MISTIC PROTOCOL USER'S GUIDE

Form 270-100823 — August, 2010

OPTO 22

43044 Business Park Drive, Temecula, CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services:
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
E-mail: support@opto22.com
Web: support.opto22.com

Mistic Protocol User's Guide
Form 270-100823 — August, 2010

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or later are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, the OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

TABLE OF CONTENTS

WELCOME

OVERVIEW

A Mistic protocol distributed control system can have one or several computers or dedicated input/output (I/O) processors controlling from one to several hundred intelligent I/O units. The I/O units respond to a protocol that we call Mistic.

By performing most of the time-critical functions at the I/O unit level, system performance is optimized and is relatively independent of overall system size. Digital functions such as latching, counting, and time delays or pulse generation are performed by a 16-bit processor dedicated to 16 or fewer I/O points. Analog PID loops, high/low limit monitoring, and engineering unit conversions are performed by the analog Mistic brain board.

This manual describes the communication protocol required to communicate with all Mistic brain board types. Every Mistic command is fully explained and illustrated by a programming example. Detailed explanations of the functions each command performs will be helpful to users of our FactoryFloor software suite, as well as to users of our MisticWare software driver. While it is possible to create code to talk directly to the Mistic protocol brain boards, the use of a high-level language or our software driver is encouraged for most applications.

ABOUT THIS MANUAL

This manual is organized as follows:

- Chapter 1: Introduction — Features and configuration information
- Chapter 2: Programming — Structure of commands and errors
- Chapter 3: Technical Information — Command overview
- Chapter 4: Command Directory — Lists all the commands
- Chapter 5: Digital Setup System Commands
- Chapter 6: Digital I/O Configuration Commands
- Chapter 7: Digital Read/Write, Latch Commands
- Chapter 8: Digital Counter Frequency Commands
- Chapter 9: Digital Time Delay Pulse Output Commands
- Chapter 10: Digital Pulse/Period Measurement Commands

- Chapter 11: Digital Event/Reaction Commands
- Chapter 12: Analog Setup/System Commands
- Chapter 13: Analog I/O Configuration Commands
- Chapter 14: Analog Read/Write/Output Commands
- Chapter 15: Analog Event/Reaction Commands
- Chapter 16: Analog PID Loop Commands
- Appendix A: ASCII Character Table
- Appendix B: Surge Protection For RS-422/485 Communication Links
- Appendix C: Troubleshooting

DOCUMENT CONVENTIONS

- **Bold** typeface indicates text to be typed. Unless otherwise noted, such text may be entered in upper or lower case. (Example: "At the DOS prompt, type **cd \windows**.")
- *Italic* typeface indicates emphasis and is used for book titles. (Example: "See the *OptoControl User's Guide* for details.")
- File names appear in all capital letters. (Example: "Open the file TEST1.TXT.")
- Key names appear in small capital letters. (Example: "Press SHIFT.")
- Key press combinations are indicated by plus signs between two or more key names. For example, SHIFT+F1 is the result of holding down the SHIFT key, then pressing and releasing the F1 key. Similarly, CTRL+ALT+DELETE is the result of pressing and holdingthe CTRL and ALT keys, then pressing and releasing the DELETE key.
- "Press" (or "click") means press and release when used in reference to a mouse button.
- Menu commands are sometimes referred to with the Menu→Command convention. For example, "Select File→Run" means to select the Run command from the File menu.
- Numbered lists indicate procedures to be followed sequentially. Bulleted lists (such as this one) provide general information.

ABOUT OPTO 22

Opto 22's goal to deliver total control to industrial automation customers dates back to its beginnings in 1974 with the introduction of optically-isolated solid-state relays. Today, Opto 22 is the number one provider of I/O systems, with more than 60 million points of I/O working reliably worldwide. After earning a reputation for consistent innovation and leadership in automation hardware, Opto 22 realized it was time to take a new approach to control software. In 1988, Opto 22 introduced the first flowchart-based control programming language. Opto 22 continues to deliver successively more advanced generations of hardware and software.

Opto 22 has targeted four key markets: Enterprise, Systems, Components, and Verticals. Opto 22 provides bi-directional data exchange software for integrating manufacturing data into the Enterprise Market, software-driven control solutions for the Systems Market, hardware and drivers for the Components Market, and Vertical Market solutions for the water, energy, and semiconductor industries.

All Opto 22 products are manufactured in the U.S. at the company's headquarters in Temecula, California, and are sold through a global network of distributors, system integrators, and OEMs. Sales offices are located throughout the United States.
For more information, contact Opto 22, 43044 Business Park Drive, Temecula, CA 92590-3614. Phone Opto 22 Inside Sales at 1-800-452-OPTO or Opto 22 headquarters at 951-695-3000. Fax us at 951-695-3095.

For fast delivery of marketing and technical documentation, visit our Web site at **www.opto22.com**.

CHAPTER 1

Introduction

Overview

Scope of Document

This manual describes the format of the data required to communicate with the family of Opto 22 I/O (input/output) controllers using the Mistic protocol. The user of this manual may either be analyzing the data created by a Mistic protocol controller in an effort to optimize the use of the high-level commands, or will be creating the data in some other program written in a high-level language. Data is exactly the same whether created by a Mistic protocol Opto 22 controller, a PC using our MisticWare I/O driver, or from custom-designed software.

Data link topology, overall system design, and the layout and connection of specific I/O units is beyond the scope of this document. Please refer to the individual hardware user's guides for specific analog and digital I/O units for this level of detail.

I/O Unit Overview

The Opto 22 Mistic protocol I/O units are a family of intelligent digital and analog input/output controllers that operate as slave devices to a host computer. Physically, each Mistic I/O unit is a modular assembly that consists of a module mounting rack, a brain board, and a number of optically-isolated analog or digital I/O modules.

The removable brain board contains a microprocessor that communicates with a Mistic processor or with a host computer equipped with the proper interface card. The brain board controls the plug-in I/O modules located on the I/O mounting rack.

Any combination of analog I/O modules may be plugged into an analog I/O unit and likewise, any combination of digital I/O modules may be plugged into a digital I/O unit. Remote analog and digital I/O units can then be combined on the same RS-485 serial data link providing endless combinations of analog and digital I/O points. In a similar way, any combination of local analog and digital I/O units can be combined on the same parallel data link.

Local analog and digital I/O units use flat ribbon cable for interconnections. The maximum cable length for local I/O units is 200 feet. Local units should be used within the same equipment rack or enclosure where cable runs can be kept short. The primary reason for choosing local I/O units (rather than remote) is the much higher bus speed of 1.4 MHz. The remote bus speed is 115 kHz. The higher speed of the local I/O unit allows data to be sent and received in a much shorter time interval. This is very important for real-time control applications.

Remote I/O units communicate with the host computer over an RS-485 serial communications link. The serial data link is composed of two twisted-pairs of cable (4 wires) that connect to each I/O unit. One pair carries commands and responses and the other pair is used for a hardware interrupt from the I/O units. It is possible to communicate with up to 256 individual I/O units on a single serial link for a total of 4,096 digital and analog I/O points.

The RS-485 data link offers excellent noise immunity and long cable lengths. The maximum cable length for remote I/O units is 3,000 feet. Wiring costs are reduced by eliminating the need for bringing large bundles of field wiring back to a central controller. Remote I/O units can be located where the actual control is needed.

System Configuration

Physical Layout

The high cost of electrical wiring and the noise susceptibility of analog signals make it desirable to place the control and monitoring points as close to the controlled device as possible. Remote I/O units offer design flexibility, with as few as eight points or as many as 4,096 points in one physical location.

Considerable installation savings and improved system performance can be realized by placing the control points (remote I/O units) close to the controlled device. The serial data link between adjacent remote I/O units consists of two twisted pairs (4 wires) which makes it practical to have a remote I/O unit located at each machine on a Opto 22 FactoryFloor®, at each heating or air conditioning unit in an energy management application, or at each instrument in a data acquisition environment.

Each Mistic I/O unit must be assigned its own unique address (0 – 255).

Communications

Remote I/O units require only two twisted pairs for the communications link. Communications are half duplex. The command and the response travel over the same twisted pair. The second twisted pair is required for hardware interrupts.

In a multidrop configuration, the last I/O unit in the network can be up to 3,000 feet from the host computer, even longer using a repeater, and up to 256 I/O units can be connected to a single serial port. In this configuration, all of the remote I/O units are wired in parallel on the same serial data link. This provides security against power failures at individual I/O unit sites. If power is lost on a remote I/O unit, communications to that board are lost; but communication to the rest of the remote I/O units on the link remains intact.

Modular Construction

Each Mistic I/O unit is composed of a mounting base with barrier strips, a brain board, and high-density, optically-isolated analog or digital I/O modules. Please refer to individual I/O unit data sheets for specific construction and connection details.

Serial Data Transmission-Remote I/O Units

Remote I/O units are connected to the host computer via a two twisted-pair cable. One pair carries command/response data and the other carries interrupts. Remote I/O units support the common baud rates (110, 150, 300, 600, 1,200, 2,400, 4,800, 9,600, 19.2K, 38.4K, 57.6K, 76.8K, and 115.2K) which are selectable via jumpers located on the remote I/O unit. Selection of baud rate often depends upon the capabilities of the host port. Many host computers have a limited baud rate. Modems and radio links often operate at 300, 1,200, or 2,400 baud. System throughput is increased by using the fastest available baud rate.

Parallel Data Transmission-Local I/O Units

The local I/O units communicate with the host computer via a 34-conductor, flat-ribbon cable. Local I/O units always operate at 1.4 MHz for high-speed data transfers. System throughput is increased by using local I/O units. M4 integral I/O is always configured on the local bus.

Protocol

All I/O units use the same command set. For remote I/O units, two types of protocols are supported, (1) ASCII and (2) binary. For local I/O units, only the binary protocol is supported. The binary protocol is used for faster data transmission and ASCII protocol is used when it is desirable to be able to view the commands and responses on an ASCII display terminal, usually an LCD or CRT type. Both protocols are 2-pass. The Mistic protocol requires the transmission of two messages on the serial data link every time a command is executed. The host sends a command to a I/O unit and then receives a response message acknowledging successful execution of the command along with any requested data, or an error message indicating that the I/O unit detected an error in the command message and was unable to execute it.

Data Verification

To ensure secure data transmission, every I/O unit command message and every response from a I/O unit includes a data verification field. There are two data verification methods. They are as follows:

1. Checksum modulo 256 (8-bit). Two checksum characters are sent and received in ASCII protocol and one byte in binary protocol.
2. CRC16. Four checksum characters are sent and received in ASCII protocol and two bytes in binary protocol.

The above data verification methods are selected by means of jumpers on the Mistic I/O unit. In addition to the jumpers, there is a command (Command C = "Set System Options") that allows you to further customize the CRC data verification method to match your system needs. This command is used to set or clear bits in the Option Control Byte. The bits that affect the CRC calculation are described below:

Option Control Byte

```
Bit 4 = CRC initialization value: 0 = 0000, 1 = FFFF.
Bit 5 = CRC method select: 0 = reverse, 1 = classical.
Bit 6 = CRC polynomial select: 0 = CRC16, 1 = CCITT.
```

The factory-default value for the Option Control Byte is 00. The default CRC data verification method will be CRC16 reverse with an initial value of 0000.

The Option Control Byte is stored in EEPROM when command E is executed and is restored upon power-up or when the RESET command is executed.

Features

System performance and throughput are increased by allowing the I/O unit to do as much processing as possible, thereby reducing communication link activity and host computer processing. The following summary of I/O unit capabilities will help to determine which tasks can be performed by the I/O unit.

Digital Commands

In addition to ON/OFF control, digital I/O units provide the following:

Input Latches

All input positions can be used to record momentary events. Each input has two associated latches that can be read or cleared. One latch is positive-edge triggered (by an OFF-to-ON transition) and the other latch is negative edge triggered (by an ON-to-OFF transition).

Time Delays

All output positions can function in the time delay mode. Outputs can be set to operate with four types of delays.

- Delay before turning OFF
- Delay before turning ON
- Pulse ON
- Pulse OFF

Time delays are programmable with a resolution of 100 microseconds. Delay times of 429,497 seconds (4.97 days) are possible (32 bits).

Pulse Generation

I/O units can be instructed to generate pulses on output channels as follows:

1. OFF Pulse.
2. ON Pulse.

3. Generate N Pulses.
4. Generate Continuous Square (Rectangular) Waves.
5. Generate Continuous Pulses With Time Proportional Output Period and Programmable TPO Percentage (Duty Cycle).

ON and OFF pulses are retriggerable. ON and OFF times are independently-programmable. You can set any output channel to generate pulses with a fixed period and then vary the duty cycle with another command (while period remains fixed). Pulses are programmable with a resolution of 100 microseconds. Pulse durations of up to 429,497 seconds (4.97 days) are possible (32 bits). Both the ON time and the OFF time must be greater than 1 millisecond for continuous square wave outputs. If the period of the square wave function is less than 10 milliseconds, then only eight channels can be active at any one time.

Event Counting

All input positions can function as event counters to return a count of external events. The count ranges from 0 to 4,294,967,296. All 32 bits can be read, or just the least significant 16 bits of the counter can be read. Each of the event counters can be individually enabled, read, stopped, and cleared. Frequencies of up to 25,000 Hz can be counted.

Pulse/Period Measurements

All of the input channels can measure the duration of a positive or negative pulse or a period. The input channel can be programmed to measure positive pulses, negative pulses, or periods. Resolution is 100 microseconds. Measurements can be read and restarted at any time. The user can read all 32 bits or just the least significant 16 bits of the measurement.

On Time/Off Time Totalization

All of the input channels can be configured to totalize the duration of ON or OFF pulses. Resolution is 100 microseconds.

Frequency Measurements

All of the input positions can be programmed to measure frequency. Sampling rates can be in increments of 0.1 seconds or 1.0 seconds. Frequency measurements are 0 to 25 kHz.

Configuration and Status

The input or output status (on or off) and channel configuration can be read at any time.

Store System Configuration

System configuration and event/reaction entries may be stored in EEPROM and restored upon power-up or reset.

Reset

A Reset command generates a hardware reset.

Watchdog Timer

The I/O unit has a watchdog timer which will monitor the communications link. If the communications link has no activity for a preset time interval, the I/O unit can set all outputs to a predetermined state. Watchdog delay times may be from 200 milliseconds to 10.9 minutes. The watchdog function is disabled unless enabled by command.

Event/Reactions

The I/O unit may be instructed to monitor the state of the I/O or the value of a counter channel and take action if needed. Up to 256 events may be monitored. Each event has an associated latch which is set when the event has occurred. These event latches may be read or read and cleared by command. A reaction command may also be set to execute upon an event occurrence. Please see Chapter 3 (Technical Information) for a detailed explanation of the event/reaction processor.

Interrupts

The I/O unit may be instructed to generate a host CPU interrupt upon any event occurrence. Remote I/O units require an additional twisted pair for interrupts. Local I/O units do not require any additional cables.

Analog Commands

Analog I/O units support input or output functions with a resolution of 12 bits. This provides a range from 0 to 4,095 counts. Additional capabilities include the following:

Offset and Gain Calibration

Each input channel may have an offset added and/or have a gain coefficient multiplied. This allows calibration to be done by software control.

Engineering unit Scaling

Each analog I/O channel may be scaled to use the engineering units of the application, as well as the 12-bit range from 0 to 4,095. The engineering unit scaling is done automatically when the I/O channel is configured with the I/O module type. Configuring a channel as a generic input or output allows the user to set any scaling range from -32,768 to 32,767.

Input Averaging and Totalization

Each analog input channel may have an averaging function which allows for digital filtering of noisy signals. Both input and output channels may be instructed to totalize (integrate) their respective data. Averaging is done every 100 milliseconds. Totalization may be computed every 100 milliseconds or any multiple thereof.

Output Ramping

Output channels can be instructed to ramp to any set point within the range of the module.

Configuration and Status

The input or output status (value) and channel configuration can be read at any time.

Store System Configuration

System configuration, initial output values, event/reaction entries, and PID parameters may all be stored in EEPROM and restored upon power-up or reset.

Reset

A Reset command generates a hardware reset.

Watchdog Timer

The I/O unit has a watchdog timer which will monitor the communications link. If the communications link has no activity for a preset time interval, the I/O unit can set all outputs to a predetermined value. Watchdog delay times may be from 200 milliseconds to 10.9 minutes. The watchdog function is disabled unless enabled by command.

Event/Reactions

The I/O unit may be instructed to monitor the values of input or output data and take action if a preset limit is exceeded. Up to 256 events may be monitored. Each event has an associated latch which is set when the event has occurred. These event latches may be read or read and cleared by command. A reaction command may also be set to execute upon an event occurrence. Please see Chapter 3 (Technical Information) for a detailed explanation of the event/reaction processor.

Interrupts

The I/O unit may be instructed to generate a host CPU interrupt upon any event occurrence.

PID Loop Control

Up to eight PID loops may be controlled by one I/O unit. The PID input (process variable) and the set point can come from any input or output point on the I/O unit as well as from the host computer via the data link. The PID output must be a channel on the I/O unit. Please see Chapter 3 (Technical Information) for a detailed explanation of the PID function performed by the I/O unit.

Summary

In laying out an installation, plan to route the RS-485 data link cable to all points where you need to install remote I/O units now, and to all points where you may need control in the future.

Make provision to supply DC power to each I/O unit location, preferably with a local power supply. If you are using Opto 22 panels, then the +24 VDC power supplies are a part of the Mistic panels.

Choose a baud rate and protocol compatible with your host computer, and make this information available to the hardware installer — it is nessary when setting the jumpers on each remote I/O unit.

Assign a unique address to each I/O unit. Addresses may appear in any order from beginning to end of the data link, however every address must be unique — no two controllers may share the same address. There is no requirement for consecutive addresses; the entire range of addresses may be used. Try to relate the address to a location or function.

Outline the functions of the application software. Take advantage of the Mistic I/O unit's processing capability to off-load host computer processing time and to reduce data link activity.

CHAPTER 2

Programming

Overview

The Mistic I/O unit is an intelligent device that acts as a slave to a host computer or a dedicated Mistic processor. The host computer issues instructions to the Mistic I/O unit by sending command message frames over a communications link. The addressed Mistic I/O unit responds to the host by sending message frames back.

Remote I/O units are connected to the host computer via a two twisted-pair cable. Therefore, the serial communications are half-duplex. Maximum baud rate is 115K baud. If interrupts are used, then another twisted pair is required. The local I/O units communicate with the host computer via a 34-conductor, flat-ribbon cable. Local I/O units always operate at 1.4 MHz for high-speed data transfers. System throughput is increased by using local I/O units.

All Mistic I/O units use the same command set. For Remote I/O units, two types of protocol are supported, (1) ASCII and (2) binary. For local I/O units, only the binary protocol is supported. The binary protocol is used for faster data transmission and ASCII protocol is used when it is desirable to be able to view the commands and responses on an ASCII display terminal, usually an LCD or CRT type. Both protocols are 2-pass. The Mistic protocol requires the transmission of two messages on the serial link every time a command is executed. The host sends a command to an I/O unit and then receives a response message acknowledging successful execution of the command along with any requested data, or an error message indicating that the I/O unit detected an error in the command message and was unable to execute it.

To ensure secure data transmission, every command message and every response from a Mistic I/O unit includes a data verification field. There are two data verification methods. They are as follows:

1. Checksum modulo 256 (8-bit). Two checksum characters are sent and received in ASCII protocol and one byte in binary protocol.
2. CRC16 or CCITT. Four checksum characters are sent and received in ASCII protocol and two bytes in binary protocol.

The host computer calculates the data verification field data (checksum or CRC) and sends it along as part of the command message to a Mistic I/O unit. When a Mistic I/O unit receives the message, it calculates its own data verification field value and compares that value with the transmitted checksum. If they match, the Mistic I/O unit can verify that the message was received correctly.

The same procedure is repeated whenever a Mistic I/O unit returns data to the host computer. When a message is received from a Mistic I/O unit, the host computer calculates the data verification field data (checksum or CRC) and compares it against the data verification field that was transmitted as part of the message.

Note: *For users who will be using a high-level language such as "C" on an IBM-PC or PC-compatible to communicate with Mistic I/O units, Opto 22 offers a software communication driver, MisticWare. The driver includes C and assembly language source code, a command tutorial, and a rich set of utilities with examples and diagnostics. The driver takes care of building command messages, calculating the data verification field data (checksum or CRC) and processing the response.*

Error Messages

Mistic I/O units are capable of detecting and reporting errors to the host computer. Mistic I/O units always return an acknowledgment message after successfully executing a command. If a Mistic I/O unit detects an error, it will *not* execute the command and an error message containing a code describing the exact nature of the error is returned.

The error codes are:

01 UNDEFINED COMMAND

The command character was not a legal command character.

02 CHECKSUM OR CRC ERROR

The checksum or CRC received by the Mistic I/O unit did not match the value calculated from the command message.

03 BUFFER OVERRUN ERROR

The receive buffer limit of 256 characters has been exceeded. The command was ignored.

04 POWER-UP CLEAR ERROR

After a power failure (4.8 VDC or lower) or a reset command, you must issue a Power-Up Clear Command before issuing any other command. If you do not, you will receive this error code. This alerts the host that a power failure has occurred.

05 DATA FIELD ERROR

Not enough characters received.

06 COMMUNICATIONS LINK WATCHDOG TIME-OUT ERROR

The communications link watchdog timer has timed out and the specified actions have been taken. The command that returns this error is not executed but it does clear the error.

07 SPECIFIED DATA INVALID ERROR

One or more data fields contains an illegal value.

08 *Busy Error*

This error is used by the LC Communicator to indicate a setup mode condition.

09 *Invalid Module Type Error*

This error code is returned when one or more specified modules is not of the type required by the command.

10 *Invalid Event Entry Error*

This error code is returned when an attempt is made to enable an event interrupt on a null entry in the event reaction table or an illegal reaction command is specified.

11 *High Resolution Time Delay Limit Reached - Digital Only*

This error code is returned when an attempt is made to start a square wave, generate N pulses or time-proportional output with a delay time value less than 10 milliseconds on more than eight output positions.

Summary

To implement an application, the user must do the following:

(1) Understand how to communicate with a Mistic I/O unit. This will require:

- The ability to build command messages.
- The ability to carry out message transactions.
- The ability to interpret data returned by a Mistic I/O unit.

(2) Understand how to manage a Mistic I/O unit network.

This will require the ability to initialize and configure each Mistic I/O unit in the network (a serial data communication link for remote I/O units or a 34-pin, flat-ribbon cable for local I/O units).

Building Command Messages

The structure of all Mistic command messages (frames) is shown in the reference sections of this manual. Use this as a guide for building commands.

Numbers Representation in ASCII Protocol

The ASCII protocol for the Mistic I/O unit uses the Hexadecimal (Hex) numbering system to represent numerical data in commands and responses. Appendix A provides an explanation of binary and Hex numbering systems.

ASCII protocol means that command messages are transmitted as a series of ASCII characters. Numbers are transmitted as the ASCII number characters 0 through 9 and the upper case ASCII characters A through F. We will refer to an ASCII character representing a Hex number as “ASCII-Hex.”

Example:

15 (decimal) = F (Hex) and is transmitted as the ASCII “F” character.

8-Bit Checksum

For ASCII protocol, the 8-bit checksum is computed by adding the decimal values of all the ASCII characters in the command frame (string), excluding the start of command character ">", the DVF (data verification field), and the CR (carriage return) character at the end. This sum is then divided by 256 and the integer remainder is converted to two ASCII-Hex digits.

Example:

>08KC01289cr is a valid command.
08KC012 is the part used to calculate the checksum.

Note: *The Start of Command character ">" is NOT part of the checksum calculation.*

The checksum is calculated by summing the decimal values of the ASCII characters which make up the command.

```
ASCII characters :   0  8      K      C      0      1      2
value of characters: 48 56     75     67     48     49     50
48+56+75+67+48+49+50 = 393
393/256 = 1 remainder 137
137 decimal = 89 hex
```

The checksum (89 Hex) is appended to the end of the command string. The complete command then becomes ">08KC01289cr."

For binary protocol, the 8-bit checksum is computed by adding the decimal values of all the bytes in the command frame EXCLUDING the one byte in the data verification field DVF at the end. This sum is then divided by 256 and the integer remainder is the checksum.

Example:

```
4E 02 5F AF (hex) is a valid command in binary protocol with
8-bit checksum.
4E 02 5F is the part used to calculate the 8-bit checksum.
4E + 02 + 5F = AF hex is the 8-bit checksum.
```

16-Bit CRC

The 16-bit CRC (cyclical redundancy check) value is computed by using one of the four algorithms shown below. The algorithms are written in C language and each uses a different lookup table.

