a link indicator is lit, there is a high probability that the communication interface is intact and the connected IOM is at least partially functional.

3.7.2 Active Port List (APL)

The gateway maintains a list of IOM ports that have active links; this list is called the *active port list* (APL). The APL serves two purposes:

- Determines which of the MM's LNK indicators to activate.
- Enables the Ethernet client to determine the states of all IOM ports.

An IOM will activate its LNK indicator upon receipt of any valid ModCmd. Similarly, the gateway will activate the MM's LNK indicator for any IOM port that is a member of the APL.

The APL is implemented as a 16-bit integer value, with one bit representing each IOM port. When an APL bit is set, the associated IOM port is active. If an APL bit is cleared, the associated IOM port is not active.

3.7.3 Link Detection Mechanism

A *LinkQuery* is a special action that is recognized by all IOMs. When an IOM receives a LinkQuery it sends a response to let the MM know that a module is present. Unlike most other actions, receipt of a LinkQuery will not light an IOM's XMT indicator, and transmission of a LinkQuery response will not light the IOM's RCV indicator.

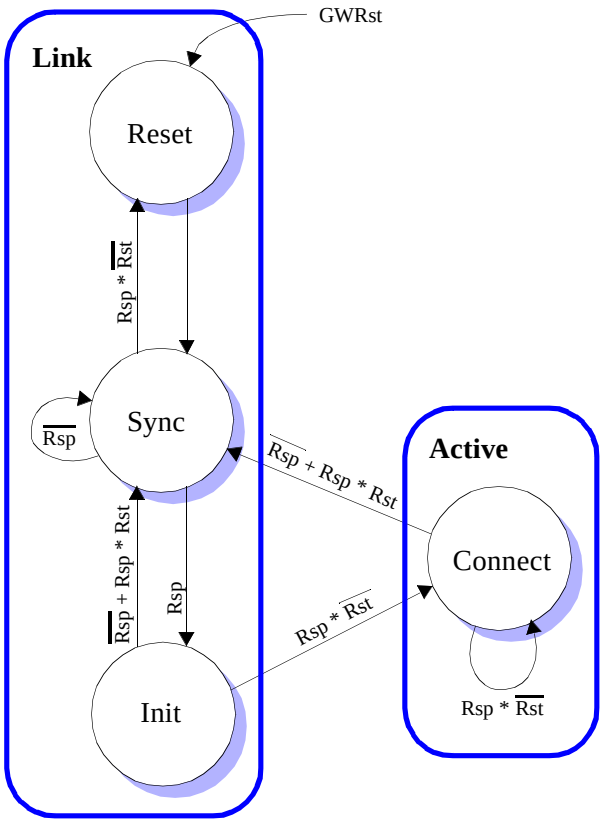
By means of LinkQuery actions, the MM periodically queries IOM ports in order to detect connection changes. These queries are performed autonomously by the MM; no client activity is required. Link queries are initiated by the MM only when no client communications are in progress.

The MM queries all IOM ports, whether or not they are members of the APL. An IOM port is automatically added to the APL if a new connection is detected on the port. Similarly, an IOM port is automatically removed from the APL if its existing gateway-to-IOM connection is broken.

3.7.4 IOM Port States

The gateway maintains a virtual state machine for each IOM port. Each IOM port state machine operates independently of the other gateway ports. A virtual IOM port state machine has four states, as shown in Figure 15.

Figure 15: IOM Port States



3.7.4.1 Signal Names

The following symbolic names are used in the state diagram to indicate state machine input signals:

- **Rst** is synonymous with the target IOMs `HRST` flag bit, which is present in all MRsps returned from the IOM. When active, this signal indicates that the IOM has executed a soft or hard reset.
- **Rsp** indicates that an MRsp was received from the target IOM.
- **GWRst** indicates that the gateway is executing a soft or hardware reset.

3.7.4.2 State Descriptions

Reset State. The gateway issues a `SoftReset` action to the target IOM and then unconditionally switches to the Sync state. The `SoftReset` action is issued even though it is not known whether an IOM is connected to the port. The Reset state is always entered in response to a gateway reset, regardless of the current state. Communications between the gateway and the connected IOM (if any) are inhibited in the Reset state.

Sync State. The gateway issues a `SetWatchdog` action to the target IOM and then waits for either an MRsp or MRsp time-out, whichever occurs first. The SetWatchdog action is issued even though it is not known whether an IOM is connected to the port. In the case of an MRsp time-out, the

Sync state will repeat. If the IOM responds, but it indicates that it has not been reset, the gateway will switch to the Reset state. If the IOM responds and indicates that it has been successfully reset, the gateway will switch to the Init state. Communications between the gateway and the connected IOM (if any) are inhibited in the Sync state.

Init State. The gateway issues a `ResetFlags` action to the target IOM to reset its `HRST` status flag. The gateway now assumes that an IOM is connected to the port because an MRsp was received in the Sync state. In the case of an MRsp time-out, or if the IOM indicates that it has been reset again, the gateway will switch back to the Sync state. If the IOM responds and indicates that it has not been reset again, the gateway will switch to the Connect state. Communications between the gateway and the connected IOM are inhibited in the Init state.

Connect State. The gateway issues a `LinkQuery` action to the target IOM. The port will remain in the Active state as long as the IOM responds and does not execute a module reset. If the IOM fails to respond to an MCmd, or if the IOM indicates it has been reset, the gateway will switch to the Sync state. Communications between the gateway and the connected IOM are allowed while the IOM port is in the Connect state.

3.7.4.3 APL Management

From the viewpoint of the Ethernet client, each of the gateway's IOM port state machines is in one of two states: *Link* or *Active*. The Link state collectively refers to the state machine's Reset, Sync and Init states, while the Active state is synonymous with the state machine's Connect state. The APL reflects the current state of each IOM port from the client's perspective.

In the APL, an IOM port is indicated active if it is in the Active state or inactive if it is in the Link (inactive) state. When a port is in the Active state, communications between the Ethernet client and connected IOM are enabled. In the Link state, all communications between client and IOM port are inhibited. The gateway will reject any client-originated MCmds that are directed to an IOM that is in the Link state.

3.8 Error Detection

Communication errors can be detected by means of two mechanisms: *integrity checking* and *content analysis*. Both of these mechanisms may be employed by an Ethernet client to detect errors in response packets. The MM detects command packet errors by means of full integrity checking and partial content analysis, and therefore the MM depends on target modules for completion of the command packet content analysis.

3.8.1 Integrity Checking

From an Ethermet client's viewpoint, integrity checking consists of ensuring that an expected, intact response packet is received from the MM in a timely manner. Integrity errors can result from a variety of conditions:

- Soft errors. Occasionally, Ethernet packets travelling between the client and MM may be corrupted by noise, either on the Ethernet physical media or at the transmitter or receiver.
- Hardware failure on the MM module, the client's Ethernet interface, or the communication path between the client and the MM.

To fully implement integrity checking, the client must employ a communication watchdog timer. The watchdog timer is started when a command packet is first sent to the MM. If the watchdog times out before a matching response packet is received, an integrity error has been detected.

The overall integrity of a command or response packet is inherently qualified by the packet's encapsulating UDP datagram header checksum. If the UDP checksum is valid, the encapsulated command or response packet is assumed to be intact. Conversely, the encapsulated packet is assumed to be corrupt if the UDP checksum is not valid.

The MM depends on its protocol stack's UDP checksum validation mechanism to detect integrity errors in command packets. In the case of a bad command packet checksum, the MM will reject the entire packet and cancel the transmission of a response packet; the client should detect the missing response packet, and hence, the integrity error, by means of a communication time-out.

In a similar fashion, the client relies on it's own UDP checksum validation mechanism to detect integrity errors in received response packets.

The possibility exists that the MM might receive and execute a valid command packet, but the encapsulating Ethernet packet could become corrupted, thereby rendering the UDP checksum invalid. In this case, the client's protocol stack will reject the packet, and the client must detect this condition by means of a communication time-out.

3.8.2 Content Analysis

Packet content errors, which are defined here as improperly formed or missing command/response packet components, can occur for a variety of reasons:

- Client programming errors.
- Soft errors that occur infrequently, resulting in the corruption of packets moving between the MM and a connected IOM.
- Hardware failure of the MM, a connected IOM, or the interface cable connecting an IOM to the MM.

3.8.2.1 MCmd Faults

The gateway is unaware of the types of modules to which its IOM ports are connected; it serves only as a gateway between the client and IOMs. As a result, the client must assume

responsibility for detecting packet content errors by checking for missing and/or incorrectly sized MRsps in response packets.

In addition, the Ethernet client must assume responsibility for the construction of valid MCmds, and the correct addressing of MCmds so as to direct them to the intended target modules.

3.8.2.2 MRsp Faults

Several conditions can lead to packet content errors that cause no MRsp to be received from a target module. In all of these cases, the gateway will detect the absence of the MRsp by means of a communication time-out:

- The client has directed an MCmd to a non-existent target module.
- The target module has developed a hardware fault.
- The connection between the MM and target module has failed.
- On the MM, the IOM port to which the target module is attached has developed a hardware fault.
- An MCmd is corrupted en-route to the target module.
- An MCmd contains at least one unsupported or improperly formatted action.

Another situation that can cause packet content errors is corruption of the MRsp. If an MRsp is corrupted en route to the gateway, the gateway will detect the faulty MRsp by means of its invalid MRsp checksum and the MRsp will be omitted from the response packet.

In all of the above mentioned cases, the gateway receives either a faulty MRsp or no MRsp at all. Regardless of the cause, the response packet will be missing an MRsp that should be present. The client must detect this condition by checking for the presence of all expected MRsps, and taking the appropriate action when a missing MRsp is encountered.

3.9 Gateway Actions

This section describes the actions that are supported by the MM gateway. As discussed in Section 3.3, the Ethernet client may direct MCmds to the gateway by specifying the special module identifier value 0xFF; this module identifier is reserved for gateway controller functions.

All supported gateway actions are listed in Table 6. In most cases, the supported actions may be executed in any order, and any arbitrary number of actions may coexist in a single MCmd so long as the MCmd does not exceed its maximum legal size of 254 bytes.

Table 6: Summary of Gateway Actions

Command	Opcode	Function
`GetLinkStatus`	`0x00`	Return link states of all IOM ports.
`GetInterlocks`	`0x01`	Return safety interlock power states.
`---`	`0x02-0xEF`	Reserved for future use.
`SoftReset`	`0xF0`	Restart module by jumping to firmware entry point.
`HardReset`	`0xF1`	Restart module by forcing a watchdog time-out.
`ResetFlags`	`0xF2`	Clear the specified bit flags in the Status byte.
`SetWatchdog`	`0xF3`	Program the watchdog interval.
`---`	`0xF4`	Reserved for future use.
`GetProductID`	`0xF5`	Return module type identifier.
`GetVersion`	`0xF6`	Return module firmware version number.
`---`	`0xF7-0xFE`	Reserved for future use.
`NOP`	`0xFF`	No operation.

3.9.1 GetLinkStatus

Function Returns the Active Port List.

Opcode `0x00`

Command `none`

Response `(LinkFlags<15:8>), (LinkFlags<7:0>)`

Parameter	Function
`LinkFlags`	Link state flags for IOM ports 0 through 15. Each bit is associated with a single port. For example, bit 4 is associated with port 4. A bit flag is set to *one* to indicate active link, or *zero* to indicate inactive link.

Notes The MM's gateway is responsible for maintaining a list of active IOM ports, called the *Active Port List* (APL). As described in Section 3.7, the gateway continuously refreshes the APL. `GetLinkStatus` returns a snapshot of the APL to the client.

The Ethernet client is permitted to communicate with IOM ports that are indicated Active in the APL. If the client attempts to communicate with

an IOM that is not Active, the gateway will reject the associated MCmd and no MRsp will be forthcoming.

3.9.2 GetInterlocks

Function Returns the signal states of all safety interlock inputs.

Opcode `0x01`

Command `none`

Response `(0,0,LockFlags<5:0>)`

Parameter	Function
`LockFlags`	State flags for all safety interlock inputs. Each bit is associated with an interlock channel number. For example, bit 4 is associated with interlock channel 4. A bit flag is set to *one* to indicate that the interlock input has applied power, or *zero* to indicate that the interlock input has no applied power.

Notes The MM includes two connectors for interlock power distribution. One connector receives power from up to six interlock contacts, while the other connector serves as a daisy-chain to distribute the interlock power to IOMs. Each interlock signal is called an *interlock channel*.

Every interlock channel occupies one circuit in each of the MM's interlock power connectors. In addition, each channel is routed to a metering circuit that enables the MM to monitor the channel's voltage level. If a channel's interlock contact is closed, the interlock will supply voltage to the input connector, which in turn will convey the voltage to the output connector and metering circuit.

3.9.3 SoftReset

Function Invokes a software reset on the MM.

Opcode `0xF0`

Command `none`

Response `none`

Notes The MM will immediately reboot upon execution of a `SoftReset` action. During the resulting reboot, the MM will not process command packets, nor will it return response packets to the client. All connected IOMs are automatically reset to their default conditions.

To ensure synchronization of the client to the MM, the Ethernet client should wait for the MM to complete its reset and reboot sequence before invoking actions on the gateway or any attached IOMs. This can be accomplished by polling the gateway, either continuously or at periodic intervals, until a response packet is received that indicates the MM has undergone a reset.

3.9.4 HardReset

Function Invokes a hardware reset on the MM.

Opcode `0xF1`

Command `none`

Response `none`

Notes `HardReset` induces a hardware reset on the MM by disabling the MM's hardware watchdog timer reset function. In response to `HardReset`, the MM will permit its watchdog circuit to time out, thereby causing a hardware reset of the entire MM module. All connected IOMs are automatically reset to their default conditions.

After executing a `HardReset`, the MM will cease to perform its gateway and server functions until the MM's watchdog timer initiates a hardware reset. During the hardware reset and the ensuing reboot, the MM will not process command packets, nor will it send response packets to the client.

This action should never be invoked by an application. It is provided strictly for diagnostic support.

3.9.5 ResetFlags

Function Resets to zero the specified bit flags in the MM's Status byte.

Opcode `0xF2`

Command `(BitMask)`

Parameter	Function
`BitMask`	Specifies the bit flags to be cleared. Set a mask bit to *one* to clear the matching Status bit. Set a mask bit to *zero* to leave the matching Status bit in its current state.

Response `none`

Notes This action clears all of the specified bit flags in the Status byte (Section 3.4.1), which is present in every ModRsp.

The specified bit flags are cleared before the ModRsp is generated. As a result, the Status byte returned in a ModRsp reflects the changes that were caused by any `ResetFlags` actions present in the associated ModCmd.

Except for the `HRST` (hard reset) bit flag, any combination of bit flags may be cleared by the Ethernet client. The client is not permitted to clear the `HRST` flag because this flag is reserved for gateway-to-IOM link control.

3.9.6 SetWatchdog

Function Programs the MM's communication time-out interval.

Opcode `0xF3`

Command `(Interval)`

Parameter	Function
`Interval`	Communication watchdog interval, specified in 100 millisecond increments. Specify *zero* to disable the communication watchdog.

Response `none`

Notes The MM employs a timer to detect the absence of communications between the Ethernet client and the gateway. If no communications are received from the client within the specified time-out interval, the communication timer will time-out and the MM will execute a hardware reset. This behavior ensures that all I/O will be turned off in the event the client shuts down abnormally.

The watchdog interval defaults to 10 seconds (`Interval` = 100) in response to a MM reset. If the default interval is suitable for the application, no `SetWatchdog` action need be issued to the MM.

Upon receipt of a `SetWatchdog` action, the new watchdog interval is effective immediately and the watchdog timer is reset so that it will time out when the new interval elapses.

3.9.7 GetProductID

Function Returns the MM's model number.

Opcode `0xF5`

Command `none`

Response `(ModelNum<15:8>),(ModelNum<7:0>)`

Parameter	Function
`ModelNum`	Always returns decimal value 2601.

3.9.8 GetVersion

Function Returns the MM's firmware version number.

Opcode `0xF6`

Command `none`

Response `(MajorVer),(MinorVer)`

Parameter	Function
`MajorVer`	Major version number. Ranges from 00 to 255 decimal.
`MinorVer`	Minor version number. Ranges from 00 to 255 decimal.

Notes On firmware labels, the major and minor version numbers are printed with a decimal point delimiter. For example, printed version "1.02" is indicated by `MajorVer` = 1, and `MinorVer` = 2.

3.10 Specifications

See Section 2.4 for gateway specifications.

Chapter 4: Serial Communication Server

4.1 Overview

In addition to its gateway role, the MM also behaves as a communication server by providing serial communication services to an Ethernet client. This chapter describes the behavior of the MM in its role as a serial communication server.

The MM includes four serial communication channels (com ports). These com ports are referenced by the symbolic channel names COM1, COM2, COM3 and COM4.

Under software control, each com port may be independently configured for either RS-232, RS-422 or RS-485 physical layer compatibility. Integral termination resistors are supplied for RS-422 and RS-485 line termination.

Each com port is allocated a 1KB transmit buffer and a 1KB receive buffer.

Standard baud rates up to 115.2 kbps are supported, as well as programmable options such as parity type, character size, and number of stop bits.

If desired, an Ethernet client may communicate with each com port via a private communication thread. Each such thread employs a unique UDP port number, thus enabling one or more Ethernet clients to carry on concurrent conversations with multiple com ports.

Each com port may be independently assigned to any arbitrary Ethernet client, although most systems are configured such that all four com ports serve a single Ethernet client. For example, COM1 could be assigned to one client and COM2 could be assigned to a different client.

4.1.1 Flow Control

Software flow control (XON/XOFF) is programmable for each com port. Hardware flow control is not supported.

When software flow control is enabled for a com port, it applies to both transmit and receive operations.

The transmitter will immediately cease transmissions (after completing the current character transmission) upon receipt of an XOFF character, and restart transmissions upon receipt of an XON character.

An XOFF will be sent by the com port transmitter when the receiver's buffer fills to within 32 bytes of full. If an XOFF has previously been sent (with no matching XON), an XON will be sent when the receiver buffer character count falls below 32.

4.1.2 Reset

As described in Chapter 3, the MM can be reset automatically by a gateway communication time-out (see Section 3.9.6), or explicitly by a client-invoked gateway reset action (Sections 3.9.3 and 3.9.4).

All com ports are reset to their default conditions when the MM is reset. As a result, all independent communication threads that are running on the Ethernet client should be notified in the event of a MM reset so that they may take the appropriate action.

4.2 Hardware Configuration

The physical layer employed by each COM port is configured under software control, so there is no need to install configuration programming shunts. It is only necessary to ensure that the proper connector pins are used for the desired electrical interface.

4.2.1 COM Port Connectors

Four male DB-9 connectors, designated "COM1" to "COM4," are provided for connecting external devices to the COM ports. Each of these connectors is assigned identical pinouts as shown in Table 7.

Table 7: COM Port Connector Pinouts

Pin	Name	Type	Signal Function
1	Rx422+	In	RS-422 receive, active high
2	Rx232	In	RS-232 receive
3	Tx232	Out	RS-232 transmit
4	Tx422+	Out	RS-422 transmit, active high
5	GND	Ref	Signal reference.
6	Rx422-	In	RS-422 receive, active low
7	RxTerm-	---	Termination resistor, receive comp.
8	TxTerm-	---	Termination resistor, transmit comp.
9	Tx422-	Out	RS-422 transmit, active low

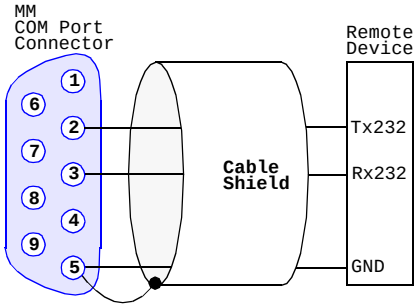
The following sub-sections detail the communication cable wiring for RS-232, RS-422 and RS-485 applications.

Shielded cable is recommended for communication cables, although unshielded cables may provide satisfactory performance. If shielded cable is used, the shield conductor should be connected to ground at one end of the cable and left disconnected at the other end of the cable. This will provide shielding without introducing ground loops. All of the following sections assume the use of shielded communication cables.

4.2.1.1 RS-232 Wiring

All RS-232 applications should be wired as shown in Figure 16.

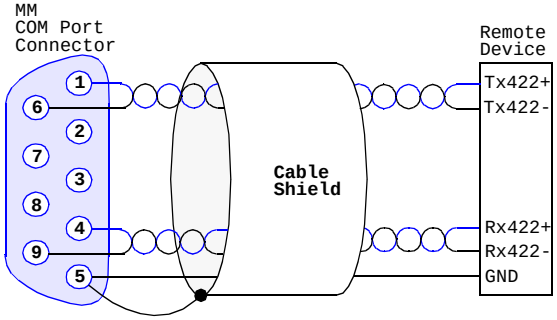
Figure 16: RS-232 Wiring



4.2.1.2 RS-422 Wiring

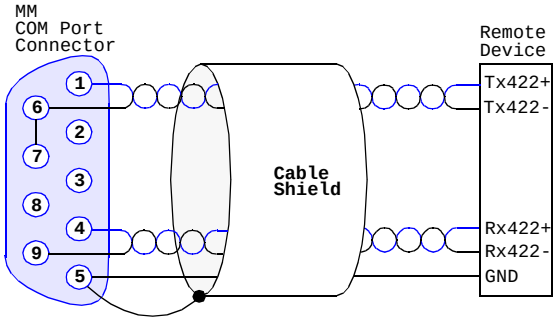
It is standard practice to use twisted pairs for RS-422 transmit and receive signal pairs, as shown in Figure 17. This will help to reduce sensitivity to the common-mode noise that inevitably results from exposure to dynamic fields.

Figure 17: RS-422, no termination



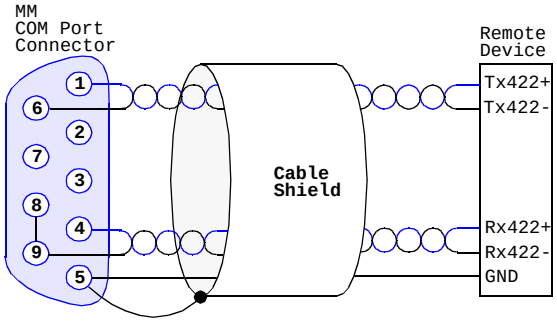
As shown in Figure 18, pins 6 and 7 may be connected together to insert a 120 ohm termination resistor across the receive pair. This termination resistor should typically be inserted only if the MM com port receiver is the last node at the end of the RS-422 network.

Figure 18: RS-422 with receiver termination



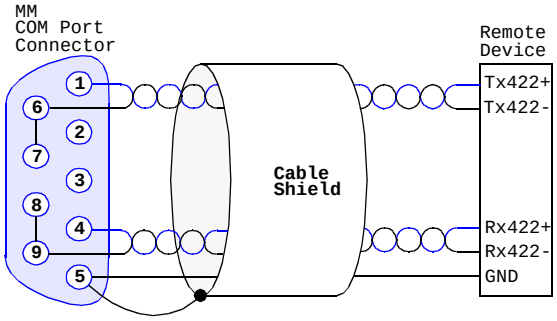
Pins 8 and 9 may be connected together (see Figure 18) to insert a 120 ohm termination resistor across the transmit pair. This termination resistor should typically be inserted only if the MM com port transmitter is the last node at the end of the RS-422 network.

Figure 19: RS-422 with transmitter termination



If termination is required on both the transmit and receive pairs, connect together pins 6 to 7 and 8 to 9 as shown in Figure 20.

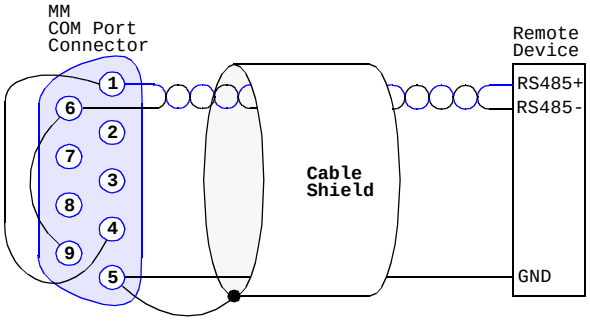
Figure 20: RS-422, transmit and receive termination



4.2.1.3 RS-485 Wiring

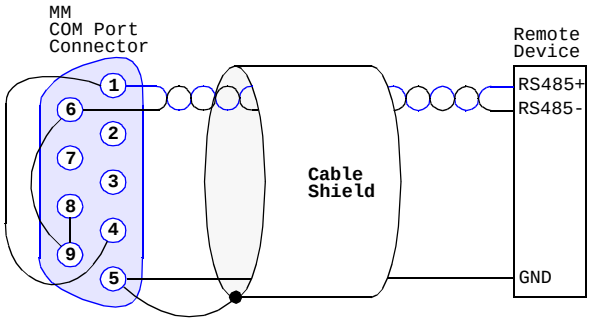
RS-485 operation is supported on Revision D (and higher) 2601 modules. It is standard practice to use a twisted pair for the bidirectional RS-485 data signal pair, as shown in Figure 21.

Figure 21: RS-485, no termination



Pins 8 and 9 may be connected together to insert a 120 ohm termination resistor across the data pair (see Figure 22). This termination resistor should typically be inserted only if the MM is the last node at the end of the RS-485 network.