/*

FUNCTION NAME: *CRCFUNS.C*

DESCRIPTION: *Calculates 16-bit CRC Values for:*

```
1. CRC16            X**16 + X**15 + X**2 + 1
2. CRC16 REVERSE    X**16 + X**14 + X**1 + 1
3. CCITT            X**16 + X**12 + X**5 + 1
4. CCITT REVERSE    X**16 + X**11 + X**4 + 1
```

OTHER MODULES: *none*

COMMENTS: *C Functions to compute 16-bit CRCs using lookup tables.*

```
*/
unsigned short crc16rev (str,cnt)
unsigned char *str;
unsigned int cnt;
{
       unsigned short crc;
       unsigned int index;
       crc = 0;
       while( cnt-- )
       {
               index = crc ^ *str++;
               crc = (crc >> 8) ^ crc16r[index & 0x00ff];
       }
       return(crc);
}

unsigned short ccittrev (str,cnt)
unsigned char *str;
unsigned int cnt;
{
       unsigned short crc;
       unsigned int index;
       crc = 0;
       while( cnt-- )
       {
               index = crc ^ *str++;
               crc = (crc >> 8) ^ ccittr[index & 0x00ff];
       }
       return(crc);
}

unsigned short crc16c (str,cnt)
unsigned char *str;
unsigned int cnt;
{
       unsigned short crc;
       unsigned int index;
       crc = 0;
       while( cnt-- )
       {
               index = (crc >> 8) ^ *str++;
               crc = (crc << 8) ^crc16[index];
       }
       return (crc);
}

unsigned short ccittc(str,cnt)
unsigned char *str;
unsigned int cnt;
{
```

```
        unsigned short crc;
        unsigned int index;
        crc = 0;
        while( cnt-- )
        {
                index = (crc >> 8) ^ *str++;
                crc = (crc << 8) ^ ccitt[index];
        ]
        return (crc);
}
```

Each of the algorithms uses a different lookup table. The tables are generated by means of the C language program shown below.

```
/*
```

PROGRAM: *TBLGEN.C*

DESCRIPTION: *Calculates 16-bit CRC lookup table values for:*

```
        1. CRC16            X**16 + X**15 + X**2 + 1
        2. CRC16 REVERSE    X**16 + X**14 + X**1 + 1
        3. CCITT            X**16 + X**12 + X**5 + 1
        4. CCITT REVERSE    X**16 + X**11 + X**4 + 1
```

OTHER MODULES: *none*

COMMENTS: *Generates the four header files crc16.h, crc16r.h, ccitt.h, and ccittr.h.*

Include these header files in your main( ) C program.

```
*/
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#define CCITT         0x1021
#define CCITT_REV     0x8408
#define CRC16         0x8005
#define CRC16_REV     0xa001
unsigned int crcch( );
unsigned int crcchrev( );
unsigned int hexasc( );
main()
{
        unsigned int  poly, data;
        unsigned int  a,b,z;
        unsigned int  *ptr;
        unsigned char buf[1000],*btr;
        unsigned char fname[20],aname[20];
        FILE    *fout;

        for( z = 0 ; z <4 ; z++ )
        {
                if( z == 0 )
```

```
{
        strcpy(fname,"CCITT.H");
        strcpy(aname,"ccitt[256]");
        poly = CCITT;
}
if( z == 1 )
{
        strcpy(fname,"CCITTR.H");
        strcpy(aname,"ccittr[256]");
        poly = CCITT_REV;
}
        if( z == 2 )

{
        strcpy(fname,"CRC16.H");
        strcpy(aname,"crc16[256]");
        poly = CRC16;
}
if( z == 3 )

{
        strcpy(fname,"CRC16R.H");
        strcpy(aname,"crc16r[256]");
        poly = CRC16_REV;
}
        if( (fout = fopen( fname , "w" )) == 0 )
{
        printf( "\n file %s cannot be opened \n%c",fname,07 );
        printf(" \n Press any key to continue...");
        getch();
        return(0);
}
strcpy(buf,"\n unsigned int " );
strcat(buf,aname);
strcat(buf," = \n\t{ \n\t" );
fwrite(buf,1,strlen(buf),fout);
printf("%s",buf);
b = 0;
btr = buf;
for( a = 0 ; a != 256 ; a++ )
{
        if( poly == CCITT || poly == CRC16 )
          data = crcch(poly,a);
        else
          data = crcchrev(poly,a);
        strcpy(btr,"0x");
        btr += 2;
        *btr++ = hexasc(data/4096);
        *btr++ = hexasc(data/256);
        *btr++ = hexasc(data/16);
        *btr++ = hexasc(data);
        if( a != 255 )
        {
          strcpy(btr,", ");
          btr += 2;
        }
          if( ++b == 0x08)
        {
```

```
                          b = 0;
                          strcpy(btr,"\n\t");
                          fwrite(buf,1,strlen(buf),fout);
                          printf("%s",buf);
                          btr = buf;
                        }
                }
                strcpy(buf,"};\n\n");
                fwrite(buf,1,strlen(buf),fout);
                printf("%s",buf);
                fclose(fout);
}
}
unsigned int crcch(poly,data)
unsigned int poly, data;
{
        unsigned int a,i,d;
        a = 0;
        data < = 8;
        for( i = 8 ; i > 0 ; i-- )    {
                        if(( data ^ a ) & 0x8000 )
                a = ( a < 1) ^ poly;
                else
                        a < = 1;
                        data < = 1;
        }
        return( a );
}
unsigned int crcchrev(poly,data)
unsigned int poly, data;
{
        unsigned int a,i,d;
        a = 0;
        data < =1;
        for( i = 8 ; i > 0 ; i-- ){
                data > = 1;
                if(( data ^ a ) & 0x0001 )
                        a = ( a > 1 ) ^ poly;
                else
                        a >= 1;
        }
return( a );
} /* convert nibble to ASCII hex char, return char */
unsigned int hexasc(nib)
unsigned int nib;
{
        unsigned char c = 48;
nib = nib & 0xf;
if(nib9)
                c += 7;
c += nib;
return c;
}
```

The program TBLGEN.C generates the following CRC lookup tables as header files:

```
unsigned int crc16 [256] =          {

    0x0000,  0x8005,  0x800F,  0x000A,  0x801B,  0x001E,  0x0014,  0x8011,
    0x8033,  0x0036,  0x003C,  0x8039,  0x0028,  0x802D,  0x8027,  0x0022,
    0x8063,  0x0066,  0x006C,  0x8069,  0x0078,  0x807D,  0x8077,  0x0072,
    0x0050,  0x8055,  0x805F,  0x005A,  0x804B,  0x004E,  0x0044,  0x8041,
    0x80C3,  0x00C6,  0x00CC,  0x80C9,  0x00D8,  0x80DD,  0x80D7,  0x00D2,
    0x00F0,  0x80F5,  0x80FF,  0x00FA,  0x80EB,  0x00EE,  0x00E4,  0x80E1,
    0x00A0,  0x80A5,  0x80AF,  0x00AA,  0x80BB,  0x00BE,  0x00B4,  0x80B1,
    0x8093,  0x0096,  0x009C,  0x8099,  0x0088,  0x808D,  0x8087,  0x0082,
    0x8183,  0x0186,  0x018C,  0x8189,  0x0198,  0x819D,  0x8197,  0x0192,
    0x01B0,  0x81B5,  0x81BF,  0x01BA,  0x81AB,  0x01AE,  0x01A4,  0x81A1,
    0x01E0,  0x81E5,  0x81EF,  0x01EA,  0x81FB,  0x01FE,  0x01F4,  0x81F1,
    0x81D3,  0x01D6,  0x01DC,  0x81D9,  0x01C8,  0x81CD,  0x81C7,  0x01C2,
    0x0140,  0x8145,  0x814F,  0x014A,  0x815B,  0x015E,  0x0154,  0x8151,
    0x8173,  0x0176,  0x017C,  0x8179,  0x0168,  0x816D,  0x8167,  0x0162,
    0x8123,  0x0126,  0x012C,  0x8129,  0x0138,  0x813D,  0x8137,  0x0132,
    0x0110,  0x8115,  0x811F,  0x011A,  0x810B,  0x010E,  0x0104,  0x8101,
    0x8303,  0x0306,  0x030C,  0x8309,  0x0318,  0x831D,  0x8317,  0x0312,
    0x0330,  0x8335,  0x833F,  0x033A,  0x832B,  0x032E,  0x0324,  0x8321,
    0x0360,  0x8365,  0x836F,  0x036A,  0x837B,  0x037E,  0x0374,  0x8371,
    0x8353,  0x0356,  0x035C,  0x8359,  0x0348,  0x834D,  0x8347,  0x0342,
    0x03C0,  0x83C5,  0x83CF,  0x03CA,  0x83DB,  0x03DE,  0x03D4,  0x83D1,
    0x83F3,  0x03F6,  0x03FC,  0x83F9,  0x03E8,  0x83ED,  0x83E7,  0x03E2,
    0x83A3,  0x03A6,  0x03AC,  0x83A9,  0x03B8,  0x83BD,  0x83B7,  0x03B2,
    0x0390,  0x8395,  0x839F,  0x039A,  0x838B,  0x038E,  0x0384,  0x8381,
    0x0280,  0x8285,  0x828F,  0x028A,  0x829B,  0x029E,  0x0294,  0x8291,
    0x82B3,  0x02B6,  0x02BC,  0x82B9,  0x02A8,  0x82AD,  0x82A7,  0x02A2,
    0x82E3,  0x02E6,  0x02EC,  0x82E9,  0x02F8,  0x82FD,  0x82F7,  0x02F2,
    0x02D0,  0x82D5,  0x82DF,  0x02DA,  0x82CB,  0x02CE,  0x02C4,  0x82C1,
    0x8243,  0x0246,  0x024C,  0x8249,  0x0258,  0x825D,  0x8257,  0x0252,
    0x0270,  0x8275,  0x827F,  0x027A,  0x826B,  0x026E,  0x0264,  0x8261,
    0x0220,  0x8225,  0x822F,  0x022A,  0x823B,  0x023E,  0x0234,  0x8231,
    0x8213,  0x0216,  0x021C,  0x8219,  0x0208,  0x820D,  0x8207,  0x0202
      };
```

```
unsigned int crc16r [256] = {
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
    0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
    0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
    0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
    0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040.
      };
```

```
unsigned int ccitt [256] =  {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
    0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
    0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
    0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
    0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
    0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
    0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
    0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
    0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
    0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
    0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
    0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
    0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
    0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
    0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
    0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
    0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
    0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
    0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
    0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
    0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
    0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
    0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
      };
```

```
unsigned int ccittr [256] = {
    0x0000, 0x1189, 0x2312, 0x329B, 0x4624, 0x57AD, 0x6536, 0x74BF,
    0x8C48, 0x9DC1, 0xAF5A, 0xBED3, 0xCA6C, 0xDBE5, 0xE97E, 0xF8F7,
    0x1081, 0x0108, 0x3393, 0x221A, 0x56A5, 0x472C, 0x75B7, 0x643E,
    0x9CC9, 0x8D40, 0xBFDB, 0xAE52, 0xDAED, 0xCB64, 0xF9FF, 0xE876,
    0x2102, 0x308B, 0x0210, 0x1399, 0x6726, 0x76AF, 0x4434, 0x55BD,
    0xAD4A, 0xBCC3, 0x8E58, 0x9FD1, 0xEB6E, 0xFAE7, 0xC87C, 0xD9F5,
    0x3183, 0x200A, 0x1291, 0x0318, 0x77A7, 0x662E, 0x54B5, 0x453C,
    0xBDCB, 0xAC42, 0x9ED9, 0x8F50, 0xFBEF, 0xEA66, 0xD8FD, 0xC974,
    0x4204, 0x538D, 0x6116, 0x709F, 0x0420, 0x15A9, 0x2732, 0x36BB,
    0xCE4C, 0xDFC5, 0xED5E, 0xFCD7, 0x8868, 0x99E1, 0xAB7A, 0xBAF3,
    0x5285, 0x430C, 0x7197, 0x601E, 0x14A1, 0x0528, 0x37B3, 0x263A,
    0xDECD, 0xCF44, 0xFDDF, 0xEC56, 0x98E9, 0x8960, 0xBBFB, 0xAA72,
    0x6306, 0x728F, 0x4014, 0x519D, 0x2522, 0x34AB, 0x0630, 0x17B9,
    0xEF4E, 0xFEC7, 0xCC5C, 0xDDD5, 0xA96A, 0xB8E3, 0x8A78, 0x9BF1,
    0x7387, 0x620E, 0x5095, 0x411C, 0x35A3, 0x242A, 0x16B1, 0x0738,
    0xFFCF, 0xEE46, 0xDCDD, 0xCD54, 0xB9EB, 0xA862, 0x9AF9, 0x8B70,
    0x8408, 0x9581, 0xA71A, 0xB693, 0xC22C, 0xD3A5, 0xE13E, 0xF0B7,
    0x0840, 0x19C9, 0x2B52, 0x3ADB, 0x4E64, 0x5FED, 0x6D76, 0x7CFF,
    0x9489, 0x8500, 0xB79B, 0xA612, 0xD2AD, 0xC324, 0xF1BF, 0xE036,
    0x18C1, 0x0948, 0x3BD3, 0x2A5A, 0x5EE5, 0x4F6C, 0x7DF7, 0x6C7E,
    0xA50A, 0xB483, 0x8618, 0x9791, 0xE32E, 0xF2A7, 0xC03C, 0xD1B5,
    0x2942, 0x38CB, 0x0A50, 0x1BD9, 0x6F66, 0x7EEF, 0x4C74, 0x5DFD,
    0xB58B, 0xA402, 0x9699, 0x8710, 0xF3AF, 0xE226, 0xD0BD, 0xC134,
    0x39C3, 0x284A, 0x1AD1, 0x0B58, 0x7FE7, 0x6E6E, 0x5CF5, 0x4D7C,
    0xC60C, 0xD785, 0xE51E, 0xF497, 0x8028, 0x91A1, 0xA33A, 0xB2B3,
    0x4A44, 0x5BCD, 0x6956, 0x78DF, 0x0C60, 0x1DE9, 0x2F72, 0x3EFB,
    0xD68D, 0xC704, 0xF59F, 0xE416, 0x90A9, 0x8120, 0xB3BB, 0xA232,
    0x5AC5, 0x4B4C, 0x79D7, 0x685E, 0x1CE1, 0x0D68, 0x3FF3, 0x2E7A,
    0xE70E, 0xF687, 0xC41C, 0xD595, 0xA12A, 0xB0A3, 0x8238, 0x93B1,
    0x6B46, 0x7ACF, 0x4854, 0x59DD, 0x2D62, 0x3CEB, 0x0E70, 0x1FF9,
    0xF78F, 0xE606, 0xD49D, 0xC514, 0xB1AB, 0xA022, 0x92B9, 0x8330,
    0x7BC7, 0x6A4E, 0x58D5, 0x495C, 0x3DE3, 0x2C6A, 0x1EF1, 0x0F78
      };
```

The following program written in "C" gives a simple example of how to use the CRC functions and the lookup tables to calculate the CRC values for Command F, using the binary protocol. See page 5-1 of this manual.

```
/*
```

PROGRAM NAME: *CRCTST.C*

DESCRIPTION: *Test program to compute 16-bit CRC values for:*

```
1.  CRC16              X**16 + X**15 + X**2 + 1
2.  CRC16 REVERSE      X**16 + X**14 + X**1 + 1
3.  CCITT              X**16 + X**12 + X**5 + 1
4.  CCITT REVERSE      X**16 + X**11 + X**4 + 1
```

OTHER MODULES: *none*

COMMENTS: *Example program illustrating the use of C functions from the file "CRCFUNS.C" to compute 16-bit CRCs.*

```
*/
#include <stdio.h>
       /* Include the header files generated by TBLGEN.C */
       #include "ccitt.h"
       #include "ccittr.h"
       #include "crc16.h"

#include "crc16r.h"
        /* Include the CRC functions for CRC computations */
#include "crcfuns.c"
main()
{
       static unsigned char message[3] = {0xC9,0x03,0x46};
       unsigned short crc;
       crc = crc16rev(message,3);
       printf("CRC16 reverse for message = %x \n",crc);
       crc = ccittrev(message,3);
       printf("CCITT reverse for message = %x \n",crc);
       crc = crc16c(message,3);
       printf("CRC16 classic for message = %x \n",crc);
       crc = ccittc(message,3);
       printf("CCITT classic for message = %x \n",crc)
}
```

The output of the above program is:

```
CRC16 reverse for message = 3c51
CCITT reverse for message = 9bde
CRC16 classic for message = 8423
CCITT classic for message = c537
```

16-Bit CRC Calculation Examples

For ASCII protocol, all characters in the command frame except for the START OF COMMAND CHARACTER “>”, the DVF (data verification field), and the CR (carriage return) character at the end are used in computing the 16-bit CRC value. Before the command frame is transmitted, the 16-bit CRC value is converted to ASCII Hex (four characters) and placed in the data verification field (DVF). Finally, a carriage return (ASCII character 0D Hex) is appended to the end of the command frame in the CR field.

Example:

For ASCII protocol >C9F7662cr is a valid command with CRC16 reverse DVF (CRC16 reverse is the factory default).

The part used to calculate the CRC checksum is C9F. The “>” character is not used for ASCII protocol CRC calculations. The data verification field DVF = 7662 is for CRC reverse. This value is converted to ASCII-Hex and placed in the data verification field DVF as the four ASCII characters “7,” “6,” “6,” and “2.” Then a carriage return (ASCII character 0D Hex) is appended to the end of the command frame in the CR field.

For binary protocol, all characters in the command frame except for the two data verification field DVF bytes at the end are used in computing the 16-bit CRC value. Before the command frame is transmitted, the 16-bit CRC value is placed in the data verification field (DVF).

Example:

For binary protocol, C903463C51 is a valid command with a 16-bit CRC value in data verification field DVF. The CRC was computed using the CRC16 reverse algorithm.

The parts used to compute the CRC value are the three bytes C9 Hex, 03 Hex, and 46 Hex. The “C” language function “crc16rev( )” was used to compute the CRC16 reverse value. The function returned the value 3C51 Hex. The two bytes 3C Hex and 51 Hex are placed in the data verification field DVF of the command frame.

Carrying Out Message Transactions

The details of how to actually write the command message out to a host adapter card in the host computer will be found in the manual that comes with your host adapter card. The method will be the same as sending characters out of a COM1 or COM2 serial port on your PC. Receiving the response messages is the same as receiving characters on your COM1 or COM2 serial port. The Opto 22 local adapter card is made to function the same as the Opto 22 remote adapter card when communicating with Mistic I/O units. Thus, communications with remote or local I/O units are virtually identical.

One difference between the remote adapter and the local adapter is that the remote adapter must be configured for the proper baud rate, number of data bits, number of stop bits, and parity. You would set these parameters to match the settings of the Mistic protocol I/O units (brain board settings).

For ASCII protocol, you would have 1 start bit, 8 data bits, 1 stop bit, and no parity. There is nothing special about data transmission in ASCII protocol.

For binary protocol, Opto 22 uses a 9th data bit (the parity bit) to indicate whether an address byte or a data byte is being transmitted. If an address byte is being transmitted, the 9th data bit is set to a one (1). If a data byte is transmitted, the 9th data bit is a zero (0). If you are using a PC with an Opto 22 adapter card *or* if you are using a host computer with an 8250 UART chip, you can program the UART to use a stuck parity bit (as the 9th data bit) which can be set to a 1 or a 0. When you are transmitting an address byte, you will need to set the stuck parity bit to a 1 and when you are transmitting a data byte, you need to set the stuck parity bit to a 0.

Opto 22 supplies a software driver which will take care of all the details of communicating to I/O units. It will do all the error checking, initialization, building the command strings, etc. If you are writing your own driver, you will need to do all this yourself.

Interpreting the Response

The Mistic I/O unit will respond to each command that contains its address in the address data field AA. The structure of the response will depend upon the command given. The reference section of this manual gives a complete description of the response to each command. If the command was not executed for some reason, the response will contain an error code. The meaning of these error codes is described on page 2-2 of this manual. Each response message frame has a field which can be examined to determine if the command was executed or if there was an error. For ASCII protocol, the first character in the response is an A if the command was successful or is a N if there was an error. The error code is given following the N. For binary protocol, the error code EC field will be zero if there was no error and non zero if there was an error.

Examples of commands and responses are given for every command in the reference section of this manual.

Interpreting Analog Data

This section applies only to analog I/O units (brain boards).

All analog input and output values are exchanged between the host computer and a Mistic I/O unit in either 16-bit or 32-bit data fields. When the data is given in counts (raw data), a 16-bit data field DDDD is used. When the data is given in engineering units, a 32-bit data field DDDDDDDD is used.

Data fields DDDDDDDD are 32 bits in length and represents a *signed* integer value in engineering units. Except for output module type 80 and input module type 00, the scaling parameters for engineering units are determined automatically when you specify the module type (by command "G" or command "a"). For example, if you specify that module 15 is an Opto 22, DA4 analog output module (0 to 5 VDC in engineering units), then the data field DDDDDDDD represents volts in increments of 1/65,536 of a volt. The range is 0 to 5 VDC. If DDDDDDDD exceeds 00050000 Hex = 5 VDC, it will be clipped to 00050000 Hex = 5 VDC. If DDDDDDDD is less than 00000000, it will be set to 00000000.

Engineering units are real-world units of measurement such as pressure, temperature, PSIG, DC volts, etc. For a 12-bit Opto 22 analog module, the numbers that you read from or write to on an analog module are normally in the range of 0 to 4,095 (some modules allow some under-range and over-range). You can scale the inputs and/or outputs from the analog modules to represent engineering units. For example, you may wish to scale the input so that the readings 0 to 4,095 correspond to 4 to 40 milliamperes. Command "f" allows you to set the engineering scaling units for anything that you wish.

Even though the engineering units are specified in increments of 1/65,536 of an engineering unit, the resolution of the analog module is still only 12 bits. The upper 16 bits of DDDDDDDD represent the integer part of the data and the lower 16 bits represents the fractional part of the data.

For output module type 80 Hex and input modules type 00, you can define your own engineering units by using command "f". The default will be 0 to 4,095 counts unless you use command "f" to set different scaling parameters. For example, assume that you have used command "a" to specify that input module 13 is generic type 00 Hex. Further assume that you want the engineering units to be 0 to 3,000 pounds per sq. in. pressure. You would use command "f" to set the proper scaling parameters. Thereafter, DDDDDDD would represent pounds pressure in increments of 1/65,536 of a pound. The maximum value that you could use for DDDDDDDD would be 0BB80000 Hex = 3,000 PSI and the minimum value would be 00000000 = 0 PSI.

The scaling parameters can be saved in EEPROM by command E. On power up, the parameters are restored from EEPROM memory.

Note that setting the scaling parameter does not change the *actual* input or *actual* output for the specified module, only the scaling factor. For example, if you have an Opto 22, DA8, 0 to 20 mA module (Configuration Type 88) and you rescale it for 0 to 20 amperes engineering units, you must still *only* apply 0 to 20 mA input to the module.

Initializing a Mistic Network

Many of the operating characteristics of a Mistic I/O unit can be selected by the host computer. On power up or after a RESET command, Mistic I/O unit initializes itself to a set of default characteristics stored in EEPROM. The defaults are shown on page 5-12 for digital and on pages 12-11, 12-12, and 12-14 for analog.

The host computer is responsible for sending each Mistic I/O unit in the network all of the commands necessary to select the desired characteristics.

For digital I/O units, initialization usually involves sending the command POWER-UP CLEAR and any setup or configuration command needed to configure your modules as counters, timers, frequency measurement, etc. You will also need to build your event/reaction tables if they are used. Many of the parameters can be saved in EEPROM and will be automatically restored upon power up. See "Store System Configuration" command on page 6-12.

For analog I/O units, initialization usually involves sending the command POWER-UP CLEAR and any setup or configuration command needed to configure your modules types. You will also need to configure your PID Loops, build your event reactions tables, set scaling parameters, set gain, offset ramping, etc. Many of the parameters can be saved in EEPROM and will be automatically restored upon power up. See "Store System Configuration" command on page 13-38.

Note: *Mistic I/O unit initialization should be always be performed when the I/O unit goes through a reset condition. The reset condition can occur whenever a RESET command is sent OR when power is lost and later restored. Unless a POWER UP CLEAR command is sent, the Mistic I/O unit will respond with an error message (error code 04) whenever you send it your first command. The purpose of this error is to warn you that Mistic I/O unit has lost its configuration and needs to be initialized. A power-up condition can also be caused by a momentary dip on the +5 VDC line which causes Mistic I/O unit's processor to reset.*

Mistic Brain Board Jumper Settings

Please refer to the data sheets for specific Mistic I/O units for hardware configurations and jumper settings.

Technical Information

Digital I/O Units

Input/Output Timing

All digital I/O units will read the I/O status or set the output state at the time the command is issued. When a "Read Module Status" command is received, the digital I/O unit will read the current status then transmit the response data. Upon receipt of a "Set Output Module State" command, the digital I/O unit will affect the new output status before the acknowledge response is sent.

Input Latching

Input latches do not need to be configured and are always enabled. A latch exists for both positive (OFF to ON) and negative (ON to OFF) transitions.

Input Counting

Any or all channels may be configured as input counters. Each counter channel has an enable/disable switch associated with it. A count is incremented on the positive (OFF to ON) transition of the input status. Maximum count rate is 25 kHz. Care should be taken with regard to signal and wire routing when dealing with high speed counters. Avoid running sensitive high speed data lines with high current/high voltage power lines.

Counter channels that are enabled upon execution of the "Store System Configuration" command will be enabled upon power-up, or execution of a "Reset" command.

Quadrature Counting

Note: G4IDC5Q module is limited to 20,000 Hz. The quadrature input module will convert a quadrature signal to a pulse stream which is output on the logic side of one of the two input channels. The channel which receives the pulse stream is determined by the direction of rotation of the encoder. The hardware of the digital I/O unit counts pulses into one channel for clockwise travel, and the other channel for counter-clockwise travel. When the two input channels are configured as a quadrature input pair, the I/O unit will automatically compute a position count by subtracting the count of the even-numbered channel from the count of the odd-numbered channel. The position count is returned when either channel is read, and both counters are cleared when either is specified in a "Clear Counter" or "Read and Clear Counter" command.

The quadrature input module will output a pulse each quadrature transition. The actual resolution of the position count is four times the encoder resolution (pulses per revolution).

Since both channels of a quadrature counter pair are always counting up, they must be cleared to zero at some point before one or the other rolls over (exceeds FFFFFFFF Hex). At a maximum input rate of 20,000 pulses per second from the G4IDC5Q module (5,000 encoder pulses per second), the time before roll-over is 59 hours.

Quadrature input counter channels must be configured in pairs, with the lower channel number being even. Therefore the only quadrature counter pairs allowed are channels 0 & 1, 2 & 3, 4 & 5, 6 & 7, 8 & 9, 10 & 11, 12 & 13, and 14 & 15.

Since the digital I/O unit has a maximum input count rate, the maximum allowable RPM in which the optical encoder may turn will be related to the number of cycles per turn the encoder outputs.

The relationship between the two numbers is such:

Maximum Allowable RPM = 300,000 / Quadrature Pulses per Revolution

Figures for Some Encoder Resolutions are Below:

Encoder Pulses/ Revolution	Max RPM
60	5,000
100	3,000
120	2,500
200	1,500
256	1,171
300	1,000
360	833
400	750
500	600
720	416
900	333
1,000	300
2,048	146

Quadrature counter channels that are enabled upon execution of the "Store System Configuration" command will be enabled upon power-up, or execution of a "Reset" command.

Pulse/Period Duration Measurement

Pulse and period duration measurement is edge-triggered. Channels configured to measure positive pulses will start timing upon the first positive edge (OFF to ON transition) and stop timing upon the next negative edge (ON to OFF transition). Channels configured to measure negative pulses will start timing upon the first negative edge (ON to OFF transition) and stop timing upon the next positive edge (OFF to ON transition). Channels configured for period measurement will start timing upon the next immediate transition, positive or negative, and complete the measurement on the next matching transition. All times will be in units of 100 microseconds.

A "Pulse Complete" bit is set after measurement is completed. If a "Read and Restart" command is issued for a channel that is in the process of measuring a pulse, measurement is stopped and will restart on the next edge for which it is configured. Data will be returned for the measurement to that point.

Pulse and period measurements are not done continuously. Only the next immediate pulse or period after a "restart" is measured. The data is held in memory for reading by the host CPU.

On Time/Off Time Totalization

Input channels may be configured to measure the total time an input has been ON or OFF. Channels configured as ON time totalizers will sum the duration of ON pulses while OFF time totalizers will sum the duration of OFF pulses. All times will be in units of 100 microseconds. Maximum total time that can be measured before rollover is 4.97 days. Totalized times are read using the "Read Pulse Duration" or "Read and Restart Pulse Duration" commands.

The "Read and Restart Pulse Duration" command will return the sum to that point and reset it to zero. If the pulse is active at the time, this command is executed, measurement will continue from zero. This differs from the single pulse measurement configuration where an edge is needed to restart a new measurement. Only after power-up, reset or configuration of the channel is an edge needed to start measurement.

Frequency Measurement

Frequency measurement resolution is selectable for either 1.0 Hz or 10 Hz. Bit 0 of the Option Control Byte is used to select frequency range and affects all channels on the I/O unit configured to measure frequency. If 1.0 Hz resolution is selected frequency measurements are updated once every second. Selecting 10 Hz resolution allows updates every 0.1 second. Maximum frequency input is 25 kHz.

Time Delay/Pulse Output

Any output channel may be instructed to perform a delay function. Possible functions include the following:

Start On Pulse/Delay Before Off

The "Start On Pulse" command instructs an output channel to turn on for a specified time, then turn off. If the output is on at the time the command is executed, it will turn OFF after the delay expires. Delay times are programmable from 500 microseconds to 4.97 days with a resolution of 100 microseconds. The delay is retriggerable.

Start Off Pulse/Delay Before On

The "Start Off Pulse" command instructs an output channel to turn off for a specified time, then turn on. If the output is off at the time the command is executed, it will turn on after the delay expires. Delay times are programmable from 500 microseconds to 4.97 days with a resolution of 100 microseconds. The delay is retriggerable.

Start Continuous Square (Rectangle) Wave

An output channel may be instructed to continuously turn on and off at regular intervals. On and off times need not be the same and are programmable from one millisecond to 4.97 days. Only eight output channels with delay times of less than 10 milliseconds may be active at any one time. This limitation applies only to channels outputting continuous square wave or generate N pulse functions.

Generate N Pulses

This function allows an output channel to turn on and off continuously for a specified number of cycles. On and off times need not be the same and are programmable from 1 millisecond to 4.97 days. Only eight output channels with delay times of less than 10 milliseconds may be active at any one time. This limitation applies only to channels performing continuous square wave or generate N pulse functions.

Time Proportional Output (TPO)

Any output channel may be set to function as a time-proportional output. The period is programmable from 100 milliseconds to 4.97 days. Once set the period remains fixed while the duty cycle may vary from 0 to 100 percent. The 'Set TPO Percentage' command selects the amount of time the output is to be on. Setting a TPO percentage to zero will cause the output to remain off. Setting it to 100 percent will cause the output to remain on.

Analog I/O Units

Analog Input/Output Timing

Analog I/O units constantly update the status of their I/O. Input modules are read every 7 milliseconds and the data is held in memory until requested by the host CPU. Output module data is held in memory and output to each module every 50 milliseconds.

All analog I/O data can be scaled to the engineering units required for the application. In the case of temperature input modules, default units are degrees centigrade. Setting bit 0 of the Option Control Byte will instruct the analog I/O unit to use degrees Fahrenheit. This bit affects all temperature conversion done by the I/O unit.

Refer to chapter 2, page 16, "Interpreting Analog Data" for an detailed explanation of the analog I/O unit engineering unit scaling.

Execution of a "Store System Configuration" command will save the current settings of all output channels. This setting will become the initial output value upon power-up or execution of a "Reset" command.

Offset and Gain

Offset

The analog I/O unit can be instructed to add an offset input module readings. This is an easy method of subtracting off minor variations. For example, if a 0- to 5-volt input is being used (G4AD6), and the lowest possible input to the module is 0.01 volts, it may be desirable to consider 0.01 volts as zero scale. By issuing a "Calculate and Set ADC Module Offset" command to the I/O unit while the input is at 0.01 volts, an offset of -0.01 volts while be added to each reading from that point on.

Gain

The analog I/O unit may also be instructed to multiply input readings with a gain coefficient. This may be necessary if it is not possible to input full-scale for a particular module. For example, if a 0- to 5-volt input is being used, but the highest possible input signal is 4.8 volts, it may be desirable to consider 4.8 volts as full-scale. By executing a "Calculate and Set ADC Module Gain" command while 4.8 volts is at the input, a Gain coefficient is calculated and multiplied with the input readings thereafter. In this case, 4.8 volts would read as 5.0 volts after the gain is set.

Offset and Gain parameters are stored in EEPROM upon execution of a "Store System Configuration" command and are restored on power-up or execution of a "Reset" command.

Note: *The Offset and Gain features should only be used to compensate for small deviations in sensor range. Use of these commands for scaling a module with a wide range to a scale with a smaller range, (for instance, using a 0 to +10 V module as a 3 to 5 V module), will NOT improve resolution. Instead, the values will tend to make large jumps for small increments of input.*

Input Averaging (Digital Filtering)

An input channel may be instructed to digitally filter its input readings. The computation is done every 100 milliseconds. The algorithm used is an implementation of a first order lag digital filter. The equation being solved is:

```
New Average = ((Current Reading-Old Average)/Weight)+Old Average
```

Weight is related to the sampling rate and time constant for the filter by the following equation:

```
Weight = (Sampling Rate+Time Constant)/Sampling Rate
```

Since the sampling rate is fixed at 100 milliseconds, this equation becomes:

```
Weight = (0.1+Time Constant) Ã 10
```

Where Time Constant is the desired time constant of the filter in seconds.

An averaged reading is only meaningful if the input module has been functioning properly continuously. Input modules which receive signals too far under-range (below -1.25 percent of span) will stop relaying data to the analog I/O unit. When this happens, the I/O unit will stop the averaging function and set the averaged reading to -32,768 as an error indication. This can occur when a 4-20 milliampere signal goes below 3.8 milliamperes, or a thermocouple input becomes an open circuit.

If the input signal is restored, and the module begins to operate properly, the averaging function must be restarted by the host CPU. This may be done by issuing another "Set Averaging Sample Weight" or "Read and Clear I/O Module Data" command for this channel.

Totalization (Integration)

Any input or output channel may be instructed to totalize its readings. The sample rate is programmable from 100 milliseconds to 54.6 minutes with a resolution of 100 milliseconds. The channel can be instructed to use the averaged (filtered) reading as the sample for totalization.

A totalized reading is only meaningful if the input module has been functioning properly continuously.

Input modules which receive signals too far under-range (below -1.25 percent of span) will stop relaying data to the analog I/O unit. When this happens, the I/O unit will stop the totalizing function and set the totalized reading to -32,768 as an error indication. This can occur when a 4-20 milliampere signal goes below 3.8 milliamperes, or a thermocouple input becomes an open circuit.

If the input signal is restored, and the module begins to operate properly, the totalization function must be restarted by the host CPU. This may be done by executing another "Set Totalization Sample Rate" or "Read and Clear I/O Module Data" command for this channel.

PID Loop Control

Up to eight PID loops may be controlled by one analog I/O unit. Any input or output channel may be specified as the loop input (process variable) while any output channel may be selected as the controlled output. The setpoint variable may be set by command or from an input or output channel. The output of one loop may be the input or setpoint of another loop, allowing cascade control. Loops may also share input or setpoint channels.

It is important to be sure that the engineering units of the setpoint channel are the same as for the input channel (process variable).

All eight loops are calculated every 100 milliseconds, unless the scan rate is set for a longer interval. Scan rate is programmable from 100 milliseconds to 109 minutes, with 100 millisecond resolution.

The PID calculation being performed is a velocity algorithm implemented in the form of the difference equation: $\Delta\ Om = \pm\ G\ [(e_m - e_{m-1}) + (I*T*e_m) + (D/T)(e_m - 2\ e_{m-1} + e_{m-2})]$

ΔOm = Change in output for the current scan cycle in engineering units.

e_m = Loop error term (setpoint - PID loop input) for the current scan cycle in engineering units.

e_{m-1} = Loop error term (setpoint - PID loop input) for the previous scan cycle in engineering units.

e_{m-2} = Loop error term (setpoint - PID loop input) for the second previous scan cycle in engineering units.

G = Loop gain. This term can be positive or negative. G is negative if the process direction is reverse. Default is 1.00. Gain must be in the range of - 32,767 to + 32,767.

I = Integral reset constant term in repeats per minute. Default is 0.0 repeats per minute.

D = Derivative rate constant term in minutes. Default is 0.0 repeats per minute.

T = Time interval between scans in seconds. Default is 100 milliseconds. Minimum time interval is 100 milliseconds.

This velocity algorithm is inherently "anti-wind-up" since it has no summation in the integral term to saturate. The algorithm is described on pages 160-162 of "Microprocessors in Instruments and Control" by Robert J. Bibbero published by John Wiley and Sons.

If for any reason the input for a PID loop is not functioning (such as an open thermocouple circuit) the calculation will not be performed and the output will remain unchanged. In this case, a flag is set in the loop status byte. If the input signal is restored, and the module begins to operate properly, the PID calculations will resume.

The setpoint and output signals may have minimum and maximum limits applied to them. If enabled, when a limit is reached, the value is clamped at the limit value until again within range. Event/reaction commands may also be used to effect alarm responses.

Manual mode for a PID also has an "Output Track Input" function which can be used as a signal converter.

Event/Reaction Processing

Event/reaction processing is a powerful feature that enables the programmer to reduce communications time by off-loading certain tasks to the I/O unit. Each I/O unit has the ability to monitor its I/O status for user-defined conditions and take action if necessary. Throughout this discussion, the condition being watched for is referred to as an "event." The action to be taken upon an event occurrence is referred to as a "reaction."

Event/reaction processing is done using a table of up to 256 (128 for each B3000 logical address) event/reaction "entries." When not executing commands from the host CPU, the I/O unit is constantly scanning its I/O and the event/reaction table for event matches. When a match is found, the reaction command for that entry is executed and a corresponding event latch is set. The event/reaction table is scanned from entry 0 on up. Entries which have not been programmed are considered "null entries" and are passed over. The I/O unit recognizes the last programmed entry and will only scan to that point. Therefore, scan time is minimized by using contiguous entries starting from 0.

Digital I/O units may be programmed to watch for the following events:

(1) A Must On - Must Off match of the I/O status.

(2) A counter channel value greater than or equal to a setpoint.

(3) A counter channel value less than or equal to a setpoint.

(4) A communications link watchdog time-out.

Digital I/O units may be programmed to execute the following commands upon an event occurrence:

(1) Null Reaction, set latch only.

(2) Set Output Module(s) State.

(3) Start On Pulse.

(4) Start Off Pulse.

(5) Enable/Disable Counter.

(6) Clear Counter.

(7) Enable/Disable Event Entry.

(8) Enable/Disable Group of Event Entries.

(9) Read and Hold Counter Value.

Analog I/O units may be programmed to watch for the following events:

(1) An I/O channel reading/setting greater than or equal to a setpoint.

(2) An I/O channel reading/setting less than or equal to a setpoint.

(3) A communications link watchdog time-out.

Analog I/O units may be programmed to execute the following commands upon an event occurrence:

(1) Null Reaction, set latch only.

(2) Set Output Channel Value.

(3) Ramp Output Channel to Setpoint.

(4) Enable/Disable PID Loop.

(5) Set PID Loop Setpoint.

(6) Set PID Loop Min-Max Output Limits.

(7) Enable/Disable Event Entry.

(9) Enable/Disable Group of Event Entries.

(10) Read and Hold I/O Channel Value.

Reaction commands are executed (and the event latch set) each time the event status changes from a false (non-matching) to a true (matching) condition. If a true condition exists at the time the event/reaction entry is enabled, the reaction command will be executed upon the next scan. Once an event has occurred, and its associated reaction command executed, the I/O unit must see the event condition change from the true to the false state before the event/reaction can occur again. Event latches are set each time the event occurs and are cleared only by command from the host CPU.

Each event/reaction table entry has an enable/disable switch. Entries that have not been enabled (or have been disabled) will be passed over when scanned. This can prevent the I/O unit from recognizing that a change has occurred from a true to a false event condition, or vice versa.

Event/Reaction table entries 0 – 31 are saved to EEPROM when a "Store System Configuration" command is executed. These are restored on power-up, or when a "Reset" command is executed. Entries which are enabled when saved to EEPROM are enabled when restored.

Interrupts

Each event/reaction table entry may be instructed to generate a host CPU interrupt upon an event occurrence. Each entry has an interrupt enable/disable switch associated with it, and the I/O unit has a global interrupt enable/disable switch that is bit 7 of the Option Control Byte.

Before an I/O unit can generate an interrupt, a number of commands must be executed. These are:

(1) Set Event Table Entry
An entry must be made instructing the I/O unit to monitor for a specific event occurrence.

(2) Enable Event Table Entry
The event entry must be enabled for scanning by the I/O unit.

(3) Set Event Interrupt Status
The event entry interrupt must be enabled.

(4) Set System Options
The Global Interrupt Enable bit must be set.

And, of course, the event being monitored must then occur.

An interrupt from an I/O unit is cleared by executing a "Read and Clear Event Latches" command. All event entries on a specific I/O unit that are set to generate an interrupt should be specified when reading and clearing as the interrupt output is cleared by the command.

If more than one I/O unit is set up to generate interrupts, the host CPU must poll each one to determine which interrupt is active. This may be done with a "Read Event Latches" or "Read and Clear Event Latches" command. If possible, it is best to put all signals that may generate interrupts into one I/O unit. This will eliminate the need to poll more than one board and speed up the host CPU interrupt response time.

Event/Reaction Applications

Digital I/O Units

Emergency Stop

A very good application for the event/reaction function is the case of an emergency stop input. Without the ability to monitor and react at the I/O unit, the host CPU must execute a read status command at frequent intervals to determine if an emergency stop input has become active. If so, it must then send commands to take corrective action. This can slow down the communications throughput for the entire process, as well as the response time for an emergency condition.

With the event/reaction function, it is possible to instruct the I/O unit to monitor for an emergency stop condition, take immediate corrective action, and interrupt the host CPU if necessary. Note that the I/O unit can only affect its own output modules as a reaction to an event occurrence. If it is necessary to change outputs on other I/O units, the same emergency stop signal can be wired to them, or the host CPU may intervene.

Counter and Gate Control

A digital I/O unit may be set up to count the number of items that pass by a sensor and divert a gate upon reaching a full count. The I/O unit can then reset the counter and start the process over again all without host CPU intervention.

Quadrature Input and Positioning Control

The Quadrature Counting function of a digital I/O unit can be used to measure the absolute position along one axis. If the motor speed can be controlled with output modules, (i.e. fast, medium, slow) then the event/reaction function can be used to slow the motor drive down as it approaches the desired position. This eliminates the need for the host CPU to constantly read the counter to determine if the required position is being approached. Allowing the I/O unit to do this function will allow faster motion speeds and eliminate the possibility of overshoot.

State Machine-Drum Sequencer

Since the digital I/O unit can change the state of any or all of its output modules when an input counter is equal to a setpoint, it is a simple matter to program a sequence of output states. The timing of the state change can be controlled by the input counter frequency. If necessary, this frequency can be generated by the I/O unit (an output connected to an input) to make a standalone sequencer.

Analog I/O Units

Input Alarm-Limit Sensing

The event/reaction function is used to detect out-of-bound conditions of input signals. The user/programmer may select any setpoint(s) within the range of the input module. If the selected setpoint is reached, an event latch is set and, if desired, a reaction command executed. Reactions may be anything from setting an output value to disabling a PID Loop. If necessary, the host CPU may be interrupted to handle the situation.

Output Ramping Waveforms

The event/reaction function may be used to create custom output waveforms by changing the slope of an output upon reaching a particular setpoint. Sawtooth and Triangle waveforms can be achieved with two event entries. Trapezoidal functions may require the use of a time delay. This can be accomplished by performing a ramp function on an unused position.

System Throughput

In most applications, system throughput is greatly affected by the rate at which I/O can be updated. For intelligent I/O units this involves sending a command string, processing of the command, then an acknowledge and/or data being returned. The time required for any transaction will be equal to the transmission time of all data in the command and response, plus the time required by the I/O unit to process the command.

The local bus has a much higher data transfer rate than RS485 serial communications, therefore, systems using local I/O units will have a greater throughput than systems using remote I/O units.

Time Required to Read 16 I/0 Positions (Typical)

I/0 Unit	Communication/ Data Transfer	Command Execution	Total
Local Digital (Non-multifunction)	.08 msec	.10 msec	.18 msec
Local Digital (Multifunction)	.08 msec	.60 msec	.68 msec
Local Analog	.28 msec,	.75 msec	1.03 msec
Remote Digital	1.06 msec *	.70 msec	1.76 msec
Remote Analog	4.23 msec *	1.30 msec	5.53 msec
* All serial communication at 115.2K baud.			

Time Required to Write 16 Output Positions (Typical)

I/0 Unit	Communication/ Data Transfer	Command Execution	Total
Local Digital (Non-multifunction)	.08 msec	.10 msec	.18 msec
Local Digital (Multifunction)	.06 msec	.94 msec	1.00 msec
Local Analog	.23 msec	2.25 msec	2.48 msec
Remote Digital	1.27 msec *	1.00 msec	2.27 msec
Remote Analog	4.32 msec *	2.20 msec	6.52 msec
* All serial communication at 115.2K baud.			

Time Required to Read 1 I/0 Position (Typical)

I/0 Unit	Communication/ Data Transfer	Command Execution	Total
Local Analog	.08 msec	.35 msec	.43 msec
Remote Analog	1.25 msec *	.38 msec	1.63 msec
* All serial communication at 115.2K baud.			

Time Required to Write 1 I/0 Position (Typical)

I/0 Unit	Communication/ Data Transfer	Command Execution	Total
Local Digital (Non-multifunction)	.05 msec	.40 msec	.45 msec
Local Analog	.08 msec,	.38 msec	.46 msec
Remote Digital	.98 msec *	.40 msec	1.38 msec
Remote Analog	1.15 msec *	.40 msec	1.55 msec
* All serial communication at 115.2K baud.			

The G4LS Non-Multifunction Local Digital I/O Units

The G4LS is a digital I/O unit designed for simple ON-OFF control. Since the G4LS has a simplified command set, it is able to execute and respond faster than multifunction I/O units.

The G4LS only executes four commands. They are:

1. Power-Up Clear, command A.
2. Identify Type, command F.
3. Read Module Status, command R.
4. Set Output Module State-Group, command J.

The G4LS only uses 8-bit checksum as its data verification method.

CHAPTER 4

Command Directory

This chapter provides a quick reference and index for all commands. Chapters 5 – 16 cover the commands in detail. Chapters 5 – 11 cover the digital commands and Chapters 12 – 16 cover the analog commands. A command description sheet is given for each command.

The command description sheet for each command in the following chapters is formatted as follows:

COMMAND NAME COMMAND CMD

The top line shows the NAME of the command at the left of the line and the command character CMD at the right following the word COMMAND. CMD is always a single ASCII character.

DESCRIPTION

Tells briefly what the command does.

VERSIONS

Indicates which I/O units support the command.

ASCII COMMAND FORMAT

Shows the correct format for the command message frame when ASCII Protocol is selected.

ASCII protocol is selected when jumper four of the group labelled "BAUD" is not installed. This protocol is used when it is desirable to have all the characters in the command string be seen on a standard computer terminal connected to the communications line. The computer terminal must be equipped with the proper hardware interface (i.e. RS-422, RS-232 etc.). All characters in the string will be displayed on the terminal except for the Ending Delimiter character which is a carriage return. The computer terminal should have the auto line feed feature enabled so that each carriage return will start a new line on the computer terminal display screen.

All the fields that make up the command message frame will be shown in the following general format. Command Name COMMAND CMD.

number of bytes →

1	2	1	0 or more	2 or 4	1
SD	AA	CMD	CDF	DVF	ED

Where: SD = Starting Delimiter; AA = Address
CMD = Command; CDF = Command Data Fields
DVF = Data Verification Field; ED = Ending Delimiter

The Starting Delimiter will always be the ASCII character > (3E Hex). The Ending Delimiter is always an ASCII carriage return (0D Hex).