Figure 22: RS-485 with termination



4.2.2 Device Networks

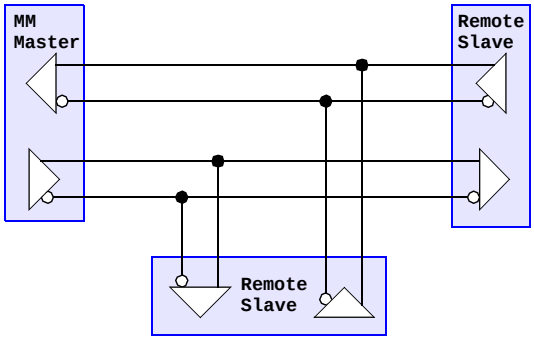
When operating as an RS-422 or RS-485 interface, a single MM com port can be used to connect to a network of remote devices. A com port's physical layer type must be configured to match the requirements of the connected device network. The physical layer type is configured by means of the PHY value, as described in Section 4.4.1.2.

Some common device network topologies and examples are discussed in the following paragraphs. All of these examples assume the MM is acting as a master on the device network.

4.2.2.1 RS-422 with a Single Master

In this case, the MM is the sole master of the device network and one or more remote slaves reside on the network. No other devices are permitted to transmit on the MM's transmit pair. A network having this topology may be configured as either PHY type 01 or 11. Figure 23 shows a network of this type with two slave devices.

Figure 23: Single-master RS-422 device network

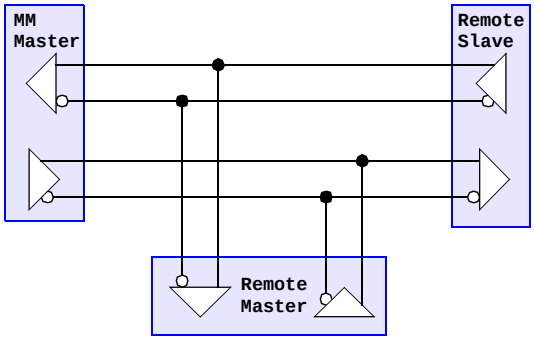


Multiple slave devices must never simultaneously drive the MM's receive pair. This is typically ensured by assigning unique addresses to the slaves and programming them so that they will drive the MM's receive pair only when responding to directed command messages from the MM.

4.2.2.2 RS-422 with Multiple Masters

In a multi-mastered device network, the MM must share its transmit pair with one or more remote masters, and one or more slave devices may reside on the device network along with the MM and other master devices. A device network having this topology should be configured as PHY type 11. Figure 24 shows an example of this type of network, in which the device network is shared by two masters (the MM and one remote master) and one slave.

Figure 24: Multi-mastered RS-422 device network

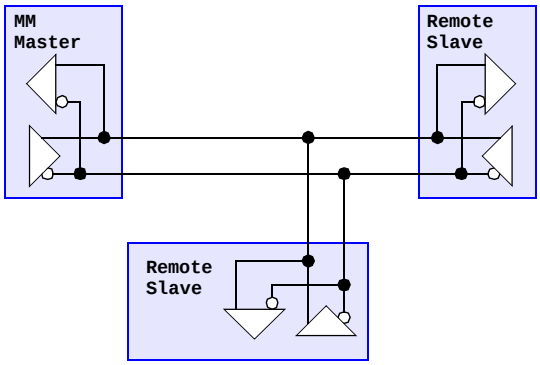


Care must be taken to ensure that the MM and the remote masters never drive the shared MM transmit pair at the same time, as transmitter collision detection is not available in this configuration.

4.2.2.3 RS-485

RS-485 networks employ a single pair for both transmit and receive. Only one transmitter is permitted to drive the common pair at a time. Figure 25 shows an RS-485 network with one master (the MM) and two slaves, although it is possible to utilize more than one master on a device network.

Figure 25: Single-master RS-485 device network



RS-485 networks may be configured as PHY type 10 or 11. Type 10 should be used for half-duplex operation in which no collisions are possible, and type 11 should be used when collisions are possible and must be detected (e.g., multiple masters) by means of full-duplex feedback.

The common pair must be driven by only one transmitter at a time. In single-master device networks, this is enforced by assigning unique addresses to the slaves and programming them so that they will drive the common pair only when responding to directed command messages from the MM.

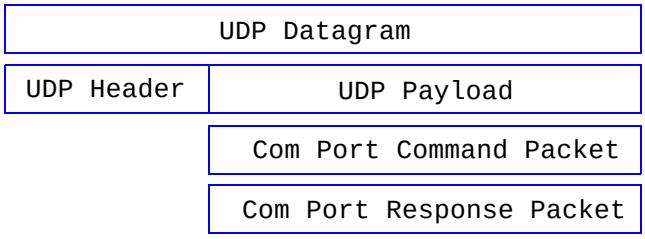
4.3 Com Port Packets

The Ethernet client communicates with the MM's serial communication ports via com port packets. There are two types of com port packets: *command* and *response*.

Com port command packets are transmitted from the client to the MM, which in turn performs the requested operation. The MM constructs a com port response packet as appropriate for the operation performed and then it transmits the response packet to the client.

Every com port packet is encapsulated in a single UDP payload as shown in Figure 26. A com port packet completely occupies the payload portion of its encapsulating UDP datagram.

Figure 26: Com Port Packet Encapsulation



4.3.1 Maximum Packet Size

Because com port packets occupy the payload region of Ethernet-based UDP datagrams, each com port packet is limited to a theoretical maximum size of 1472 bytes. As a practical matter, however, com port packets are limited to a maximum of 1024 bytes, which is the maximum amount of available transmit buffer space when sending data, or the largest possible received character count associated with the target com port.

4.3.2 Command Packets

Each com port command packet is directed to a specific com port for execution. The com port to which a command packet is directed is determined by the destination UDP port number specified in the encapsulating UDP datagram. For example, any UDP datagram addressed to UDP port 10001 will be directed to COM1. See Section 1.2.2 for a discussion of UDP port assignments on the MM.

Exactly one operation is invoked per com port command packet. The operation that is to be performed is denoted by a one-byte opcode at the beginning of the packet. The opcode is followed by zero or more command-specific data bytes.

Figure 27: Com Port Command Packet Structure

Field	Function
`Opcode`	Command opcode. This opcode indicates the function to be performed. The opcode byte may be modified as described in Section 3.1.1.2.
`Data[]`	Command-specific data. This field contains zero or more bytes, depending on the function to be performed.

The size and meaning of the values contained in the `Data[]` field depends on the function that is to be performed, as dictated by the `Opcode` byte. See Section 4.4 for descriptions of supported commands.

The opcode byte may be modified so as to enable the MM's retry mechanism. See Section 3.1.1.2 for details.

4.3.3 Response Packets

Each com port response packet is sent to the client's originating IP address and UDP port number. As shown in Figure 28, a com port response is composed of a status byte followed by one or more command-specific data bytes.

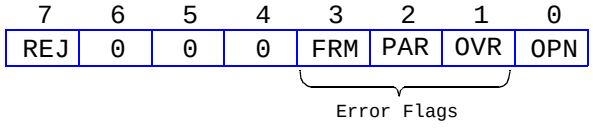
Figure 28: Com Port Response Packet Structure

Field	Function
`Status`	Status flags associated with the associated com port. See Section 4.3.3.1 for details.
`Data[]`	Command-specific response data. This field contains zero or more bytes, depending on the function that was performed.

4.3.3.1 Status Byte

The first byte of every com port response packet contains a set of active-high bit flags (see Figure 29) that indicate the state of the associated com port. Unless stated otherwise, the returned status byte indicates the state that is in effect immediately *after* the command executed.

Figure 29: Com Port Status Byte



Status bit flags are defined as follows:

- **`REJ`** indicates that the most recent com port command was rejected. This flag will be set if execution of the command would have caused an error or if the opcode is illegal.

This flag is automatically reset at the beginning of each command.

- **OPN** indicates that the com port is open (i.e., transmit and receive operations are enabled). This flag is set by the Open action and reset by the Close action. All com ports default to the closed state upon module reset.
- **FRM** indicates that a character framing error was detected on a received character. The faulty character is discarded.
- **PAR** indicates that a parity error was detected on a received character. The faulty character is discarded.
- **OVR** indicates that the receive buffer overflowed. This will happen if either (1) the UART receiver overflowed because incoming characters occurred at a rate that is faster than the MM can process them, or (2) the receive buffer is full and another receive character arrived at the UART.

In most applications, assertion of any of the error flags will be cause for discarding all data received from the remote device.

Error flags may be reset by issuing a `ClearFlags` command to the target com port. See Section 4.4.4 for details.

4.4 Com Port Commands

This section describes the commands that are supported by the MM communication server.

Table 8: Summary of Com Port Commands

Command	Opcode	Function
`SetMode`	`0x00`	Program the port's operating mode.
`Send`	`0x01`	Transmit a character string.
`Receive`	`0x02`	Receive a character string.
`ClearFlags`	`0x03`	Reset error flags.
`Open`	`0x04`	Enable transmit and receive.
`Close`	`0x05`	Disable transmit and receive.
`Flush`	`0x06`	Dump receiver buffer.
`GetRxCount`	`0x07`	Return receive buffer character count.
`GetTxCount`	`0x08`	Return transmit buffer character count.
`StartBreak`	`0x09`	Enable break transmission.
`EndBreak`	`0x0A`	Disable break transmission.
`BreakChar`	`0x0B`	Set break character.

4.4.1 SetMode

Function Programs the operating mode.

Opcode `0x00`

Command `(ClockDiv[0]),(ClockDiv[1]), (Attrib),(LEDs)`

Parameter	Function
`ClockDiv`	Clock divisor for the baud rate generator.
`Attrib`	Collection of bit fields that set various operating attributes.
`LEDs`	Specifies the behavior of the com port's status LED.

In the following subsections, bold type indicates the default settings that are in effect after a module reset.

4.4.1.1 Baud Rate

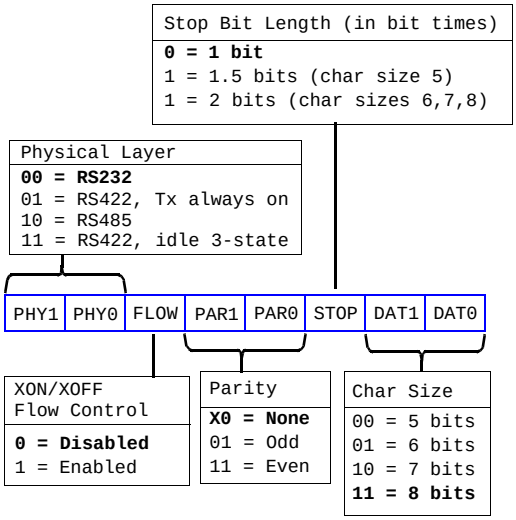
`ClockDiv` specifies the baud rate for both transmitter and receiver. It may be set to any of the following values in order to program standard baud rates. Setting `ClockDiv` to any value not shown in the table will result in a non-standard baud rate.

Baud Rate (bits/second)	`ClockDiv[0]` (hex)	`ClockDiv[1]` (hex)
300	00	03
600	80	01
1200	C0	00
2400	60	00
4800	30	00
9600	**18**	**00**
19.2K	0C	00
38.4K	06	00
57.6K	04	00
115.2K	02	00

4.4.1.2 Operational Attributes

The `Attrib` byte (Figure 30) specifies various operational attributes for the communication interface.

Figure 30: Attributes Byte



PHY

These two bits specify the physical layer type and behavior as follows:

PHY	Duplex	Type	Tx Enabled
00	**Full**	**RS-232**	**Always**
01	Full	RS-422	Always
10_1	Half	RS-422 / RS-485	When Sending
11_1	Full		

Notes

1. PHY values 10 and 11 are available only on model 2601 Revision D (or later) circuit boards.

PHY=00: Full-duplex RS-232.

PHY=01: Full-duplex RS-422. Both receiver and transmit line driver are enabled at all times.

PHY=10: Half-duplex RS-422 or RS-485. The transmit line driver is enabled only while characters are being sent; it is automatically disabled (tri-stated) when the transmitter is idle. The receiver is enabled only when the transmitter is idle, so that receive characters are blocked while the transmitter is active. The receiver turn-around time is 250 microseconds, maximum.

PHY=11: Full-duplex RS-422 or RS-485. The transmit line driver is enabled only while characters are being sent; it is automatically disabled (tri-stated) when the transmitter is idle. The receiver is enabled at all times. This mode can be useful for multi-master RS-485 device networks that require collision detection. The receiver turn-around time is 250 microseconds, maximum.

Refer to Section 4.2.2 for information about how to configure PHY for various device network topologies.

FLOW

This bit enables XON/XOFF flow control. See Section 4.1.1 for information about flow control.

PAR

These two bits specify whether even, odd, or no parity will be applied to transmitted and received characters. When even or odd parity is enabled, all incoming characters are parity checked and any parity errors will be detected and reported.

STOP

This bit specifies the number of stop bits to send when transmitting a character.

DAT

These two bits specify the number of data bits to be used in both transmitted and received characters.

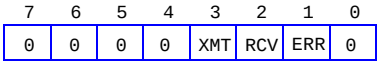
4.4.1.3 LED Behavior

The LEDs byte is shown in Figure 31. When set to logic one, each bit will cause the com port's status LED to light for approximately 100 milliseconds in response to the associated event.

Any combination of these bits may be specified. For example, the XMT and RCV bits may both be set, in which case the LED will light when a character is sent or received at the com port.

After a module reset, the RCV flag is set and all other flags are reset to zero.

Figure 31: LEDs Byte



XMT causes the LED to light when a character is transmitted.

RCV causes the LED to light when a character is received.

ERR causes the LED to light when a receiver break condition is detected or when an error (framing, overrun or parity) is detected.

Response (Status)

Parameter	Function
Status	Com port status. The REJ bit flag will be set if the com port is open.

Notes The target com port must be closed when this command is issued. If the com port is open, the command will be rejected and the returned status byte's REJ flag will be set.

4.4.2 Send

Function Transmits a character string.

Opcode `0x01`

Command `Data[]`

Parameter	Function
`Data[]`	Array of byte values to be transmitted to the remote serial device.

Response `(Status)`

Parameter	Function
`Status`	Com port status. The `REJ` bit flag will be set if the transmit buffer has insufficient free space to accommodate the specified byte string.

Notes All bytes in `Data[]` are buffered on the MM for transmission to the remote serial device. The transmit buffer is a FIFO, so any bytes that are already pending in the transmit buffer will be sent first, before the new data bytes.

The MM determines the number of data bytes in the com port command by means of the UDP datagram size. The MM assumes that all bytes following the first com port command byte are data bytes that are to be transmitted to the remote serial device.

Note that the MM does not employ any delimiter characters, such as carriage returns, to denote logical partitions in the serial data stream.

4.4.3 Receive

Function Returns a received character string.

Opcode `0x02`

Command `(MaxSize[0]),(MaxSize[1])`

Parameter	Function
`MaxSize`	Maximum number of bytes to return. The MM will return the smaller of `MaxSize` bytes or all unread bytes that have been received from the remote serial device.

Response `(Status),Data[]`

Parameter	Function
`Status`	Com port status.
`Data[]`	Array of byte values received from the remote serial device.

Notes The client can determine the number of bytes received from the MM by examining the size of the UDP datagram that comprises the com port response. The size of `Data[]` equals the size of the datagram minus one. The size of `Data[]` is guaranteed to never exceed `MaxSize`.

All of the bytes returned in `Data[]` are removed from the com port receive buffer.

4.4.4 ClearFlags

Function Resets to zero the `BRK` bit and all error flags in the status byte.

Opcode `0x03`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status condition that is effective after this action executes. The `FRM`, `PAR`, `OVR` and `WRP` bits are all reset to zero.

4.4.5 Open

Function Enables the serial transmitter and receiver.

Opcode `0x04`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is already open.

Notes Before issuing an `Open` command, the `SetMode` command should be used to configure the target com port for proper operation.

Upon module reset, all com ports default to the closed (disabled) state. Each com port must be enabled by `Open` before it will communicate with a remote serial device.

4.4.6 Close

Function Disables the serial transmitter and receiver.

Opcode `0x05`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is already closed.

Notes All of the status flags are reset to zero.

`Close` flushes any buffered characters in the transmit and receive queues, effectively emptying these buffers.

Any character transmission that is in progress will be completed before the transmit buffer is flushed.

All com ports are initialized to the closed state upon module reset. In the closed state, both the transmit and receive buffers are initialized to the empty state, and all com port serial transmitters and receivers are disabled.

4.4.7 Flush

Function Dumps the receive character buffer and resets all error flags.

Opcode `0x06`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status.

Notes This command discards all characters in the receive character buffer and resets to zero all of the receiver's error status flags.

4.4.8 GetRxCount

Function Returns the receive buffer character count.

Opcode `0x07`

Command `none`

Response `(Status),(Count[0]),(Count[1])`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is closed.
`Count`	Number of bytes pending in the receive buffer.

Notes `GetRxCount` returns the total number of characters pending in the receive buffer.

4.4.9 GetTxCount

Function Returns the transmit buffer character count.

Opcode `0x08`

Command `none`

Response `(Status),(Count[0]),(Count[1])`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is closed.
`Count`	Number of bytes pending in the transmit buffer.

Notes `GetTxCount` returns the total number of characters pending in the transmit buffer. This command can be used to determine whether the transmit buffer has sufficient free space for a new character string before executing a `Send` command.

4.4.10 StartBreak

Function Causes the transmitter to send a continuous break.

Opcode `0x09`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is closed.

4.4.11 EndBreak

Function Terminates break transmission.

Opcode `0x0A`

Command `none`

Response `(Status)`

Parameter	Function
`Status`	Com port status. The `REJ` flag will be set if the port is closed.

4.4.12 BreakChar

Function Specifies the Break character.

Opcode `0x0B`

Command `(BreakChar)`

Parameter	Function
`BreakChar`	Character that is to be stored in the receive buffer upon detection of an incoming break condition.

Response `(Status)`

Parameter	Function
`Status`	Com port status.

Notes The default break character is a NULL (0x00). The break character may be changed at any time, regardless of whether the target comport is open.

4.5 Specifications

See Section 2.4 for com port specifications.

Chapter 5: Common IOM Actions

5.1 Overview

This chapter discusses the actions that are uniformly supported by all IOMs.

Table 9 lists a summary of the common actions. Action opcodes in the range `0x00-0xEF` are unique to each module type, while opcodes in the range `0xF0-0xFF` are common to all modules. The "LEDs" column lists which of the indicator LEDs light in response to each action. This column uses the following abbreviations for the indicator LEDs:

- L = LNK. Indicates that the IOM and its client have established a connection.
- R = RCV. Indicates that a valid action has been received and executed by the target IOM.
- X = XMT. Indicates that an executed action produced a response, and the response has been send to the IOM's client.
- D = Depends on the connected module type. In most cases, module-specific actions will light the `LNK` and `RCV` indicators, and any action that produces a response will light `XMT` as well.

Table 9: Summary of Common IOM Actions

Command	Opcode	LEDs	Function
`---`	`0x00-0xEF`	D	Module dependent.
`SoftReset`	`0xF0`	-	Restart module by jumping to firmware entry point.
`HardReset`	`0xF1`	-	Restart module by forcing a watchdog time-out.
`ResetFlags`	`0xF2`	L R	Clear the specified bit flags in the Status byte.
`SetWatchdog`	`0xF3`	L R	Program the watchdog interval.
`---`	`0xF4`	-	Reserved for future use.
`GetProductID`	`0xF5`	L R X	Return the module type identifier.
`GetVersion`	`0xF6`	L R X	Return the module firmware version number.
`GetAddress`	`0xF7`	L R X	Return the module address shunt configuration.
`---`	`0xF8-0xFD`	-	Reserved for future use.
`LinkQuery`	`0xFE`	L	Test for module presence.
`NOP`	`0xFF`	L R	No operation.

5.2 Actions

5.2.1 SoftReset

Function Invokes a software reset by forcing the target IOM to restart its firmware.

Opcode `0xF0`

Command `none`

Response `none`

Notes After executing this action, the target module will immediately undergo a reboot. During the resulting reboot, the module will not process command packets, nor will it return response packets to the client. To maintain client-MM synchronization during `SoftReset`, the client should use a polling process similar to that described in the `HardReset` action.

`SoftReset` does not cause the target module's `RCV` or `XMT` indicators to light, as this action is not intended for use by the Ethernet client.

5.2.2 HardReset

Function Invokes a hardware reset on the target IOM.

Opcode `0xF1`

Command `none`

Response `none`

Notes This action disables the resetting of the target module's hardware watchdog timer. Upon receipt of a `HardReset` action, and until the resulting reboot is finished, the target module will not process command packets, nor will it return response packets to the client.

To ensure synchronization between the client and the target IOM, the client should wait for the IOM to complete its reboot before invoking other actions on the IOM. This can be accomplished by polling the IOM, either continuously or at periodic intervals, until a response packet is received that indicates the module has undergone a reset.

5.2.3 ResetFlags

Function Resets to zero the specified bit flags in the Status byte.

Opcode `0xF2`

Command `(BitMask)`

Parameter	Function
`BitMask`	Specifies the bit flags to be cleared. Set a mask bit to *one* to clear the matching Status bit. Set a mask bit to *zero* to leave the matching Status bit in its current state.

Response `none`

Notes This action clears all of the specified bit flags in the Status byte (Section 3.4.1), which is present in every ModRsp.

Except for the `HRST` (hard reset) bit flag, any combination of bit flags may be cleared by the Ethernet client. The client is not permitted to clear the `HRST` flag because this flag is reserved for gateway-to-IOM link control.

5.2.4 SetWatchdog

Function Programs the communication watchdog time-out interval.

Opcode `0xF3`

Command `(Interval)`

Parameter	Function
`Interval`	Communication watchdog interval, specified in 2 millisecond increments. Specify *zero* to disable the communication watchdog.

Response `none`

Notes The communication watchdog timer is used to detect interruptions in communications between the gateway and IOM. If no communications are received from the gateway within the specified time-out interval, an IOM will execute a hardware watchdog reset. This behavior ensures the shutdown of all I/O on any IOM which has lost communication with the gateway.

The gateway automatically invokes this action for an IOM when it first establishes communication with the module. As a result, there is no need for the client to explicitly invoke this action. In a typical application system, the client will never need to, nor should it, invoke this action.

5.2.5 GetProductID

Function Returns the model type identifier.

Command `(0xF5)`

Response `(ModelNum<15:8>),(ModelNum<7:0>)`

Parameter	Function
`ModelNum`	Model number of the target module.

Notes The returned value is a 16-bit, unsigned integer that indicates the model number of the target module. For example, the decimal value 2650 is returned by a model 2650 relay module.

5.2.6 GetVersion

Function Returns the firmware version number.

Opcode `0xF6`

Command `none`

Response `(MajorVer),(MinorVer)`

Parameter	Function
`MajorVer`	Major version number. Ranges from 00 to 99 decimal.
`MinorVer`	Minor version number. Ranges from 00 to 99 decimal.

Notes On firmware labels, the major and minor version numbers are printed with a decimal point delimiter. For example, printed version “1.02” is indicated by `MajorVer` = 1 and `MinorVer` = 2.

5.2.7 GetAddress

Function Returns the address shunt settings.

Opcode `0xF7`

Command `none`

Response `(Address)`

Parameter	Function
`Address`	Address shunt settings. Ranges from 0x00 to 0x0F.

Notes Most IOMs have provision for installing address shunts. These shunts enable the system integrator to specify an address for the module, with a value in the range 0 to 15 decimal.

The MM does not use the board address; it is provided for exclusive use by the Ethernet client. For example, the client may use the module address to differentiate between IOMs of the same type, or to verify that IOMs are connected to the correct IOM ports on the MM.

5.2.8 LinkQuery

Function Queries an IOM port to determine if a module is present.

Opcode `0xFE`

Command `none`

Response `none`

Notes In normal operation, `LinkQuery` should never be invoked by the client. `LinkQuery` is invoked by the gateway, in a manner that is transparent to the Ethernet client, to determine if a module is connected to a specific IOM port.

`LinkQuery` does not cause the target module's RCV or XMT indicators to light as this action is not intended for use by the Ethernet client.

5.2.9 NOP

Function No operation.

Opcode `0xFF`

Command `none`

Response `none`

Notes `NOP` may be used to acquire IOM status when no other actions are required.

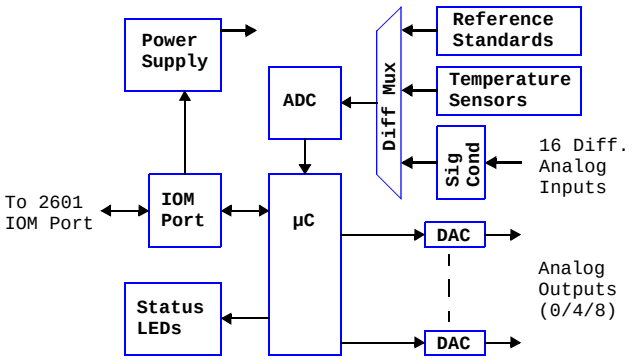
Chapter 6: Model 2608 Analog I/O Module

6.1 Introduction

The model 2608 module is a smart I/O module (IOM) that has 16 fully differential analog input (AIN) channels and either zero (2608-0), four (2608-4) or eight (2608-8) analog output (AOUT) channels. By means of a single Cat-5 cable, the 2608 may be connected to any IOM port on a model 2601 main module (MM).

Analog I/O channels are referenced by channel number. AIN channel numbers range from 0 to 15, and AOUT channel numbers range from 0 to 3.

Figure 32: Model 2608 Block Diagram



Analog input and output subsystems include special circuitry that implements fully electronic calibration. Excellent long-term accuracy is achieved through the combination of high quality on-board reference standards and the total absence of mechanical trims. Calibration constants are stored on-board in non-volatile memory so that modules can be changed out without the need for system-level recalibration.

Module connectors are optimized for direct connection to field wiring, in most cases eliminating the need for external field wiring termination systems. All field wiring is attached directly to the module by means of pluggable connectors so that downtime will be minimized should it become necessary to replace a module.

The entire module is automatically reset by a built-in watchdog timer if an internal fault develops. Also, the module automatically resets if the MM fails to communicate within a programmable time interval.

Diagnostic LEDs include Power, Heartbeat, and communication status. A self-test is performed automatically at power-on and reset, and any detected fault conditions are displayed on the status indicators.

A high-efficiency, on-board power supply produces all required operating voltages from a single 24VDC source, which is obtained from the MM via the module's IOM port. Test points are provided for all power supply voltages and analog I/O signals.

6.2 Analog Outputs

AOUT channels generate programmed voltages spanning the range from -10V to +10V. Each AOUT channel employs a dedicated digital-to-analog converter (DAC) with an effective resolution of 15 bits, minimum. All AOUT channels default to zero volts out upon reset or power-up.

Each AOUT channel is provided with a connector that includes circuits for analog output, remote output sense, output return, and power for external circuitry such as active sensors.

6.3 Analog Inputs

Analog inputs utilize fully differential signal paths to help maintain measurement accuracy in electrically noisy environments. AIN channels may be independently software configured for either a ±10V or ±100mV input range.

Various reference standards, as well as thermocouple reference junction temperature sensors, are built into the model 2608. All of these references occupy dedicated input channels so as to not consume application input channels.

AIN signals connect to the module by means of pluggable terminal blocks. Each channel is provided with terminal block circuits for the differential analog input signal pair and power out to external signal-producing devices.

6.3.1 Conversion Process

A 10 µsec, 16-bit ADC is used to digitize all external input signals and the internal references. Data conversions are performed periodically, at two millisecond intervals, independent of client activity.

During each two-millisecond (mS) interval, all 16 external inputs and all internal references are acquired. External inputs are acquired sequentially in order of increasing channel number, beginning with channel 0. The following process is applied to each external input:

- The analog input signal is routed to the ADC and gain is applied as appropriate.
- The ADC's input signal path is given time to settle.
- The input is digitized four times in rapid succession, and the results are averaged to produce a *snapshot* value. The snapshot value is then stored in on-board memory.
- Snapshot values are accumulated for 16 mS (or 20 mS, if the power line frequency is declared to be 50 Hz). The accumulated snapshots are averaged to produce the *integrated* value, which is then stored in on-board memory. Because this process effectively integrates the measurement for a duration of one line cycle period, differential line frequency noise is greatly attenuated.

6.3.1.1 Returned Data

In the case of the 16 external input channels, both the snapshot and integrated values may be accessed by the client. Snapshot values are most useful if minimum data age is required, while integrated values provide superior noise reduction and line frequency rejection.

Only integrated values are available from the on-board temperature sensor and reference standards channels. This is because these values are not intended for use by real-time applications.

Although the snapshot and integrated values may be directly accessed by a client, it is usually preferable to use the middleware data acquisition functions supplied on the distribution media; these functions perform offset and gain corrections, return acquired voltages in engineering units and provide extensive support for thermocouples.

6.3.2 Thermocouples

Onboard temperature sensors are mounted near the analog input terminations to provide reference junction compensation for thermocouples (TCs). These sensors occupy dedicated input channels so that all of the application channels will remain available for external signal measurements.

When interfaced to a TC, an AIN channel must be configured for the ±100mV input range. This provides sufficient gain for resolving the TC signal, and it is the input range that is expected by the distribution media, which translates TC voltage into temperature units. Refer to the documentation supplied with the distribution media for information about TC software support.

The TC's common mode voltage (CMV) must not exceed the maximum specified CMV limits of the 2608 AIN channels. To prevent excessive CMV, the TC should have no electrical connection at its hot end (i.e., it should be an "isolated" TC) and hardware programming shunts should be installed as described in Section 6.4.2.1.

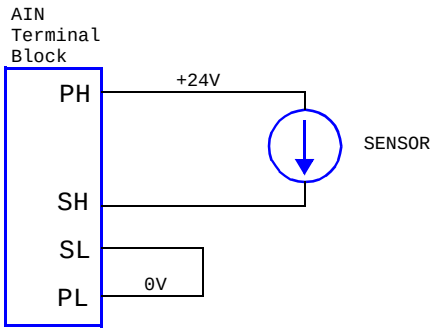
6.3.3 4-20 mA Current Loops

An AIN channel must be configured for the ±10V input range when it will be used to measure a 4-20 mA current loop; this is the input range that is expected by the middleware, which translates the measured loop current into engineering units. Refer to the documentation supplied with the distribution media for details.

To guarantee conformance to the maximum AIN common mode voltage specification, only the grounded end of the current loop should be connected to the AIN channel.

The AIN channel may be configured to provide excitation for current loops, as shown in the following schematic. Note that in this case, the loop current power is provided directly by the AIN channel; see Section 6.4.2.2 for an explanation of how to configure option shunts so that the AIN channel will provide power for a current loop.

Figure 33: Typical Current Loop Schematic



6.4 Hardware Configuration

6.4.1 Analog Outputs

Each AOUT channel is provided with a 6-pin header—called the *output programming block*, or *OPB*—that is designed to accept hardware programming shunts. Various configuration options are programmed by installing shunts in the appropriate positions on the OPB, as described in the following sections.

6.4.1.1 Remote Sense Shunts

AOUT channels must be individually configured for either local or remote output sensing.

Many control systems generate control outputs based on process feedback. Local sensing is well suited for such applications, as the absolute accuracy of the control signal is not a concern. To enable local sensing, install a shunt on the AOUT channel's OPB at pins 1-2. When local sensing is employed, there should be no connection to the Sense pin on the AOUT channel's connector.

Remote sensing should be used if high output voltage accuracy is required in the presence of widely varying or large load currents. Remote sensing requires two conductors for the output's high side signal: one to supply load current and another to sense the applied voltage at the load. If remote sensing is employed, make sure to remove the shunt (if there is one) on the AOUT channel's OPB at pins 1-2.

6.4.1.2 Power Distribution Shunts

It is sometimes necessary to supply additional operating power to devices that are connected to AOUT channels. The OPB may be programmed to route either +24V or +10V to an external device via the associated AOUT connector. The selected voltage is routed to the connector on pin 5.

Install a shunt on the OPB at pins 3-4 to route 24VDC to the connector. Install a shunt on the OPB at pins 4-6 to route 10VDC to the connector.

6.4.1.3 Connectors

Each AOUT channel is provided with a dedicated, five-pin connector. The connectors are labeled "AOUT0" through

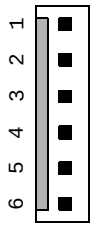
"AOUT7," which correspond to analog output channels 0 through 7, respectively.

Table 10: AOUT Connectors

Chan	Connector
AOUT0	P1
AOUT1	P2
AOUT2	P7
AOUT3	P8
AOUT4	P9
AOUT5	P10
AOUT6	P15
AOUT7	P16

The connector pinout is identical for all eight AOUT channels:

Table 11: Pinouts of AOUT Connectors

Layout (top view)	Pin	Name	Function
	1	Shld	This may be connected to a cable shield, but the shield conductor must be left disconnected at the remote end of the cable in order to avoid ground loops.
	2	Out	Analog output signal that supplies current to the load.
	3	Sense	Analog output remote sense.
	4	Com	Analog output current return. The load is connected between this pin and Out.
	5	Pwr	Optional power output, either 24VDC or 10VDC.

6.4.2 Analog Inputs

Each AIN channel is provided with a 10-pin header—called the *input programming block*, or *IPB*—that is designed to accept hardware programming shunts. Various signal conditioning options are enabled by installing shunts on the IPB as described in the following sections.

6.4.2.1 Thermocouple Shunts

Each IPB has options for installing either one or two shunts for TC support. A shunt should always be installed at pins 5-6 when the affiliated AIN channel will be connected to a TC. This will prevent the TC's common-mode voltage (CMV) from exceeding the maximum specified input CMV for the AIN channel, and it provides a low impedance current path to ground for RF noise.

To enable open-TC detection, install a second shunt at pins 7-8. This will force the AIN channel to return a large, positive digitized value in the event of an open-circuit failure at the TC's hot junction. If this shunt is not installed, the returned digitized value will be indeterminate in the case of an open TC.

6.4.2.2 4-20mA Termination Shunts

When an AIN channel is interfaced to a 4-20mA current loop, the IPB may be programmed to insert a 500 ohm, 1% resistor into the loop to convert the loop current to a voltage. Install a shunt at pins 9-10 to enable this option.

6.4.2.3 Power Distribution Shunts

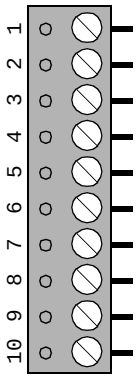
In many applications, it is necessary to supply operating power to devices that are connected to AIN channels. The IPB may be programmed to route either 24VDC or 10VDC to an external device via the associated AIN channel's terminal block. The selected voltage is routed to the channel's "Pwr" terminal.

Install a shunt on the IPB at pins 3-4 to route 24VDC to the Pwr terminal. Install a shunt on the IPB at pins 2-4 to route 10VDC to the Pwr terminal.

6.4.2.4 Terminal Blocks

Eight 10-pin pluggable terminal blocks (TBs) provide all connections required by the sixteen AIN channels. Each TB provides connections for two adjacent AIN channels, as shown in Table 12.

Table 12: Pinouts of AIN Terminal Blocks

Layout (top view)	Pin No.	Func.	Channel							
	1	Pwr	1	3	5	7	9	11	13	15
	2	Shld								
	3	Ain+								
	4	Ain-								
	5	Com								
	6	Pwr	0	2	4	6	8	10	12	14
	7	Shld								
	8	Ain+								
	9	Ain-								
	10	Com								
	TB:		P6	P5	P4	P3	P11	P12	P13	P14

Each AIN channel has five signals, as detailed in Table 13.

Table 13: AIN Signals

Name	Function
Pwr	Optional power output, either 24VDC or 10VDC.
Shld	This may be connected to a cable shield, but the shield conductor must be left disconnected at the remote end of the cable in order to avoid ground loops.
Ain+	Positive sense input. The differential input signal is applied to Ain+ and Ain-.
Ain-	Negative sense input.
Com	Analog ground reference. If connecting to an isolated signal source (e.g., isolated power supply, battery, etc.) this should be connected to Ain+ or Ain- to prevent excessive common mode voltages. This signal may also be used as a current return for Pwr.

6.5 Reset

Upon module reset, all analog input channels default to low gain and all analog outputs switch to zero volts out. A 2608 module will experience a module reset in response to any of the following conditions:

- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.
- Communication time-out, which will occur if the client fails to communicate with the DIO module within a programmable communication watchdog interval.

6.6 Actions

This section describes the programmable actions that are supported exclusively by AIO modules. These actions may be employed, in any sequence or combination, in MCmd action lists. In addition to the actions listed here, the AIO module also supports all of the common actions that are universally supported by all IOMs.

Table 14: Summary of Model 2608 Actions

Command	Opcode	Function
SetOutput	0x00	Program analog output.
GetOutput	0x01	Return analog output.
SetInputRanges	0x02	Select input ranges.
GetInputRanges	0x03	Return input ranges.
GetSnapshots	0x04	Return snapshot values.
GetIntegrateds	0x05	Return integrated values.
GetInternals	0x0B	Return digitized references.
SetLineFreq	0x0C	Declare line frequency.
ReadEeprom	0x0D	Return byte from EEPROM
WriteEeprom	0x0E	Write byte to EEPROM

Note: Model 2608 has a maximum MRsp size of 38 bytes.

6.6.1 SetOutput

Function Programs the output setpoint of one analog output channel.

Opcode 0x00

Command (Chan),(Value<7:0>),(Value<15:8>)

Parameter	Function
Chan	Analog output channel to be programmed. Values may range from 0 to 3, inclusive. The six high-order bits should be zero for future compatibility, but are ignored in this version.
Value	Signed, 16-bit value, LSB first, that specifies the output setpoint for the target analog output channel.

Value, which specifies the exact value that will be written to the DAC, is not corrected for DAC gain and offset errors. The following table illustrates the approximate relationship between the Value parameter and the resulting output voltage.

Value	Output Voltage (V)
0x8000	-10.0003
0x8001	-10.0000
0xFFFF	- 0.0003052
0x0000	0.0000
0x0001	+ 0.0003052
0x7FFF	+10.0000

Response none

Notes Although the action response is generated immediately, the actual updating of the analog output is deferred. The analog output will be set to its target output voltage within two milliseconds after this action executes.

All analog output setpoints default to zero in response to a module reset, causing all analog outputs to produce zero volts out.

6.6.2 GetOutput

Function Returns the programmed output setpoint of one analog output channel.

Opcode 0x01

Command (Chan)

Parameter	Function
Chan	Analog output channel to be addressed. Values may range from 0 to 3, inclusive. The six high-order bits should be zero for future compatibility, but are ignored in this version.

Response `(Value<7:0>),(Value<15:8>)`

Parameter	Function
`Value`	Signed, 16-bit value, LSB first, that indicates the programmed output setpoint for the target analog output channel. This is the value set by the most recent `SetOutput` action, or zero if the module has been reset.

6.6.3 SetInputRanges

Function Programs the input ranges for all 16 analog input channels.

Opcode `0x02`

Command `(Range<7:0>),(Range<15:8>)`

Parameter	Function
`Range`	Bit flags that specify the voltage input range to use when measuring analog input channels. Each bit corresponds to one channel (i.e., bit 9 corresponds to channel 9). Set a flag to 0 to select the ±10V range, or 1 to select the ±100mV range.

Response `none`

Notes Once set, the input range settings apply until they are changed by another `SetInputRange` action or a module reset occurs.

After a module reset, all channels default to the ±10V range.

Any channel that has its range changed by this action will have its stored snapshot and integrated sample values reset. The next snapshot will be stored as both the current snapshot value and as the integrated value; the next integrated value update will occur 16 (or 20, if the line frequency is 50 Hz) milliseconds later.

The input range must be set to ±10V on any channel that has the 4-20mA termination option enabled. In this case, the channel will measure +10V at 20mA, and +2V at 4mA.

The input range must be set to ±100mV on any channel that is connected to a thermocouple. Refer to the distribution media for detailed information about thermocouple usage with the model 2608.

6.6.4 GetInputRanges

Function Returns the programmed input ranges of all 16 analog input channels.

Opcode `0x03`

Command `none`

Response `(Range<7:0>),(Range<15:8>)`

Parameter	Function
`Range`	Bit flags that specify the voltage input range to use when measuring analog input channels. Each bit corresponds to one channel (i.e., bit 9 corresponds to channel 9). Flag set to 0 indicate the ±10V range; flags set to 1 indicate the ±100mV range.

Notes The returned Range flags will indicate the values set by the most recent `SetInputRange` action, unless a module reset has occurred, in which case all Range flags will be reset to zero.

6.6.5 GetSnapshots

Function Returns the most recently acquired snapshot values from the 16 external AIN channels.

Opcode `0x04`

Command `none`

Response
```
(Data[0]<7:0>),(Data[0]<15:8>),
 ... ,
(Data[15]<7:0>),(Data[15]<15:8>)
```

Parameter	Function
`Data`	Word array containing the acquired values from all analog input channels. Data ordering begins with channel 0 and progresses by increasing channel number. All word values are returned least significant byte first.

Notes This action does not cause AIN channels to be digitized; it only returns previously digitized values. Returned values range in age from 0 to 2 milliseconds.

One 16-bit, signed binary value is returned for each of the module's 16 AIN channels. For each channel, the value that is returned is the average of four samples acquired in rapid succession.

Returned values are not corrected to eliminate gain and offset errors. Each returned `Data` word value is scaled approximately as follows:

Input Range	Data Scaling, Per Bit
±10 V range	305.1758 µV
±100 mV range	3.051758 µV

6.6.6 GetIntegrateds

Function Returns the most recently acquired integrated values from the 16 external AIN channels.

Opcode `0x05`

Command `none`

Response
```
(Data[0]<7:0>),(Data[0]<15:8>),
 ... ,
(Data[15]<7:0>),(Data[15]<15:8>)
```

Parameter	Function
`Data`	Word array containing the acquired values from all analog input channels. Data ordering begins with channel 0 and progresses by increasing channel number. All word values are returned least significant byte first.

Notes Like the `GetSnapshots` action, this action does not cause AIN channels to be digitized; it simply returns previously digitized values. The ages of the returned values range from 0 to 16 milliseconds.

The differences between these data values and those returned by `GetSnapshots` is the age of the data and the amount and nature of the applied filtering.

6.6.7 GetInternals

Function Returns the most recently acquired integrated values from all internal reference standards and the on-board temperature sensors.

Opcode `0x0B`

Command `none`

Response
```
(LgMax[0]),(LgMax[1]),
(LgMin[0]),(LgMin[1]),
(HgMax[0]),(HgMax[1]),
(HgMin[0]),(HgMin[1]),
(Temp[0][0]),(Temp[0][1]),
(Temp[1][0]),(Temp[1][1]),
(Temp[2][0]),(Temp[2][1]),
(Temp[3][0]),(Temp[3][1]),
(Temp[4][0]),(Temp[4][1]),
(Temp[5][0]),(Temp[5][1]),
(Temp[6][0]),(Temp[6][1]),
(Temp[7][0]),(Temp[7][1])
```

Parameter	Function
`LgMax`	+10V ref @ low gain, LSB first.
`LgMin`	0V @ low gain, LSB first.
`HgMax`	+100mV ref @ high gain, LSB first.
`HgMin`	0V @ high gain, LSB first.
`Temp[0]`	Chan 0-1 temperature, LSB first.
`Temp[1]`	Chan 2-3 temperature, LSB first.
`Temp[2]`	Chan 4-5 temperature, LSB first.
`Temp[3]`	Chan 6-7 temperature, LSB first.
`Temp[4]`	Chan 8-9 temperature, LSB first.
`Temp[5]`	Chan 10-11 temperature, LSB first.
`Temp[6]`	Chan 12-13 temperature, LSB first.
`Temp[7]`	Chan 14-15 temperature, LSB first.

Notes This action does not cause the reference standards to be digitized; it simply returns the previously digitized standards values. The returned data are the uncorrected binary values obtained from the digitizer. These values are not valid immediately following a module reset; allow at least 32 milliseconds following a module reset before executing this action.

All reference standards are measured at low gain, except `HgMax` and `HgMin`, which are measured at high gain. Each returned word value—which represents the measured output voltage produced by the associated voltage or temperature reference—is scaled to a resolution of approximately 305 µV (low gain) or 3.05 µV (high gain).

Reference voltage values may be used to normalize measurements from external AIN channels and the on-board temperature sensors so as to eliminate circuit gain and offset errors. Refer to the distribution media middleware documentation for details.

6.6.8 SetLineFreq

Function Declares the power line frequency used in the application environment.

Opcode `0x0C`

Command `(FCode)`

Parameter	Function
`FCode`	Enumerated value that specifies the power line frequency: set to 0 (default) for 60 Hz, set to 1 for 50 Hz.

Response `none`

6.6.9 ReadEeprom

Function Returns a byte value from the module's non-volatile memory.

Opcode `0x0D`

Command `(Adrs)`

Parameter	Function
`Adrs`	Memory address that is to be read from.

Response `(Data)`

Parameter	Function
`Data`	Byte value retrieved from the EEPROM.

Notes This function is used by the middleware to store calibration data for the module. Refer to the distribution media for details.

6.6.10 WriteEeprom

Function Stores a byte value in the module's non-volatile memory.

Opcode `0x0E`

Command `(Adrs),(Data)`

Parameter	Function
`Adrs`	Memory address that is to be written to.
`Data`	Byte value that is to be stored.

Response `none`

Notes This function is used by the middleware to store calibration data for the module. Refer to the distribution media for details.

Due to the lengthly time required to write a byte to the EEPROM, this action should be invoked only once per command packet.

6.7 Specifications

Table 15: Model 2608 Specifications

Analog Inputs	Resolution	16 bits
	Input channels	16
	Conversion rate	500 samples/second for each input channel.
	Input range	±10V, ±100mV software selectable per channel. 4-20mA selectable by option shunts per channel.
	Input impedance	100 megohm with no thermocouple shunts installed.
	Sensor support	Thermocouple cold-junction compensation. Thermocouple open-circuit detection.
Analog Outputs	Resolution	16 bits
	Output channels	0 (2608-0), 4 (2608-4) or 8 (2608-8).
	Output range	±10V
	Output current	2mA, max
Auxilliary Power Output	Power rating	Shunt selectable for each channel: 10VDC ±5% @ 400mA (total for all channels), or 24VDC @ 500mA (total for all channels)
	Fuse	Socketed 500mA fuse per channel. Wickman 396-0500-044 or equivalent.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent
	Analog outputs (qty 0, 4 or 8)	(*) Molex 22-01-3067 with 08-50-0114 pin contacts (6), or equivalent
	Analog inputs (qty 8)	(*) Pluggable TB, 10-pin, RIA 31166110 or equivalent
Temperature	Operating range	0 to 70°C
Power	Input power	+24V ±5% @ 80mA.
Dimensions	Outer dimensions	5.0H x 8.7W x 1.8D inches, including DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel; additional clearance required for field wiring.

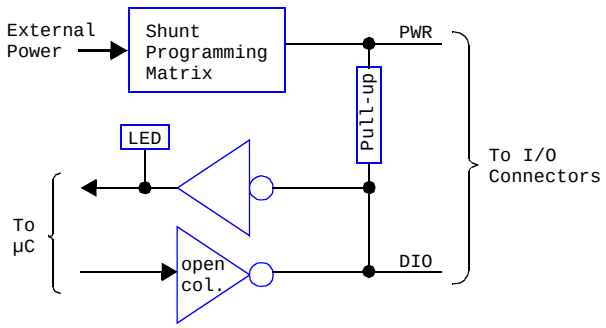
Chapter 7: Model 2610 Digital I/O Module

7.1 Introduction

The model 2610 is a smart I/O module (IOM) that has 48 programmable digital I/O (DIO) channels, numbered 0 to 47. The module's microcontroller, which provides I/O services to a remote client, communicates with the client by means of a standard Category-5 STP/UTP cable over an optically isolated, asynchronous serial interface.

As shown in Figure 34, each channel consists of an active-low, open-collector driver, a pull-up resistor and an active-low receiver. A shunt programming matrix enables one of six possible external power sources to supply power to the channel. For simplicity, some circuitry is excluded from the block diagram, including a diode that suppresses induced EMF from inductive loads.

Figure 34: DIO Channel Block Diagram



The client can program and read back the state of each channel's output driver, and it can acquire the debounced state of each channel's physical signal. The microcontroller debounces all acquired channel states by means of a 10 millisecond software debounce filter. In addition, 32 of the DIO channels have the ability to autonomously generate PWM output signals at a rate and duty cycle specified by the client.

Each DIO channel includes an LED indicator that lights when the channel is active. The indicator lights regardless of whether the channel is driven by an external signal source or by the channel's output driver.

On-board connectors enable direct connection to field wiring without the need for intermediate terminal blocks. Each connector, which uses a standard RJ-12 modular jack, includes connections for three DIO channels to support direct wiring to devices that require the services of multiple DIO channels.

Integral support is provided for system interlock circuits and multiple DIO power supply voltages. Each DIO channel may be independently powered from any of up to six external power sources.

The communication interface between the client and the on-board microcontroller is optically isolated to ensure error-free operation in demanding industrial environments.

7.2 Hardware Configuration

The DIO module is configured by installing programming shunts and connecting various cables to the module as described in this section.

7.2.1 IOM Network Connector

A single RJ-45 connector, J49, is used to connect the DIO module to its client. This should be mated to a standard UTP (unshielded twisted pair) Category-5 cable.

The other end of this cable will be connected to one of the sixteen IOM ports on a model 2601 module.

7.2.2 Interlock Power Connectors

Each DIO channel is powered from an external power source. At least one external power source is required to make the DIO channels functional. Up to six independent external power sources are supported. Due to a forward diode drop, *each power source is reduced by approximately 0.8V* before it is applied to its target DIO channel.

Connectors P1 and P2 supply power to DIO channels from external power sources. All external power sources must supply positive DC voltages with respect to system ground. The current return of each external power source must be connected to the system ground.

P1 and P2 are identical in function and pinout. The external power sources may be connected to either P1 or P2. If other IOMs require access to these external power sources, the unused connector may be used to daisy-chain power out to other IOMs. In this manner, power can be distributed to any arbitrary number of external IOMs without the use of dedicated power distribution terminal blocks.