The Address field is two bytes long. It contains two ASCII characters that represent an address (00 to FF Hex). For example, if the address of the I/O unit is 29 (1D Hex), then the two bytes transmitted would be an ASCII 1 (31 Hex) and an ASCII D (44 Hex). The ASCII character 1 is transmitted first.

The CMD field will be one or two ASCII characters, the command characters. The name of the command and the command characters are shown on the top line.

Some commands require that additional Command Data Fields (CDF) be transmitted. These Command Data Fields provide the additional information needed by the I/O unit before it can execute the command. For example, the SET OUTPUT command requires a Command Data Field that tells the I/O unit which output channel to set. Some commands do not require any additional data fields while other more powerful commands require many data fields. Some data fields are optional and if not provided by the host computer, default values are used. The command description sheet will show the format of each Command Data Field (CDF).

The Data Verification Field (DVF) requires two bytes or four bytes depending upon which of data verification method is selected. One 8-bit checksum method and one 16-bit CRC16 method is provided. See Chapter 2 for an explanation of how these bytes can be constructed by the host computer for transmission to the I/O unit.

ASCII RESPONSE FORMAT

Shows the format of the response message frame when ASCII protocol is selected.

The response message frame is sent from the I/O unit to the host in response to the above ASCII command.

The fields that make up the response message frame will be shown in two general formats. The first shows the format of the response message frame transmitted by the addressed I/O unit for a successful completion of the command that was sent by the host. The second shows the format of the response message frame transmitted by the addressed I/O unit when an error occurred.

Success Response Message Frame

number of bytes →

1	0 or more	2 or 4	1
A	RDF	DVF	CR

Where: A = ASCII Character A (41 Hex) RDF = Response DataFields
DVF = Data Verification Field CR = ASCII Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = ASCII Carriage Return (0D Hex)

The Response Data Fields (RDF) are used for commands that require data to be transmitted back to the host computer. The command description sheet will show the format of each Response Data Field (RDF) when return data is required. For example, the “Read Frequency Measurement” command requires 32-bit frequency measurement data to be returned to the host computer.

When the I/O unit encounters an error in executing a command, it places an Error Code (EC) in the EC field of the Error Response Message Frame. See Chapter 2 for the meaning of the Error Codes.

The Data Verification Field (DVF) field will have two bytes or four bytes depending upon which of data verification method is selected. The will compute the checksum or CRC per the selected method and place it in the Data Verification Field of the response message frame.

BINARY COMMAND FORMAT:

Shows the correct format for the command message frame when binary Protocol is selected.

The command frame is sent from the host to the addressed I/O unit where it is executed. The binary protocol is selected when jumper four of the group labelled "BAUD" is installed. This protocol is used when speed is important. It reduces the number of characters in the command frame and the response frame. It also frees the I/O unit from the task of having to check all characters on the communications line.

The fields that make up the command message frame will be shown in the following general format.

number of bytes →

1	1	1 or 2	0 or more	1 or 2
AA	LEN	CMD	CDF	DVF

Where: AA = Address, LEN = Length Field
CMD = Command, CDF = Command Data Fields
DVF = Data Verification Field

The Address is a one-byte binary number (unsigned integer) in the range 0 to 255.

The Length field is a one-byte binary number (unsigned integer) in the range 0 to 255. It gives the number of bytes that will be transmitted following the Length field. The length will be the frame length (FL) - 2 or stated as an equation LEN = FL - 2.

CMD is the command field, one or two ASCII command characters.

Some commands require additional Command Data Fields (CDF) to provide the additional information needed by the I/O unit before it can execute the command. When Command Data Fields (CDF) are required, the command description sheet will show the format of each additional field. There are no optional data fields in the binary protocol.

The Data Verification Field (DVF) field requires one byte or two bytes depending upon which of data verification method is selected. One 8-bit checksum method and one 16-bit CRC16 method is provided. Jumper 5 of the group labelled "BAUD" is used to select the data verification method. When the jumper is omitted, the data verification is 8-bit checksum. When the jumper is installed, the data verification is 16-bit CRC16. See Chapter 2 for an explanation of the jumper selections and how these bytes can be constructed by the host computer for transmission to the I/O unit.

BINARY RESPONSE FORMAT:

Shows the correct format for the response message frame when binary protocol is selected.

The response message frame is sent from the I/O unit to the host in response to the above binary command.

All the fields that make up the response message frame will be shown in two general formats. The first shows the format of the response message frame transmitted by the addressed I/O unit for a successful completion of the command that was sent by the host. The second shows the format of the response message frame transmitted by the addressed I/O unit when an error occurred.

Success Response Message Frame

number of bytes →

1	1	0 or more	1 or 2
LEN	EC=0	RDF	DVF

Where: LEN = Length Field EC = Error Code = 0 (Zero)
RDF = Response Data Fields DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field EC = Error Code (Non Zero)
DVF = Data Verification Field

The Length field is a one-byte binary number (unsigned integer) in the range 0 to 255. It gives the number of bytes that will be transmitted following the Length field. The length will be the frame length (FL) - 1 or stated as an equation LEN = FL - 1.

The Error Code field is a one-byte binary number (unsigned integer). The I/O unit places a zero (00 Hex) in the Error Code field of the Success Response Message Frame to indicate that no errors occurred and the command was successfully executed. A non-zero value is placed in the Error Code field of the Error Response Message Frame to indicate that an error occurred. See Chapter 2 for the meaning of the Error Codes.

Some commands require data to be transmitted back to the host computer. One or more Response Data Fields (RDF) are added to the Success Response Message Frame when return data is required. The command description sheet will show the format of each Response Data Field (RDF).

The Data Verification Field (DVF) field will have one bytes or two bytes depending upon which data verification method is selected. The I/O unit will compute an 8-bit checksum or a 16-bit CRC16 per the selected method and place it in the Data Verification Field of the response message frame.

Remarks

Describes in detail how the command is constructed and used.

Example

Shows actual command messages to and responses from the I/O unit that demonstrate the use of the command.

Command Summary

The following summary lists the commands by related groups. The version indicates whether the command is an A(nalog) or D(igital) command. The digital commands are used with the local and remote *Digital* I/O units. The analog commands are used with the local and remote *Analog* I/O units.

Chapter 5: Digital Setup / System Commands

Command Name	Command Format	Version
IDENTIFY TYPE	F	D
POWER UP CLEAR	A	D
REPEAT LAST RESPONSE	^	D
RESET	B	D
RESET ALL PARAMETERS TO DEFAULT	x	D
SET WATCHDOG MOMO AND DELAY (Digital)	D MMMM NNNN TTTT	D
SET RESPONSE DELAY	~	D
SET SYSTEM OPTIONS	C SS CC	D

Chapter 6: Digital I/O Configuration Commands

Command Name	Command Format	Version
READ MODULE CONFIGURATION	Y	D
SET CHANNEL CONFIGURATION	a CC TT	D
SET I/O CONFIGURATION-GROUP	G MMMM TT	D
STORE SYSTEM CONFIGURATION	E	D

Chapter 7: Digital Read/Write, Latch Commands

Command Name	Command Format	Version
CLEAR OUTPUT (DEACTIVATE OUTPUT)	e CC	D
READ AND OPTIONALLY CLEAR LATCHES GROUP	S FF	D
READ AND OPTIONALLY CLEAR LATCH	w CC FF	D
READ MODULE STATUS	R	D
SET OUTPUT MODULE STATE-GROUP	J MMMM NNNN	D
SET OUTPUT (ACTIVATE OUTPUT)	d CC	D

Chapter 8: Digital Counter, Frequency Commands

Command Name	Command Format	Version
CLEAR COUNTER	c CC	D
ENABLE/DISABLE COUNTER GROUP	H EEEE DDDD	D
ENABLE/DISABLE COUNTER	b CC SS	D
READ 16 BIT COUNTER	m CC	D
READ 32 BIT COUNTER GROUP	T [MMMM]	D
READ 32 BIT COUNTER	l CC	D
READ AND CLEAR 16 BIT COUNTER	o CC	D
READ AND CLEAR 32 BIT COUNTER GROUP	U [MMMM]	D
READ AND CLEAR 32 BIT COUNTER	n CC	D
READ COUNTER ENABLE/DISABLE STATUS	u	D
READ FREQUENCY MEASUREMENT	t CC	D
READ FREQUENCY MEASUREMENT GROUP	Z [MMMM]	D

Chapter 9: Digital Time Delay/Pulse Output Commands

Command Name	Command Format	Version
GENERATE N PULSES	i CC NNNNNNNN FFFFFFFF X[XXXXXXX]	D
READ OUTPUT TIMER COUNTER	k CC	D
SET TIME PROPORTIONAL OUTPUT PERIOD	] CC T[TTTTTTT]	D
SET TPO PERCENTAGE	j CC PPPPPPPP	D
START CONTINUOUS SQUARE WAVE	h CC NNNNNNNN FFFFFFFF	D
START OFF PULSE	g CC T[TTTTTTT]	D
START ON PULSE	f CC T[TTTTTTT]	D

Chapter 10: Digital Pulse/Period Measurement Commands

Command Name	Command Format	Version
READ 16 BIT PULSE/PERIOD MEASUREMENT	q CC	D
READ 32 BIT PULSE/PERIOD GROUP	W [MMMM]	D
READ 32 BIT PULSE/PERIOD MEASUREMENT	p CC	D
READ AND RESTART 16 BIT PULSE/PERIOD	s CC	D
READ AND RESTART 32 BIT PULSE/PERIOD	r CC	D
READ AND RESTART 32 BIT PULSE/PERIOD GROUP	X [MMMM]	D
READ PULSE/PERIOD COMPLETE STATUS	V	D

Chapter 11: Digital Event/Reaction Commands

Command Name	Command Format	Version
CLEAR EVENT/REACTION TABLE	-	D
CLEAR EVENT TABLE ENTRY	\ EE	D
CLEAR INTERRUPT	zB	D
ENABLE/DISABLE EVENT ENTRY GROUP	{ GG MMMM NNNN	D
ENABLE/DISABLE EVENT TABLE ENTRY	N EE SS	D
READ AND CLEAR EVENT LATCHES	Q EE	D
READ EVENT DATA HOLDING BUFFER	I EE	D
READ EVENT ENTRY ENABLE/DISABLE STATUS	v EE	D
READ EVENT LATCHES	P EE	D
READ AND OPTIONALLY CLEAR EVENT LATCH	zA EE FF	D
READ EVENT TABLE ENTRY	O EE	D
SET EVENT INTERRUPT STATUS	I EE SS	D
SET EVENT ON COMM LINK WATCHDOG TIMEOUT	y EE	D
SET EVENT ON COUNTER/TIMER >=	L EE CC NNNNNNNN	D
SET EVENT ON COUNTER/TIMER<>=	} EE CC NNNNNNNN	D
SET EVENT ON MOMO MATCH	K EE MMMM NNNN	D
SET EVENT REACTION COMMAND	M EE [REACTION COMMAND]	D
REACTION COMMANDS (To be used with digital command M)		
CLEAR COUNTER	05 CC	D
ENABLE/DISABLE COUNTER	04 CC SS	D
ENABLE/DISABLE EVENT ENTRY GROUP	07 GG MMMM NNNN	D
ENABLE/DISABLE EVENT TABLE ENTRY	06 EE SS	D
NULL REACTION (DO NOTHING)	00	D
READ AND HOLD COUNTER VALUE	08 CC	D
SET OUTPUT MODULE STATE-GROUP	01 MMMM NNNN	D
START OFF PULSE	03 CC TTTTTTTT	D
START ON PULSE	02 CC TTTTTTTT	D

Chapter 12: Analog Setup/System Commands

Command Name	Command Format	Version
IDENTIFY TYPE	F	A
POWER UP CLEAR	A	A
REPEAT LAST RESPONSE	^	A
RESET	B	A
RESET ALL PARAMETERS TO DEFAULT	J	A
SET COMM LINK WATCHDOG AND DELAY	D TTTT	A
SET COMM LINK WATCHDOG TIMEOUT DATA	H MMMM DDDDDDDD	A
SET RESPONSE DELAY	~	A
SET SYSTEM OPTIONS	C SS CC	A

Chapter 13: Analog I/O Configuration Commands

Command Name	Command Format	Version
CALCULATE AND SET ADC MODULE OFFSET	d CC	A
CALCULATE AND SET ADC MODULE GAIN	e CC	A
READ MODULE CONFIGURATION	Y	A
SET ADC MODULE OFFSET	b CC OOOO	A
SET ADC MODULE GAIN	c CC GGGG	A
SET AVERAGING SAMPLE WEIGHT (DIG. FILTERING)	h CC DDDD	A
SET CHANNEL CONFIGURATION	a CC TT	A
SET ENGINEERING UNIT SCALING PARAMETERS	f CC HHHHHHHH LLLLLLLL	A
SET I/O CONFIGURATION - GROUP	G MMMM TT ...	A
SET TOTALIZATION SAMPLE RATE	g CC DDDD	A
SET TPO RESOLUTION	] CC SS	A
STORE SYSTEM CONFIGURATION	E	

Chapter 14: Analog Read/Write/Output Commands

Command Name	Command Format	Version
RAMP DAC OUTPUT TO ENDPOINT	Z CC EEEEEEEE SSSSSSSS	A
READ AND CLEAR I/O MODULE DATA	s CC TT	A
READ AND CLEAR I/O MODULE DATA-GROUP	S MMMM TT	A
READ I/O MODULE MAGNITUDE	r CC TT	A
READ I/O MODULE MAGNITUDE-GROUP	R MMMM TT	A
SET DAC MODULE MAGNITUDE, ENG. UNITS	w CC DDDDDDDD	A
SET DAC MODULE MAGNITUDE, ENG. UNITS-GRP.	W MMMM DDDDDDDD .	A
SET DAC MODULE MAGNITUDE, COUNTS	x CC DDDD	A
SET DAC MODULE MAGNITUDE, COUNTS-GROUP	X MMMM DDDD .	A

Command Name	Command Format	Version
CLEAR EVENT/REACTION TABLE	_	A
CLEAR EVENT TABLE ENTRY	\ EE	A
CLEAR INTERRUPT	zB	A
ENABLE/DISABLE EVENT ENTRY GROUP	{ GG MMMM NNNN	A
ENABLE/DISABLE EVENT TABLE ENTRY	N EE SS	A
READ AND CLEAR EVENT LATCHES	Q EE	A
READ EVENT DATA HOLDING BUFFER	l EE	A
READ EVENT ENTRY ENABLE/DISABLE STATUS	v EE	A
READ EVENT LATCHES	P EE	A
READ AND OPTIONALLY CLEAR EVENT LATCH	zA EE FF	A
READ EVENT TABLE ENTRY	O EE	A
SET EVENT INTERRUPT STATUS	I EE SS	A
SET EVENT ON COMM LINK WATCHDOG TIMEOUT	y EE	A
SET EVENT ON I/O >= SETPOINT	K EE CC TT DDDD[DDDD]	A
SET EVENT ON I/O <= SETPOINT	L EE CC TT DDDD[DDDD]	A
SET EVENT REACTION COMMAND	M EE [REACTION COMMAND]	A
REACTION COMMANDS (To be used with analog command M)		
ENABLE/DISABLE EVENT ENTRY GROUP	07 GG MMMM NNNN	A
ENABLE/DISABLE EVENT TABLE ENTRY	06 EE SS	A
ENABLE/DISABLE PID LOOP	04 LL SS	A
NULL REACTION (DO NOTHING)	00	A
RAMP DAC OUTPUT TO ENDPOINT	03 CC EEEEEEEE SSSSSSSS	A
READ AND HOLD I/O DATA	08 CC TT	A
SET DAC MODULE MAGNITUDE, COUNTS	01 CC DDDD	A
SET DAC MODULE MAGNITUDE, ENG. UNITS	02 CC DDDDDDDD	A
SET PID LOOP MIN-MAX OUTPUT LIMITS	09 LL HHHHHHHH LLLLLLLL	A
SET PID LOOP SETPOINT	05 LL SSSSSSSS	A

Chapter 16: PID Loop Commands

Command Name	Command Format	Version
INITIALIZE PID LOOP	i LL II SS OO TTTT	A
READ ALL PID LOOP PARAMETERS	T LL	A
READ PID LOOP PARAMETER	t LL PP	A
SET PID LOOP CONTROL OPTIONS	j CC SSSS CCCC	A
SET PID LOOP DERIVATIVE RATE	n LL DDDDDDDD	A
SET PID LOOP GAIN	l LL GGGGGGGG	A
SET PID LOOP INTEGRAL RESET RATE	m LL IIIIIII	A
SET PID LOOP MAXIMUM RATE OF CHANGE	u LL RRRRRRRR	A
SET PID LOOP MIN-MAX OUTPUT LIMITS	p LL HHHHHHHH LLLLLLLL	A
SET PID LOOP MIN-MAX SETPOINT LIMITS	o LL HHHHHHHH	A
SET PID LOOP PROCESS VARIABLE	q LL SSSSSSSS	A
SET PID LOOP SETPOINT	k LL SSSSSSSS	A

CHAPTER 5

Digital Setup System Commands

IDENTIFY TYPE — COMMAND F

DESCRIPTION:

This command causes the addressed unit to send back a response to the host that identifies the type of I/O unit.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	F	DVF	CR

Where: > = ASCII Character > (3E Hex)
F = ASCII Character F (46 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex)
DDDD = Brainboard Type - 4 Byte Data Field
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	46 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
46 Hex = Command F (46 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DD = B4 Type - 2 Byte Data Field DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error
DVF = Data Verification Field

REMARKS:

Response contains a four (4) byte (ASCII) or a two (2) byte (Binary) data field that identifies the brainboard type as follows:

0010 = Remote 16-Point Digital Multifunction I/O Unit
0011 = G4D32RS Digital Remote Simple I/O Unit
0012 = Remote 16-Point Analog I/O Unit
0013 = Remote 8-Point Analog I/O Unit
0020 = Local 16-Point Digital Multifunction I/O Unit
0021 = Local 16-Point Digital Non-multifunction I/O Unit
0022 = Local 16-Point Analog I/O Unit
0023 = Local 8-Point Analog I/O Unit
0048 = B3000 (digital address) Multifunction I/O Unit
0050 = B3000 (analog address) Multifunction I/O Unit

EXAMPLES:

ASCII Protocol:

Command	>C9FC2cr
Response	A001002cr

Sends an Identify Type command in ASCII protocol to the I/O unit at address C9. Data verification method is 8-bit checksum. Response indicates that the addressed unit is a Remote 16-Point digital multifunction I/O unit.

Binary Protocol:

Command	C9 03 46 3C 51
Response	05 00 00 10 00 01

Sends a Identify Type command in binary protocol to the I/O unit at address C9. Data verification method is 16-bit CRC. Response indicates that the addressed unit is a Remote 16-Point digital multifunction I/O unit.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

POWER-UP CLEAR — COMMAND A

DESCRIPTION:

Prevents the I/O unit from returning a Power-Up Clear Expected error message in response to instructions following application of power or the Reset command.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	A	DVF	CR

Where:
> = ASCII Character > (3E Hex)
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data VerificationField

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	41 hex	DVF

Where: AA = Address (00 to FF Hex)
41 Hex = Command A (41 Hex)
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=0	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 0 (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

This command *must* be sent after power-up, a Reset command or any other reset condition has occurred (i.e. watchdog timeout, low voltage, etc.). If any other command is sent first, a Power-Up Clear Expected error is returned, and the command is NOT executed. A Power Up Clear Expected error will be returned after every command until the Power-Up Clear command is executed.

The Power-Up Clear Expected error provides an indication to the host that there has been a power failure and that the I/O unit has been reset to the power-up configuration stored in EEPROM. When the host receives a Power-Up Clear Expected error, it must send a Power-Up Clear command. For digital modules, the host should (1) enable counters, (2) set the outputs to the desired on/off state, (3) set the outputs to perform special functions such as on/off pulse generation, N pulse generation, square wave generation and time proportional output generation, (4) set event/reaction table entries 32-255, if needed and (5) execute any other power failure recovery routines. See analog Reset command in Chap. 12.

EXAMPLES:

ASCII Protocol:

Command	>79AB1cr
Response	A41cr

Sends a Power Up Clear command in ASCII protocol to the I/O unit at address 79. Data verification method is 8-bit checksum.

Binary Protocol:

Command	79 03 41 19 11
Response	03 00 F0 00

Sends a Power Up Clear command in binary protocol to the I/O unit at address 79. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

REPEAT LAST RESPONSE — COMMAND ^

DESCRIPTION:

This command causes the addressed unit to repeat the response to the previous command.

VERSIONS:

Digital, Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	^	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- ^ = ASCII Character ^ (5E Hex)
- CR = Carriage Return (0D Hex)
- AA = Address
- DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 or more	2 or 4	1
A	RDF	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- RDF = Response Data Field<R>
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- DVF = Data Verification Field
- EC = Error Code<R>
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	5E hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
5E = Command ^ (5E Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error code (Non Zero)
DVF = Data Verification Field

REMARKS:

The addressed I/O unit repeats its previous response, Note that this response is not necessarily the last response that was received by the host computer. The last response to the host computer could have come from another I/O unit on the same communication link.

EXAMPLES:

ASCII Protocol:

Command	>50^C3cr
Response	A000001cr

Sends a Repeat Last response command in ASCII protocol to the I/O unit at address 50 Hex. Data verification method is 9-bit checksum. Response is the response to the last command sent (in this case, the last response was to a Set System Options command - Command C).

Binary Protocol:

Command	50 03 5E 19 81
Response	05 00 00 00 CC 00

Sends a Repeat Last Response command in binary protocol to the I/O unit at address 50 Hex. Data verification method is 16-bit CRC. Response is to the last command sent (in this case, the last response was to a Set System Options command - Command C).

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command forces a hardware reset. The I/O unit is restored to the configuration stored in EEPROM. The factory default is used if nothing was previously saved to EEPROM.

VERSIONS:

Digital, Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	B	DVF	CR

Where:
> = ASCII Character > (3E Hex)
B = ASCII Character B (42 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	42 hex	DVF

Where: AA = Address (00 to FF Hex)
42 Hex = Command B (42 Hex)
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

A delay of 1 second minimum is required before issuing another command. The first command sent after this command must be the POWER-UP CLEAR command. The I/O unit will issue an acknowledgment to the host of the RESET command, prior to performing the hardware reset. When the RESET command is sent to a Digital I/O unit, the following actions will be initiated:

DIGITAL

All outputs are turned off
Configuration is restored from EEPROM
The Option Byte is restored from EEPROM
Response Delay Setting is restored from EEPROM
Counter Enable/Disable Status is restored from EEPROM
Counters are cleared
Latches are cleared
Event latches are cleared
Pulse/Square Wave outputs are canceled
Time proportional outputs are canceled
Event/Reaction table entries 0-31 are restored from EEPROM
Event/Reaction table entries 32-255 are cleared
Global Event Interrupt is disabled

EXAMPLES:

ASCII Protocol:

Command	>22BA6cr
Response	A41cr

Sends a Reset command in ASCII protocol to the I/O unit at address 22 Hex and restores it to the configuration that was last saved in EEPROM. See command E. Data verification method is 8-bit checksum.

Binary Protocol:

Command	22 03 42 CB 20
Response	03 00 F0 00

Sends a Reset command in Binary protocol to the I/O unit at address 22 Hex and restores it to the configuration that was last stored in EEPROM. See command E. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

RESET ALL PARAMETERS TO DEFAULT — COMMAND x

DESCRIPTION:

This command will cause all modules and event parameters to be reset to factory default conditions. See Remarks for a list of the conditions.

VERSIONS:

Digital Version 1.03 or later.

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	x	DVF	CR

Where: > = ASCII Character > (3E Hex)
x = ASCII Character x (78 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	78 hex	DVF

Where: AA = Address (00 to FF Hex)
78 Hex = Command x (78 Hex)
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

This command will cause all modules and event parametersto be reset to the following factory default conditions:

All modules are configured as counter input, Type 00.
All counters are disabled and cleared.
Any output modules will be set to the off state.
The event/reaction table is cleared.
Global event interrupt is disabled.
Communications link watchdog is disabled.
Response Delay is set to zero.

EXAMPLES:

ASCII Protocol:

Command	>22xDCcr
Response	A41cr

Sends a Reset All Parameters To Default command in ASCII protocol to the I/O unit at address 22 Hex. The addressed I/O unit is restored to the factory default configuration. Data verification method is 8-bit checksum.