Table 16: Pinouts of Connectors P1 and P2

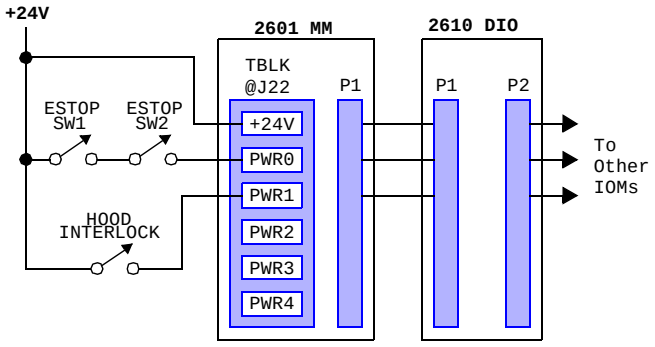
PWB Label	Pin	Function
+24V	1	+24V power, always on.
0	2	Optional positive DC power #0.
1	3	Optional positive DC power #1.
2	4	Optional positive DC power #2.
3	5	Optional positive DC power #3.
4	6	Optional positive DC power #4.

The external power sources need not be derived from interlock contacts, but by employing interlocked power sources, the system interlock cabling can be minimized. This feature is

especially useful for guaranteeing the fail-safe shutdown of select DIO channels in the event of a critical condition such as emergency stop (ESTOP) activation, open safety hood, etc.

Figure 35 illustrates a system in which PWR0 is connected to multiple system emergency stop contacts, and PWR1 is routed through a safety hood interlock contact. Note that all of the external power sources are daisy-chained from module to module. Any DIO that is configured to use PWR0 will automatically lose power when either ESTOP contact is opened, and any DIO that is configured to use PWR1 will lose power when the hood opens. All other DIOs use the +24V "always on" power source and thus are unaffected by interlock contacts. PWR2, PWR3 and PWR4 are not used in this application.

Figure 35: Wiring Example With Interlock Contacts



Use Sensoray cable assembly, part number 2600C1, to connect P1 or P2 to the power daisy chain

7.2.3 DIO Power Selection Shunt

Each DIO channel may be independently operated from any of the six external power sources (that are connected to P1 and P2) by installing the appropriate programming shunt on the module.

A shunt receptacle matrix is provided for each channel. Each matrix has six shunt positions, corresponding to the six external power sources. As shown in Table 17, a programming shunt must be installed at the position corresponding to the DIO channel's desired supply voltage. Only one shunt should be installed per block.

Table 17: Interlock Power Programming Matrix

PWB Label	Shunt Pins	Selected Power Source
+24V	1-2	+24V power, always on.
0	3-4	Optional positive DC power #0.
1	5-6	Optional positive DC power #1.
2	7-8	Optional positive DC power #2.
3	9-10	Optional positive DC power #3.
4	11-12	Optional positive DC power #4.

Each channel's shunt receptacle matrix is located to the rear of that channel's I/O connector.

7.2.4 I/O Connectors

The DIO module employs 48 RJ-12 modular jacks for direct connection to field wiring, designated J1 to J48.

As shown in Table 18, each jack has five signal circuits: the shunt-selected interlock power source (PWR) on pin 2, common ground (GND) on pin 6, and three contiguous DIO channels (except J24 and J48, which have a reduced number of DIO channel connections) on pins 3, 4 and 5. Pin signals labeled "NC" are not connected to any circuitry on the module.

Table 18: Pinouts of the I/O Connectors

PWB Label	RJ-12 Jack	Pin Signal					
		1	2	3	4	5	6
CH0	J1	NC	PWR	CH0	CH1	CH2	GND
CH1	J25	NC	PWR	CH1	CH2	CH3	GND
CH2	J2	NC	PWR	CH2	CH3	CH4	GND
CH3	J26	NC	PWR	CH3	CH4	CH5	GND
CH4	J3	NC	PWR	CH4	CH5	CH6	GND
CH5	J27	NC	PWR	CH5	CH6	CH7	GND
CH6	J4	NC	PWR	CH6	CH7	CH8	GND
CH7	J28	NC	PWR	CH7	CH8	CH9	GND
CH8	J5	NC	PWR	CH8	CH9	CH10	GND
CH9	J29	NC	PWR	CH9	CH10	CH11	GND
CH10	J6	NC	PWR	CH10	CH11	CH12	GND
CH11	J30	NC	PWR	CH11	CH12	CH13	GND
CH12	J7	NC	PWR	CH12	CH13	CH14	GND
CH13	J31	NC	PWR	CH13	CH14	CH15	GND
CH14	J8	NC	PWR	CH14	CH15	CH16	GND
CH15	J32	NC	PWR	CH15	CH16	CH17	GND
CH16	J9	NC	PWR	CH16	CH17	CH18	GND
CH17	J33	NC	PWR	CH17	CH18	CH19	GND
CH18	J10	NC	PWR	CH18	CH19	CH20	GND
CH19	J34	NC	PWR	CH19	CH20	CH21	GND
CH20	J11	NC	PWR	CH20	CH21	CH22	GND
CH21	J35	NC	PWR	CH21	CH22	CH23	GND
CH22	J12	NC	PWR	CH22	CH23	CH24	GND
CH23	J36	NC	PWR	CH23	CH24	CH25	GND
CH24	J13	NC	PWR	CH24	CH25	CH26	GND
CH25	J37	NC	PWR	CH25	CH26	CH27	GND
CH26	J14	NC	PWR	CH26	CH27	CH28	GND
CH27	J38	NC	PWR	CH27	CH28	CH29	GND
CH28	J15	NC	PWR	CH28	CH29	CH30	GND
CH29	J39	NC	PWR	CH29	CH30	CH31	GND
CH30	J16	NC	PWR	CH30	CH31	CH32	GND
CH31	J40	NC	PWR	CH31	CH32	CH33	GND
CH32	J17	NC	PWR	CH32	CH33	CH34	GND
CH33	J41	NC	PWR	CH33	CH34	CH35	GND
CH34	J18	NC	PWR	CH34	CH35	CH36	GND
CH35	J42	NC	PWR	CH35	CH36	CH37	GND
CH36	J19	NC	PWR	CH36	CH37	CH38	GND

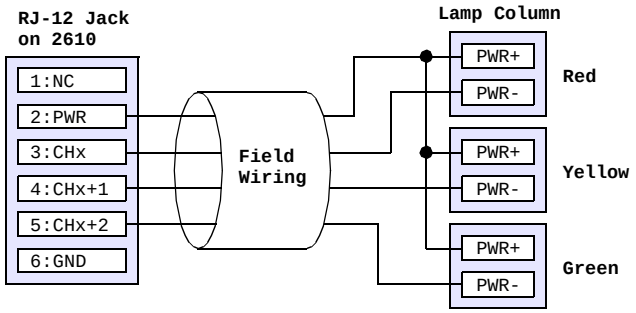
Table 18: Pinouts of the I/O Connectors

PWB Label	RJ-12 Jack	Pin Signal					
		1	2	3	4	5	6
CH37	J43	NC	PWR	CH37	CH38	CH39	GND
CH38	J20	NC	PWR	CH38	CH39	CH40	GND
CH39	J44	NC	PWR	CH39	CH40	CH41	GND
CH40	J21	NC	PWR	CH40	CH41	CH42	GND
CH41	J45	NC	PWR	CH41	CH42	CH43	GND
CH42	J22	NC	PWR	CH42	CH43	CH44	GND
CH43	J46	NC	PWR	CH43	CH44	CH45	GND
CH44	J23	NC	PWR	CH44	CH45	CH46	GND
CH45	J47	NC	PWR	CH45	CH46	CH47	GND
CH46	J24	NC	PWR	CH46	CH47	NC	GND
CH47	J48	NC	PWR	CH47	NC	NC	GND

Except for J24 and J48 (channels 46 and 47), each jack exposes connections to three DIO channels. For example, J1 provides connections to channels 0, 1 and 2. This feature can simplify wiring in cases where a single field cable must carry more than one DIO signal.

To get an idea of how this feature might be used, consider the case of a three-lamp lintern assembly with integrated cable (Figure 36). The source end of the cable can terminate to a single jack on the DIO module, which in turn provides access to all three lamp control signals.

Figure 36: Multiple Channels on Each I/O Jack



7.2.4.1 Restrictions

There are a few restrictions related to connecting multiple DIO channels through a single RJ-12 jack:

1. When multiple DIO channels connect to field wiring through a single jack, the channels that share the jack need not be powered from the same power source, but *only one power source per jack is available to the field wiring*.

   An example of this can be seen in Figure 36, in which all three lamps are powered from a common PWR source; this would not be possible if the lamps required differing supply voltages, because only one supply voltage is available from the jack.

2. If power can be interrupted on a DIO channel, the channel must connect to field wiring through a dedicated jack that is not shared by any other channels.

   For example, suppose PWR3 may be interrupted by an interlock contact, and channel 42 is powered from PWR3. In this case, the jack that connects field wiring to channel 42 (e.g., J21, J22 or J40) may connect to CH42, GND and PWR, but it must not make field wiring connections to any other DIO channels.

7.3 Channel Operating Modes

Each channel may be independently operated as an input or a programmed, standard output.

7.3.1 Input Operation

To use a channel as an input, the client must ensure that it never programs the channel to its active state. Since the client never enables the channel's output driver, the channel may be driven by an external, active-low driver. When the external driver is not driving the channel low, the channel's pull-up resistor will force the channel to its inactive state.

7.3.2 Standard Output Operation

When a channel is used as a standard output, the client simply programs the channel's output driver to the desired state. The channel is either driven low by the channel's on-board driver, or it is pulled high by the channel's pull-up resistor.

7.3.3 Wired-or Operation

Since a DIO channel's output driver is open-collector, it is possible to "wire-or" a channel by operating it both as a programmed output and by connecting the channel to one or more external open-collector drivers. If any of the connected drivers asserts its active-low output, the channel will be driven to its active state.

7.3.4 Output Modes

DIO channels 0 to 23 are unique in that they support two different output modes: Standard and PWM. In the Standard mode, a channel may be driven as described earlier: either by its explicitly programmed onboard driver or by an external current sink. In the PWM mode, the channel's output driver is cycled on and off at a client-specified duty cycle.

The output mode may be independently configured for each channel. For example, channel 0 can operate in the PWM mode while channels 1 to 23 operate in the Standard output mode.

All channels support the Standard operating mode, which is the default mode following a reset. Only channels 0 through 31 may be configured for the PWM mode; channels 32 through 47 support only the Standard mode.

7.3.5 Input Debounce

All 48 DIO channel physical states are sampled every 10 milliseconds. The on-board microcontroller applies a debounce filter to the sampled states, resulting in a 10 millisecond debounce period.

When acquiring physical channel states from the DIO module, the client always receives the debounced image of the physical states, which is delayed 10 milliseconds by the debounce function.

7.4 Reset

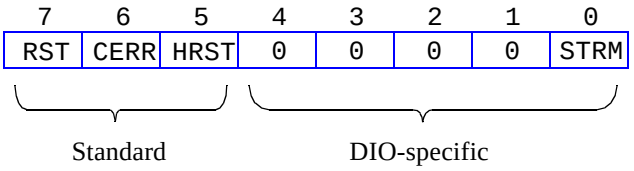
Upon module reset, all DIO drivers default to the inactive state and channels 0 to 23 default to the Standard operating mode. A DIO module will experience a module reset in response to any of the following conditions:

- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.
- Communication time-out, which will occur if the client fails to communicate with the DIO module within a programmable communication watchdog interval.

7.5 Status Flags

In addition to the standard flag bits that are common to all IOMs (see Section 3.4.1), the status byte returned by a DIO module includes one fault flag that is specific to DIO modules only. The DIO status byte is structured as shown in Figure 37.

Figure 37: DIO Status Byte



STRM - Active-high bit flag that indicates an error was detected in the serial data stream that is used to control the DIO output drivers. This flag can be cleared by invoking a `ResetFlags` action.

7.6 Actions

This section describes the programmable actions that are supported exclusively by the DIO module. These actions may be employed, in any sequence or combination, in MCmd action lists.

DIO modules support all of the common actions that are universally recognized by all IOMs, such as `HardReset`, `SoftReset`, etc. Refer to Chapter 5 for details.

Table 19: Summary of DIO Module Actions

Command	Opcode	Function
`SetModes`	`0x00`	Set operating modes for DIO channels 0-7.
`GetModes`	`0x01`	Return operating modes for DIO channels 0-7.
`SetPwmRatio`	`0x02`	Program PWM duty cycle and period.
`GetPwmRatio`	`0x03`	Return PWM duty cycle and period.
`GetInputs`	`0x04`	Return all DIO input states.
`GetOutputs`	`0x05`	Return all DIO output states.
`SetOutputs`	`0x06`	Program all DIO output states.
`SetModes32`	`0x07`	Set operating modes for DIO channels 0-31 (Vers. 1.02 or higher).
`GetModes32`	`0x08`	Return operating modes for DIO channels 0-31 (Vers. 1.02 or higher).

Note: Model 2610 has a maximum MRsp size of ten bytes.

7.6.1 SetModes

Function Programs the operating modes for DIO channels 0 through 7.

Opcode `0x00`

Command `(ModeFlags)`

Parameter	Function
`ModeFlags`	Operating modes for channels 0 to 7. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Set a bit to *one* to select the PWM mode, or to *zero* to select the Standard mode.

Response `none`

Notes Each of channels 0 through 7 may operate in either the Standard mode or the PWM mode. Following a reset, all channels default to the Standard operating mode.

When a channel's operating mode is first changed to the PWM mode, its output driver duty cycle is reset to zero; this prevents the output from going active until the duty cycle is explicitly set by a `SetPwmRatio` action.

Similarly, when a channel's operating mode is changed to the Standard mode, its output driver is reset to the inactive state to prevent its output from going active until explicitly set by a `SetOutputs` action.

7.6.2 GetModes

Function Returns the operating modes for DIO channels 0 through 7.

Opcode `0x01`

Command `none`

Response `(ModeFlags)`

Parameter	Function
`ModeFlags`	Operating modes for channels 0 to 7. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Set a bit to *one* to select the PWM mode, or to *zero* to select the Standard mode.

Notes `GetModes` returns the current operating modes of DIO channels 0-7. The operating modes may be changed by design (e.g., in response to a `SetModes` action) or unexpectedly (e.g., all channel operating modes revert to the default when a module reset occurs).

7.6.3 SetPwmRatio

Function Programs the PWM timing attributes for one DIO channel.

Opcode `0x02`

Command `(Channel),(OnTime),(OffTime)`

Parameter	Function
`Channel`	Byte value that specifies the DIO channel that is to be configured. Value may range from 0 to 23.
`OnTime`	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments. Value may range from 0 to 255.
`OffTime`	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments. Value may range from 1 to 255.

Response `none`

Notes `SetPwmRatio` applies only to channels that have previously been configured for the PWM operating mode. This action has no effect on channels that are configured for the Standard operating mode.

Only channels 0 through 23 can operate in the PWM output mode. The CERR status flag will be set if an illegal channel number is specified.

Two values are specified to establish the duty cycle: `OnTime` and `OffTime`. Each of these values specifies a time interval, expressed in 2 millisecond increments. For example, by programming `OnTime` to 1 and `OffTime` to 4, the duty cycle will be set to 20% with a period of 10 milliseconds (2 milliseconds active and 8 milliseconds inactive per output cycle).

The output duty cycle can be set to zero by declaring the `OnTime` value to be zero. It is not possible to set the duty cycle to 100% as the `OffTime` parameter must be at least 1. The `OffTime` value will automatically default to 1 if the client attempts to set it to zero.

After this action executes, the present channel output state will continue to be asserted for the duration of its previously programmed time, and then the new `OnTime` and `OffTime` will take effect.

7.6.4 GetPwmRatio

Function Returns the PWM timing attributes of one DIO channel.

Opcode `0x03`

Command `(Channel)`

Parameter	Function
`Channel`	Byte value that specifies the DIO channel to be addressed. Value may range from 0 to 23.

Response `(OnTime),(OffTime)`

Parameter	Function
`OnTime`	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments.
`OffTime`	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments.

Notes `GetPwmRatio` applies only to channels that have previously been configured for the PWM operating mode. The two returned byte values

are meaningful only if the specified channel is configured for the PWM operating mode.

Only channels 0 through 23 can operate in the PWM output mode. The CERR status flag will be set if an illegal channel number is specified.

Two values are returned: `OnTime` and `OffTime`. Each of these values specifies a time interval, expressed in 2 millisecond increments, that are currently in effect. These values will be the values programmed by the most recent `SetPwmRatio` action, or the default values if no `SetPwmRatio` action has executed since the channel's operating mode was switched to PWM.

7.6.5 GetInputs

Function Returns the debounced, physical channel states of all DIO channels.

Opcode `0x04`

Command `none`

Response `(S<7:0>),(S<15:8>),(S<23:16>),(S<31:24>),(S<39:32>),(S<47:40>)`

Parameter	Function
`S[]`	Byte array containing 48 bit flags that represent the debounced, physical states of all DIO channels. Each bit is associated with one DIO channel. For example, bit 17 is associated with DIO channel 17. Any bit set to *one* indicates the associated channel is set to the active state; any bit set to *zero* indicates the channel is set to the inactive state.

Notes The returned state values represent the debounced, physical states of all DIO channels. Because the inputs are sampled every 2 milliseconds, and the debounce period is 10 milliseconds, the returned state values will all have an age ranging from 10 to 12 milliseconds, plus any network communication latency.

Note that the physical states of *all* DIO channels are returned, regardless of their respective operating modes.

7.6.6 GetOutputs

Function Returns the programmed output driver states of all DIO channels. Note that the programmed output driver states may not correspond exactly to the physical channel states (that would be returned by `GetInputs`) as some channels may be driven by external signal sources. In the case of channels that have been configured for the PWM mode, this function returns indeterminate state values.

Opcode `0x05`

Command `none`

Response `(S<7:0>),(S<15:8>),(S<23:16>),(S<31:24>),(S<39:32>),(S<47:40>)`

Parameter	Function
`S[]`	Byte array containing 48 bit flags that represent the programmed output driver states of all DIO channels. Each bit is associated with one DIO channel. For example, bit 17 is associated with DIO channel 17. Any bit set to *one* indicates the associated channel is set to the active state; any bit set to *zero* indicates the channel is set to the inactive state.

7.6.7 SetOutputs

Function Programs the output driver states of all DIO channels that are configured for the Standard operating mode.

Opcode `0x06`

Command `(S<7:0>),(S<15:8>),(S<23:16>),(S<31:24>),(S<39:32>),(S<47:40>)`

Parameter	Function
`S[]`	Byte array containing 48 bit flags. Each flag represents the target output state for one DIO channel. For example, bit 17 is associated with DIO channel 17. Set a bit to *one* to program the associated channel to the active state, or to *zero* to program the channel to the inactive state.

Response `none`

Notes `SetOutputs` programs the output driver states of channels that are configured for the Standard operating mode; it has no effect on channels that are configured for the PWM operating mode.

In order to optimize the network response time, `SetOutputs` invokes a deferred processing function on the DIO module. When the `SetOutputs` action executes, it immediately generates a response to acknowledge receipt of the action, but the physical driver states may not change immediately. Instead, the output drivers will change to their target values within two milliseconds of executing this action.

All modified outputs will change state synchronously so that arbitrary groups of DIO channels can be used in applications that require "state machine" type behavior.

When a channel's operating mode is first changed to the PWM mode, its duty cycle is reset to zero to force the output off until the duty cycle is explicitly set by a `SetPwmRatio` action.

7.6.8 SetModes32

Function Programs the operating modes for DIO channels 0 through 31.

Opcode `0x07`

Command `(M<7:0>),(M<15:8>),(M<23:16>),(M<24:31>)`

Parameter	Function
`M[]`	Operating modes for channels 0 to 31. Each bit is associated with a channel number. Set a bit to *one* for PWM mode, or *zero* for Standard mode.

Response `none`

Notes This command is available only on 7410 firmware version 1.02 or higher.

Each of channels 0 through 31 may operate in either the Standard mode or the PWM mode. Following a reset, all channels default to the Standard operating mode.

7.6.9 GetModes32

Function Returns the operating modes for DIO channels 0 through 31.

Opcode `0x08`

Command `none`

Response `(M<7:0>),(M<15:8>),(M<23:16>),(M<24:31>)`

Parameter	Function
`M[]`	Operating modes for channels 0 to 31. Each bit is associated with a channel number. Set a bit to *one* for PWM mode, or *zero* for Standard mode.

Notes This command is available only on 7410 firmware version 1.02 or higher.

`GetModes` returns the current operating modes of DIO channels 0-31. The operating modes may be changed by design (e.g., in response to a `SetModes` action) or unexpectedly (e.g., all channel operating modes revert to the default when a module reset occurs).

7.7 Specifications

Table 20: Model 2610 Specifications

Interlock Power Daisy Chain	Voltage range	+5.8VDC to +24VDC, as required by DIO channel loads.
	Current	10A maximum, per circuit.
DIO	Channels	48
	Pull-up resistor	47K ohms, 5%.
	On state output current	100mA sink per output, max. 2A total sink (sum of all output currents), max.
	Off state output leakage	400µA @ 24V 200µA @ 12V 80µA @ 5V
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent
	Digital I/O (qty 48)	RJ-12 plug, AMP 520425-3 or equivalent
	Power daisy chain (qty 2)	AMP 770849-6 with 640706-1 pin contacts (6), or equivalent Accepts Sensoray cable assembly, part number 2600C1.
Temperature	Operating temperature	0 to 70°C operating.
Power	Power	+24V @ 120mA, max., with no output current, all output LEDs turned on.
Dimensions	Outer dimensions	5.0H x 14.3W x 2.0D inches, including DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel; additional clearance required for field wiring.

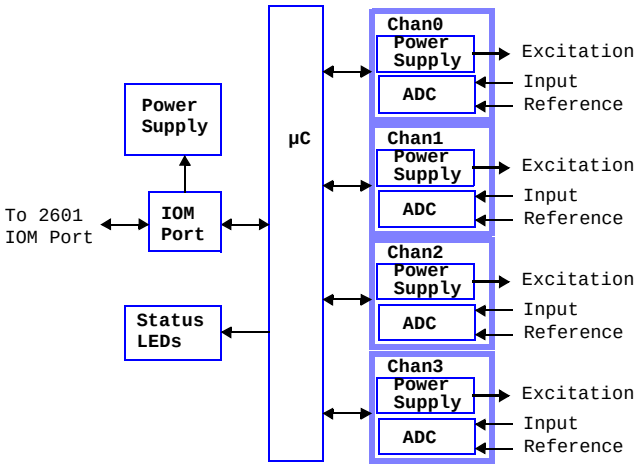
Chapter 8: Model 2612 Strain Gauge/RTD Module

8.1 Introduction

Model 2612 is a smart I/O module (IOM) that has four sensor interface channels that are optimized for precision measurement of passive resistive devices such as strain gauges, RTDs and thermistors. The 2612 may be connected to any IOM port on a model 2601 main module (MM) by means of a single Cat-5 cable.

Each channel consists of an independent, fully differential analog-to-digital converter (ADC) and a programmable excitation generator. The ADC has inputs for both the differential sensor signal and a differential reference voltage, thus enabling the ADC to eliminate gain errors via ratiometric conversion.

Figure 38: Model 2612 Block Diagram



Analog input subsystems include special circuitry that implement fully electronic calibration. Excellent long-term accuracy is achieved through the combination of high quality on-board reference standards and the total absence of mechanical trims.

The module's electrical connectors are optimized for direct connection to field wiring, in most cases eliminating the need for external field wiring termination systems. All field wiring is attached directly to the module by means of pluggable connectors so that downtime will be minimized should it become necessary to replace a module.

The entire module is automatically reset by a built-in watchdog timer if an internal fault develops. Also, the module automatically resets if the MM fails to communicate to it within a programmable time interval.

Diagnostic LEDs include Power, Heartbeat, and communication status. A self-test is performed automatically at power-on and reset, and any detected fault conditions are displayed on the status indicators.

A high-efficiency, on-board power supply produces all required operating voltages from a single 24VDC source, which is obtained from the MM via the module's IOM port. Test points are provided for all power supply voltages and analog I/O signals.

8.2 Conversion Modes

Each channel is assigned a "conversion mode," which specifies its input conversion rate. This mode specifier has two parts: the Oversample Ratio and Speed Multiplier.

8.2.1 Oversample Ratio

The oversample ratio (OSR) determines a number of operational parameters as shown in Table 21.

Table 21: Behavior for various OSR settings

OSR	RMS Noise	Conversion Rate (Hz)	First Notch (Hz)	–3dB Point (Hz)	Effective Bits
64	23µV	3515.6	28125	1696	17
128	4.5µV	1757.8	14062.5	848	20.1
256	2.8µV	878.9	7031.3	424	20.8
512	2µV	439.5	3515.6	212	21.3
1024	1.4µV	219.7	1757.8	106	21.8
2048	1.1µV	109.9	878.9	53	22.1
4096	720nV	54.9	439.5	26.5	22.7
8192	530nV	27.5	219.7	13.2	23.2
16384	350nV	13.75	109.9	6.6	23.8
32768	280nV	6.875	54.9	3.3	24.1

For example, with OSR=32768, the first null (notch) of the digital filter occurs at approximately 55 Hz. This setting provides better than 80dB rejection in the range from 49Hz to 61Hz.