Binary Protocol:

Command	22 03 78 D8 A0<~>
Response	03 00 F0 00

Sends a Reset All Parameters To Default command in binary protocol to the I/O unit at address 22 Hex. The addressed I/O unit is restored to the factory default configuration. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets the communications line watchdog parameters for the addressed I/O unit. When enabled, if a command (or > character for ASCII protocol) is not received after the specified delay, output modules may be instructed to turn on, turn off or do nothing. Anytime delays on specified output channels are canceled upon watchdog timeout. After the specified modules are turned on or turned off, a complete scan of the event/reaction table occurs, starting from 0. A delay of zero (0) disables the watchdog function.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	1	1	4	4	4	2 or 4	1
>	AA	D	MMMM	NNNN	TTTT	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- D = ASCII Character D (44 Hex)
- MMMM = Modules to go ON upon timeout
- NNNN = Modules to go OFF upon timeout
- TTTT = Time Delay x 10 ms. (Minimum delay is 200 milliseconds, TTTT = 0014 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	2	2	1 or 2
AA	LEN	44 hex	MMMM	NNNN	TTTT	DVF

Where: AA = Address LEN = Length Field (08 or 09 Hex)
44 Hex = Command D (44 Hex)
MMMM<MJ251> = Modules to go ON upon timeout
NNNN = Modules to go OFF upon timeout
TTTT = Time Delay x 10 ms. (Minimum delay is 200 milliseconds. TTTT = 0014 Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

You can use the MOMO (Must ON = MMMM, Must OFF = NNNN) to instruct the I/O unit to turn ON output modules, to turn OFF output modules or to do nothing upon timeout.

The MOMO consists of two 16 bits words. Each bit of the 16-bit word corresponds to a particular output module. The most significant bit corresponds to module #15 and the least significant bit corresponds to module #00.

A module is selected by setting a '1' in the corresponding MOMO. The correspondence is as follows:

Modules which must go ON

MOMO	M	M	M	M
Module No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 0 0	0 0 0 0	1 1 0 1	0 0 0 1
Hex Data	8	0	D	1

Modules which must go OFF

Data Field	N	N	N	N
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 1 0 0	0 0 1 0	0 0 0 0	0 0 0 0
Hex Data	4	1	0	0

In the above example, data bits are set for modules 15, 07, 06, 04 and 00 to go ON (80D1 Hex) and for modules 14 and 09 to go OFF (NNNN = 4100 Hex) if the watchdog timer for the selected I/O unit times out.

In the above example, for ASCII protocol, the Command Data Field (CDF) will contain the eight ASCII characters '80D14100' plus four characters which set the watchdog delay. For binary protocol, the Command Data Field (CDF) will contain the four binary data bytes '80', 'D1', '41' and '00' (Hex) plus two data bytes which set the watchdog delay. Before this command can turn output I/O modules ON or OFF, you must first configure the appropriate I/O modules for outputs. Commands 'G' or 'a' can be used to set the configuration.

The minimum delay for the watchdog timer is 200 milliseconds (TTTT = 0014 Hex).

If the watchdog timer for a particular I/O unit times out, the next instruction sent to the I/O unit will not be executed. Instead an error code will be sent back to the host computer. This error code is sent as a warning to let the host know that a timeout occurred. Subsequent commands will be executed in a normal manner, provided the time interval between commands is shorter than the watchdog timer delay time.

On power up this parameter is restored from EEPROM memory.

EXAMPLES:

ASCII Protocol:

Command	>F2DFF00000004002Ccr
Response	A41cr

Instructs the I/O unit at address F2 to turn ON output modules 15 — 08. No module is turned OFF. Modules 07 — 00 are neither turned ON or turned OFF. The watchdog delay time is set for 10,240 milliseconds (4 x 256 x 10 ms.). Protocol is ASCII. Data verification method is 8-bit checksum.

Binary Protocol:

Command	F2 09 44 FF 00 00 00 04 00 C5 CF
Response	03 00 F0 00

Instructs the I/O unit at address F2 to turn ON output modules 08 — 15. No modules are specified to be turned OFF. Modules 00 — 07 are neither turned ON or turned OFF. They remain in their previous state. The watchdog delay time is set for 10,240 milliseconds (4 x 256 x 10 ms.). Protocol is Binary. Data verification method is 16-bit CRC. The Length Field is 09 indicating that 9 bytes are to follow the length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET RESPONSE DELAY — COMMAND ~

DESCRIPTION:

This command is used to set a delay time for command responses. Data field DD specifies the delay time in units of 10 milliseconds.

VERSIONS:

Digital Version 1.06 or later.

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	~	DD	DVF	CR

Where:
> = ASCII Character > (3E Hex)
~ = ASCII Character ~ (7E Hex)
DVF = Data Verification Field
AA = Address
DD = Response Delay
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	7E hex	DD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
7E Hex = Command ~ (7E Hex) DD = Response Delay
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

After acknowledging the "Set Response Delay" command, for all subsequent commands, the I/O unit will wait DD x 10 milliseconds before responding. Note that any particular command is executed immediately and that only the acknowledge or data response is delayed.

The default response delay setting is zero unless a different value has been set and saved in EEPROM with a "Store System Configuration" command.

EXAMPLES:

ASCII Protocol:

Command	>38~054Ecr
Response	A41cr

Sends a Set Response Delay command in ASCII protocol to the I/O unit at address 38 Hex. The response delay will be set to 50 milliseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 7E 05 02 AD
Response	03 00 F0 00

Sends a Set Response Delay command in binary protocol to the I/O unit at address 38 Hex. The response delay will be set to 50 milliseconds. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set (SS) or clear (CC) the bits in the Option Control Byte. The Option Control Byte is used to select certain system options. The system options and the controlling bits are as follows:

Bit 0 = Frequency resolution setting: 0 = 1 Hz, 1 = 10 Hz.
Bit 1 = Not used.
Bit 2 = Not used.
Bit 3 = Not used.
Bit 4 = CRC initialization value: 0 = 0000, 1 = FFFF.
Bit 5 = CRC method select: 0 = reverse, 1 = classical.
Bit 6 = CRC polynomial select: 0 = CRC16, 1 = CCITT.
Bit 7 = Global event interrupt enable: 0 = disabled, 1 = enabled.

The factory default is 00. This byte is stored in EEPROM when command E is executed and is restored upon power-up or when the RESET command (B) is executed.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	C	SS	CC	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
C = ASCII Character C (43 Hex)
SS = Bits to set in Option Control Byte
CC = Bits to clear in Option Control Byte
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDD = Option Control Byte (last 8 bits)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	43 hex	SS	CC	DVF

Where: AA = Address
43 Hex = Command C (43 Hex)
CC = Bits to clear in Option Control Byte
DVF = Data Verification Field
LEN = Length Field (04 or 05 Hex)
SS = Bits to set in Option Control Byte

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex)
DDDD = Option Control Byte (last 8 bits)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

You can use this command to set (SS), clear (CC) or do nothing to the bits in the Option Control Byte. You can use the Store System Configuration (Command E) to save the Option Control Byte in EEPROM. The Option Control Byte is always restored from EEPROM upon power-up or when the RESET (Command B) is executed.

The response from this command is a 16-bit word. The current value of the Option Control Byte is returned as the least significant byte of the 16-bit word. The most significant byte can be ignored. It is for future expansion.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

EXAMPLES:

ASCII Protocol:

Command	>50C00FF94cr
Response	A000001cr

Instructs the I/O unit at address 50 to clear all bits of the Option Control Byte. This is the factory default condition. Protocol is ASCII. Data verification method is 8-bit checksum.

Binary Protocol:

Command	50 05 43 00 FF 54 71
Response	05 00 00 00 CC 00

Instructs the I/O unit at address 50 to clear all bits of the Option Control Byte. This is the factory default condition. Protocol is binary. Data verification method is 16-bit CRC. The Length Field is 05 indicating that 5 bytes are to follow the length field.

Digital I/O Configuration Commands

READ MODULE CONFIGURATION — COMMAND Y

DESCRIPTION:

This command causes the addressed unit to send back a response to the host that identifies the module configuration type for each of the 16 channels. Data is returned as 2 ASCII Hex digits for each of the 16 channels for ASCII protocol. For binary protocol, data is returned as 1 data byte for each channel. See Remarks for possible configuration types.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	Y	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
Y = ASCII Character Y (59 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	32	2 or 4	1
A	DD	DVF	CR

Where: A = ASCII Character A (41 Hex) DD = Module Configuration
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Note: *Two ASCII characters (representing Hex data) are returned for each channel.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	59 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
59 Hex = Command Y (59 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	16	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (12 to 13 Hex) DD = Module Configuration
EC = Error Code = 00 Hex (Zero) DVF = Data Verification Field

Note: *One data byte is returned for each channel.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

For ASCII protocol, the Return Data Field (DD) will contain 32 ASCII (Hex) characters. For binary protocol, the Return Data Field (DD) will contain 16 binary data bytes, one for each channel. The data is returned in a specific order from the highest channel number to the lowest. Data for channel 15 is returned first and data for channel 00 is returned last.

The returned data is the Configuration Type for each channel (module). The data is interpreted as follows:

00 Hex = Counter Input
01 Hex = Positive Pulse Measurement Input
02 Hex = Negative Pulse Measurement Input
03 Hex = Period Measurement Input
04 Hex = Frequency Measurement Input
05 Hex = Quadrature Counter Input
06 Hex = On Time Totalizer Input
07 Hex = Off Time Totalizer Input
80 Hex = Standard Output

EXAMPLES:

ASCII Protocol:

Command	>EDYE2cr
Response	A80808080808080800000000000000000081cr

Sends a Read Module Configuration command in ASCII protocol to the I/O unit at address ED. Data verification method is 8-bit checksum. Response indicates that channels 15 — 08 are configured as standard output (80) and channels 07 — 00 are configured as counter inputs (00). Unless the counters are enabled with the 'H' command or the 'b' command, the counter input is the same as standard input.

Binary Protocol:

Command	ED 03 59 FF 50
Response	13 00 80 80 80 80 80 80 80 80
	00 00 00 00 00 00 00 00 52 A6

Sends a Read Module Configuration command in binary protocol to the I/O unit at address ED. Data verification method is 16-bit CRC. Response indicates that channels 15 — 08 are configured as standard output (80) and channels 07 — 00 are configured as counter inputs (00). Unless the counters are enabled with the 'H' command or the 'b' command, the counter input is the same as standard input. The length field of 13 Hex = 19 digital indicates that 19 data bytes will follow the length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET CHANNEL CONFIGURATION — COMMAND a

DESCRIPTION:

This command is used to configure a specific channel (module) to a Configuration Type. The Configuration Types are as follows:

00 Hex = Counter Input
01 Hex = Positive Pulse Measurement Input
02 Hex = Negative Pulse Measurement Input
03 Hex = Period Measurement Input
04 Hex = Frequency Measurement Input
05 Hex = Quadrature Counter Input
06 Hex = On Time Totalizer Input
07 Hex = Off Time Totalizer Input
80 Hex = Standard Output

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	a	CC	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
a = ASCII Character a (61 Hex)
TT = Configuration Type
CR = Carriage Return (0D Hex)
AA = Address
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	61 hex	CC	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
61 Hex = Command a (61 Hex) CC = Channel Number (00 to 0F Hex)
TT = Configuration Type DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

This command is similar to the Set I/O Configuration - Group, command ‘G’, except that it operates on one specified channel only. See remarks for command ‘G’ for information concerning quadrature counting.

Examples:

ASCII Protocol:

Command	>6Ca0F80B8cr
Response	A41cr

Sends a Set Channel Configuration command in ASCII protocol to the I/O unit at address 6C. Configures channel 15 (0F Hex) for standard output (80 Hex). Data verification method is 8-bit checksum.

Binary Protocol:

Command	6C 05 61 0F 80 4B C5
Response	03 00 F0 00

Sends a Set Channel Configuration command in binary protocol to the I/O unit at address 6C. Configures channel 15 (0F Hex) for standard output (80 Hex). Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET I/O CONFIGURATION-GROUP — COMMAND G

DESCRIPTION:

This command configures the modules specified by data field MMMM to the Configuration Type specified by data field TT. For each bit in data field MMMM that is set to a 1, there must be a corresponding data field TT.

The Configuration Types (TT) are as follows:

00 Hex = Counter Input
01 Hex = Positive Pulse Measurement Input
02 Hex = Negative Pulse Measurement Input
03 Hex = Period Measurement Input
04 Hex = Frequency Measurement Input
05 Hex = Quadrature Counter Input
06 Hex = On Time Totalizer Input
07 Hex = Off Time Totalizer Input
80 Hex = Standard Output

VERSIONS:

Digital, Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	0 to 32	2 or 4	1
>	AA	G	MMMM	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
A = Address
G = ASCII Character G (47 Hex)
MMMM = Channels to be Configured
TT = Configuration Type (One field TT for each bit in MMMM that is set to a 1)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	0 to 16	1 or 2
AA	LEN	47 hex	MMMM	TT	DVF

Where: AA = Address (00 to FF Hex)
LEN = Length Field
47 Hex = Command G (47 Hex)
MMMM = Channels to be Configured
TT = Configuration Type (One field TT for each bit in MMMM that is set to a 1)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

A channel (module) is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondance is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 1 1 1	0 0 0 0	0 0 0 0	0 0 0 1
Hex Data	F	0	0	1

In the above example, data bits are set for channels (modules) 15, 14, 13, 12, and 00. For the ASCII protocol, the Command Data Field (CDF) will contain the four ASCII characters 'F001'. For the binary protocol, the Command Data Field (CDF) will contain the two Hex data bytes 'F0' and '01'.

For each bit set to a one (1) in data field MMMM, there must be a corresponding data field TT that specifies the configuration type. In the above example MMMM = F001, therefore five (5) data fields TT are required. Data fields TT are sent in the same order as the one (1) bits in MMMM, most significant first and least significant last. In the above example, the first data field, TT, corresponds to the most significant bit in MMMM that is set to a one (1) (Channel 15) and the last data field, TT, corresponds to the least significant bit in MMMM that is set to a one (1) (Channel 00).

Notes On Using The Quadrature Input Module, G4IDC5Q:

The quadrature input module will convert a quadrature signal to a pulse stream which is output on the logic side of one of the two input channels. The channel which receives the pulse stream is determined by the direction of rotation of the encoder. The hardware on the brain card counts pulses into one channel for clockwise travel, and the other channel for counter-clockwise travel. When the two input channels are configured as a quadrature input pair, the CPU on the brain card will automatically compute a position count by subtracting the count of the even numbered channel from the count of the odd numbered channel. The position count is returned when either channel is read, and both counters are cleared when either is specified in a clear or read-and-clear counter command.

The quadrature input module will output a pulse each quadrature transition. The actual resolution of the position count is four times the encoder resolution (pulse per revolution).

Since both channels of a quadrature counter pair are always counting up, they must be cleared to zero at some point before one or the other rolls over (exceeds FFFFFFFF Hex).

At a maximum input rate of 12,500 quadrature pulses per second, the time before roll-over is 23.8 hours.

Quadrature counter input channels must be configured in pairs, with the lower channel number being even. Therefore the only quadrature counter pairs allowed are channels 0 & 1, 2 & 3, 4 & 5, 6 & 7, 8 & 9, 10 & 11, 12 & 13, and 14 & 15.

EXAMPLES:

ASCII Protocol:

Command	>75 G F001 80 80 80 80 01 8B cr
Response	A41cr

Sends a Set I/O Configuration-Group command in ASCII protocol to the I/O unit at address 75 Hex. Channels 15, 14, 13, 12, and 01 are selected (MMMM = F001) to be configured as follows:

Channel 15 = 80 Hex (standard output)
Channel 14 = 80 Hex (standard output)
Channel 13 = 80 Hex (standard output)
Channel 12 = 80 Hex (standard output)
Channel 00 = 01 Hex (positive pulse measurement input)

Data verification method is 8-bit checksum.

Binary Protocol:

Command	75 0A 47 F001 80 80 80 80 01 6551
Response	03 00 F0 00

Sends a Set I/O Configuration-Group command in binary protocol to the I/O unit at address 75 Hex. Channels 15, 14, 13, 12, and 01 are selected (MMMM = F001) to be configured as follows:

Channel 15 = 80 Hex (standard output)
Channel 14 = 80 Hex (standard output)
Channel 13 = 80 Hex (standard output)
Channel 12 = 80 Hex (standard output)
Channel 00 = 01 Hex (positive pulse measurement input)

Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command saves the current system parameters to EEPROM. The parameters saved to EEPROM are:

(1) Module configuration.

(2) Counter enable/disable status.

(3) Communications link watchdog parameters.

(4) Option Control Byte.

(5) Response Delay Setting

(6) Event/Reaction table entries 00 thru 20 Hex.

This command requires one (1) second to execute. The command response is sent after the command has finished executing.

VERSIONS:

Digital, Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	E	DVF	CR

Where: > = ASCII Character > (3E Hex)
E = ASCII Character E (45 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	45 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
45 Hex = Command E (45 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The parameters stored in EEPROM are restored upon power up or if a RESET (Command B) is issued. This command requires one (1) second to execute. The command response is sent after the command has finished executing.

EXAMPLES:

ASCII Protocol:

Command	>31EA9cr
Response	A41cr

Sends a Store System Configuration command in ASCII protocol to the I/O unit at address 31 Hex. Data verification method is 8-bit checksum.

Binary Protocol:

Command	31 03 45 CC 90
Response	03 00 F0 00

Sends a Store System Configuration command in binary protocol to the I/O unit at address 31 Hex. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Digital Read/Write, Latch Commands

CLEAR OUTPUT (DEACTIVATE OUTPUT) COMMAND e

DESCRIPTION:

This command clears (deactivates) the output channel (module) specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	e	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- e = ASCII Character e (65 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)
- AA = Address
- CC = Channel Number

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	65 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
65 Hex = Command e CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command is similar to the Set Output Module State-Group (Command J) command except that it acts on one module only. The channel specified must be configured for a standard output before issuing this command. Any time delay or TPO functions on the selected channel will be canceled by this command.

EXAMPLES:

ASCII Protocol:

Command	>30e0F3Ecr
Response	A41cr

Sends a Clear Output command in ASCII protocol to the I/O unit at address 30 Hex. Command data field (CC) specifies that channel 0F Hex (15 decimal) is to be cleared. Data verification method is 8-bit checksum.

BINARY PROTOCOL:

Command	30 04 65 0F 95 25
Response	03 00 F0 00

Sends a Clear Output command in binary protocol to the I/O unit at address 30 Hex. Command data field (CC) specifies that channel 0F Hex (15 decimal) is to be cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND OPTIONALLY CLEAR LATCHES GROUP — COMMAND S

DESCRIPTION:

This command returns input latch data (32 bits) for both the positive and the negative input latches and then clears the group of latches specified by data field FF (see Remarks). Latches function independently and concurrently with all other input functions. The latches are always functional.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	S	FF	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- S = ASCII Character S (53 Hex)
- FF = Latch Group to be Cleared
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	4	2 or 4	1
A	PPPP	NNNN	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- PPPP = Positive Latch Data
- NNNN = Negative Latch Data
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	53 hex	FF	DVF

Where: AA = Address (00 to FF Hex)
53 Hex = Command S
DVF = Data Verification Field
LEN = Length Field (02 or 03 Hex)
FF = Latch Group to be Cleared

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	2	1 or 2
LEN	EC=00	PPPP	NNNN	DVF

Where: LEN = Length Field (06 or 07 Hex)
PPPP = Positive Latch Data
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)
NNNN = Negative Latch Data

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

The latches are read and the data returned *before* any latches are cleared. Data field FF specifies which group of latches are to be cleared as follows:

FF = 00 : No latches are cleared.
FF = 01 : All positive latches are cleared.
FF = 02 : All negative latches are cleared.
FF = 03 : Both positive and negative latches are cleared.

Positive latches are set (latched) by an OFF to ON (0 to 1) input transition. Negative latches are set (latched) by an ON to OFF (0 to 1) transition of the input.

The return data consists of two 16-bit data fields PPPP and NNNN. A '1' in the data field indicates that the latch is set and a '0' in the data field indicates that the latch is cleared. The correspondence between the bits in the returned data fields and the channel numbers is as follows:

Positive latch data PPPP

Data Field	P	P	P	P
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 1 1	0 0 0 0	1 0 0 1	0 0 0 0
Hex Data	B	0	9	0

Negative latch data NNNN

Data Field	N	N	N	N
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
Hex Data	8	0	0	8

In the above example positive latches for channels 15, 13, 12, 07 and 04 are set and negative latches for channels 15 and 00 are set. All others are clear.

EXAMPLES:

ASCII Protocol:

Command >25S031Dcr
Response A000000D0D5cr

Sends a Read and Optionally Clear Latches Group command in ASCII protocol to the I/O unit at address 25 Hex. Data field FF=03 specifies that both positive and negative latches are to be cleared after read. The returned data field PPPP= 0000 Hex indicates that all positive latches are cleared. The returned data field NNNN= 00D0 Hex indicates that negative latches 7, 6 and 4 are set. All other negative latches are cleared. Data verification method is 8-bit checksum. The checksum is DVF=D5 Hex.

Binary Protocol:

Command 25 04 53 02 FC F7
Response 07 00 00 09 00 08 3A 00

Sends a Read and Optionally Clear Latches Group command in binary protocol to the I/O unit at address 25 Hex. Data field FF=02 specifies that all negative latches are to be cleared after read. The returned data field PPPP= 0009 Hex indicates that positive latches 03 and 00 are set. All the other positive latches are cleared. The returned data field NNNN= 0008 Hex indicates that negative latch 03 is set. All the other negative latches are cleared. Data verification method is 16-bit CRC. Data field DVF is 3A00 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND OPTIONALLY CLEAR LATCH — COMMAND w

DESCRIPTION:

This command returns input latch data and the current module status (ON or OFF) for the channel specified by data field CC. Data field FF specifies which latch will be cleared (see Remarks). Latches function independently and concurrently with all other input functions. The latches are always functional. Each channel has one positive and one negative latch.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	w	CC	FF	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
w = ASCII Character w (77 Hex) CC = Channel Number
FF = Latch or Latches to be Cleared DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2 or 4	1
A	DD	DVF	CR

Where:
A = ASCII Character A (41 Hex) DD = Latch Data and Status
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	77 hex	CC	FF	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
77 Hex = Command w CC = Channel Number (00 to 0F Hex)
FF = Latch or Latches to be ClearedDVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (06 or 07 Hex) EC = Error Code = 00 Hex (Zero)
DD = Latch Data and Status DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel to be read by this command must be configured as an input channel, otherwise an error is returned.

Both the positive and the negative latches for the specified channel are read and the data returned *before* any latches are cleared. The ON/OFF status of the input module is also returned. Data field FF specifies which of latches are to be cleared as follows:

FF = 00 : No latches are cleared.
FF = 01 : The positive latch is cleared.
FF = 02 : The negative latch is cleared.
FF = 03 : Both positive and negative latches are cleared.

Positive latches are set (latched) by an OFF to ON (0 to 1) input transition. Negative latches are set (latched) by an ON to OFF (0 to 1) transition of the input.

The return data consists of an 8-bit data field DD. The interpretation of the bits in the data field DD are as follows:

Bit 0 is the state of the positive latch.
0 = latch is cleared. No positive transition (OFF to ON) has occurred.
1 = latch is set. At least one positive transition (OFF to ON) has occurred.

Bit 1 is the state of the negative latch.
0 = latch is cleared. No negative transition (ON to OFF) has occurred.
1 = latch is set. At least one negative transition (ON to OFF) has occurred.

Bit 2 is the current module status.
0 = input module is OFF. Input voltage is below the logic dropout voltage.
1 = input module is ON. Input voltage is above the logic pickup voltage.

Bits 3 to 7 are reserved for future use and will read as 0 (zero).

EXAMPLES:

ASCII Protocol:

Command	>25w0600A4cr
Response	A02A3cr

Sends a Read and Optionally Clear Latch command in ASCII protocol to the I/O unit at address 25 Hex. Data field CC=06 specifies that channel 06 Hex (6 decimal) is to be read. Data field FF=00 specifies that no latches are to be cleared. The returned data field DD=02 indicates that both the positive latch is clear (bit 0=0), the negative latch is set (bit 1=1) and the input module is OFF (bit 2=0). Data verification method is 8-bit checksum. The checksum is DVF=A3 Hex.