Rejection of the first notch frequency and its multiples (up to 1.8MHz) exceeds 120dB. Effective noise bandwidth is a measure of how the ADC will reject wideband input noise up to the sample rate.

8.2.2 Speed Multiplier

The speed multiplier, which is used to double the input conversion rate, may be enabled or disabled for each channel. All other parameters (e.g., RMS Noise, Effective Bits) in Table 21 are unaffected by the speed multiplier. One additional cycle of latency will be inserted between input signal and output data when the speed multiplier is active.

8.3 Excitation

Each channel has a dedicated, programmable power supply which may be used to to provide excitation for a passive sensor such as a strain gauge, RTD or thermistor. Under software control, the power supply may be programmed to output one of four constant voltages: 1.25V, 2V, 3V or 5V. See section Section 8.7.3 for programming details.

The excitation supplies are designed to power isolated resistive devices, so you should not connect the excitation outputs to external power supplies as this is likely to damage the module's circuitry.

To ensure good voltage regulation by the excitation supply, it is necessary to draw the minimum output current as specified in Section 8.8. This minimum current can be drawn entirely through the serial sensor/resistor comination, or an additional shunt resistance may be placed across the supply for this purpose.

In the case of RTDs, thermistors and other single-element devices, an external fixed resistor must be supplied to form a voltage divider so that the device will produce a voltage that is a function of its resistance.

8.3.1 Self-Heating Errors

In the process of measuring a passive resistive device, an electric current must flow through the device to produce a measurable voltage. Since the device has electrical resistance, this current results in power consumption within the device that will tend to raise its temperature (i.e., "self-heating").

In the case of temperature sensors such as RTDs and thermistors, self-heating can cause a measurement error because the sensor resistance may not accurately reflect the ambient temperature.

Self-heating can be minimized by minimizing the current through the sensor and/or attaching the sensor to a heat sink. Sensor current can be minimized by connecting a large resistance in series with the sensor. Chose the largest possible series resistance that will give the required measurement resolution. Keep in mind that an additional shunt resistor may be required across the excitation supply to ensure that its minimum required load current is satisfied.

8.4 Analog Inputs

Each channel incorporates a dedicated ADC to convert a differential sensor voltage to a digital value. Analog input channels utilize fully differential signal paths to help maintain measurement accuracy in electrically noisy environments. In addition, each analog input channel is provided with a pair of reference inputs; this pair comprises a differential reference that sets the input range for the channel.

The ADC produces a value that is proportional to the difference voltage between the channel's differential input pair, within the input range defined by the reference inputs. The digitized value is the ratio of the differential input voltage to the reference voltage.

8.4.1 Reference Voltage Range

The reference voltage is a differential voltage that is applied to the Ref+ and Ref- inputs. This differential voltage (Ref+ minus Ref-) must fall within the range from 0V to +5V.

The Ref+ and Ref- absolute input voltage must be within the range from 0V to +5V with respect to Power-.

8.4.2 Input Voltage Range

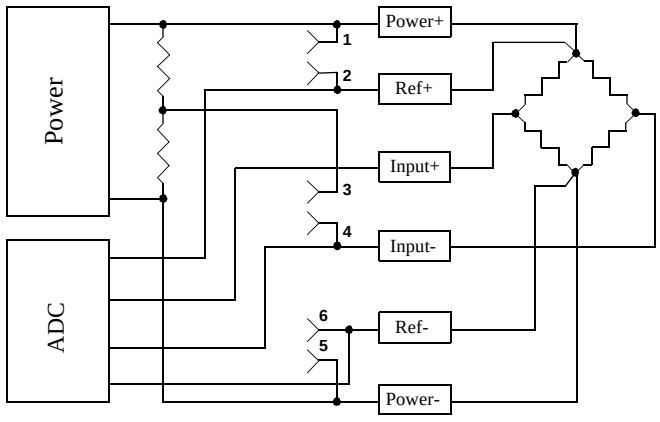
The differential input voltage applied to Input+ and Input- must not exceed half of the reference voltage. For example, with Ref+ set to 5V and Ref- set to 0V, the differential input voltage must be in the range from -2.5V (full scale negative) to +2.5V (full scale positive). Outside this range, the converter indicates an overrange or underrange condition.

To prevent circuit damage, the voltage on all input pins (with respect to the `Power-` pin) must fall within the voltage range 0V to +5V.

8.5 Hardware Configuration

Each input channel is provided with a 6-pin header (called the *input programming block*, or *IPB)* that is designed to accept hardware programming shunts. Various configuration options are programmed by installing shunts in the appropriate positions on the IPB, as described in the following sections.

Figure 39: Default Configuration: All Shunts Removed



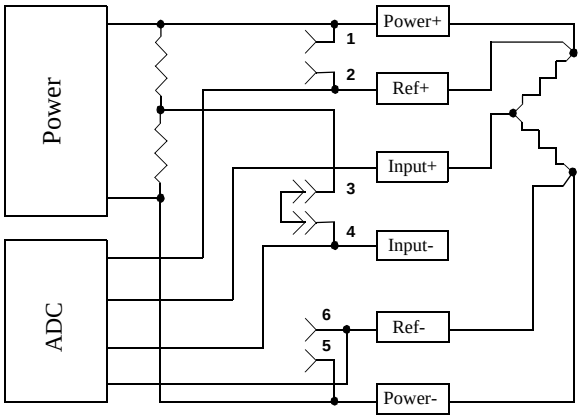
Full-bridge strain gauges may be connected as shown in Figure 39 to implement a 6-wire interface with remote excitation sensing.

8.5.1 Bridge Completion Shunt

Each channel has a 2:1 (2.5+2.5 KOhm, 0.1%) divider connected across the negative and positive power outputs. This divider serves as bridge completion resistors for half-bridge strain gauges.

Install a shunt on the IPB at pins 3-4 to enable the bridge completion network and connect the divider output to Input-. In this case, the voltage on Input+ must be in the range between Ref+ and Ref-.

Figure 40: Half-bridge Device With Internal Bridge Completion



Half-bridge strain gauges, as well as RTDs and thermistors, may be connected as shown in Figure 40 to implement remote excitation sensing. Of the two external resistors shown here, the bottom one is the sensor element and the top one is a fixed, low tempco resistor.

8.5.2 Remote Sense Shunts

Each channel must be configured for either local or remote excitation sensing. Remote sensing (Figure 39 and Figure 40) makes it possible to compensate voltage drop on the excitation supply wires. If this drop is not a concern, local sensing may be used.

To enable local sensing, install a shunt on the channel's IPB at pins 1-2 (positive) and pins 5-6 (negative). When local sensing is employed, there should be no connections to the Ref pins.

Figure 41: Full-bridge Device With Local Sensing

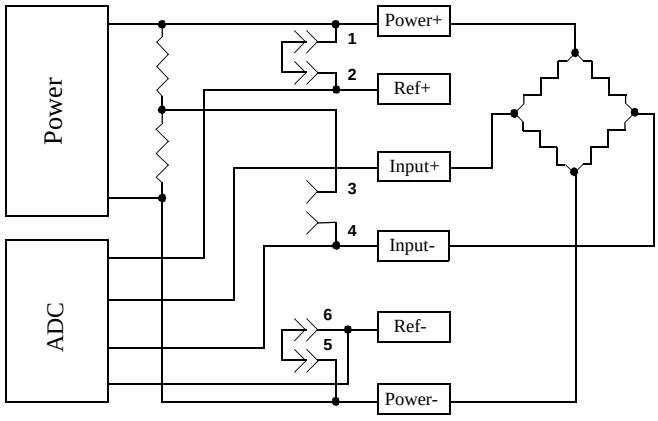
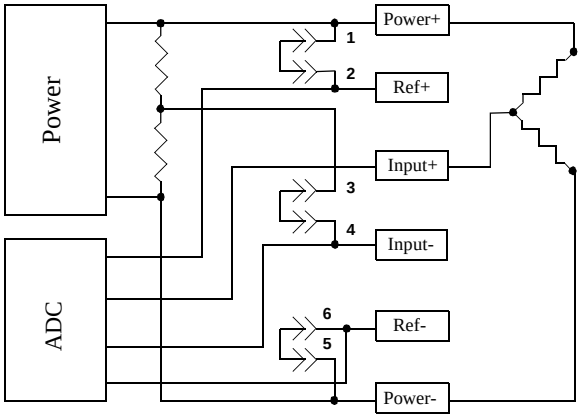


Figure 42: Half-bridge Device With Local Sensing



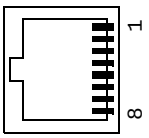
Half-bridge strain gauges, as well as RTDs and thermistors, may be connected as shown in Figure 42 if remote sensing is not needed. Of the two external resistors shown here, the bottom one is the sensor element and the top one is a fixed, low tempco resistor.

8.5.3 Field Wiring Connectors

Each channel has two dedicated, eight-pin connectors for field wiring: an RJ-45 socket and a pluggable terminal block. Connectors are labeled "CH0" through "CH3." The connector pinout is identical for all channels. Either of a channel's two connectors may be used, as best fits the system requirements.

8.5.3.1 RJ-45 Connectors

Table 22: Pinouts of RJ-45 Channel Connectors

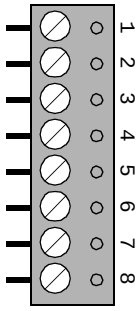
Layout (top view)	Pin	Name	Function
	1	Power-	Negative excitation output.
	2	Power+	Positive excitation output.
	3,6	Shld	This may be connected to a cable shield, but the shield conductor must be unconnected at the remote end of the cable to avoid ground loops.
	4	In+	Positive signal input.
	5	In-	Negative signal input.
	7	Vref-	Negative reference input.
	8	Vref+	Positive reference input.

8.5.3.2 Terminal Blocks

Pin 1 of a TB may be physically located by orienting the board so that the TB's silkscreen label reads from left to right; pin 1

is the left-most pin on the TB. TB pins are numbered in sequential order, beginning with pin 1.

Table 23: Pinouts of Channel Terminal Blocks

Layout (top view)	Pin	Name	Function
	1,8	Shld	This may be connected to a cable shield, but the shield conductor must be unconnected at the remote end of the cable in order to avoid ground loops.
	2	Power+	Positive power output.
	3	Vref-	Negative reference input.
	4	In+	Positive signal input.
	5	In-	Negative signal input.
	6	Vref+	Positive reference input.
	7	Power-	Negative power output.

8.6 Reset

Upon module reset, all power outputs switch to 1.25 volts out. A 2612 module will experience a module reset in response to any of the following conditions:

- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.
- Communication time-out, which will occur if the client fails to communicate with the 2612 module within a programmable communication watchdog interval.

8.7 Actions

This section describes the programmable actions that are supported exclusively by 2612 modules. These actions may be employed, in any sequence or combination, in MCmd action lists. In addition to the actions listed here, the 2612 module also supports all of the common actions that are universally supported by all IOMs.

Table 24: Summary of Model 2612 Actions

Command	Opcode	Function
`SetMode`	`0x00`	Program input mode.
`GetInputs`	`0x01`	Return ADC values.
`SetVoltages`	`0x02`	Select output voltages.
`ReadEeprom`	`0x0D`	Return data from EEPROM
`WriteEeprom`	`0x0E`	Write data to EEPROM

Note: Model 2612 has a maximum MRsp size of 26 bytes.

8.7.1 SetMode

Function Programs the conversion mode of one analog input channel.

Opcode `0x00`

Command `(Chan),(Mode<7:0>),(Mode<15:8>),(Mode<23:16>),(Mode<31:24>)`

Parameter	Function
`Chan`	Analog output channel to be programmed. Values may range from 0 to 3, inclusive. The six high-order bits are reserved and should be set to zero.
`Mode`	32-bit value that specifies the conversion mode for the target channel.

`Mode:` bits 31 to 20 specify the oversample ratio (OSR), as shown in the following table. See Section 8.2.1 for details. Bit 19 specifies the speed multiplier mode. Bits 18 to 0 must be set to zero.

Value	OSR
`0xA01`	`64`
`0xA02`	`128`
`0xA03`	`256`
`0xA04`	`512`
`0xA05`	`1024`
`0xA06`	`2048`
`0xA07`	`4096`
`0xA08`	`8192`
`0xA09`	`16384`
`0xA0F`	`32768`

Response `none`

8.7.2 GetInputs

Function Returns the digitized values of all analog input channels.

Opcode `0x01`

Command `none`

Response
```
(s[0][0]),(s[0][1]),
(s[0][2]),(s[0][3]),
(s[1][0]),(s[1][1]),
(s[1][2]),(s[1][3]),
(s[2][0]),(s[2][1]),
(s[2][2]),(s[2][3]),
(s[3][0]),(s[3][1]),
```

```
(s[3][2]),(s[3][3]),
(n[0]),(n[1]),(n[2]),(n[3]),
```

Parameter	Function
s[]	Array of four 32-bit status/data values, LSB first, associated with analog input channels 0 (first) to 3 (last).
n[]	Array of four 8-bit sample numbers, LSB first, associated with analog input channels 0 (first) to 3 (last).

`s:` Each of these four 32-bit status/data values has the following structure:

31	30	29	28 . . . 0
0	0	SN	msb Data lsb

The three most significant bits of a status/data value indicates the data polarity and conversion state. The remaining 29 bits are the conversion result.

Bits 31 and 30 are always LOW.

`SN:` bit 29 is the conversion result sign indicator. If VIN is >0, this bit is HIGH. If VIN is <0, this bit is LOW.

`Data:` bit 28 is the most significant bit (msb) of the result. This bit, in conjunction with Bit 29, also indicates an underrange or overrange condition. If bits 29 and 28 are both HIGH, the channel's differential input voltage is above +FS (full scale). If bits 29 and 28 are both LOW, the input voltage is below –FS.

8.7.3 SetVoltages

Function Programs the output power voltages for all four channels.

Opcode `0x02`

Command `(Volts<7:0>)`

Parameter	Function
Volts	Four 2-bit fields that specify the output voltages. Each bit field corresponds to one channel.

`Volts` is an 8-bit value, passed LSB first, that is composed of bit fields that specify the output voltages. It has the following structure:

7	6	5	4	3	2	1	0
Out3		Out2		Out1		Out0	

Bit Field Value	Output Voltage
00	1.25V
01	2V
10	3V
11	5V

Response `none`

Notes Once set, the output voltages apply until they are changed by another `SetVoltanges` action or a module reset occurs.

After a module reset, all power outputs default to 1.25V output.

8.7.4 ReadEeprom

Function Returns a byte value from the module's non-volatile memory.

Opcode `0x0D`

Command `(Adrs)`

Parameter	Function
Adrs	Memory address that is to be read from.

Response `(Data)`

Parameter	Function
Data	Byte value retrieved from the EEPROM.

Notes This function is used by the middleware to store calibration data for the module. Refer to the distribution media for details.

8.7.5 WriteEeprom

Function Stores a byte value in the module's non-volatile memory.

Opcode `0x0E`

Command `(Adrs),(Data)`

Parameter	Function
`Adrs`	Memory address that is to be written to.
`Data`	Byte value that is to be stored.

Response `none`

Notes This function is used by the middleware to store calibration data for the module. Refer to the distribution media for details.

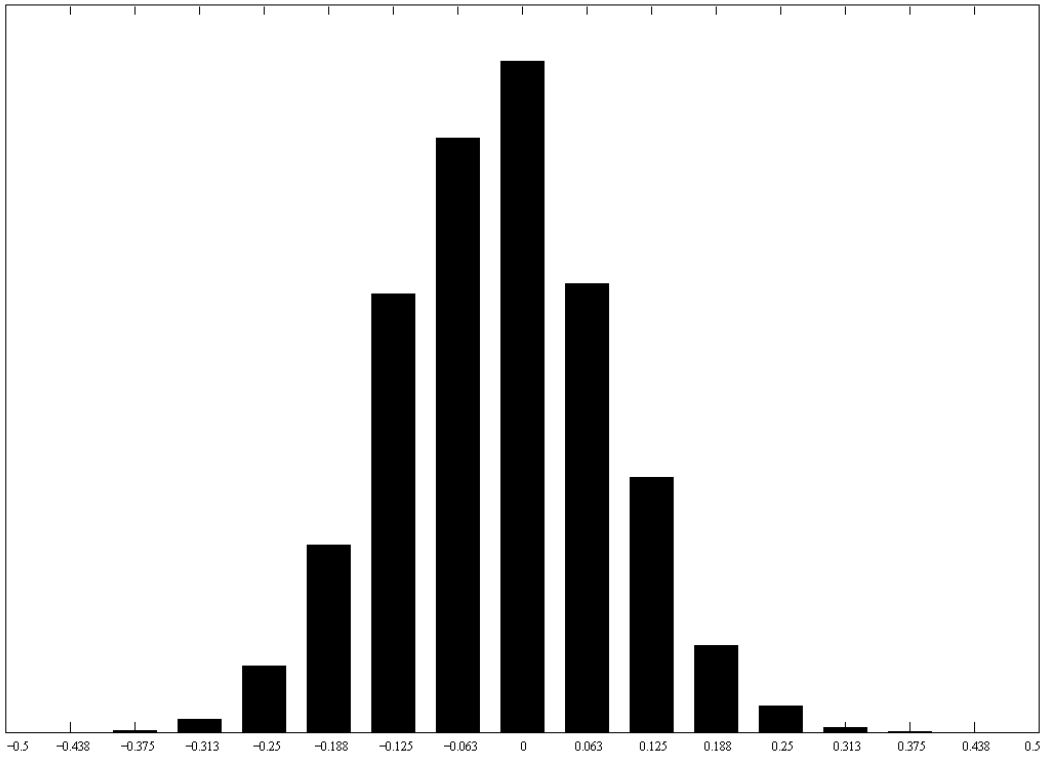
Due to the lengthy time required to write a byte to the EEPROM, this action should be invoked only once per command packet.

8.8 Specifications

Table 25: Model 2612 Specifications

Analog Inputs	Resolution	24 bits
	Input channels	4
	Conversion rate	6.875 to 3500 samples/second for each channel.
	Input range	±1/2 of reference.
	Input impedance	110 kOhm, minimum, measured from either sensor input to reference voltage.
Power Outputs	Output channels	4
	Output power	1.25V, 2V, 3V or 5V, ±10%@ 1 to 25 mA for each channel.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent
	Analog inputs (qty 4)	RJ-45 plug, AMP 554169 or equivalent
	Analog inputs (qty 4)	(*)Pluggable TB, 8-pin, RIA 31166110 or equivalent
Temperature	Operating range	0 to 70°C
Power	Input power	+24V ±5% @ 60mA.
Dimensions	Outer dimensions	5.0H x 5.0W x 1.8D inches, including DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel; additional clearance required for field wiring.

Figure 43: Typical noise distribution, ppm (13.75 samples per second)



Chapter 9: Model 2620 Counter Module

9.1 Introduction

The model 2620 is a smart I/O module (IOM) that has four identical 32-bit programmable counter channels, numbered 0 to 3.

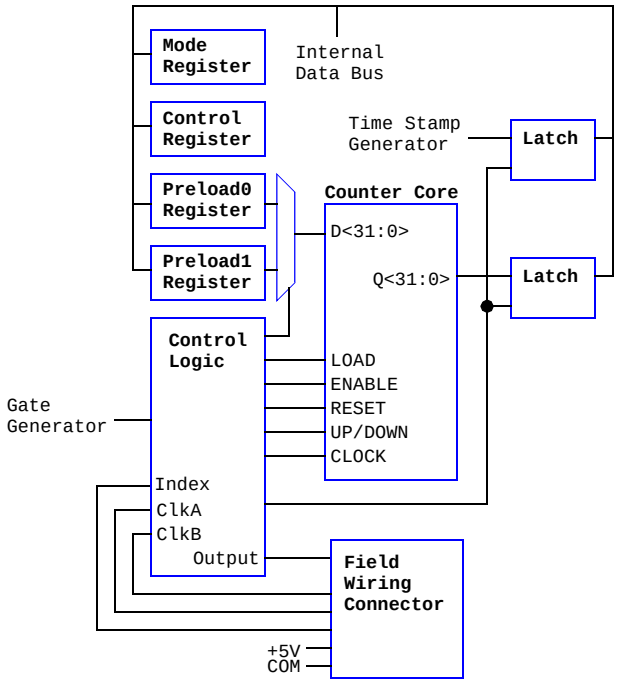
By means of a single Cat-5 cable, the 2620 may be connected to any IOM port on a model 2601 main module (MM). The onboard microcontroller communicates with the MM by means of an optically isolated, asynchronous serial interface.

Counter channels are designed to support these applications:

- Incremental encoder interface, with quadrature decoder and clock multiplier (x1, x2 and x4).
- Pulse generator, with software, hardware or periodic interval triggering.
- PWM generator.
- Pulse width measurement.
- Period measurement.
- Frequency measurement.

As shown in Figure 44, each channel has two clock inputs ("A" and "B"), one index input, one output, and a fused 5 Volt DC output for supplying operating power to an incremental encoder or other low-power device. The channel infrastructure consists of a 32-bit counter core, two 32-bit preload registers, 32-bit latch register, mode register and control register.

Figure 44: Counter Channel Block Diagram



Each channel is independently programmable for clock type (quadrature x4/x2/x1 or single phase) and index function. The index input can be programmed to behave in various ways, including setting the counter core to a specific value and/or latching counts upon index edge events.

The maximum count rate for each counter channel is 10 MHz. Single-phase clock sources having a 50 percent duty cycle can be counted at this rate. For other duty cycles, the maximum count rated must be derated based on the clock's pulse width. Quadrature clocks up to 2.5 MHz are supported, derated for non-symmetrical phasing or deviations from 50 percent duty cycles.

A free-running 16-bit counter serves as a shared time stamp generator for all channels. The time stamp resolution, which is common to all channels, may be programmed to 1, 10, 100 or 1000 microseconds. Each channel synchronously latches the time stamp counts whenever its counter core is transferred to its output data latch.

The free-running counter also generates pulses that may be used as gate generators in frequency counting applications. A clock divisor, which is common to all channels, may be programmed to to set the pulse width to any value from one to 16,382 milliseconds, in one-millisecond increments.

All clock and index inputs employ fully differential RS-422 line receivers for superior noise immunity. The line receivers include termination resistors for enhanced noise rejection in noisy environments. Although clock and index inputs are optimized for RS-422 differential pairs, they are also TTL and 5V CMOS compatible.

9.1.1 Diagnostic LEDs

- **LNK** - Lights when the 2620 has established a communication link with the 2601 main module.
- **HBT** - "Heartbeat" indicator flashes to show that the on-board CPU is healthy.
- **ACT** - Lights briefly when an Ethernet client command is executed.
- **RST** - Lights briefly when the module is reset.
- **Channel Activity LEDs** - These four LEDs indicate the presence of a dynamic signal at the counter core's clock input. Each channel has one activity LED, located near to its terminal block. Each LED flashes at a constant rate while the associated core's clock signal is changing.

9.2 Hardware Configuration

The 2620 module is configured by connecting cables and devices as described in this section.

9.2.1 IOM Network Connector

A single RJ-45 connector, J1, is used to connect the 2620 module to its client. This should be mated to a standard UTP (unshielded twisted pair) Category-5 cable.

The other end of this cable will be connected to one of the sixteen IOM ports on a model 2601 module.

9.2.2 Field Wiring Terminal Blocks

Each counter channel is associated with a removable 10-pin screw terminal block (TB) on the 2620 circuit board. The TB's symbolic pin names are silk-screened onto the circuit board to simplify the connection of field wiring. TB pin functions are described in Table 26.

Table 26: Pinouts of Terminal Blocks

PWB Label	Pin	Function
A+	1	Clock "A" positive input (RS-422, TTL/CMOS). If you are using the "A" clock, connect this to the clock "A" input signal. Otherwise, you may leave disconnected, or connect to +5V or GND.
A-	2	Clock "A" negative input (RS-422). Leave disconnected for single-ended (TTL/CMOS) signal sources or if you are not using the "A" clock.
B+	3	Clock "B" positive input (RS-422, TTL/CMOS). If you are using the "B" clock, connect this to the clock input signal. Otherwise, you may leave disconnected, or connect to +5V or GND.
B-	4	Clock "B" negative input (RS-422). Leave disconnected for single-ended (TTL/CMOS) signal sources or if you are not using the "B" clock.
X+	5	Index positive input (RS-422, TTL/CMOS). If you are using the index, connect this to the index input signal. Otherwise, you may leave disconnected, or connect to +5V or GND.
X-	6	Index negative input (RS-422). Leave disconnected for single-ended (TTL/CMOS) signal sources or if you are not using the counter index.
ENAB	7	Count Enable input (active high, built-in pull-up, TTL/CMOS). When negated, this signal inhibits counting unconditionally. When asserted (or left disconnected), counting is enabled according to permissives dictated by the channel's configuration.
OUT	8	Counter channel's output (TTL/CMOS). The behavior and polarity of this signal is programmable.
GND	9	5V power supply return for pin 10. This is the ground reference for all logic signals passing into or out of the TB.
+5V	10	+5VDC power. This can be used to power external devices, such as incremental encoders or tachometers. A 500mA socketed fuse protects against short circuits. The total 5V current for all four counter channels is limited to 500mA under normal operating conditions.