Binary Protocol:

Command	25 05 77 06 03 B0 BE
Response	04 00 07 03 00

Sends a Read and Optionally Clear Latch command in Binary protocol to the I/O unit at address 25 Hex. Data field CC=06 specifies that channel 06 Hex (6 decimal) is to be read. Data field FF=03 specifies that both the positive latch and the negative latch are to be cleared. The returned data field DD=07 indicates that the positive latch is set (bit 0=1), the negative latch is set (bit 1=1) and the input module is ON (bit 2=1). Data verification method is 16-bit CRC. Data field DVF is 0300 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ MODULE STATUS — COMMAND R

DESCRIPTION:

This command reads and returns the current state of all module channels.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	R	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
R = ASCII Character R (52 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDD = Module Status
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	52 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
52 Hex = Command R DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Module Status DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command causes the addressed I/O unit to return a 16-bit data field, one bit for each I/O module. A '1' indicates that the module is ON. A '0' indicates that it is OFF. The correspondence between the data bits and the module numbers is as follows:

Data Field	D	D	D	D
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 1 0	0 0 0 0	0 0 0 1	0 1 0 0
Hex Data	A	0	1	4

The above example indicates that modules 15, 13, 04, and 02 are ON. All other modules are OFF. This command will read the current status (ON or OFF state) of all modules regardless of how they are configured and regardless of whether they are inputs or outputs.

EXAMPLES:

ASCII Protocol:

Command	>67RBFcr
Response	A000001cr

Sends a Read Module Status command in ASCII protocol to the I/O unit at address 67 Hex. Response indicates that all modules are OFF. Data verification method is 8-bit checksum.

Binary Protocol:

Command	67 03 52 D2 30
Response	05 00 00 E0 44 01

Sends a Read Module Status command in binary protocol to the I/O unit at address 67 Hex. Response indicates that modules 07, 06, and 05 are ON. All other modules are OFF. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the state of any or all output channels. The specified channels must have previously been configured as standard outputs by using the 'G' or 'a' command. Returns an error if a channel is specified which is not a standard output. An error is also returned if the command specifies a module to be both turned ON and OFF. Any time delay or TPO function on a specified channel will be canceled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	4	2 or 4	1
>	AA	J	MMMM	NNNN	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
J = ASCII Character J (4A Hex)
MMM = Modules to go ON
NNNN = Modules to go OFF
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	2	2	1 or 2
AA	LEN	4A hex	MMMM	NNNN	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (06 or 07 Hex)
4A Hex = Command J MMMM = Modules to go ON
NNNN = Modules to go OFF DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

A channel (module) is selected by setting a '1' in the corresponding 16-bit data field MMMM or NNNN. The correspondence is as follows:

Modules which are to go ON:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 1 1 1	0 0 0 0	0 0 0 0	1 1 0 1
Hex Data	7	0	0	D

Modules which are to go OFF:

Data Field	N	N	N	N
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 1 0 0	0 0 0 0	0 0 1 0
Hex Data	0	4	0	2

In the above example, channels (modules) 14, 13, 12, 03, 02, and 00 will be turned ON and channels 10 and 01 will be turned OFF. The other channels are neither turned On nor turned OFF. They will remain in their present state.

For the ASCII protocol, the Command Data Field (CDF) will contain the eight ASCII characters '700D0402'. For the binary protocol, the Command Data Field (CDF) will contain the four binary data bytes '70', '0D', '04', and '02'.

EXAMPLES:

ASCII Protocol:

Command	>C0JFF00000069cr
Response	A41cr

Sends a Set Output Module State-Group command in ASCII protocol to the I/O unit at address C0 Hex. Channels 15 — 08 will be turned ON. No channels are to be turned OFF. Data verification method is 8-bit checksum.

Binary Protocol:

Command	C0 07 4A FF 00 00 00 61 68
Response	03 00 F0 00

Sends a Set Output Module State-Group command in binary protocol to the I/O unit at address C0 Hex. Channels 15 — 08 will be turned ON. No channels are to be turned OFF. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET OUTPUT (ACTIVATE OUTPUT) — COMMAND d

DESCRIPTION:

This command turns ON (activates) the output channel (module) specified by data field CC. Any time delay or TPO function on the specified channel will be canceled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	d	CC	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
d = ASCII Character d (64 Hex) CC = Channel Number
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	64 hex	CC	DVF

Where: AA = Address (00 to FF Hex)
64 Hex = Command d
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

This command is similar to the Set Output Module State-Group (Command J) command except that it acts on one module only. The channel specified must be configured for standard output before issuing this command.

EXAMPLES:

ASCII Protocol:

Command	>21d0E3Ccr
Response	A41cr

Sends a Set Output Module State-Group command in ASCII protocol to the I/O unit at address 21 Hex. Specifies that output module 0E Hex (14 decimal) is to be turned ON. Data verification method is 8-bit checksum.

Binary Protocol:

Command	21 04 64 0E F9 E0
Response	03 00 F0 00

Sends a Set Output Module State-Group command in binary protocol to the I/O unit at address 21 Hex. Specifies that output module 0E Hex (14 decimal) is to be turned ON. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Digital Counter Frequency Commands

CLEAR COUNTER — COMMAND c

DESCRIPTION:

This command clears the counter specified by data field CC. Count is reset to zero.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	c	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- c = ASCII Character c (63 Hex)
- CC = Channel Number
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Vertification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	63 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
63 Hex = Command c (63 Hex) CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a counter before issuing this command.

EXAMPLES:

ASCII Protocol:

Command	>39c0433cr
Response	A41cr

Sends a Clear Counter command in ASCII protocol to the I/O unit at address 39 Hex. Specifies that counter channel 04 Hex is to be cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	39 04 63 04 6E 64
Response	03 00 F0 00

Sends a Clear Counter command in binary protocol to the I/O unit at address 39 Hex. Specifies that counter channel 04 Hex is to be cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

ENABLE/DISABLE COUNTER GROUP — COMMAND H

DESCRIPTION:

This command instructs the addressed I/O unit to enable (SS <> 0) or disable (SS = 0) counter channels specified by the 16-bit data field MMMM. Before an input module can act as a counter, it must be configured as a counter input (default configuration) and the counter channel must be enabled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	2	2 or 4	1
>	AA	H	MMMM	SS	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- H = ASCII Character H (48 Hex)
- MMMM = Mask of Modules Affected
- SS = Enable/Disable Flag, 0 to Disable or <> 0 to Enable
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1			
AA	LEN	48 Hex	MMMM	SS	DVF

Where: AA = Address (00 to FF Hex)
LEN = Length Field (05 or 06 Hex)
48 Hex = Command H (48 Hex)
MMMM = Mask of Modules Affected
SS = Enable/Disable Flag, 0 to Disable or <> 0 to Enable
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

The channel specified must be configured as a counter (default configuration) before issuing this command otherwise an error is returned. Disabling a counter makes it act as a standard input. Upon power-up all counter channels are disabled.

Maximum counter frequency is 25 KHz

A channel (module) is selected by setting a '1' in the corresponding 16-bit data field MMMM. The correspondence between a bit in the data fields and a channel number is as follows:

Counter Channels to be Enabled (SS <> 0) or Disabled (SS = 0)

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 1	0 0 0 0	0 0 0 1	0 0 0 0
Hex Data	1	0	1	0

In the above example, counter channels 12 and 04 are selected. The selected channels will be enabled if SS is not equal to 0. The selected channels will be disabled if SS is equal to 0.

EXAMPLES:

ASCII Protocol:

Command	>38H101001D6cr
Response	A41cr

Sends a enable/disable Counter Group command in ASCII protocol to the I/O unit at address 38 Hex. Command data field MMMM = 1010 specifies that counter channel numbers 12 and 04 are to be selected. Data field SS = 01 specifies that the selected channels are to be enabled. Data verification method is 8-bit checksum. Checksum is D6 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	38 06 48 10 10 01 DD 56
Response	03 00 F0 00

Sends a enable/disable Counter Group command in binary protocol to the I/O unit at address 38 Hex. Command data field MMMM = 1010 specifies that counter channel numbers 12 and 04 are to be selected. Data field SS = 01 specifies that the selected channels are to be enabled. Data verification method is 16-bit CRC. Data verification field is DD 56 Hex for the command and F0 00 Hex for
the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

ENABLE/DISABLE COUNTER — COMMAND b

DESCRIPTION:

This command instructs the addressed I/O unit to enable or disable a counter channel specified by data field CC. Enable or disable is specified by data field SS. Before an input module can act as a counter, it must be configured as a counter input (default configuration) and the counter channel must be enabled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	b	CC	SS	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
b = ASCII Character b (62 Hex)
CC = Channel Number (00 to 0F Hex)
SS = Enable Setting, 0 to Disable or <> 0 to Enable
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	62 hex	CC	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
62 Hex = Command b (62 Hex) CC = Channel Number (00 to 0F Hex)
SS = Enable Setting, 0 to Disable or not 0 to Enable
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by CC must be configured as a counter (default configuration) before issuing this command otherwise an error is returned. Upon power up all counter channels are disabled. Disabling a counter makes it act as a standard input. Maximum counter frequency is 25 KHz

The channel number as specified by CC is a hexadecimal number between 00 and 0F.

EXAMPLES:

ASCII Protocol:

Command	>38b050193cr
Response	A41cr

Sends a enable/disable Counter command in ASCII protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies counter channel number 05. Command data field SS = 01 specifies that the channel is to be enabled. Data verification method is 8-bit checksum. Checksum is 93 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	38 05 62 05 01 47 C2
Response	03 00 F0 00

Sends a enable/disable Counter command in binary protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies counter channel number 05. Command data field SS = 01 specifies that the channel is to be enabled. Data verification method is 16-bit CRC. Data verification field (16-bit CRC) is 47 C2 Hex for the command and F0 00 for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ 16-BIT COUNTER — COMMAND m

DESCRIPTION:

This command is used to read the least significant 16 bits of the current counter value for the counter input channel number specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	m	CC	DVF	CR

Where:
> = ASCII Character > (3E Hex)
m = ASCII Character m (6D Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
DDDD = Current Counter Value
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	6D hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
6D = Command m (6D Hex) CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Current Counter Value DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a counter before issuing this command. Only the least significant 16 bits of the current counter value are returned.

EXAMPLES:

ASCII Protocol:

Command	>38m053Dcr
Response	A001507cr

Sends a Read 16 Bit Counter command in ASCII protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies that counter channel 05 is to be read. Returns a 16-bit counter value of 0015 Hex (21 decimal). Data verification method is 8-bit checksum. Checksum is 3D Hex for the command and 07 Hex for the response.

Binary Protocol:

Command	38 04 6D 05 32 A0
Response	05 00 00 15 03 C1

Sends a Read 16 Bit Counter command in binary protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies that counter channel 05 is to be read. Returns a 16-bit counter value of 0015 Hex (21 decimal). Data verification method is 16-bit CRC. The data verification field (16-bit CRC) is 32 A0 Hex for the command and 03 C1 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ 32-BIT COUNTER GROUP

DESCRIPTION:

This command causes the addressed unit to read the counter channels specified by data field MMMM and to send back the current 32-bit counter value for each of the specified counter channels.

The data field [MMMM] is optional for ASCII protocol but mandatory for binary protocol. When omitted (ASCII protocol), data is returned for all 16 channels.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	0 or 4	2 or 4	1
>	AA	T	[MMMM]	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
T = ASCII Character T (54 Hex)
[MMMM] = Channels Affected (Optional)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = Counter Data - 32 bits
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Eight ASCII Hex data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	54 hex	MMMM	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
54 Hex = Command T (54 Hex) MMMM = Channels Affected
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC	DDDDDDDD	DVF

Where: LEN = ength Field
EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Counter Data - 32-bit
DVF = Data Verification Field

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Four binary data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a counter before issuing this command. The current 32-bit counter value for each of the specified channels is returned.

A channel is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
Hex Data	0	0	3	0

In the above example, data bits are set for channels (modules) 05 and 04. For ASCII protocol, the Return Data Field (DD) will contain 16 ASCII (Hex) characters for the 2 counter channels specified. For binary protocol, the Return Data Field (DD) will contain 8 binary data bytes, four for each of the 2 modules specified. The data is returned in a specific order from the highest channel number requested to the lowest. In the above example, data for the specified channels (modules) is returned in the following order 05 and 04.

EXAMPLES:

ASCII Protocol:

Command   >38T003082cr
Response   A000000150000002B5Bcr

Sends a Read 32 Bit Counter Group command in ASCII protocol to the I/O unit at address 38 Hex. Command data field MMMM = 0030 Hex specifies that counter data is to be read from data channels 05 and 04. The response contains two 32-bit data fields. The field DDDDDDDD = 00000015 Hex is the data for channel 05 and the field DDDDDDDD = 0000002B Hex is for channel 04. Data verification method is 8-bit checksum. Checksum is 82 Hex for the command and 5B Hex for the response.

Binary Protocol:

Command   38 05 54 00 30 CD E0
Response   0B 00 00 00 00 15 00 00 00 2B 39 3C

Sends a Read 32-Bit Counter Group command in binary protocol to the I/O unit at address 38 Hex. Command data field MMMM = 0030 Hex specifies that counter data is to be read from data channels 05 and 04. The response contains two 32-bit data fields. The field DDDD = 00 00 00 15 Hex is the data for channel 05 and the field DDDD = 00 00 00 2B Hex is for channel 04. Data verification method is 16-bit CRC. The data verification field (16-bit CRC) is CD E0 Hex for the command and 39 3C Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ 32-BIT COUNTER — COMMAND l

DESCRIPTION:

This command returns the current counter value (32 bits) for the counter input channel number specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	l	CC	DVF	CR

Where:
> = ASCII Character > (3E Hex)
l = ASCII Character l (6C Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = Current Counter Value
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	6C hex	CC	DVF

Where: AA = Address (00 to FF Hex)
6C Hex = Command l (6C Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	DDDDDDDD	DVF

Where: LEN = Length Field (06 or 07 Hex)
EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Current Counter Value
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

The channel specified must be configured as a counter before issuing this command. All 32 bits of the current counter value are returned.

EXAMPLES:

ASCII Protocol:

Command	>38I053Ccr
Response	A00000015C7cr

Sends a Read 32-Bit Counter command in ASCII protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies that counter channel 05 is to be read. Returns a 32-bit counter value of 00000015 Hex (21 decimal). Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 6C 05 A2 A1
Response	07 00 00 00 00 15 78 C0

Sends a Read 32-Bit Counter command in binary protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies that counter channel 05 is to be read. Returns a 32-bit counter value of 00 00 00 15 Hex (21 decimal). Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to read the least significant 16 bits of the current counter value for the counter input channel number specified by data field CC. Counter is cleared after the count is read.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	o	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- o = ASCII Character o (6F Hex)
- DVF = Data Verification Field
- AA = Address
- CC = Channel Number
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- DDDD = Current Counter Value
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- DVF = Data Verification Field
- EC = Error Code
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	6F hex	CC	DVF

Where: AA = Address (00 to FF Hex)
6F = Command o (6F Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex)
DDDD = Current Counter Value
EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

The channel specified must be configured as a counter before issuing this command. Only the least significant 16 bits of the current counter value are returned. However, all 32 bits of the counter are cleared.

EXAMPLES:

ASCII Protocol:

Command	>5Co044Bcr
Response	A00590Fcr

Sends a Read and Clear 16-Bit Counter command in ASCII protocol to the I/O unit at address 5C Hex. Command data field CC = 04 specifies that counter channel 04 is to be read and then cleared. Returns a 16-bit counter value of 0059 Hex (89 decimal). Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 6F 05 52 A1
Response	05 00 00 2D D1 C0

Sends a Read 16-bit Counter command in binary protocol to the I/O unit at address 38 Hex. Command data field CC = 05 specifies that counter channel 05 is to be read and then cleared. Returns a 16-bit counter value of 00 2D Hex (45 decimal). Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND CLEAR 32-BIT COUNTER GROUP — COMMAND U

DESCRIPTION:

This command causes the addressed unit to read the counter channels specified by data field MMMM and to send back the current 32-bit counter value for each of the specified counter channels. Counters are cleared after they are read.

The data field [MMMM] is optional for ASCII protocol but mandatory for binary protocol. When omitted (ASCII protocol), data is returned for all 16 channels and all counters are cleared.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	0 or 4	2 or 4	1
>	AA	U	[MMMM]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- U = ASCII Character U (55 Hex)
- [MMMM] = Channels Affected (Optional)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDDDDDD = Counter Data - 32 bits
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Eight ASCII Hex data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	55 hex	MMMM	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
55 = Command U (55 Hex) MMMM = Channels Affected
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC	DDDDDDDD	DVF

Where: LEN = Length Field
EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Counter Data – 32-bit
DVF = Data Verification Field

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Four binary data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a counter before issuing this command. The current 32-bit counter value for each of the specified channels is returned.

A channel is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
Hex Data	0	0	3	0

In the above example, data bits are set for channels (modules) 05 and 04. For ASCII protocol, the Return Data Field (DDDDDDDD) will contain 8 ASCII (Hex) characters for each the two counter channels specified (16 characters total). For binary protocol, the Return Data Field (DDDDDDDD) will contain eight binary data bytes, four for each of the two modules specified. The data is returned in a specific order from the highest channel number requested to the lowest. In the above example, data for the specified channels (modules) is returned in the following order 05 and 04.

EXAMPLES:

ASCII Protocol:

Command	>38U003083cr
Response	A000000320000001950cr

Sends a Read 32-Bit Counter Group command in ASCII protocol to the I/O unit at address 38 Hex. Command specifies that counter data is to be read from data channels 05 and 04. The response contains two 32-bit data fields. The field 00000032 Hex is the data for channel 05 and the field 00000019 Hex is for channel 04. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 05 55 00 30 0D B1
Response	0B 00 00 00 00 00 00 00 00 29 3B B0

Sends a Read 32-Bit Counter Group command in binary protocol to the I/O unit at address 38 Hex. Command specifies that counter data is to be read from data channels 05 and 04. The response contains two 32-bit data fields. The field 00 00 00 00 Hex is the data for channel 05 and the field 00 00 00 29 Hex is for channel 04. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND CLEAR 32-BIT COUNTER — COMMAND n

DESCRIPTION:

This command returns the current counter value (32 bits) for the counter channel specified by command data field CC. Counter is cleared after the counter is read.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	n	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- n = ASCII Character n (6E Hex)
- DVF = Data Verification Field
- AA = Address
- CC = Channel Number (00 to 0F Hex)
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDDDDDD = Counter Data - 32 bits
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- DVF = Data Verification Field
- EC = Error Code
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	6E hex	CC	DVF

Where: AA = Address (00 to FF Hex)
6E = Command n (6E Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number (00 to 0F Hex)

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC	DDDDDDDD	DVF

Where: LEN = Length Field (06 or 07 Hex)
EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Counter Data – 32-bit
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

The channel specified must be configured as a counter before issuing this command. The channel number specified must be in the range of 00 to 0F Hex. Returns the current counter value (32 bits).

EXAMPLES:

ASCII Protocol:

Command	>38n053Ecr
Response	A0000004EDAcr

Sends a Read and Clear 32-Bit Counter command in ASCII protocol to the I/O unit at address 38 Hex. Command data field specifies that channel 05 is to be read. Returns a 32-bit counter value of 0000004E Hex (78 decimal). Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 6E 04 02 61
Response	07 00 00 00 00 1E BF 81

Sends a Read 32-Bit Counter command in binary protocol to the I/O unit at address 38 Hex. Command data field specifies that channel 04 is to be read. Returns a 32-bit counter value of 00 00 00 1E Hex (30 decimal). Data verification method is 16-bit CRC. The 16-bit CRC is BF 81 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ COUNTER ENABLE/DISABLE STATUS — COMMAND u

DESCRIPTION:

This command causes the addressed unit to send back a 16-bit response data field to the host that identifies the counters that are currently enabled. A 1 indicates that the counter is enabled and a 0 indicates that the counter is disabled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	u	DVF	CR

Where: > = ASCII Character > (3E Hex)
u = ASCII Character u (75 Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex)
DVF = Data Verification Field
DDDD = Counter Channels Enabled
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	75 hex	DVF

Where: AA = Address (00 to FF Hex)   LEN = Length Field (02 or 03 Hex)
75 Hex = Command u   DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex)   DDDD = Counter Channels Enabled
EC = Error Code = 00 Hex (Zero)   DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

A '1' in the 16-bit return data field indicates that a counter channel is enabled. A zero indicates that the channel is either disabled or is not a counter channel. The correspondence between the bits in the returned data field and the corresponding channel number is as follows:

Data Field	D	D	D	D
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	1 1 0 0	0 0 0 0	0 0 0 0
Hex Data	0	C	0	0

In the above example, data bits are set for channels (modules) 11 and 10 indicating that these counter channels have been enabled to act as counters.

EXAMPLES:

ASCII Protocol:

Command	>38uE0cr
Response	A0C0014cr

Sends a Read Counter Enable/Disable Status command in ASCII protocol to the I/O unit at address 38. Response data field DDDD = 0C00 Hex indicates that channels 11 and 10 are enabled as input counters. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 03 75 13 81
Response	05 00 0F 00 3C 05

Sends a Read Counter Enable/Disable Status command in binary protocol to the I/O unit at address 38. Response data field DDDD = 0F 00 Hex indicates that channels 11, 10, 09, and 08 are enabled as input counters. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command returns the current frequency measurement for the channel specified by command data field CC. Returns a 16-bit data field which represents frequency in units of 1 Hertz or 10 Hertz. Maximum frequency input is 25 KHz.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	t	CC	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
t = ASCII Character t (74 Hex) CC = Channel Number
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = Current Frequency Measurement
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	74 hex	CC	DVF

Where: AA = Address (00 to FF Hex)
74 Hex = Command t
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex)
DDDD = Current Frequency Value
EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

The channel specified must be configured for Frequency Measurement Input before issuing this command. Use Command G or Command a. The current frequency value (16 bits) for the specified channel number (CC) is returned. Bit 0 in the Option Control Byte determines the frequency units. A 0 in bit 0 will return a frequency value in units of 1 Hertz. A 1 in bit 0 will return a frequency value in units of 10 Hertz. Use Command C (Set System Options) to set or clear bits in the Option Control Byte.

EXAMPLES:

ASCII Protocol:

Command	>38t0645cr
Response	A000001cr

Sends a Read Frequency Measurement command in ASCII protocol to the I/O unit at address 38 Hex. Command data field specifies that channel 06 is to be read. Returns a 16-bit frequency value of 0000 Hex. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 74 04 62 6A
Response	05 00 00 00 CC 00

Sends a Read Frequency Measurement command in binary protocol to the I/O unit at address 38 Hex. Command data field specifies that channel 04 is to be read. Returns a 16-bit frequency value of 0000 Hex. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ FREQUENCY MEASUREMENT-GROUP — COMMAND Z

DESCRIPTION:

This command causes the addressed unit to read the current frequency value for the channels specified by data field MMMM. A 16-bit data field is returned for each channel specified. Frequency is in units of 1 Hertz or 10 Hertz.

The data field [MMMM] is optional for ASCII protocol but mandatory for binary protocol. When omitted (ASCII protocol), data is returned for all 16 channels.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	0 or 4	2 or 4	1
>	AA	Z	[MMMM]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- Z = ASCII Character Z (5A Hex)
- [MMMM] = Channels Affected (Optional)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 64	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- DDDD = Current Frequency Value
- CR = Carriage Return (0D Hex)

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Four ASCII Hex data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	5A hex	MMMM	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
5A Hex = Command Z MMMM = Channels Affected
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 32	1 or 2
LEN	EC	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Current Frequency Value DVF = Data Verification Field

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Two binary data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured for Frequency Measurement Input before issuing this command. Use Command G or Command a. The current frequency value (16 bits) for each of the specified channels (MMMM) is returned. Bit 0 in the Option Control Byte determines the frequency units. A 0 in bit 0 will return a frequency value in units of 1 Hertz. A 1 in bit 0 will return a frequency value in units of 10 Hertz. Use Command \ (Set System Options) to set or clear bits in the Option Control Byte.

A channel is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	1 0 1 1	0 0 0 0
Hex Data	0	0	B	0

In the above example, data bits are set for channels 07, 05 and 04. For ASCII protocol, the Return Data Field (DDDD) will contain 4 ASCII (Hex) characters for each the channels specified. For binary protocol, the Return Data Field (DDDD) will contain 2 binary data bytes for each of the channels specified. The data is returned in a specific order from the highest channel number requested to the lowest. In the above example, data for the specified channels is returned in the following order 07, 05, and 04.