Refer to Section 9.5 for tips on how to connect the channel's input and output signals for various applications.

9.3 Reset

The 2620 assumes the following condition upon module reset:

- All counter operating modes are reset to zero.
- All preload registers are reset to zero.
- The common control register is reset to zero.

A 2620 module will experience a module reset in response to any of the following conditions:

- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.
- Communication time-out, which will occur if the client fails to communicate with the DIO module within a programmable communication watchdog interval.

9.4 Actions

This section describes the programmable actions that are supported exclusively by 2620 modules. These actions may be employed, in any sequence or combination, in MCmd action lists.

In addition to the actions listed here, the model 2620 also supports all of the common actions that are universally supported by all IOM types.

Table 27: Summary of 2620 Module Actions - Base Opcodes

Command	Opcode	Function
SoftTrigger	0x00	Triggers counter latch and/or load.
SetMode	0x01	Program the channel operating mode.
GetStatus	0x02	Return the channel's status.
SetPreload0	0x03	Store a value in the preload0 register.
SetPreload1	0x04	Store a value in the preload1 register.
GetCounts	0x05	Return the channel's count.

Table 27: Summary of 2620 Module Actions - Base Opcodes

Command	Opcode	Function
GetCountsTs	0x06	Return the channel's count with time stamp.
SetCommonControl	0x0F	Program the common control register.

Unlike most other IOM types in the 2600 family, nearly all of the 2620-specific actions employ a special opcode format that designates the counter channel to which the action applies. The opcode byte that is issued to the 2620 is formed by adding the counter channel number, multiplied by sixteen, to a base opcode. For example, opcode 0x31 would be used to apply a SetMode action (base opcode 0x01) to counter channel 3.

Note: Model 2620 has a maximum MRsp size of twenty bytes.

9.4.1 SetMode

Function Programs a counter channel's operating mode.

Opcode `0x00 + Chan * 16`

Command `(Mode<7:0>),(Mode<15:8>)`

Parameter	Function
Mode	Collection of bit fields that specify the counter's functional attributes.

`Mode` is a 16-bit value, passed LSB first, that is composed of bit fields that specify the functional attributes of the counter channel.

Figure 45: Counter Mode MSB

15	14	13	12	11	10	9	8
RUN	OM1	OM0	XP	PL1	PL0	LAT	CET

RUN: Enable channel operations.

0 - (default upon module reset) Halt channel, force core to zero (including bit 32), force status bits to their default states, reset trigger latches. Preload and latch registers are not modified. After writing to the mode register with this bit cleared, it is necessary to write to it again with this bit set to start the channel running.

1 - Run or continue to run in the specified mode.

OM: Output pin's mode (2-bit field):

0 - Counter bit 31.

1 - Counter bit 32 (toggles at zero counts).

2 - Active when counts are zero.

3 - Active during counter under/overflow.

XP: Index input polarity:

0 - Active high.

1 - Active low.

PL: Preload trigger (2-bit field):

0 - Preload on soft trigger only.

1 - Preload on index leading edge or soft trigger.

2 - Preload on zero counts reached or soft trigger.

3 - Reserved.

LAT: Latch trigger:

0 - Latch on soft trigger only.

1 - Latch on index leading edge or soft trigger.

CET: Count enable trigger:

0 - Enable upon configuration (no trig needed).

1 - Enable on index leading edge.

Figure 46: Counter Mode LSB

7	6	5	4	3	2	1	0
OP	M2	M1	M0	CD1	CD0	PLM	XC

OP: Output pin's polarity:

0 - Active high.

1 - Active low.

M: Mode (3-bit field). Modes 0-3 use quadrature-encoded two-phase clock, modes 4-6 use single-phase clock, and mode 7 uses the internal clock:

0 - quad x1, clock on rising A.

1 - quad x1, clock on falling A.

2 - quad x2, clock on either edge of A.

3 - quad x4, clock on either edge of A or B.

4 - mono, clock on rising A, B controls count direction.

5 - mono, clock on falling A, B controls count direction.

6 - mono, clock on either edge A, B controls count direction.

7 - internal clock (10MHz), A is the gate (enables counting while asserted), B controls count direction.

CD: Count disable trigger:

0 - Never disabled by any trigger.

1 - Disable on index trailing edge (if enabled).

2 - Disable when zero counts reached.

PLM: Select preload register:

0 - Only preload register 0.

1 - Use both preload registers.

XC: Index source:

0 - External Index pin.

1 - Internal free-running gate generator.

Response none

Notes After the mode is changed to its target value, the counter core is automatically reset to zero. The channel's preload registers are not changed by this action.

The target channel's mode changes immediately to the specified mode when the SetMode action executes. As a result, the client may invoke another action that will modify the counter core (i.e., SetCounter) immediately after a SetMode action.

9.4.2 SetCommonControl

Function Programs the gate time divisor and time stamp resolution common to all counter channels.

Opcode 0x0F

Command (Ctrl<7:0>),(Ctrl<15:8>)

Parameter	Function
Ctrl	Common counter configuration values.

Ctrl is a 16-bit value, passed LSB first, that is composed of two bit fields that specify the gate time divisor (G) and time stamp resolution (TS). These values are common to all counter channels.

Figure 47: Ctrl MSB

15	14	13	12	11	10	9	8
G13	G12	G11	G10	G9	G8	G7	G6

Figure 48: Ctrl LSB

7	6	5	4	3	2	1	0
G5	G4	G3	G2	G1	G0	TS1	TS0

G: Gate time divisor (14-bit field):

2 to 16,383 - Specifies the gate time divisor that is common to all channels. The gate interval is set to 2*(G+1) milliseconds; this is the period of the square-wave output produced by the gate generator.

TS: Time stamp resolution (2-bit field):

0 - 1 microsecond.

1 - 10 microseconds.

2 - 100 microseconds.

3 - 1 millisecond.

Response none

Notes This action does not require a counter channel number to be attached to the opcode byte.

When using the time gate in frequency counter applications, set the gate time divisor so that the gate generator period equals the desired gate time. For example, set the gate time divisor to 499 to generate a one-second gate period.

The time stamp resolution should be programmed so as to meet the needs of the channel having the highest resolution; all other channels must then use the same time stamp resolution.

The application must trade-off time stamp resolution for longer time stamp intervals, as shown in the following table:

Resolution	Maximum Interval
1 µs	65.536 ms
10 µs	655.36 ms
100 µs	6.5536 seconds
1 ms	65.536 seconds

9.4.3 SoftTrigger

Function Triggers counter latch and/or load.

Opcode 0x01 + Chan * 16

Command (Flags)

Parameter	Function
Flags	Bits that specify the soft trigger actions that are being invoked.

Flags contains bits that specify the action or actions that are being triggered.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	LAT	LOD

LAT: Transfer counter core to data latch.

LOD: Force counter load from a preload register. If only one preload register is in use, Preload0 is always used. If both preload registers are in use, the active preload register will be used and the register pointers will be toggled so that the other preload register will be used next time.

Response none

9.4.4 GetStatus

Function Returns a counter channel's status information.

Opcode `0x02 + Chan * 16`

Command `none`

Response `(Status<15:8>),(Status<7:0>)`

Figure 49: Counter Status MSB

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

Figure 50: Counter Status LSB

7	6	5	4	3	2	1	0
QDE	LAT	GO	LOD	EXT	UF	OF	ZER

QDE: Quadrature decoder error. This bit is automatically reset by this GetStatus action.

LAT: Counter core was latched. This bit is automatically reset by a ReadLatch action.

GO: Counter was enabled by a trigger.

LOD: Counter was pre-loaded. This bit is automatically reset by this GetStatus action.

EXT: Counter extension bit 32.

UF: Counter underflowed. This bit is automatically reset by this GetStatus action.

OF: Counter overflowed. This bit is automatically reset by this GetStatus action.

ZER: Counter value is now zero.

All other bits are reserved for future use.

9.4.5 SetPreload0

Function Programs a counter channel's Preload0 register to the specified value.

Opcode `0x03 + Chan * 16`

Command `(Counts[0]),(Counts[1]),`
`(Counts[2]),(Counts[3])`

Parameter	Function
`Counts`	The 32-bit value that is to be stored in the counter channel's preload register. Bytes are ordered least-significant byte first.

Response `none`

Notes The preload register is loaded immediately when this action executes. The counter core is not affected; it will continue to operate in its programmed mode.

9.4.6 SetPreload1

Function Programs a counter channel's Preload1 register to the specified value.

Opcode `0x04 + Chan * 16`

Command `(Counts[0]),(Counts[1]),`
`(Counts[2]),(Counts[3])`

Parameter	Function
`Counts`	The 32-bit value that is to be stored in the counter channel's preload register. Bytes are ordered least-significant byte first.

Response `none`

Notes The preload register is loaded immediately when this action executes. The counter core is not affected; it will continue to operate in its programmed mode.

9.4.7 GetCounts

Function Returns a channel's count value.

Opcode `0x05 + Chan * 16`

Command `none`

Response `(Counts[0]),(Counts[1]),`
`(Counts[2]),(Counts[3])`

Parameter	Function
`Counts`	Byte array containing the 32-bit value that is contained in the counter's output latch when the action executes. Bytes are ordered least-significant byte first.

Notes Counting is not interrupted by this action.

9.4.8 GetCountsTs

Function Returns a channel's count value and time stamp.

Opcode `0x06 + Chan * 16`

Command `none`

Response `(Counts[0]),(Counts[1]),`
`(Counts[2]),(Counts[3]),`
`(Tstamp[0]),(Tstamp[1])`

Parameter	Function
`Counts`	Byte array containing the 32-bit value that is contained in the counter's output latch when the action executes. Bytes are ordered least-significant byte first.
`Tstamp`	Byte array contains the 16-bit value that is contained in the counter's time stamp latch when the action executes. Bytes are ordered least-significant byte first.

Notes Counting is not interrupted by this action.

9.5 Applications

This section explains how to configure the counter channels for various common applications.

9.5.1 Incremental Encoder Interface

This configures a counter channel so that it will interface to either a quadrature-encoded or a single-phase clock source, with optional index input.

Quadrature-encoded sources are typically used to track the physical position of a mechanical process that can dither or travel in two directions, while single-phase sources are for mechanical processes that travel in only one direction (and where dithering is either not possible or is not a problem).

The counter core can be forced to a reference value by triggering a transfer from the Preload0 register. The index input may be used as a hardware trigger by connecting it to an external "home position" sensor (e.g., hall-effect sensor, encoder index output, etc.). If enabled, an index transition will preload the counter core. The core can always be preloaded by invoking a SoftTrigger action with the LOD bit asserted.

Set the Mode register as follows:

Field	Value	Notes
OM	X	
XP	*	Set as appropriate if using external index to trigger preloads.
PL	*	Set to 1 to enable preloads upon external index, or to 0 for soft preloads only.
LAT	0	Latch on soft trigger.
CET	0	
OP	X	
M	*	Single-phase clock: set to 4, 5 or 6. Quadrature clock: set to 0 or 1 (x1 multiplier), 2 (x2 multiplier) or 3 (x4 multiplier).
CD	0	
PLM	0	Use only the Preload0 register. Preload0 must be programmed to the value that will be preloaded upon preload trigger.
XC	0	External index signal.

Connect the channel as follows:

Signal	Connect To
ClkA	Clock input for single-phase clocks. For quadrature clocks, this is the phase-A input.
ClkB	Phase-B input for quadrature clocks. Tie to either GND or +5V (same as open) to set count direction.
Index	Hardware trigger for preloads. Leave disconnected if hardware-triggered preloads are not required.
Output	NC

9.5.2 Pulse Generator

This configures a counter channel so that it will generate a single output pulse in response to a hardware or software trigger (i.e., a "one-shot"). The duration of the output pulse is determined by the value stored in the Preload0 register.

Set the Mode register as follows:

Field	Value	Notes
OM	2	Output active only during pulse.
XP	*	Set as appropriate if using hardware trigger.
PL	*	Set to 1 to enable preloads upon external index, or to 0 for soft preloads only.
LAT	0	Latch on soft trigger.
CET	0	
OP	*	Set output polarity as required.
M	7	Use internal clock.
CD	2	
PLM	0	Use only the Preload0 register. Preload0 must be programmed to set the pulse duration.
XC	0	External index signal.

Connect the channel as follows:

Signal	Connect To
ClkA	NC
ClkB	GND (count down)
Index	Hardware trigger for starting an output pulse. Leave disconnected if hardware-triggering is not required
Output	Pulse output signal.

9.5.3 PWM Generator

This configures a counter channel so that it will toggle its output signal at periodic intervals, with programmable period and duty cycle.

Set the Mode register as follows:

Field	Value	Notes
OM	1	Toggle output when core transitions to zero counts.
XP	X	
PL	2	Preload on zero counts reached.
LAT	0	Latch on soft trigger.
CET	0	
OP	*	Set output polarity as required.
M	7	Use internal clock.
CD	0	
PLM	1	Use both preload registers. Preload0/Preload1 must be programmed with the on/off output times, respectively, for the desired period and duty cycle.
XC	0	External index signal.

Connect the channel as follows:

Signal	Connect To
ClkA	NC
ClkB	GND (count down)
Index	NC
Output	PWM output signal.

9.5.4 Pulse Width Measurement

This configures a counter channel so that it will measure the width of pulses applied to the index input. When a measurement is completed, the result is latched and the next measurement begins automatically. The most recently acquired measurement value may be read from the latch at any time.

Set the Mode register as follows:

Field	Value	Notes
OM	X	
XP	0	
PL	1	
LAT	1	Latch on index leading edge.
CET	1	Start counting on index leading edge.
OP	X	
M	7	Use internal clock.
CD	1	Stop counting on index trailing edge.
PLM	0	Use only the Preload0 register. Preload0 must be programmed to zero.
XC	0	External index signal.

Connect the channel as follows:

Signal	Connect To
ClkA	NC
ClkB	+5V or open (count up)
Index	Pulse input signal to be measured.
Output	NC

9.5.5 Period Measurement

This configures a counter channel so that it will measure the period of a squarewave applied to the index input. When a measurement is completed, the result is latched and the next measurement begins automatically. The most recently acquired measurement value may be read from the latch at any time.

Set the Mode register as follows:

Field	Value	Notes
OM	X	
XP	*	Doesn't matter, unless one signal edge has more jitter than the other edge.
PL	1	Preload on index leading edge.
LAT	1	Latch on index leading edge.
CET	1	Enable on first index leading edge.
OP	X	
M	7	Use internal clock.
CD	0	
PLM	0	Use one preload register, which must contain 0.
XC	0	External index signal.

Connect the channel as follows:

Signal	Connect To
ClkA	NC
ClkB	+5V or open (count up)
Index	Periodic input signal to be measured.
Output	NC

9.5.6 Frequency Measurement

This configures a counter channel so that it will measure the frequency of a periodic signal. In this example, the count gate is generated by the internal time gate generator, which is common to all counter channels. It is assumed that the gate

time has been programmed so that the appropriate gate period will be generated.

If the internal time gate generator can't be used for some reason (e.g., irregular gate time or different channels must operate with different gate times) then a second counter channel must be configured to serve as a gate generator for the frequency counting channel.

The frequency counting channel is configured as a "gated pulse counter." It counts pulses from an external signal while the counting gate is active. When a measurement is completed, the result is latched and the next measurement begins automatically. The most recently acquired measurement value may be read from the latch at any time

Set the Mode register as follows:

Field	Value	Notes
OM	X	
XP	0	Index (gate) polarity is active high.
PL	1	Preload on index (gate) leading edge.
LAT	1	Latch on index (gate) leading edge.
CET	1	Enable counting on index (gate) leading edge.
OP	X	
M	4	Count external, single-phase clock.
CD	0	
PLM	0	Use only Preload0, which must contain 0.
XC	1	Index connected to internal time gate generator.

Connect the channel as follows:

Signal	Connect To
ClkA	Input signal to be measured.
ClkB	+5V or open (count up)
Index	NC
Output	NC

9.6 Specifications

Table 28: Model 2620 Specifications

Counters	Resolution	32 bits
	Counter channels	4
	Count rate	10 MHz, maximum.
	Input clock	Maximums (derate for deviations from 50% duty cycle): 2.5 MHz @ quadrature x4. 5 MHz @ quadrature/mono x2. 10 MHZ @ quadrature/mono x1.
	Internal clock	10 MHz.
Input/Output Signals	Clock input	RS-422 differential, TTL/5VCMOS compatible, ±7V CMV maximum.
	Index input	RS-422 differential, TTL/5VCMOS compatible, ±7V CMV maximum..
	Count Enable input	TTL/5VCMOS compatible.
	Counter output	TTL/5VCMOS compatible.
Device Power Outputs	Output voltage	5VDC ±5%.
	Output current	400mA, max., total for all channels.
	Fuse	Socketed 500mA fuse per channel.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent
	Counter I/O (qty 4)	(*) Pluggable TB, 10-pin, RIA 31166110 or equivalent
Temperature	Operating range	0 to 70°C
Power	Input power	+24V ±5% @ 30mA, not including external device power.
Dimensions	Outer dimensions	5.0H x 4.1W x 2.0D inches, including DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel; additional clearance required for field wiring.

Chapter 10: Model 2650 Relay Module

10.1 Introduction

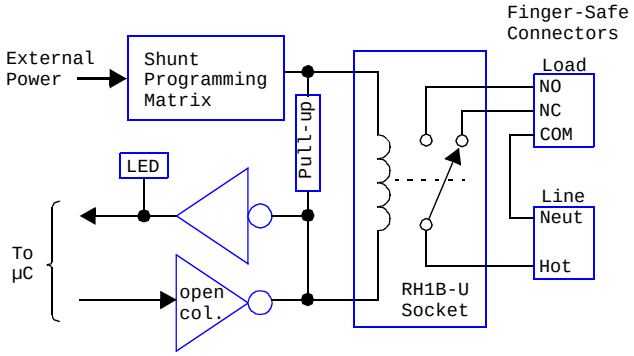
Models 2650 and 2650-0 are smart I/O modules (IOM) that control up to eight mechanical relays. Each relay switches single-phase power—up to 10A, 250V—through normally open and/or normally closed contacts. Relay channels are referenced by channel number, ranging from 0 to 7.

The module's microcontroller, which provides I/O services to a remote client, communicates with the client by means of a standard Category-5 UTP cable over an optically isolated, asynchronous serial interface.

As shown in Figure 51, each relay channel includes:

- Finger-safe socket for a standard RH1B-U relay.
- Active-low, open-collector coil driver with induced-EMF suppression diode for inductive load protection.
- Coil driver monitor circuit. This feature, which is provided primarily for diagnostics support, enables the driver's physical output state to be acquired by the client.
- Indicator LED that lights when the coil driver is energized.
- Two finger-safe connectors per channel, one for the line connection and one for the load. Loads may have either a normally open or normally closed connection to line power.
- Shunt programming matrix for implementing interlocked fail-safe operation.

Figure 51: Relay Channel Block Diagram



In addition, the following components are supplied for each relay channel on model 2650 (not supplied with model 2650-0):

- Single-pole, double throw relay.
- Relay hold-down clip.

10.2 Hardware Configuration

The 2650 relay module (RLY) is configured by installing programming shunts and connecting various cables to the module as described in this section.

10.2.1 IOM Network Connector

A single RJ-45 connector, J9, is used to connect the 2650 module to its client. This should be mated to a standard UTP (unshielded twisted pair) Category-5 cable.

The other end of this cable will be connected to one of the sixteen IOM ports on a model 2601 module.

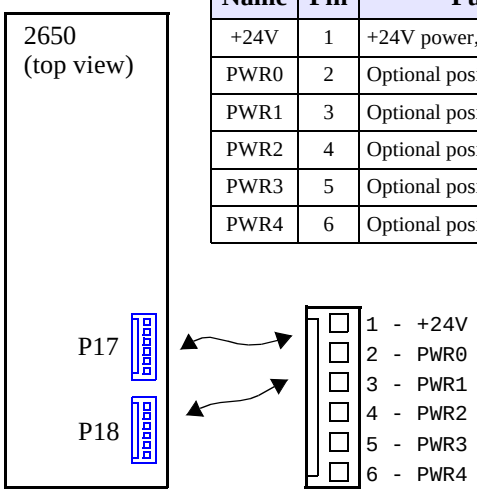
10.2.2 Interlock Power Connectors

Each relay channel is powered from an external 24VDC power source. At least one external 24VDC power source is required to make the relay channels functional. Up to six independent, external power sources are supported; as many as five of these sources may be left open or wired to arbitrary, positive DC voltage sources up to 24VDC, but any source that will power relays must supply 24VDC.

Connectors P17 and P18 supply power to relay channels from external power sources. All external power sources must supply positive DC voltages with respect to system ground. The current return of each external power source must be connected to the system ground.

As shown in Figure 52, connectors P17 and P18 are identical in function and pinout. The external power sources may be connected to either P17 or P18. If other IOMs require access to these external power sources, the unused connector may be used to daisy-chain power out to other IOMs. In this manner, power can be distributed to any arbitrary number of external IOMs without the use of dedicated power distribution terminal blocks.

Figure 52: Pinouts of Connectors P17 and P18



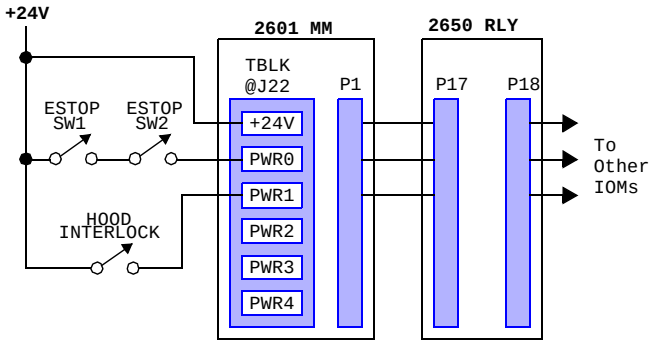
Name	Pin	Function
+24V	1	+24V power, always on.
PWR0	2	Optional positive DC power #0.
PWR1	3	Optional positive DC power #1.
PWR2	4	Optional positive DC power #2.
PWR3	5	Optional positive DC power #3.
PWR4	6	Optional positive DC power #4.

The external power sources need not be derived from interlock contacts, but by employing interlocked power sources, the system interlock cabling can be minimized. This feature is especially useful for guaranteeing the fail-safe shutdown of

select relay channels in the event of a critical condition such as emergency stop (ESTOP) activation, open safety hood, etc.

Figure 35 illustrates a system in which PWR0 is connected to multiple system emergency stop contacts, and PWR1 is routed through a safety hood interlock contact. Note that all of the external power sources are daisy-chained from module to module. Any relay that is configured to use PWR0 will de-energize when either ESTOP contact is opened, and any relay that is configured to use PWR1 will de-energize when the hood opens. All other relays use the +24V "always on" power source and thus are unaffected by interlock contacts. PWR2, PWR3 and PWR4 are not used in this application.

Figure 53: Wiring Example with Interlock Contacts



Use Sensoray cable assembly, part number 2600C1, to connect P17 or P18 to the power daisy chain.

10.2.3 Relay Power Selection Shunt

Each relay channel may be independently operated from any of the six external power sources (that are connected to P17 and P18) by installing the appropriate programming shunt on the module.

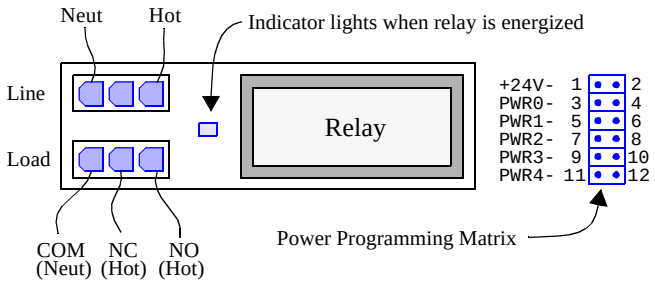
A shunt receptacle matrix is provided for each channel. Each matrix has six shunt positions, corresponding to the six external power sources. As shown in Table 17, a 2mm programming shunt (supplied with the 2650) must be installed at the position corresponding to the relay channel's target power source. Only one shunt should be installed per matrix.

Table 29: Interlock Power Programming Matrix

PWB Label	Shunt Pins	Selected Power Source
+24V	1-2	+24V power, always on (default)
0	3-4	Optional positive DC power #0.
1	5-6	Optional positive DC power #1.
2	7-8	Optional positive DC power #2.
3	9-10	Optional positive DC power #3.
4	11-12	Optional positive DC power #4.

Each channel's shunt receptacle matrix is located adjacent to its relay socket as shown in Figure 54.

Figure 54: Relay Channel Layout (top view)



10.2.4 I/O Connectors

The 2650 module employs sixteen connectors for interfacing relay line and load power signals. Each relay channel uses two of these connectors, one for line and one for load, as shown in Table 30.

For AC power switching, the hot and neutral line-side signals connect to the top connector (see Figure 54) at the terminals shown as Hot and Neut. The AC load is typically connected to the bottom connector across the COM and NO (normally open) terminals, and thus hot power is switched onto the NO terminal only when the relay is energized. Terminals Neut and COM are connected together by circuit board traces.