EXAMPLES:

ASCII Protocol:

Command	>38Z00F09Bcr
Response	A0000000000000000041cr

Sends a Read Frequency Measurement Group command in ASCII protocol to the I/O unit at address 38 Hex. Command data field MMMM = 00F0 Hex specifies that frequency data is to be read from data channels 07, 06, 05, and 04. The response contains four 16-bit data fields. Data indicates that the frequency is 0000 for all of the specified channels. Data verification method is 8-bit checksum. The checksum is 9B Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	38 05 5A 00 F0 5E 81
Response	0B 00 00 00 00 00 00 00 00 00 E5 71

Sends a Read Frequency Measurement Group command in ASCII protocol to the I/O unit at address 38 Hex. Command specifies that frequency data is to be read from data channels 07, 06, 05, and 04. The response contains four 16-bit data fields. Data indicates that the frequency is 00 00 for all the specified channels. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Digital Time Delay Pulse Output Commands

GENERATE N PULSES — COMMAND i

DESCRIPTION:

This command causes the addressed unit to generate an output pulse stream having a specified ON time, OFF time and number of pulses. Command data field NNNNNNNN specifies the ON time in 100 microsecond units. Command data field FFFFFFFF specifies the OFF time in 100 microsecond units. Command data field XXXXXXXX specifies the number of pulses to be output. Both the ON time and the OFF time must be greater than or equal to 1 millisecond.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	1	8	8	1 to 8	2 or 4	1
>	AA	i	CC	NNNNNNNN	FFFFFFFF	X[XXXXXXX]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- i = ASCII Character i (69 Hex)
- CC = Channel Number
- NNNNNNNN = Pulse ON Time
- FFFFFFFF = Pulse OFF Time
- X[XXXXXXX] = Number of Pulses
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Note: *Pulse ON times and pulse OFF times are in units of 100 microseconds. The length of datafield X [XXXXXXX] can optionally be 1 ASCII Hex data byte (0 to F Hex) to 8 ASCII Hex bytes.*

ASCII RESPONSE FORMAT:

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	4	4	1 or 2
AA	LEN	69 hex	CC	NNNNNNNN	FFFFFFFF	XXXXXXXX	DVF

Where: AA = Address LEN = Length Field (0E or 0F Hex)
69 = Command i (69 Hex) CC = Channel Number
NNNNNNNN = Pulse ON Time FFFFFFFF = Pulse OFF Time
XXXXXXXX = Number of Pulses DVF = Data Verification Field

Note: *Pulse ON times and pulse OFF times are in units of 100 microseconds.*

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero))
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a standard output before issuing this command. Command 'G' or command 'a' may be used. ON or OFF delay times of less than 1 millisecond will be ignored and an error is returned to the host. The output pulse stream is terminated by the set or clear output commands.

Data field X[XXXXXXX] may optionally be 1 byte to 8 bytes for ASCII protocol but must be 4 binary data bytes (32 bits) for binary protocol.

EXAMPLES:

ASCII Protocol:

Command	>14i0F000027100000271000000019E2cr
Response	A41cr

Sends a Generate N Pulses command in ASCII protocol to the I/O unit at address 14. The fields for the command frame are as follows:

>	=	Start of command frame character
14	=	Address field AA = 14 Hex
i	=	Command i
0F	=	Channel number. CC = 0F Hex (15 decimal)
00002710	=	Pulse ON time x 100 uS. Field NNNNNNNN = 00002710 Hex
00002710	=	Pulse OFF time x 100 uS. Field FFFFFFFF = 00002710 Hex
00000019	=	Number of pulses. Field XXXXXXXX = 00000019 Hex
E2	=	Data verification field (8-bit checksum). DVF = E2 Hex
cr	=	Carriage return (0D Hex)

Specifies that channel 0F Hex is to generate a pulse train with an ON time of 1 second (00002710 Hex) and an OFF time of 1 second (00002710 Hex). Specifies that 25 pulses (00000019 Hex) are to be output. Data verification method is 8-bit checksum. Checksum is E2 Hex for the command.

Binary Protocol:

Command	14 10 69 0F 00 00 27 10 00 00 27 10 00 00 00 19 CB 19
Response	03 00 F0 00

Sends a Generate N Pulses command in binary protocol to the I/O unit at address 14. The length LEN field is 10 Hex (16 decimal) indicating that 16 bytes follow the length field. The fields for the command frame are as follows:

14	=	Address field AA = 14 Hex
10	=	Length field. LEN = 10 Hex (16 decimal)
69	=	Command i (69 Hex)
0F	=	Channel number. CC = 0F Hex (15 decimal)
00002710	=	Pulse ON time x 100 uS. Field NNNNNNNN = 00002710 Hex
00002710	=	Pulse OFF time x 100 uS. Field FFFFFFFF = 00002710 Hex
00000019	=	Number of pulses. Field XXXXXXXX = 00000019 Hex
CB19	=	Data verification field (16-bit CRC). DVF = CB 19 Hex

Specifies that channel 0F Hex is to generate a pulse train with an ON time of 1 second (00 00 27 10 Hex) and an OFF time of 1 second (00 00 27 10 Hex). Specifies that 25 pulses (00 00 00 19 Hex) are to be output. Data verification method is 16-bit CRC. Data Verification field is CB 19 Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ OUTPUT TIMER COUNTER — COMMAND k

DESCRIPTION:

This command returns the current time delay value of an output channel which may have a delay in progress. The value represents the time remaining for the delay. A time of zero indicates the delay has finished or is not active.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	k	CC	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
k = ASCII Character k (6B Hex) CC = Channel Number
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	TTTTTTTT	DVF	CR

Where: A = ASCII Character A (41 Hex)
TTTTTTTT = Delay count remaining
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *Delay count remaining is in units of 100 μS. Error Response Message Frame*

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	6B hex	CC	DVF

Where: AA = Address (00 to FF Hex)
6B Hex = Command k (6B Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
CC = Channel Number

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	TTTTTTTT	DVF

Where: LEN = Length Field (06 or 07 Hex)
TTTTTTTT = Time count remaining
EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Note: *Time count is in units of 100 µS.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

The channel specified must be configured as a standard output before issuing this command or an error is returned. Returned data is in units of 100 microseconds.

EXAMPLES:

ASCII Protocol:

Command >50k0f46cr
Response A00052A65E4cr

Sends a Read Output Timer Counter command in ASCII protocol to the I/O unit at address 50 Hex. Specifies that counter for channel 0F is to be read. Returns a 32-bit count of 00052A65 Hex (338,533 decimal). Units are are 100 microseconds, therefore the measurement represents 33,853,300 microseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command 50 04 6B 0F F5 3F
Response 07 00 00 03 2E 2F CB AD

Sends a Read Output Timer Counter command in binary protocol to the I/O unit at address 50 Hex. Specifies that channel 0F Hex is to be read. Returns a 32-bit count of 00032E2F Hex (208,431 decimal). Units are are 100 microseconds, therefore the measurement represents 20,843,100 microseconds. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET TIME PROPORTIONAL OUTPUT PERIOD

DESCRIPTION:

This command is used to set the period of a time proportional output. It is the first part of a two part command set. The other part is the Set TPO Percentage command (Command j). This command must be executed before using the Set TPO Percentage command. Time setting is in units of 100 microseconds. A minimum setting of 100 milliseconds is required.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	]	CC	TT[TTTTTT]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
-] = ASCII Character] (5D Hex)
- CC = Channel Number (00 to 0F Hex)
- TT[TTTTTT] = Period Time Setting in 100 Microsecond Units
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	5D hex	CC	TTTTTTTT	DVF

Where: AA = Address (00 to FF Hex)   LEN = Length Field (07 or 08 Hex)
5D Hex = Command] (5D Hex)   CC = Channel Number
TTTTTTTT = Period Time Setting in 100 Microsecond Units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a standard output before issuing this command otherwise an error is returned. For ASCII protocol, the data field (TTTTTTTT) can optionally be 2 to 8 ASCII Hex data bytes. For binary protocol, the data field (TTTTTTTT) must be 4 binary data bytes.

EXAMPLES:

ASCII Protocol:

Command	>14]0F00002710C2cr
Response	A41cr

Sends a Set Time Proportional Output Period command in ASCII protocol to the I/O unit at address 14 Hex. Output channel number is 0F Hex (15 digital). Output pulses are to have a period of 1 second (00002710 Hex). Data verification method is 8-bit checksum. Value of checksum is C2 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	14 08 5D 0F 00 00 27 10 4F CA
Response	03 00 F0 00

Sends a Set Time Proportional Output Period command in binary protocol to the I/O unit at address 14 Hex. The length LEN field is 08 indicating that 8 bytes follow the length field. Output channel number is 0F Hex (15 digital). Output pulses are to have a period of 1 second (00 00 27 10 Hex). Data verification method is 16-bit CRC. Value of data verification field (DVF) is 4F CA Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET TPO* PERCENTAGE — COMMAND j

DESCRIPTION:

This command instructs the addressed I/O unit to set the time proportional output percentage to the value specified by data field PPPPPPPP. Data field PPPPPPPP is a 32-bit unsigned integer. Units are in 1/65,536 of a percent thus the most significant 16 bits represent the whole number part and the least significant 16 bits represent the fractional part of the percentage. The channel number is specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	j	CC	PPPPPPPP	DVF	CR

Where:
> = ASCII Character > (3E Hex)
j = ASCII Character j (6A Hex)
PPPPPPPP = Percentage ON Setting
CR = Carriage Return (0D Hex)
AA = Address
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

ASCII RESPONSE FORMAT:

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

* Time Proportional Output

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	6A hex	CC	PPPPPPPP	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
6A Hex = Command j (6A Hex) CC = Channel Number (00 to 0F Hex)
PPPPPPPP = Percentage ON Setting DVF = Data Verification Field

BINARY RESPONSE FORMAT:

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The Set Time Proportional Output Period command must be executed before this command otherwise you will get an error. This command can be used as many times as you wish to set the TPO percentage without having to issue the Set Time Proportional Output Period command again (unless you want to change the period of the TPO output).

The channel specified by CC must be configured as a standard output before issuing this command otherwise an error is returned.

Data field PPPPPPPP is a 32-bit unsigned integer. Units are in 1/65,536 of a percent thus the most significant 16 bits represent the whole number part and the least significant 16 bits represent the fractional part of the percentage. Percentage values of 640000 Hex (6,553,600 decimal) or greater will cause the TPO channel to remain ON continuously. The value 6,553,600 corresponds to 6,553,600 x 1/65,536 = 100.000 percent. A value of 00000000 Hex will cause the TPO channel to be OFF continuously.

Any ON time value which calculates to less than 1 millisecond will result in the module remaining OFF.

Any OFF time value which calculates to less than 1 millisecond will result in the module remaining ON.

EXAMPLES:

ASCII Protocol:

Command	>14 j 0F 00328000 D2 cr
Response	A41cr

Sends a Set TPO Percentage command in ASCII protocol to the I/O unit at address 14 Hex. Output channel number 0F Hex (15 decimal) is specified. The percentage ON is 00328000 Hex = 3,309,568 decimal (in units of 1/65,536 of a percent) = 50.50 percent. Data verification method is 8-bit checksum. The checksum is D2 Hex.

Binary Protocol:

Command	14 08 6A 0F 00328000 E295
Response	03 00 F000

Sends a Set TPO Percentage command in binary protocol to the I/O unit at address 14 Hex. Output channel number 0F Hex (15 decimal) is specified. The percentage ON is 00328000 Hex = 3,309,568 decimal (in units of 1/65,536 of a percent) = 50.50 percent. Data verification method is 16-bit CRC. The data verification field DVF is E295 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

START CONTINUOUS SQUARE WAVE — COMMAND h

DESCRIPTION:

This command is used to start a continuous square wave at the output channel specified by data field CC. Data field NNNNNNNN specifies the ON time in 100 microsecond units. Data field FFFFFFFF specifies the OFF time in 100 microsecond units.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	8	2 or 4	1
>	AA	h	CC	NNNNNNNN	FFFFFFFF	DVF	CR

Where:

>	=	ASCII Character > (3E Hex)	AA	=	Address
h	=	ASCII Character h (68 Hex)	CC	=	Channel Number
NNNNNNNN	=	ON Time x 100 µS.	FFFFFFFF	=	OFF Time x 100 µS.
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT:

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:

A	=	ASCII Character A (41 Hex)	DVF	=	Data Verification Field
CR	=	Carriage Return (0D Hex)			

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:

N	=	ASCII Character N (4E Hex)	EC	=	Error Code
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	4	4	1 or 2
AA	LEN	68 hex	CC	NNNNNNNN	FFFFFFFF	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (0B or 0C Hex)
68 = Command h (68 Hex) CC = Channel Number
NNNNNNNN = ON Time x 100 µS. FFFFFFFF = OFF Time x 100 µS.
DVF = Data Verification Field

BINARY RESPONSE FORMAT:

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by data field CC must be configured as a standard output before issuing this command. Command "G" or command "a" may be used. ON or OFF delay times of less than 1 millisecond will be ignored and an error is returned to the host. *Timing begins with the OFF state*. If a continuous square wave is currently running on the specified channel, new timing values will become effective upon the next state transition (ON to OFF or OFF to ON). The output pulse stream is terminated by the set or clear output commands.

Continuous square wave functions are permitted on any and all output channels, however square wave functions with delay times of less than 10 milliseconds are limited to a total of eight channels at one time.

EXAMPLES:

ASCII Protocol:

Command	>5Ch0F00002710000027106Acr
Response	A41cr

Sends a Start Continuous Square Wave command in ASCII protocol to the I/O unit at address 5C. Specifies that channel 0F Hex is to generate a continuous pulse train output with an ON time of 1 second (00002710 Hex) and an OFF time of 1 second (00002710 Hex). Data verification method is 8-bit checksum. Checksum is 6A Hex.

Binary Protocol:

Command	5C 0C 68 0E 00 00 27 10 00 00 27 10 9C FC
Response	03 00 F0 00

Sends a Start Continuous Square Wave command in binary protocol to the I/O unit at address 5C. Length data field of 0C Hex indicates that 12 (0c\C Hex) bytes will follow the LEN field. Specifies that channel 0E Hex is to generate a continuous pulse train output with an ON time of 1 second (00 00 27 10 Hex) and an OFF time of 1 second (00 00 27 10 Hex). Data verification method is 16-bit CRC. Data Verification field is 9C FC Hex for the command.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

START OFF PULSE — COMMAND g

DESCRIPTION:

This command is used to generate a retriggerable OFF pulse at the standard output channel specified by data field CC. The delay time is specified by the data field TTTTTTTT. Time setting is in units of 100 micro seconds. Minimum pulse time is 0.5 milliseconds. An error is returned if the time is set for less than 0.5 milliseconds (TTTTTTTT < 5).

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	g	CC	T[TTTTTTT]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- g = ASCII Character g (67 Hex)
- CC = Channel Number (00 to 0F Hex)
- T[TTTTTTT] = Pulse OFF Time Setting in 100 Microsecond Units
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	67 hex	CC	TTTTTTTT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
67 Hex = Command g (67 Hex) CC = Channel Number
TTTTTTTT = Pulse OFF Time Setting in 100 Microsecond Units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a standard output before issuing this command otherwise an error is returned. For ASCII protocol, the data field (TTTTTTTT) can optionally be 1 to 8 ASCII Hex data bytes. For binary protocol, the data field (TTTTTTTT) must be 4 binary data bytes.

If the output module is ON before this command is issued, then it will go OFF at the start of the time delay and go back ON at the end of the delay. If the output module is OFF before this command is issued, then it will remain OFF for the duration of the time delay and it will go ON at the end of the delay (and remain ON).

EXAMPLES:

ASCII Protocol:

Command	>5Cg0E00004E20EFcr
Response	A41cr

Sends a Start OFF Pulse command in ASCII protocol to the I/O unit at address 5C Hex. Output channel number is 0E Hex (14 digital). Output OFF pulse is to be 2 (00004E20 Hex x 100 uS = 20,000 x 100 uS) seconds long. Data verification method is 8-bit checksum. Value of checksum is EF Hex.

Binary Protocol:

Command	5C 08 67 0E 00 00 4E 20 07 D9
Response	03 00 F0 00

Sends a Start OFF Pulse command in binary protocol to the I/O unit at address 5C Hex. Output channel number is 0E Hex (14 digital). Output OFF pulse is to be 2 seconds long Data.

Note: *Blank spaces in the above example are added for clarity only.They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to generate a retriggerable ON pulse at the standard output channel specified by data field CC. The delay time is specified by the data field TTTTTTTT. Time setting is in units of 100 micro seconds. Minimum pulse time is 0.5 milliseconds. An error is returned if the time is set for less than 0.5 milliseconds (TTTTTTTT < 5).

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	f	CC	T[TTTTTTT]	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
f = ASCII Character f (66 Hex) CC = Channel Number (00 to 0F Hex)
T[TTTTTTT] = Pulse ON Time Setting in 100 Microsecond Units
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	66 hex	CC	TTTTTTTT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
66 Hex = Command f (66 Hex) CC = Channel Number
TTTTTTTT = Pulse ON Time Setting in 100 Microsecond Units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured as a standard output before issuing this command otherwise an error is returned. For ASCII protocol, the data field (TTTTTTTT) can optionally be 1 or 8 ASCII Hex data bytes. For binary protocol, the data field (TTTTTTTT) must be 4 binary data bytes.

If the output module is OFF before this command is issued, then it will go ON at the start of the time delay and go back OFF at the end of the delay. If the output module is ON before this command is issued, then it will remain ON for the duration of the time delay and it will go OFF at the end of the delay (and remain OFF).

EXAMPLES:

ASCII Protocol:

Command	>5Cf0E000186A0F2cr
Response	A41cr

Sends a Start ON Pulse command in ASCII protocol to the I/O unit at address 5C Hex. Output channel number is 0E Hex (14 digital). Output ON pulse is to be 10 (000186A0 Hex x 100 uS = 100,000 x 100 uS) seconds long. Data verification method is 8-bit checksum. Value of checksum is F2 Hex.

Binary Protocol:

Command	5C 08 66 0D 00 01 86 A0 76 9B
Response	03 00 F0 00

Sends a Start ON Pulse command in binary protocol to the I/O unit at address 5C Hex. Output channel number is 0D Hex (13 digital). Output ON pulse is to be 10 seconds long. Data verification method is 16-bit CRC. Data verification field is 76 9B Hex for the command.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Digital Pulse/Period Measument Commands

READ 16-BIT PULSE/PERIOD MEASUREMENT — COMMAND q

DESCRIPTION:

This command is used to read the least significant 16 bits of the current pulse/period measurement data for the input channel number specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	q	CC	DVF	CR

Where:
> = ASCII Character > (3E Hex
q = ASCII Character q (71 Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
DDDD = Pulse/Period Measurement Data
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	71 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
71 = Command q (71 Hex) CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Pulse/Period Measurement Data DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. Only the least significant 16 bits of the pulse/period measurement data are returned. Returned data is in units of 100 microseconds.

EXAMPLES:

ASCII Protocol:

Command	>14q073Dcr
Response	A001507cr

Sends a Read 16-bit Pulse/Period Measurement command in ASCII protocol to the I/O unit at address 14 Hex. Specifies that channel 07 is to be read. Returns a 16-bit pulse/period measurement value of 0015 Hex (21 decimal). Units are are 100 microseconds, therefore the measurement represents 2,100 microseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	77 04 71 0E E1 FF
Response	05 00 34 06 0E 96

Sends a Read 16-bit Pulse/Period Measurement command in Binary protocol to the I/O unit at address 77 Hex. Length field of 04 Hex indicates that 4 bytes will follow. Specifies that channel 0E is to be read. Returns a 16-bit pulse/period measurement value of 34 06 Hex (13,318 decimal). Units are are 100 microseconds, therefore the measurement represents 1,331,800 microseconds. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command causes the addressed unit to read the channels specified by data field MMMM and to send back the 32-bit pulse/period measurement data for each of the specified input channels.

The data field [MMMM] is optional for ASCII protocol but mandatory for binary protocol. When omitted (ASCII protocol), data is returned for all 16 channels.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	0 or 4	2 or 4	1
>	AA	W	[MMMM]	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
W = ASCII Character W (57 Hex)
[MMMM] = Channels Affected (Optional)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = Pulse/Period Data - 32-bits
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Eight =ASCII Hex data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	57 hex	MMMM	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
57 Hex = Command W (57 Hex) MMMM = Channels Affected
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC	DDDDDDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Pulse/Period Data - 32-bit DVF = Data Verification Field

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Four binary data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The specified channels must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. All 32-bits of the pulse/period measurement data are returned for each specified channel. Returned data is in units of 100 microseconds.

A channel is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
Hex Data	0	0	3	0

In the above example, data bits are set for channels (modules) 05 and 04. For ASCII protocol, the Return Data Field (DDDDDDDD) will contain 16 ASCII (Hex) characters for the 2 channels specified. For Binary protocol, the Return Data Field (DDDDDDDD) will contain eight binary data bytes, four for each of the two modules specified. The data is returned in a specific order from the highest channel number requested to the lowest. In the above example, data for the specified channels (modules) is returned in the following order: 05 and 04.

EXAMPLES:

ASCII Protocol:

Command >14W00307Fcr
Response A000000150000002B5Bcr

Sends a Read 32-bit Pulse/Period Group command in ASCII protocol to the I/O unit at address 14 Hex. Command specifies that counter data is to be read from data channels 05 and 04. The response contains two 32-bit data fields. The field 00000015 Hex (21 decimal) is the data for channel 05. Units are 100 microseconds, therefore the measurement represents 2,100 microseconds. The field 0000002B Hex (43 decimal) is for channel 04. Units are are 100 microseconds, therefore the measurement represents 4,300 microseconds. Data verification method is 8-bit checksum. Checksum is 5B Hex for the response.

Binary Protocol:

Command 14 05 57 00 C0 4F 81
Response 0B 00 00 00 00 15 00 00 00 2B 39 3C

Sends a Read 32-bit Pulse/Period Group command in Binary protocol to the I/O unit at address 14 Hex. Command specifies that counter data is to be read from data channels 07 and 06. The response contains two 32-bit data fields. The field 00 00 00 15 Hex (21 decimal) is the data for channel 07. Units are are 100 microseconds, therefore the measurement represents 2,100 microseconds. The field 0000002B Hex (43 decimal) is for channel 06. Units are are 100 microseconds, therefore the measurement represents 4,300 microseconds. Data verification method is 16-bit CRC. The CRC value is 39 3C Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command returns the pulse/period measurement data (32-bits) for the input channel number specified by data field CC.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	p	CC	DVF	CR

Where:
> = ASCII Character > (3E Hex)
p = ASCII Character p (70 Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = Pulse/Period Data
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	70 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
70 Hex = Command p (70 Hex) CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	DDDDDDDD	DVF

Where: LEN = Length Field (06 or 07 Hex) EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Pulse/Period Data DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. All 32-bits of the pulse/period measurement data are returned. Returned data is in units of 100 microseconds.

EXMPLES:

ASCII Protocol:

Command	>14p0439cr
Response	A00000015C7cr

Sends a Read 32-bit Pulse/Period Measurement command in ASCII protocol to the I/O unit at address 14 Hex. Specifies that channel 04 is to be read. Returns a 32-bit pulse/period measurement of 00000015 Hex (21 decimal). Units are are 100 microseconds, therefore the measurement represents 2,100 microseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	14 04 70 07 33 20
Response	07 00 00 00 00 15 78 C0

Sends a Read 32-bit Pulse/Period Measurement command in binary protocol to the I/O unit at address 14 Hex. Specifies that channel 07 is to be read. Returns a 32-bit pulse/period measurement of 00000015 Hex (21 decimal). Units are are 100 microseconds, therefore the measurement represents 2100 microseconds. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are* added *for clarity only. They do not actually appear in the command or in the response.*

READ AND RESTART 16-BIT PULSE/PERIOD — COMMAND s

DESCRIPTION:

This command is used to read the least significant 16 bits of the current pulse/period measurement data for the input channel number specified by data field CC. Pulse/Period measurement is restarted after the current data is read. Measurement begins on the next appropriate edge.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	s	CC	DVF	CR

Where: > = ASCII Character > (3E Hex)
s = ASCII Character s (73 Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT:

Sucess Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex)
DVF = Data Verification Field
DDDD = Pulse/Period Data
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	73 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
73 Hex = Command s (73 Hex) CC = Channel Number
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Pulse/Period Data DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The channel specified must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. Only the least significant 16 bits of the pulse/period measurement data are returned. Returned data is in units of 100 microseconds.

If this command is received while the measurement is in progress, the measurement is terminated, the data is returned and a new pulse/period measurement is started. The new measurement will start when the specified positive or negative edge is received. Period measurements will start on the next positive *or* negative edge.