Both AC and DC loads may be controlled. Please refer to the RH1B relay data sheet for detailed specifications.

Table 30: Line and Load Connectors

Relay Channel	Connector	
	Line	Load
CH0	P15	P16
CH1	P13	P14
CH2	P11	P12
CH3	P9	P10
CH4	P7	P8
CH5	P5	P6
CH6	P3	P4
CH7	P1	P2

Sixteen mating connector shells (AMP part number 172166-1) are supplied with the module for line and load field wiring termination. Three crimp pins (AMP part number 770988-1) are provided for each shell. Each crimp pin must be crimped onto a field wire and then inserted into its connector shell.

10.2.5 Reserved Connectors

Connectors JP1 and JP10 are reserved for manufacturing programming and test. Application installations should leave these unconnected.

10.3 Reset

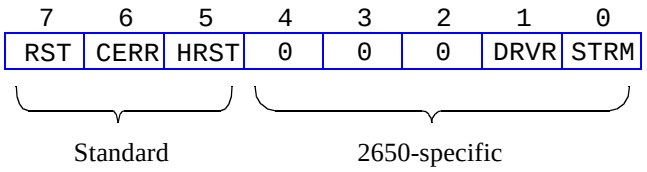
All drivers default to the inactive state (relays coils are all de-energized) upon module reset. 2650 modules undergo module reset in response to any of the following conditions:

- Module power-up.
- Reset action requested by the Ethernet client.
- Watchdog time-out due to soft or hardware fault.
- Communication time-out, which will occur if the client fails to communicate with the 2650 module within a programmable communication watchdog interval.

10.4 Status Flags

In addition to the standard flag bits that are common to all IOMs (Section 3.4.1), the returned status byte includes two fault flags that are specific to 2650 modules only. The status byte is structured as shown in Figure 55.

Figure 55: RLYx Status Byte:



DRVR - Active-high bit flag that indicates one or more relay coil drivers failed to go to the commanded state. This may be caused by a driver fault, a shorted relay coil or a serial data stream problem.

STRM - Active-high bit flag that indicates an error was detected in the serial data stream that is used to control relay drivers and monitor driver states.

2650-specific fault flag bits can be cleared by invoking a `ResetFlags` action.

10.5 Actions

This section describes the programmable actions that are supported exclusively by 2650 modules. These actions may be employed, in any sequence or combination, in MCmd action lists.

2650 modules support all of the common actions that are universally recognized by all IOMs, such as `HardReset`, `SoftReset`, etc. Refer to Chapter 5 for details.

Table 31: Summary of Relay Module Actions

Command	Opcode	Function
`GetInputs`	`0x00`	Return the physical states of all relay coil drivers.

Table 31: Summary of Relay Module Actions

Command	Opcode	Function
`GetOutputs`	`0x01`	Return the programmed states of all relay coil drivers.
`SetOutputs`	`0x02`	Program all relay coil drivers.

Note: Model 2650 has a maximum MRsp size of eleven bytes.

10.5.1 GetInputs

Function Returns the physical states of all relay channels.

Opcode `0x00`

Command `none`

Response `(States)`

Parameter	Function
`States`	Bit flags that represent the physical relay states. Each bit is associated with one relay channel. For example, bit 7 is associated with relay channel 7. Any bit set to *one* indicates the associated channel is set to the active state; any bit set to *zero* indicates the channel is set to the inactive state.

Notes Each channel includes a monitoring circuit that enables the on-board processor to determine the physical state of the relay coil driver. This action returns the monitored physical state of each coil driver, even if the relay is not present or its coil winding has opened.

Coil driver states are acquired periodically at two millisecond intervals. Consequently, `States` may not accurately reflect the state of a coil driver that has changed its physical state within the last two milliseconds.

After the coil drivers have been stable for at least two milliseconds, the `States` value returned by `GetInputs` should agree with the `States` value last programmed by `SetOutputs` (unless a module reset has occurred).

10.5.2 GetOutputs

Function Returns the programmed coil driver states of all relay channels.

Opcode `0x01`

Command `none`

Response `(States)`

Parameter	Function
`States`	Bit flags that represent the programmed driver states of all relay channels. Each bit is associated with one relay channel. For example, bit 7 is associated with relay channel 7. Any bit set to *one* indicates the associated coil driver is set to the active state; any bit set to *zero* indicates the driver is set to the inactive state.

10.5.3 SetOutputs

Function Programs the output driver states of all relay channels.

Opcode `0x02`

Command `(States)`

Parameter	Function
`States`	Bit flags that represent the target output states for all relay channels. Each bit is associated with one relay channel. For example, bit 7 is associated with relay channel 7. Set a bit to *one* to program the associated channel to the active state, or to *zero* to program the channel to the inactive state.

Response `none`

Notes Although the `SetOutputs` action response is generated immediately, the actual updating of the outputs is deferred. All output drivers will be set to their target values within two milliseconds after this action executes.

10.6 Specifications

Table 32: Model 2650 Specifications

Interlock Power Daisy Chain	Voltage range	+5.8VDC to +24VDC. Relay coils must be routed to +24VDC circuits.
	Current	10A maximum, per circuit.
Relays	Channels	8, with finger-safe relay sockets.
	Contact resistance	50 milliohms, max.
	Form	Single pole, double throw, 24V coil.
Relay Components (not included with 2650-0)	Relays (qty 8)	Idec RH1B-U DC24V or equivalent.
	Hold-down clips (qty 8)	Idec SY4S-51F1 or equivalent.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent.
	Relay line/load (qty 16)	(*) AMP 172-166 with 770988-1 socket contacts (3), or equivalent.
	Power daisy chain (qty 2)	AMP 770849-6 with 640706-1 pin contacts (6), or equivalent. Accepts Sensoray cable assembly, part number 2600C1.
Temperature	Operating range	0 to 70°C.
Power	Input power	+24V ±5% @ 310mA, with all relay coils energized.
Dimensions	Outer dimensions	5.0H x 8.6W x 3.0D inches, including relays and DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel.

Table 33: Load Limits

Load Type	Line Voltage	Maximum Load Current
Resistive	30V DC	10A
	110V DC	0.5A
	120V AC	10A
	240V AC	7A
Inductive	30V DC	7A
	110V DC	0.3A
	120V AC	7A
	240V AC	5A

Chapter 11: Model 2652 SSR Module

11.1 Introduction

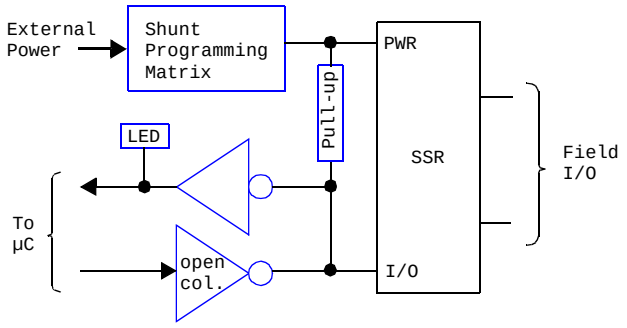
Model 2652 is a smart I/O module (IOM) that monitors and controls up to eight solid state relays (SSRs). All SSRs are socketed so that SSR types may be mixed or left unpopulated, as required. Each SSR socket may be populated with any supported SSR type, including AC in, AC out, DC in and DC out. SSRs are secured to the IOM via integral hold-down screws that are retained by pem nuts on the IOM.

The module's microcontroller, which provides I/O services to a remote client, communicates with the client by means of a standard Category-5 UTP cable over an optically isolated, asynchronous serial interface.

As shown in Figure 56, each relay channel includes:

- Standard SSR socket.
- Active-low, open-collector driver for output SSR types.
- Channel monitor circuit, to enable the client to acquire the channel's physical state.
- Indicator LED that lights when the channel is energized by either the output driver or by an externally generated signal arriving from an input SSR.
- Two finger-safe field I/O connectors per channel.
- Shunt programming matrix for implementing interlocked fail-safe operation.

Figure 56: SSR Channel Block Diagram



The client can program and read back the state of each channel's output driver, and it can acquire the debounced state of each channel's physical signal. The microcontroller debounces all acquired channel states with a 10 millisecond software debounce filter. When used in conjunction with output SSR types, all channels have the ability to autonomously produce PWM type output signals at a rate and duty cycle specified by the client.

Each SSR channel includes an LED indicator that lights when the channel is active. The indicator lights regardless of whether the channel is driven by an external signal source (through an input SSR type) or by the channel's output driver.

On-board, finger-safe connectors enable direct connection to field wiring without the need for intermediate terminal blocks. One connector is used for the line connection and the other for load.

Integral support is provided for system interlock circuits and multiple power supply voltages. Each SSR channel may be independently powered from any of up to six external DC power sources.

The communication interface between the client and the on-board microcontroller is optically isolated to ensure error-free operation in demanding industrial environments.

11.2 Hardware Configuration

The 2652 module is configured by installing programming shunts and connecting various cables to the module as described in this section.

11.2.1 IOM Network Connector

A single RJ-45 connector, J49, is used to connect the 2652 module to its client. This should be mated to a standard UTP (unshielded twisted pair) Category-5 cable.

The other end of this cable will be connected to one of the sixteen IOM ports on a model 2601 module.

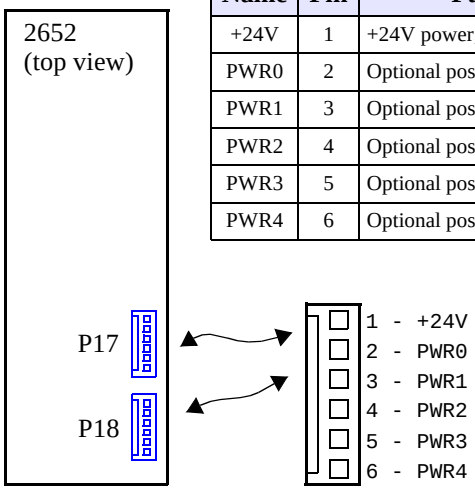
11.2.2 Interlock Power Connectors

The monitoring and control circuitry on each SSR channel is powered from an external DC power source. At least one external power source is required to make the SSR channels functional. Up to six independent DC power sources are supported. Note that an SSR's control voltage must match its selected DC power supply voltage. For example, use an SSR such as the 70-OAC24 (24VDC control, 120VAC output) for AC output applications that will be operating from a 24VDC supply.

Connectors P17 and P18 convey power to the SSR monitoring and control circuitry from external DC power sources. All external power sources must supply positive DC voltages with respect to system ground. The current return of each external power source must connect to the system ground.

As shown in Figure 57, connectors P17 and P18 are identical in function and pinout. The external DC power sources may be connected to either P17 or P18. If other IOMs require access to these external power sources, the unused connector may be used to daisy-chain power out to other IOMs. In this manner, power can be distributed to any arbitrary number of external IOMs without the use of dedicated power distribution terminal blocks.

Figure 57: Pinouts of Connectors P17 and P18

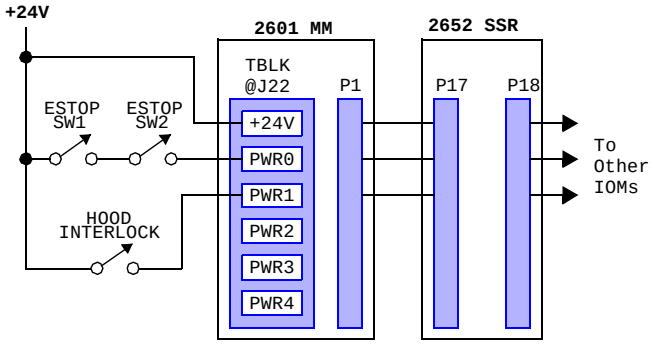


Name	Pin	Function
+24V	1	+24V power, always on.
PWR0	2	Optional positive DC power #0.
PWR1	3	Optional positive DC power #1.
PWR2	4	Optional positive DC power #2.
PWR3	5	Optional positive DC power #3.
PWR4	6	Optional positive DC power #4.

The external DC power sources need not be derived from interlock contacts, but by employing interlocked power sources, the system interlock cabling can be minimized. This feature is especially useful for guaranteeing the fail-safe shutdown of select SSR channels in the event of a critical condition such as emergency stop (ESTOP) activation, open safety hood, etc.

Figure 58 illustrates a system in which PWR0 is connected to multiple system emergency stop contacts, and PWR1 is routed through a safety hood interlock contact. Note that all of the external power sources are daisy-chained from module to module. Any SSR that is configured to use PWR0 will automatically lose power when either ESTOP contact is opened, and any SSR that is configured to use PWR1 will lose power when the hood opens. All other SSRs use the +24V "always on" power source and thus are unaffected by interlock contacts. PWR2, PWR3 and PWR4 are not used in this application.

Figure 58: Wiring Example With Interlock Contacts



Use Sensoray cable assembly, part number 2600C1, to connect P1 or P2 to the power daisy chain

11.2.3 Relay Power Selection Shunt

Each relay channel may be independently operated from any of the six external DC power sources (that are connected to P17 and P18) by installing the appropriate programming shunt.

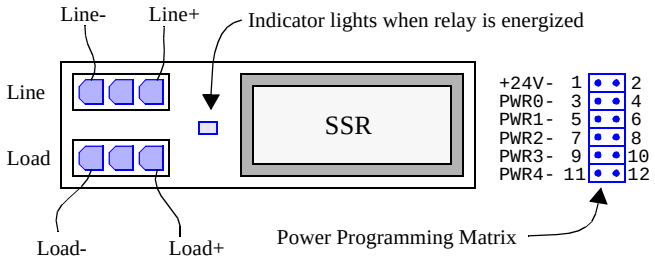
A shunt receptacle matrix is provided for each channel. Each matrix has six shunt positions, corresponding to the six external DC power sources. As shown in Table 34, a 2mm programming shunt (supplied with the 2652) must be installed at the position corresponding to the relay channel's target power source. Only one shunt should be installed per matrix.

Table 34: Interlock Power Programming Matrix

PWB Label	Shunt Pins	Selected Power Source
+24V	1-2	+24V power, always on (default)
0	3-4	Optional positive DC power #0.
1	5-6	Optional positive DC power #1.
2	7-8	Optional positive DC power #2.
3	9-10	Optional positive DC power #3.
4	11-12	Optional positive DC power #4.

Each channel's shunt receptacle matrix is located adjacent to its SSR socket as shown in Figure 59.

Figure 59: SSR Channel Layout (top view)



11.2.4 I/O Connectors

The 2652 module employs sixteen connectors for field wiring. Each channel uses one connector for line and one for load (see Figure 59).

Table 35: Line and Load Connectors

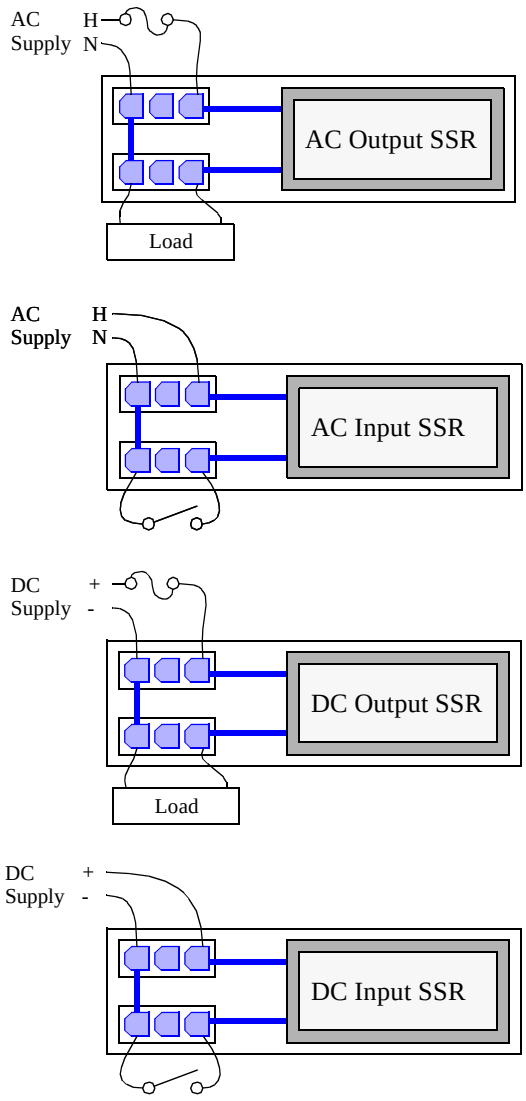
Relay Channel	Connector	
	Line	Load
CH0	P15	P16
CH1	P13	P14
CH2	P11	P12
CH3	P9	P10
CH4	P7	P8
CH5	P5	P6
CH6	P3	P4
CH7	P1	P2

The Line- and Load- signals are connected together by circuit board traces as indicated by the heavy blue lines in Figure 60.

For AC and DC output modules, the power supply should be wired to the Line connector and the load device should be wired to the Load connector. The Line+ and Load+ signals are connected together by the SSR only when the SSR is energized. It is recommended that the circuit be protected with a fuse or circuit breaker on the Line side.

For AC and DC input modules, the line supply should be wired to the Line connector and the contact should be wired to the Load connector.

Figure 60: Field Wiring Diagrams



Sixteen mating connector shells (AMP part number 172166-1) are supplied with the module for field wiring termination. Two crimp pins (AMP part number 770988-1) are provided for each shell. Each crimp pin must be attached to a field wire and then inserted into its connector shell.

11.2.5 Reserved Connectors

Connectors JP9 and JP10 are reserved for manufacturing programming and test. Application installations should leave these unconnected.

11.3 Channel Operating Modes

Each channel may be independently operated as an input or output, with AC or DC power and load by installing the appropriate SSR type into the channel's SSR socket.

11.3.1 Output Operation

When a channel is using an output SSR, the client simply programs the channel's output driver to the desired state.

11.3.2 Input Operation

When used with an input SSR, the client must ensure that it never programs a channel's output driver to its active state, as this would override the SSR. Instead, the client must leave the output driver in its default, inactive state and allow the SSR to drive the channel's physical state.

11.3.3 Output Modes

SSR channels support two different output modes: Standard and PWM. In the Standard mode, a channel's output driver is directly controlled by the client as described in Section 11.3.1. In the PWM mode, the channel's output driver is cycled on and off at a client-specified rate and duty cycle.

The output mode may be independently configured for each channel. For example, channel 0 can operate in the PWM mode while channels 1 to 7 operate in the Standard output mode.

11.3.4 Input Debounce

All SSR channel physical states are sampled every 10 milliseconds. The on-board microcontroller applies a debounce filter to the sampled states, resulting in a 10 millisecond debounce period.

When acquiring physical channel states from the 2652 module, the client always receives the debounced image of the physical states, which is delayed 10 milliseconds by the debounce function.

11.4 Reset

Upon module reset, all channels default to the Standard operating mode and all SSR output drivers default to the inactive state. A 2652 module will experience a module reset in response to any of the following conditions:

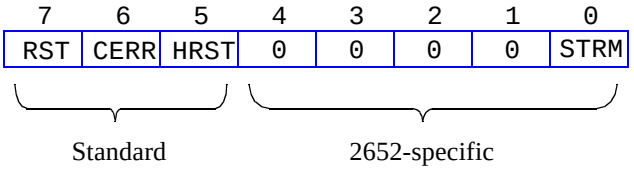
- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.

- Communication time-out, which will occur if the client fails to communicate with the 2652 module within a programmable communication watchdog interval.

11.5 Status Flags

In addition to the standard flag bits that are common to all IOMs (see Section 3.4.1), the status byte returned by a 2652 module includes one fault flag that is specific to 2652 module types. The 2652 status byte is structured as shown in Figure 61.

Figure 61: 2652 Status Byte



STRM - Active-high bit flag that indicates an error was detected in the serial data stream that is used to control the SSR output drivers. This flag can be cleared by invoking a `ResetFlags` action.

11.6 Actions

This section describes the programmable actions that are supported exclusively by the 2652 module. These actions may be employed, in any sequence or combination, in MCmd action lists.

2652 modules support all of the common actions that are universally recognized by all IOMs, such as `HardReset`, `SoftReset`, etc. Refer to Chapter 5 for details.

Table 36: Summary of 2652 Module Actions

Command	Opcode	Function
`GetInputs`	`0x00`	Return all physical channel states.
`GetOutputs`	`0x01`	Return all programmed output states.
`SetOutputs`	`0x02`	Program all output driver states.
`SetModes`	`0x03`	Set operating modes.
`GetModes`	`0x04`	Return operating modes.
`SetPwmRatio`	`0x05`	Program PWM duty cycle and period.
`GetPwmRatio`	`0x06`	Return PWM duty cycle and period.

Note: Model 2610 has a maximum MRsp size of eleven bytes.

11.6.1 GetInputs

Function Returns the debounced, physical channel states of all SSR channels.

Opcode `0x00`

Command `none`

Response `(States)`

Parameter	Function
`States`	Bit flags that represent the debounced, physical states of all SSR channels. Each bit is associated with one channel. For example, bit 7 is associated with SSR channel 7. Any bit set to *one* indicates the associated channel is set to the active state; any bit set to *zero* indicates the channel is set to the inactive state.

Notes The returned state values represent the debounced, physical states of all SSR channels. Because the inputs are sampled every 2 milliseconds, and the debounce period is 10 milliseconds, the returned state values will all have an age ranging from 10 to 12 milliseconds, plus any network communication latency.

Note that the physical states of *all* SSR channels are returned, regardless of their respective operating modes.

11.6.2 GetOutputs

Function Returns the programmed output driver states of all SSR channels. Note that the programmed output driver states may not be identical to the physical channel states (that would be returned by `GetInputs`) as some channels may be driven by external signal sources. In the case of channels that have been configured for the PWM mode, this function returns indeterminate state values.

Opcode `0x01`

Command `none`

Response `(States)`

Parameter	Function
`States`	Bit flags that represent the programmed output driver states of all SSR channels. Each bit is associated with one SSR channel. For example, bit 7 is associated with SSR channel 7. Logic *one* indicates the associated channel is set to the active state; logic *zero* indicates the inactive state.

11.6.3 SetOutputs

Function Programs the output driver states of all channels that are configured for the Standard operating mode.

Opcode `0x02`

Command `(States)`

Parameter	Function
`States`	Bit flags. Each flag represents the target output state for one SSR channel. For example, bit 7 is associated with SSR channel 7. Set a bit to *one* to program the associated channel to the active state, or to *zero* to program the channel to the inactive state.

Response `none`

Notes `SetOutputs` programs the output driver states of channels that are configured for the Standard operating mode; it has no effect on channels that are configured for the PWM operating mode.

In order to optimize the network response time, `SetOutputs` invokes a deferred processing function on the 2652 module. When the `SetOutputs` action executes, it immediately generates a response to acknowledge receipt of the action, but the physical driver states may not change immediately. Instead, the output drivers will change to their target values within two milliseconds of executing this action.

All modified outputs will change state synchronously so that arbitrary groups of SSR channels can be used in applications that require "state machine" type behavior.

11.6.4 SetModes

Function Programs the operating modes for all channels.

Opcode `0x03`

Command `(ModeFlags)`

Parameter	Function
`ModeFlags`	Operating modes for all channels. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Set a bit to *one* to select the PWM mode, or to *zero* to select the Standard mode.

Response `none`

Notes Each channel may operate in either the Standard mode or the PWM mode. Following a reset, all channels default to the Standard operating mode.

When a channel's operating mode is first changed to the PWM mode, its output driver duty cycle is reset to zero; this prevents the output from going active until the duty cycle is explicitly set by a `SetPwmRatio` action.

Similarly, when a channel's operating mode is changed to the Standard mode, its output driver is reset to the inactive state to prevent its output from going active until explicitly set by a `SetOutputs` action.

11.6.5 GetModes

Function Returns the operating modes for all channels.

Opcode `0x04`

Command `none`

Response `(ModeFlags)`

Parameter	Function
`ModeFlags`	Operating modes for all channels. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Set a bit to *one* to select the PWM mode, or to *zero* to select the Standard mode.

Notes `GetModes` returns the current operating modes of all SSR channels. The operating modes may be changed under program control (e.g., in response to a `SetModes` action) or unexpectedly (e.g., all channel operating modes revert to the default when a module reset occurs).

11.6.6 SetPwmRatio

Function Programs the PWM timing attributes for one SSR channel.

Opcode `0x05`

Command `(Channel),(OnTime),(OffTime)`

Parameter	Function
`Channel`	Byte value that specifies the SSR channel that is to be configured. Value may range from 0 to 7.
`OnTime`	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments. Value range: 0 to 255.
`OffTime`	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments. Value range: 1 to 255.

Response `none`

Notes `SetPwmRatio` applies only to channels that have previously been configured for the PWM operating mode. This action has no effect on channels that are configured for the Standard operating mode.

The CERR status flag will be set if an illegal channel number is specified.

Two values are specified to establish the duty cycle: `OnTime` and `OffTime`. Each of these values specifies a time interval, expressed in 2 millisecond increments. For example, by programming `OnTime` to 1 and `OffTime` to 4, the duty cycle will be set to 20% with a period of 10 milliseconds (2 milliseconds active and 8 milliseconds inactive per output cycle).

The output duty cycle can be set to zero by declaring the `OnTime` value to be zero. It is not possible to set the duty cycle to 100% as the `OffTime` parameter must be at least 1. The `OffTime` value will automatically default to 1 if the client attempts to set it to zero.

After this action executes, the present channel output state will continue to be asserted for the duration of its previously programmed time, and then the new `OnTime` and `OffTime` will take effect.

11.6.7 GetPwmRatio

Function Returns the PWM timing attributes of one SSR channel.

Opcode `0x06`

Command `(Channel)`

Parameter	Function
`Channel`	Byte value that specifies the SSR channel to be addressed. Value may range from 0 to 7.

Response `(OnTime),(OffTime)`

Parameter	Function
`OnTime`	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments.
`OffTime`	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments.

Notes `GetPwmRatio` applies only to channels that have previously been configured for the PWM operating mode. The two returned byte values are meaningful only if the specified channel is configured for the PWM operating mode.

The CERR status flag will be set if an illegal channel number is specified.

Two values are returned: `OnTime` and `OffTime`. Each of these values specifies a time interval, expressed in 2 millisecond increments, that are currently in effect. These values will be the values programmed by the most recent `SetPwmRatio` action, or the default values if no `SetPwmRatio` action has executed since the channel's operating mode was switched to PWM.

11.7 Specifications

Table 37: Model 2652 Specifications

Interlock Power Daisy Chain	Voltage range	+5VDC to +24VDC, as required by SSRs.
	Current	10A maximum, per circuit.

Table 37: Model 2652 Specifications

Relays	Channels	8, with sockets.
	Form factor	Industry standard digital SSRs: 0.6W x 1.7L x 1.25H (inches).
	SSR types (not included with 2652)	Any combination of Grayhill 70- series or equivalent. Examples: 70-OAC24 - 120V AC output 70-OAC24A - 240V AC output 70-ODC24 - 60V DC output 70-ODC24R - mechanical relay output 70-IAC24 - 120V AC input 70-IAC24A - 240V AC input 70-IDC24 - 32V DC input Note: All of the above SSR types employ 24V logic. Part numbers will be different for other logic levels. Example: use 70-OAC5 for 120V AC output if the associated daisy-chained DC power supply is 5V.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent.
	Relay line/load (qty 16)	(*) AMP 172-166 with 770988-1 socket contacts (2), or equivalent.
	Power daisy chain (qty 2)	AMP 770849-6 with 640706-1 pin contacts (6), or equivalent. Accepts Sensoray cable assembly, part number 2600C1.
Temperature	Operating range	0 to 70°C.
Power	Input power	+24V ±5% @ 310mA, with eight 70-OAC24 SSRs energized.
Dimensions	Outer dimensions	5.0H x 8.6W x 3.0D inches, including relays and DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel.

Chapter 12: Model 2653 SSR Module

12.1 Introduction

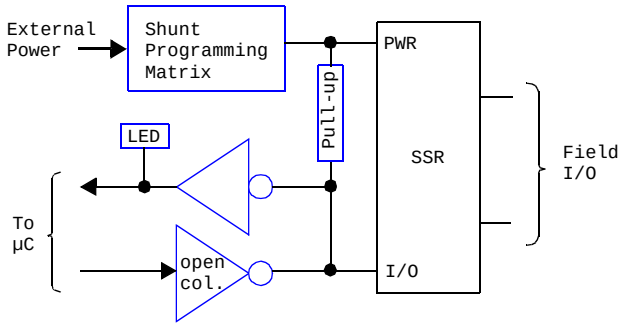
Model 2653 is a smart I/O module (IOM) that monitors and controls up to sixteen solid state relays (SSRs). All SSRs are socketed so that SSR types may be mixed or left unpopulated, as required. Each SSR socket may be populated with any supported SSR type, including AC in, AC out, DC in and DC out. SSRs are secured to the IOM via integral hold-down screws that are retained by pem nuts on the IOM.

The module's microcontroller, which provides I/O services to a remote client, communicates with the client by means of a standard Category-5 UTP cable over an optically isolated, asynchronous serial interface.

As shown in Figure 62, each relay channel includes:

- Standard SSR socket.
- Active-low, open-collector driver for output SSR types.
- Channel monitor circuit, to enable the client to acquire the channel's physical state.
- Indicator LED that lights when the channel is energized by either the output driver or by an externally generated signal arriving from an input SSR.
- Two finger-safe field I/O connectors per channel.
- Shunt programming matrix for implementing interlocked fail-safe operation.

Figure 62: SSR Channel Block Diagram



The client can program and read back the state of each channel's output driver, and it can acquire the debounced state of each channel's physical signal. The microcontroller debounces all acquired channel states with a 10 millisecond software debounce filter. When used in conjunction with output SSR types, all channels have the ability to autonomously produce PWM type output signals at a rate and duty cycle specified by the client.

Each SSR channel includes an LED indicator that lights when the channel is active. The indicator lights regardless of whether the channel is driven by an external signal source (through an input SSR type) or by the channel's output driver.

On-board, finger-safe connectors enable direct connection to field wiring without the need for intermediate terminal blocks. One connector is used for the line connection and the other for load.

Integral support is provided for system interlock circuits and multiple power supply voltages. Each SSR channel may be independently powered from any of up to six external DC power sources.

The communication interface between the client and the on-board microcontroller is optically isolated to ensure error-free operation in demanding industrial environments.

12.2 Hardware Configuration

The 2653 module is configured by installing programming shunts and connecting various cables to the module as described in this section.

12.2.1 IOM Network Connector

A single RJ-45 connector, J1, is used to connect the 2653 module to its client. This should be mated to a standard UTP (unshielded twisted pair) Category-5 cable.

The other end of this cable will be connected to one of the sixteen IOM ports on a model 2601 module.

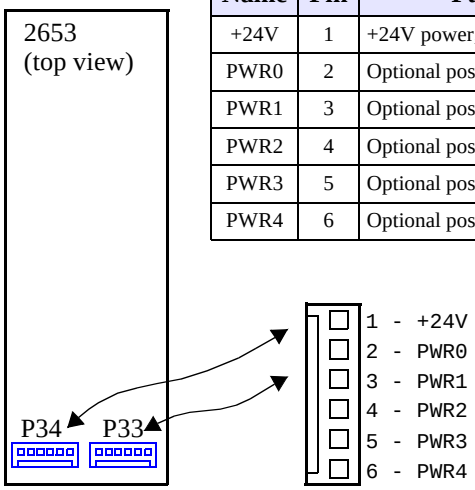
12.2.2 Interlock Power Connectors

The monitoring and control circuitry on each SSR channel is powered from an external DC power source. At least one external power source is required to make the SSR channels functional. Up to six independent DC power sources are supported. Note that an SSR's control voltage must match its selected DC power supply voltage. For example, use an SSR such as the 70-OAC24 (24VDC control, 120VAC output) for AC output applications that will be operating from a 24VDC supply.

Connectors P33 and P34 convey power to the SSR monitoring and control circuitry from external DC power sources. All external power sources must supply positive DC voltages with respect to system ground. The current return of each external power source must connect to the system ground.

As shown in Figure 63, connectors P33 and P34 are identical in function and pinout. The external DC power sources may be connected to either P33 or P34. If other IOMs require access to these external power sources, the unused connector may be used to daisy-chain power out to other IOMs. In this manner, power can be distributed to any arbitrary number of external IOMs without the use of dedicated power distribution terminal blocks.

Figure 63: Pinouts of Connectors P33 and P34

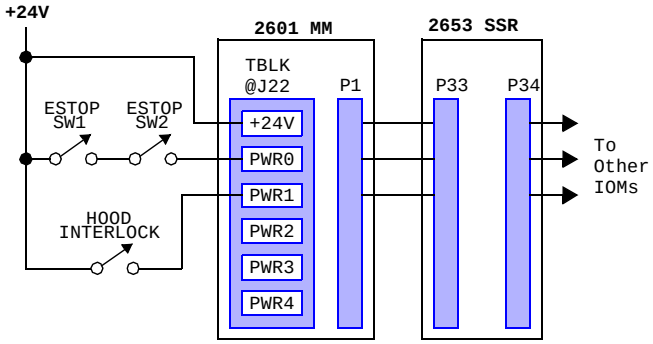


Name	Pin	Function
+24V	1	+24V power, always on.
PWR0	2	Optional positive DC power #0.
PWR1	3	Optional positive DC power #1.
PWR2	4	Optional positive DC power #2.
PWR3	5	Optional positive DC power #3.
PWR4	6	Optional positive DC power #4.

The external DC power sources need not be derived from interlock contacts, but by employing interlocked power sources, the system interlock cabling can be minimized. This feature is especially useful for guaranteeing the fail-safe shutdown of select SSR channels in the event of a critical condition such as emergency stop (ESTOP) activation, open safety hood, etc.

Figure 64 illustrates a system in which PWR0 is connected to multiple system emergency stop contacts, and PWR1 is routed through a safety hood interlock contact. Note that all of the external power sources are daisy-chained from module to module. Any SSR that is configured to use PWR0 will automatically lose power when either ESTOP contact is opened, and any SSR that is configured to use PWR1 will lose power when the hood opens. All other SSRs use the +24V "always on" power source and thus are unaffected by interlock contacts. PWR2, PWR3 and PWR4 are not used in this application.

Figure 64: Wiring Example With Interlock Contacts



Use Sensoray cable assembly, part number 2600C1, to connect P1 or P2 to the power daisy chain

12.2.3 Relay Power Selection Shunt

Each relay channel may be independently operated from any of the six external DC power sources (that are connected to P33 and P34) by installing the appropriate programming shunt.

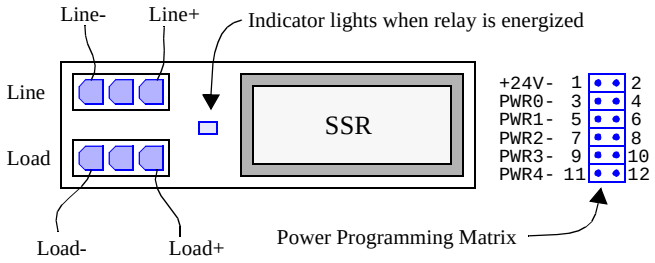
A shunt receptacle matrix is provided for each channel. Each matrix has six shunt positions, corresponding to the six external DC power sources. As shown in Table 38, a 2mm programming shunt (supplied with the 2653) must be installed at the position corresponding to the relay channel's target power source. Only one shunt should be installed per matrix.

Table 38: Interlock Power Programming Matrix

PWB Label	Shunt Pins	Selected Power Source
+24V	1-2	+24V power, always on (default)
0	3-4	Optional positive DC power #0.
1	5-6	Optional positive DC power #1.
2	7-8	Optional positive DC power #2.
3	9-10	Optional positive DC power #3.
4	11-12	Optional positive DC power #4.

Each channel's shunt receptacle matrix is located adjacent to its SSR socket as shown in Figure 65.

Figure 65: SSR Channel Layout (top view)



12.2.4 I/O Connectors

The 2653 module employs thirty two connectors for field wiring. Each channel uses one connector for line and one for load (see Figure 65).

Table 39: Line and Load Connectors

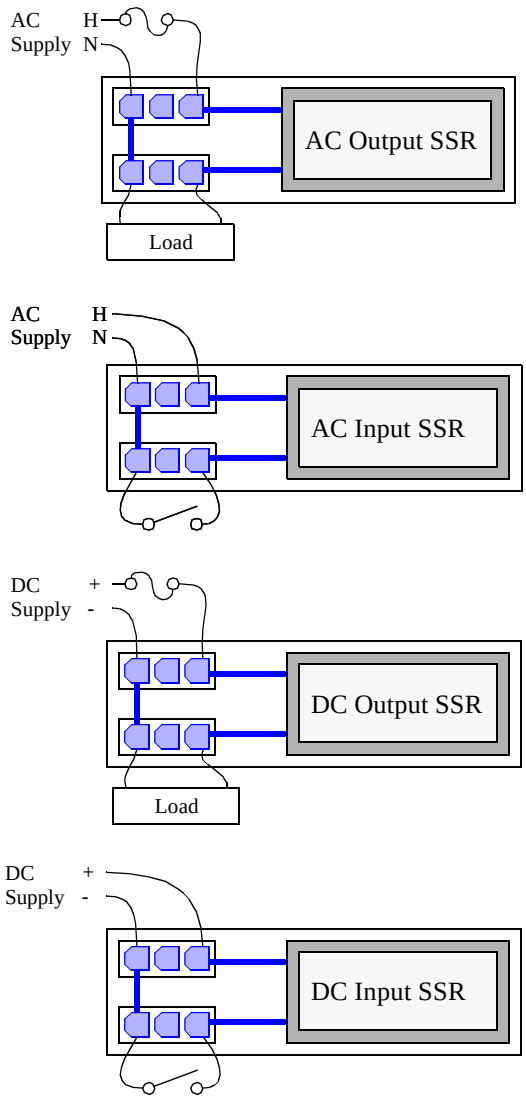
Relay Channel	Connector	
	Line	Load
CH0	P2	P1
CH1	P4	P3
CH2	P6	P5
CH3	P8	P7
CH4	P10	P9
CH5	P12	P11
CH6	P14	P13
CH7	P16	P15

Table 39: Line and Load Connectors

Relay Channel	Connector	
	Line	Load
CH8	P18	P17
CH9	P20	P19
CH10	P22	P21
CH11	P24	P23
CH12	P26	P25
CH13	P28	P27
CH14	P30	P29
CH15	P32	P31

The Line- and Load- signals are connected together by circuit board traces as indicated by the heavy blue lines in Figure 66.

Figure 66: Field Wiring Diagrams



For AC and DC output modules, the power supply should be wired to the Line connector and the load device should be wired to the Load connector. The Line+ and Load+ signals are connected together by the SSR only when the SSR is energized. It is recommended that the circuit be protected with a fuse or circuit breaker on the Line side.

For AC and DC input modules, the line supply should be wired to the Line connector and the contact should be wired to the Load connector.

Thirty two mating connector shells (AMP part number 172166-1) are supplied with the module for field wiring termination. Two crimp pins (AMP part number 770988-1) are provided for each shell. Each crimp pin must be attached to a field wire and then inserted into its connector shell.

12.2.5 Reserved Connectors

Connectors JP17 and JP18 are reserved for manufacturing programming and test. Application installations should leave these unconnected.

12.3 Channel Operating Modes

Each channel may be independently operated as an input or output, with AC or DC power and load by installing the appropriate SSR type into the channel's SSR socket.

12.3.1 Output Operation

When a channel is using an output SSR, the client simply programs the channel's output driver to the desired state.

12.3.2 Input Operation

When used with an input SSR, the client must ensure that it never programs a channel's output driver to its active state, as this would override the SSR. Instead, the client must leave the output driver in its default, inactive state and allow the SSR to drive the channel's physical state.

12.3.3 Output Modes

SSR channels support two different output modes: Standard and PWM. In the Standard mode, a channel's output driver is directly controlled by the client as described in Section 12.3.1. In the PWM mode, the channel's output driver is cycled on and off at a client-specified rate and duty cycle.

The output mode may be independently configured for each channel. For example, channel 0 can operate in the PWM mode while channels 1 to 15 operate in the Standard output mode.

12.3.4 Input Debounce

All SSR channel physical states are sampled every 10 milliseconds. The on-board microcontroller applies a debounce filter to the sampled states, resulting in a 10 millisecond debounce period.

When acquiring physical channel states from the 2653 module, the client always receives the debounced image of the physical

states, which is delayed 10 milliseconds by the debounce function.

12.4 Reset

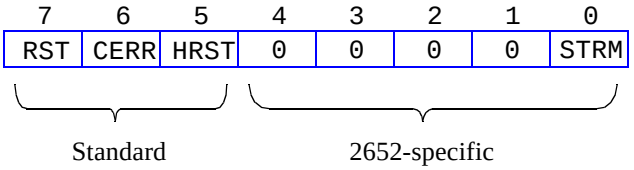
Upon module reset, all channels default to the Standard operating mode and all SSR output drivers default to the inactive state. A 2653 module will experience a module reset in response to any of the following conditions:

- Module power-up.
- Watchdog time-out due to soft or hard fault.
- SoftReset or HardReset action request from the client.
- Communication time-out, which will occur if the client fails to communicate with the 2653 module within a programmable communication watchdog interval.

12.5 Status Flags

In addition to the standard flag bits that are common to all IOMs (see Section 3.4.1), the status byte returned by a 2653 module includes one fault flag that is specific to 2653 module types. The 2653 status byte is structured as shown in Figure 67.

Figure 67: 2653 Status Byte



STRM - Active-high bit flag that indicates an error was detected in the serial data stream that is used to control the SSR output drivers. This flag can be cleared by invoking a `ResetFlags` action.

12.6 Actions

This section describes the programmable actions that are supported exclusively by the 2653 module. These actions may be employed, in any sequence or combination, in MCmd action lists.

2653 modules support all of the common actions that are universally recognized by all IOMs, such as `HardReset`, `SoftReset`, etc. Refer to Chapter 5 for details.

Table 40: Summary of 2653 Module Actions

Command	Opcode	Function
`GetInputs`	`0x00`	Return physical channel states.
`GetOutputs`	`0x01`	Return programmed channel states.
`SetOutputs`	`0x02`	Program output driver states.
`SetModes`	`0x03`	Set channel operating modes.
`GetModes`	`0x04`	Return channel operating modes.
`SetPwmRatio`	`0x05`	Program PWM duty cycle and period for one channel.
`GetPwmRatio`	`0x06`	Return PWM duty cycle and period for one channel.

Note: Model 2610 has a maximum MRsp size of eleven bytes.

12.6.1 GetInputs

Function Returns the debounced, physical states of all SSR channels.

Opcode `0x00`

Command `none`

Response `(S<7:0>),(S<15:8>)`

Parameter	Function
`S[]`	Byte array containing 16 bit flags that represent the debounced, physical states of all SSR channels. Each bit is associated with one SSR channel. For example, bit 14 is associated with SSR channel 14. Any bit set to *one* indicates the associated channel is set to the active state; any bit set to *zero* indicates the channel is set to the inactive state.

Notes The returned state values represent the debounced, physical states of all SSR channels. Because the inputs are sampled every 2 milliseconds, and the debounce period is 10 milliseconds, the returned state values will all have an age ranging from 10 to 12 milliseconds, plus any network communication latency.

Note that the physical states of *all* SSR channels are returned, regardless of their respective operating modes.

12.6.2 GetOutputs

Function Returns the programmed output driver states of all SSR channels. Note that the programmed states may match the physical states (that would be returned by `GetInputs`) because some channels may be driven by external signal sources. In the case of channels that have been configured for the PWM mode, this function returns indeterminate values.

Opcode `0x01`

Command `none`

Response `(S<7:0>),(S<15:8>)`

Parameter	Function
`S[]`	Byte array containing 16 bit flags that represent the programmed output driver states of all SSR channels. Each bit is associated with one SSR channel. For example, bit 14 is associated with SSR channel 14. Any bit set to *one* indicates the associated channel is programmed to the active state; any bit set to *zero* indicates the channel is programmed to the inactive state.

12.6.3 SetOutputs

Function Programs the output driver states of all channels that are configured for the Standard operating mode.

Opcode `0x02`

Command `(S<7:0>),(S<15:8>)`

Parameter	Function
`S[]`	Byte array containing 16 bit flags that specify the desired output driver states for all channels. Each flag represents the target output state for one SSR channel. For example, bit 7 is associated with SSR channel 7. Set a bit to *one* to program the associated channel to the active state, or to *zero* to program the channel to the inactive state.

Response `none`

Notes `SetOutputs` programs the output driver states of channels that are configured for the Standard operating mode; it has no effect on channels that are configured for the PWM operating mode.

In order to optimize the network response time, `SetOutputs` invokes a deferred processing function on the 2653 module. When the `SetOutputs` action executes, it immediately generates a response to acknowledge receipt of the action, but the physical driver states may not change immediately. Instead, the output drivers will change to their target values within two milliseconds of executing this action.

All modified outputs will change state synchronously so that arbitrary groups of SSR channels can be used in applications that require "state machine" type behavior.

12.6.4 SetModes

Function Programs the operating modes for all channels.

Opcode `0x03`

Command `(M<7:0>),(M<15:8>)`

Parameter	Function
`M[]`	Byte array containing 16 bit flags that specify the operating modes for all channels. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Set a bit to *one* to select the PWM mode, or to *zero* to select the Standard mode.

Response `none`

Notes Each channel may operate in either the Standard mode or the PWM mode. Following a reset, all channels default to the Standard operating mode.

When a channel's operating mode is first changed to the PWM mode, its output driver duty cycle is reset to zero; this prevents the output from going active until the duty cycle is explicitly set by a `SetPwmRatio` action.

Similarly, when a channel's operating mode is changed to the Standard mode, its output driver is reset to the inactive state to prevent its output from going active until explicitly set by a `SetOutputs` action.

12.6.5 GetModes

Function Returns the operating modes for all channels.

Opcode `0x04`

Command `none`

Response `(M<7:0>),(M<15:8>)`

Parameter	Function
`M[]`	Byte array containing 16 bit flags that indicate the operating modes for all channels. Each bit is associated with a channel number. For example, bit 4 is associated with channel 4. Logic *one* indicates PWM mode, and *zero* indicates Standard mode.

Notes `GetModes` returns the current operating modes of all SSR channels. The operating modes may be changed under program control (e.g., in

response to a SetModes action) or unexpectedly (e.g., all channel operating modes revert to the default when a module reset occurs).

12.6.6 SetPwmRatio

Function Programs the PWM timing attributes for one SSR channel.

Opcode 0x05

Command (Channel),(OnTime),(OffTime)

Parameter	Function
Channel	Byte value that specifies the SSR channel that is to be configured. Value may range from 0 to 15.
OnTime	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments. Value may range from 0 to 255.
OffTime	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments. Value may range from 1 to 255.

Response none

Notes SetPwmRatio applies only to channels that have previously been configured for the PWM operating mode. This action has no effect on channels that are configured for the Standard operating mode.

The CERR status flag will be set if an illegal channel number is specified.

Two values are specified to establish the duty cycle: OnTime and OffTime. Each of these values specifies a time interval, expressed in 2 millisecond increments. For example, by programming OnTime to 1 and OffTime to 4, the duty cycle will be set to 20% with a period of 10 milliseconds (2 milliseconds active and 8 milliseconds inactive per output cycle).

The output duty cycle can be set to zero by declaring the OnTime value to be zero. It is not possible to set the duty cycle to 100% as the OffTime parameter must be at least 1. The OffTime value will automatically default to 1 if the client attempts to set it to zero.

After this action executes, the present channel output state will continue to be asserted for the duration of its previously programmed time, and then the new OnTime and OffTime will take effect.

12.6.7 GetPwmRatio

Function Returns the PWM timing attributes of one SSR channel.

Opcode 0x06

Command (Channel)

Parameter	Function
Channel	Byte value that specifies the SSR channel to be addressed. Value may range from 0 to 15.

Response (OnTime),(OffTime)

Parameter	Function
OnTime	On-time (output active) for each PWM cycle, expressed in 2 millisecond increments.
OffTime	Off-time (output inactive) for each PWM cycle, expressed in 2 millisecond increments.

Notes GetPwmRatio applies only to channels that have previously been configured for the PWM operating mode. The two returned byte values are meaningful only if the specified channel is configured for the PWM operating mode.

The CERR status flag will be set if an illegal channel number is specified.

Two values are returned: OnTime and OffTime. Each of these values specifies a time interval, expressed in 2 millisecond increments, that are currently in effect. These values will be the values programmed by the most recent SetPwmRatio action, or the default values if no SetPwmRatio action has executed since the channel's operating mode was switched to PWM.

12.7 Specifications

Table 41: Model 2653 Specifications

Interlock Power Daisy Chain	Voltage range	+5VDC to +24VDC, as required by SSRs.
	Current	10A maximum, per circuit.
Relays	Channels	16, with sockets.
	Form factor	Industry standard digital SSRs: 0.6W x 1.7L x 1.25H (inches).
	SSR types (not included with 2653)	Any combination of Grayhill 70- series or equivalent. Examples: 70-OAC24 - 120V AC output 70-OAC24A - 240V AC output 70-ODC24 - 60V DC output 70-ODC24R - mechanical relay output 70-IAC24 - 120V AC input 70-IAC24A - 240V AC input 70-IDC24 - 32V DC input Note: All of the above SSR types employ 24V logic. Part numbers will be different for other logic levels. Example: use 70-OAC5 for 120V AC output if the associated daisy-chained DC power supply is 5V.
Mating Connectors (*) = included with module	IOM port (qty 1)	RJ-45 plug, AMP 554169 or equivalent.
	Relay line/load (qty 16)	(*) AMP 172-166 with 770988-1 socket contacts (2), or equivalent.
	Power daisy chain (qty 2)	AMP 770849-6 with 640706-1 pin contacts (6), or equivalent. Accepts Sensoray cable assembly, part number 2600C1.
Temperature	Operating range	0 to 70°C.
Power	Input power	+24V ±5% @ 420mA, with eight 70-OAC24 SSRs energized.
Dimensions	Outer dimensions	5.0H x 11.1W x 3.0D inches, including relays and DIN mounting frame. Depth is specified with respect to surface of DIN rail support panel.