EXAMPLES:

ASCII Protocol:

Command	>14s063Ecr
Response	A00590Fcr

Sends a Read and Restart 16-bit Pulse/Period command in ASCII protocol to the I/O unit at address 14 Hex. Command data field specifies that channel 06 is to be read. Returns a 16-bit pulse/period measurement of 0059 Hex (89 decimal). Units are are 100 microseconds, therefore the measurement represents 8,900 microseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	14 04 73 06 03 E1
Response	05 00 00 2D D1 C0

Sends a Read and Restart 16-bit Pulse/Period command in binary protocol to the I/O unit at address 14 Hex. Command data field specifies that channel 06 is to be read. Returns a 16-bit pulse/period measurement of 00 2D Hex (45 decimal). Units are are 100 microseconds, therefore the measurement represents 4,500 microseconds. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND RESTART 32-BIT PULSE/PERIOD — COMMAND r

DESCRIPTION:

This command returns the pulse/period measurement data (32-bits) for the input channel number specified by data field CC. The pulse/period measurement is restarted after the current data is read.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	r	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- r = ASCII Character r (72 Hex)
- DVF = Data Verification Field
- AA = Address
- CC = Channel Number
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDDDDDD = Pulse/Period Data
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- DVF = Data Verification Field
- EC = Error Code
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	72 hex	CC	DVF

Where: AA = Address (00 to FF Hex)   LEN = Length Field (03 or 04 Hex)
72 Hex = Command r (72 Hex)   CC = Channel Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	DDDDDDDD	DVF

Where: LEN = Length Field (06 or 07 Hex)
EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Pulse/Period Data
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. All 32-bits of the pulse/period measurement data are returned. Returned data is in units of 100 microseconds.

If this command is received while the measurement is in progress, the measurement is terminated, the data is returned and a new pulse/period measurement is started. The new measurement will start when the specified positive or negative edge is received. Period measurements will start on the next positive *or* negative edge.

ON/OFF time totalization will continue without the need for an edge.

EXAMPLES:

ASCII Protocol:

Command	>14r043Bcr
Response	A00000015C7cr

Sends a Read and Restart 32-bit Pulse/Period command in ASCII protocol to the I/O unit at address 14 Hex. Specifies that channel 04 is to be read. Returns a 32-bit pulse/period measurement of 00000015 Hex (21 decimal). Units are are 100 microseconds, therefore the measurement represents 2,100 microseconds. Restarts pulse/period measurement. Data verification method is 8-bit checksum.

Binary Protocol:

Command	14 04 72 04 52 61
Response	07 00 00 00 00 15 78 C0

Sends a Read 32-bit Pulse/Period Measurement command in binary protocol to the I/O unit at address 14 Hex. Specifies that channel 04 is to be read. Returns a 32-bit pulse/period measurement of 00000015 Hex (21 decimal). Units are are 100 microseconds, therefore the measurement represents 2,100 microseconds. Restarts pulse/period measurement. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND RESTART 32-BIT PULSE/PERIOD GROUP — COMMAND X

DESCRIPTION:

This command causes the addressed unit to read the counter channels specified by data field MMMM and to send back the 32-bit pulse/period measurement data for each of the specified counter channels. The pulse/period measurement is restarted after the current data is read.

The data field [MMMM] is optional for ASCII protocol but mandatory for binary protocol. When omitted (ASCII protocol), data is returned for all 16 channels and all counters are cleared.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	0 to 4	2 or 4	1
>	AA	X	[MMMM]	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
X = ASCII Character X (58 Hex)
[MMMM] = Channels Affected (Optional)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = Pulse/Period Data - 32-bits
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Eight ASCII Hex data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	58 hex	MMMM	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
58 = Command X (58 Hex) MMMM = Channels Affected
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC	DDDDDDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDDDDDD = Pulse/Period Data DVF = Data Verification Field

Note: *The length of the return data field will depend upon the number of channels specified by the 16-bit Command Data Field (MMMM). Four binary data bytes are returned for each channel specified.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The channel specified must be configured for pulse measurement, period measurement or pulse totalization before issuing this command. All 32 bits of the pulse/period measurement data are returned. Returned data is in units of 100 microseconds.

A channel is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
Hex Data	0	0	3	0

In the above example, data bits are set for channels (modules) 05 and 04. For ASCII protocol, the Return Data Field (DDDDDDDD) will contain 8 ASCII (Hex) characters for each the 2 channels specified (16 characters total). For binary protocol, the Return Data Field (DDDDDDDD) will contain eight binary data bytes, four for each of the two modules specified. The data is returned in a specific order from the highest channel number requested to the lowest. In the above example, data for the specified channels (modules) is returned in the following order 05 and 04.

If this command is received while the measurement is in progress, the measurement is terminated, the data is returned and a new pulse/period measurement is started. The new measurement will start when the specified positive or negative edge is received. Period measurements will start on the next positive *or* negative edge.

ON/OFF time totalization will continue without the need for an edge.

Examples:

ASCII Protocol:

Command	>14X009086cr
Response	A000000320000001950cr

Sends a Read and Restart 32-bit Pulse/Period Group command in ASCII protocol to the I/O unit at address 14 Hex. Command specifies that counter data is to be read from data channels 07 and 04. The response contains two 32-bit data fields. The field 00000032 Hex (50 decimal) is the data for channel 07. Units are 100 microseconds, therefore the measurement represents 5,000 microseconds. The field 00000019 Hex (25 decimal) is for channel 04. Units are 100 microseconds, therefore the measurement represents 2,500 microseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	14 05 58 00 C0 4C B1
Response	0B 00 00 00 00 00 00 00 00 29 3B B0

Sends a Read and Restart 32-bit Pulse/Period Group command in binary protocol to the I/O unit at address 38 Hex. Command specifies that counter data is to be read from data channels 07 and 06. The response contains two 32-bit data fields. The field 00 00 00 00 Hex is the data for channel 07 and the field 00 00 00 29 Hex (41 decimal) is for channel 06. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ PULSE/PERIOD COMPLETE STATUS — COMMAND V

DESCRIPTION:

This command causes the addressed unit to send back a 16-bit response data field to the host that identifies the channels which have measured one complete pulse or one complete period (cycle). A 1 indicates that the measurement is complete and that data is ready. A 0 indicates that the measurement is in progress or waiting for an edge trigger to occur or else the channel is not configured for pulse/period measurements.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	V	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
V = ASCII Character V (56 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = Channels with Data Ready
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	56 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
56 Hex = Command V (56 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) DDDD = Channels with Data Ready
EC = Error Code = 00 Hex (Zero) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The correspondence between the bits in the returned data field and the corresponding channel number is as follows:

Data Field	D	D	D	D
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	1 1 1 1	0 0 0 0	0 0 0 0
Hex Data	0	F	0	0

In the above example, data bits are set for channels (modules) 11, 10, 09, and 08, indicating that these channels have completed their pulse/period measurements and that data is ready. All pulse/period measurements are edge triggered.

EXAMPLES:

ASCII Protocol:

Command	>14VBBcr
Response	A0C0014cr

Sends a Read Pulse/Period Complete Status command in ASCII protocol to the I/O unit at address 14. Response data field DDDD = 0C00 indicates that channels 11 and 10 have completed their measurements and that data is ready. Data verification method is 8-bit checksum.

Binary Protocol:

Command	14 03 56 CA C0
Response	05 00 0F 00 3C 05

Sends a Read Pulse/Period Complete Status command in binary protocol to the I/O unit at address 14. Response data field DD = 0F 00 indicates that channels 11, 10, 09, and 08 have completed their measurements and that data is ready. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Digital Event/Reaction Commands

CLEAR EVENT/REACTION TABLE — COMMAND _

DESCRIPTION:

This command is used to clear the entire event/reaction Table. All event latches and interrupts are also cleared.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	_	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
_ = ASCII Character _ (5F Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	5F hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
5F Hex = Command _ (5F Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The first 32 entries in the event/reaction table are restored from EEPROM upon power-up or when the Reset command is executed.

EXAMPLES:

ASCII Protocol:

Command	>4E_D8cr
Response	A41cr

Sends a Clear Event/Reaction Table command in ASCII protocol to the I/O unit at address 4E Hex. Clears the event/reaction table for the addressed unit. Event latches and interrupts are also cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	4E 03 5F DF 20
Response	03 00 F0 00

Sends a Clear Event/Reaction Table command in Binary protocol to the I/O unit at address 4E Hex. Clears the Event/Reaction Table for the addressed unit. Event latches and interrupts are also cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

CLEAR EVENT TABLE ENTRY — COMMAND \

DESCRIPTION:

This command is used to clear a single entry in the Event/Reaction Table. Data field EE specifies the entry to be cleared.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	\	EE	DVF	CR

Where:					
>	=	ASCII Character > (3E Hex)	AA	=	Address
\	=	ASCII Character \ (5C Hex)	EE	=	Event Table Entry Number
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:					
A	=	ASCII Character A (41 Hex)	DVF	=	Data Verification Field
CR	=	Carriage Return (0D Hex)			

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:					
N	=	ASCII Character N (4E Hex)	EC	=	Error Code
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	5C hex	EE	DVF

Where:
AA = Address (00 to FF Hex)
5C Hex = Command \ (5C Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
EE = Event Table Entry Number

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where:
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where:
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

The command data field EE specifies the event/reaction table entry to be cleared. If the specified entry is 00 to 1F Hex, then it will be restored from EEPROM upon power-up or any type of reset.

EXAMPLES:

ASCII Protocol:

Command	>38\1D3Ccr
Response	A41cr

Sends a Clear event Table Entry command in ASCII protocol to the I/O unit at address 38 Hex. Specifies that entry number 1D Hex (29 decimal) is to be cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 5C 1D A8 B5
Response	03 00 F0 00

Sends a Clear event Table Entry command in binary protocol to the I/O unit at address 38 Hex. Specifies that entry number 1D Hex (29 decimal) is to be cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

CLEAR INTERRUPT

COMMAND zB

Description:

This command is used to clear the Event/Reaction interrupt output. Event latches are not affected.

Versions:

Digital Version 1.08 or later

ASCII Command Format:

number of bytes →

1	2	2	2 or 4	1
>	AA	zB	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
zB = ASCII Characters zB (7A42 Hex)DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII Response Format

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	2	1 or 2
AA	LEN	7A42 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
7A42 Hex = Command zB (7A42 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command will only clear the interrupt output from the I/O unit. It will not clear any event latch or return data.

DESCRIPTION:

This command is used to enable and/or disable a selected group of 16 event table entries in the event/reaction table. Data field GG specifies which group of entries is to be enabled (or disabled). The most significant nibble (the upper 4 bits) of GG is used to determine which group of 16 entries will be enabled (or disabled). The lower nibble is ignored. See Remarks for examples. Data field MMMM is a bitmask representing entries to be enabled. Data field NNNN is a bitmask representing entries to be disabled. Setting the same bit in both enable and disable bitmasks will result in an error being returned and the command is not executed. An entry which is not enabled is not checked for it's event occurrence by the CPU.

VERSIONS:

Digital - Firmware 1.02 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	4	2 or 4	1
>	AA	{	GG	MMMM	NNNN	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address (00 to FF Hex)
- { = ASCII Character { (7B Hex)
- GG = Event Entry Group Number
- MMMM = Bitmask Representing Entries to be Enabled
- NNNN = Bitmask Representing Entries to be Disabled
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	2	1 or 2
AA	LEN	7B hex	GG	MM	NN	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
7B Hex = Command { (7B Hex) GG = Event Entry Group Number
MM = Bitmask Representing Entries to be Enabled
NN = Bitmask Representing Entries to be Disabled
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

ERROR RESPONSE MESSAGE FRAME

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Once the event/reaction table entries are enabled, the CPU will begin comparing for an event match. An entry which is not enabled is not checked for it's event occurrence by the CPU, and the event latch will not be set.

If GG (Event Entry Group Number) is set to FF Hex, then ALL event table entries are specified and bitmask MMMM is checked for enable/disable status. If MMMM is equal to zero (0), then all entries are disabled. If MMMM is not equal to zero (0), then all entries are enabled. In either case, bitmask NNNN *must* be sent, but is ignored.

Data field GG specifies which group of entries is to be enabled (or disabled). If GG is not equal to FF Hex, then the most significant nibble (the upper 4 bits) of GG is used to determine which group of 16 entries will be enabled (or disabled). The lower nibble is ignored. For example, if GG = 23 Hex, MMMM = 0006 Hex and NNNN = C300 Hex, then the 2 event/reaction table entries 21 Hex and 22 Hex will be enabled and the 4 event/reaction table entries 2F Hex, 2E Hex, 29 Hex and 28 Hex will be disabled. If GG = FF Hex and if MMMM = 0001 Hex, then all entries will be enabled.

The correspondence between the bits in the data field MMMM and the entry offset is as follows:

Data Field	M	M	M	M
Entry Offset	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Offset Hex	0F 0E 0D 0C	0B 0A 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for entry offset numbers 02 Hex and 01 Hex. To get the actual event/reaction table entry number rather than just the entry offset, you must add the group number given by the upper nibble of GG to the entry offset value in Hex. For example, if the group number is 20 Hex, then the actual event/reaction table entry numbers for this example would be
20 + 01 = 21 Hex and 20 + 02 = 22 Hex.

The correspondence between the bits in the data field NNNN and the entry offset is as follows:

Data Field	N	N	N	N
Entry Offset	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Offset Hex	0F 0E 0D 0C	0B 0A 09 08	07 06 05 04	03 02 01 00
Example	1 1 0 0	0 0 1 1	0 0 0 0	0 0 0 0
Hex Data	C	3	0	0

In the above example, data bits are set for entry offset numbers 0F Hex, 0E Hex, 09 Hex and 08 Hex. To get the actual event/reaction table entry number rather than just the entry offset, you must add the group number given by the upper nibble of GG to the entry offset value in Hex. For example, if the group number is 20 Hex, then the actual event/reaction table entry numbers for this example would be 2F Hex, 2E Hex, 29 Hex, and 28 Hex.

EXAMPLES:

ASCII Protocol:

Command	> 8F { 23 0006 C300 FA cr
Response	A41cr

Sends an enable/disable Event Entry Group command in ASCII protocol to the I/O unit at address 8F Hex. Data field GG = 23 Hex, specifies that the 16 event/reaction Table entries 20 Hex to 2F Hex will be selected. Data field MMMM = 0006 specifies that the event/reaction table entries 22 Hex and 21 Hex shall be enabled. Data field NNNN = C300 specifies that the event/reaction table entries 2F Hex, 2E Hex, 29 Hex, and 28 Hex shall be disabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	8F 08 7B AF 0001 1000 B3 43
Response	03 00 F0 00

Sends an enable/disable Event Entry Group command in Binary protocol to the I/O unit at address 8F Hex. Data field GG = AF Hex, specifies that the 16 event/reaction Table entries A0 Hex to AF Hex will be selected. Data field MMMM = 0001 specifies that the event/reaction table entry A0 Hex shall be enabled. Data field NNNN = 1000 specifies that the event/reaction table entry AF Hex shall be disabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

ENABLE/DISABLE EVENT TABLE ENTRY — COMMAND N

DESCRIPTION:

This command is used to enable or to disable an event table entry specified by data field EE. A zero value in data field SS specifies that the entry is to be disabled. A non-zero value specifies that the entry is to be enabled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	N	EE	SS	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- N = ASCII Character N (4E Hex)
- EE = Event Entry Number
- SS = Enable Status, zero to disable, non-zero to enable
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	4E hex	EE	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
4E Hex = Command N (4E Hex) EE = Event Entry Number
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

Before issuing this command, you must first issue a K, L, y or } command to place an entry in the event/reaction table, otherwise an error is returned.

Once enabled, the CPU will begin comparing for an event match. If disabled, the entry is skipped and no action is taken upon an event match and the event latch will not be set.

EXAMPLES:

ASCII Protocol:

Command	> 64 N 16 01 80 cr
Response	A41cr

Sends an enable/disable Event Table Entry command in ASCII protocol to the I/O unit at address 64 Hex. event/reaction Table Entry number is 16 Hex. The enable status field SS = 01 specifies that the event/reaction Table Entry is to be enabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 05 4E 16 01 B3 DE
Response	03 00 F0 00

Sends an enable/disable Event Table Entry command in binary protocol to the I/O unit at address 64 Hex. event/reaction Table Entry number is 16 Hex. The enable status field SS = 01 specifies that the table entry is to be enabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND CLEAR EVENT LATCHES — COMMAND Q

DESCRIPTION:

This command is used to read and then clear the event latches. If command data field EE = FF Hex, then all 256 latch bits (one for each latch) are returned, otherwise only 16 bits are returned. The most significant 4 bits of data field EE determine which group of 16 latch bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	Q	EE	DVF	CR

Where:

> =	ASCII Character > (3E Hex)	AA =	Address
Q =	ASCII Character Q (51 Hex)	EE =	Latch Group
DVF =	Data Verification Field	CR =	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where:

A =	ASCII Character A (41 Hex)	DDDD =	Latch Bits (16 Bits or 256 Bits)
DVF =	Data Verification Field	CR =	Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:

N =	ASCII Character N (4E Hex)	EC =	Error Code
DVF =	Data Verification Field	CR =	Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	51 hex	EE	DVF

Where: AA = Address (00 to FF Hex)
51 Hex = Command Q (51 Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
EE = Latch Group

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field
DDDD = Latch Bits (16 or 256)
EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

There are 256 event latches, one for each event/reaction table entry. The 256 latches are grouped into 16 groups of 16 latches each. Each bit of the return data field represents one latch. If the latch is set, the data bit will be a '1'. If the latch is cleared, the data bit will be a '0'. Latches are set each time an event changes from a non-matching to a matching condition. If command data field EE = FF then all 256 latch bits are returned otherwise only one group of latch bits (16 bits) are returned. The upper nibble (most significant 4 bits) of command data field EE determines which one of the 16 groups of latch bits will be returned. The lower nibble (least significant 4 bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the relative latch number is as follows:

Data Field	D	D	D	D
Entry Offset	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative latch numbers 02 and 01. To get the actual latch numbers rather than just relative latch numbers, you must add an offset value to the relative latch number. First, multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the channel number. For example, assume that the group number is 5 and the relative latch number is 2. The actual latch number is (5x16)+2=82.

The latches are cleared after they are read. If EE = FF Hex all 256 latches are cleared otherwise only the 16 latches for the specified group are cleared.

EXAMPLES:

ASCII Protocol:

Command	> 64 Q 00 1B cr
Response	A 0006 07 cr

Sends a Read and Clear Event Latches command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16 bits of latch data, one bit for each of the first 16 latches. The return data field of DDDD = 0006 indicates that latches 02 and 01 are set. All others are cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 51 00 61 63
Response	05 00 00 03 CD 40

Sends a Read and Clear Event Latches command in binary protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16 bits of latch data, one bit for each of the first 16 latches. The return data field of DDDD = 0003 indicates that latches 01 and 00 are set. All others are cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to read data which has been previously read and held by the execution of digital reaction command 08. The returned data will be a 32-bit number. See Remarks.

VERSIONS:

Digital-Firmware 1.02 and Later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	I	EE	DVF	CR

Where:
> = ASCII Character > (3E Hex)
| = ASCII Character | (7C Hex)
DVF = Data Verification Field
AA = Address
EE = Event Entry Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = 32 Bit Data Field
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	7C hex	EE	DVF

Where: AA = Address (00 to FF Hex)
7C Hex = Command | (7C Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
EE = Event Entry Number

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (06 or 07 Hex)
DDDD = 32-Bit Data Field
EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

Data DDDDDDDD will only be valid if the digital reaction command 08 has been *executed*. See Digital event/reaction Command M. Data field DDDDDDDD is 32 bits. Units are in counts.

EXAMPLES:

ASCII Protocol:

Command	>64|0248cr
Response	A00450000CAcr

Sends a Read Event Data Holding Buffer command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 02 specifies that the event/reaction table entry number is 02 Hex. Data verification method is 8-bit checksum. The 8 checksum is 48 for the command. The response of A indicates a successful command. The returned data DDDDDDDD = 00 45 00 00 Hex (4,521,984 decimal) is in counts. The checksum for the response is CA Hex.

Binary Protocol:

Command	60 04 7C 00 01 7F
Response	07 00 0A AA 00 00 4F 22

Sends a Read Event Data Holding Buffer command in binary protocol to the I/O unit at address 60 Hex. Data field EE = 00 specifies that the event/reaction table entry number is 00 Hex. Data verification method is 16-bit CRC. The 16-bit CRC for the command is 01 7F Hex. The response of 07 indicates that 07 Hex bytes are to follow. A error code of EC=00 indicates a successful command. The returned data DDDD = 0A AA 00 00 Hex (178,913,280 decimal) is in counts. The 16-bit CRC for the response is 4F22 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ EVENT ENTRY ENABLE/DISABLE STATUS — COMMAND v

DESCRIPTION:

This command returns 16 status bits or 256 status bits. Each status bit represents one event table entry. The status bit will be a '1' for event table entries which are enabled and a '0' for entries which are disabled. If data field EE = FF Hex, then 256 status bits are returned, otherwise only 16 status bits are returned. The most significant 4 bits of data field EE determine which group of 16 event table entry status bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	v	EE	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
v = ASCII Character v (76 Hex) EE = Event Table Entry Group
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = Status Bits (16 Bits or 256 Bits)
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	76 hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
76 Hex = Command v (76 Hex) EE = Event Table Entry Group
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Status Bits (16 or 256) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

There are 256 event table entries. They can be grouped into 16 groups of 16 event table entries each. Each status bit in the return data field represents one event table entry. If the event table entry is enabled, the data bit will be a '1'. If the event table entry is disabled, the data bit will be a '0'. If command data field EE = FF, then all 256 status bits are returned, otherwise only one group of status bits (16 bits) are returned. The upper nibble (most significant 4 bits) of command data field EE determines which one of the 16 groups of event table entries status bits will be returned. The lower nibble (least significant 4 bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the relative event table entry number is as follows:

Data Field	D	D	D	D
Event No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative event table entry numbers 02 and 01. To get the actual event table entry numbers rather than just relative event table entry numbers, you must add an offset value to the relative event table entry number. First, multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the relative event table entry number. For example; assume that the group number is 5 and the relative event table entry number is 2. The actual event table entry number is (5x16)+2=82.

EXAMPLES:

ASCII Protocol:

Command	>64v0040cr
Response	A000203cr

Sends a Read Event Entry enable/disable Status command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the status bits for the first group of 16 event table entries are to be returned. The return data field of DDDD = 0002 indicates that event table entry 01 is enabled. All others in the group are disabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 76 00 91 78
Response	05 00 00 03 CD 40

Sends a Read Event Entry Enable/Disable Status command in Binary protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the status bits for the first group of 16 event table entries are to be returned. The return data field of DDDD = 0003 indicates that event table entries 01 and 00 are enabled. All others in the group are disabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ EVENT LATCHES — COMMAND P

DESCRIPTION:

This command is used to read the event latches. If command data field EE = FF Hex, then all 256 latch bits (one for each latch) are returned, otherwise only 16 bits are returned. The most significant 4 bits of data field EE determine which group of 16 latch bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	P	EE	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
P = ASCII Character P (50 Hex) EE = Latch Group
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = Latch Bits (16 Bits or 256 Bits)
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	50 hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
50 Hex = Command P (50 Hex) EE = Latch Group
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Latch Bits (16 or 256) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

There are 256 event latches, one for each event/reaction table entry. The 256 latches are grouped into 16 groups of 16 latches each. Each bit of the return data field represents one latch. If the latch is set, the data bit will be a '1'. If the latch is cleared, the data bit will be a '0'. Latches are set each time an event changes from a non-matching to a matching condition. If command data field EE = FF then all 256 latch bits are returned otherwise only one group of latch bits (16 bits) are returned. The upper nibble (most significant 4 bits) of command data field EE determines which one of the 16 groups of latch bits will be returned. The lower nibble (least significant 4 bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the relative latch number is as follows:

Data Field	D	D	D	D
Latch No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative latch numbers 02 and 01. To get the actual latch numbers rather than just relative latch numbers, you must add an offset value to the relative latch number. First, multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the relative latch numbers. For example, assume that the group number is 5 and the relative latch number is 2. The actual latch number is (5x16)+2=82.

EXAMPLES:

ASCII Protocol:

Command >64P001Acr
Response A000607cr

Sends a Read Event Latches command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16 bits of latch data, one bit for each of the first 16 latches. The return data field of DDDD = 0006 indicates that latches 02 and 01 are set. All others are cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command 64 04 50 00 F1 62
Response 05 00 00 03 CD 40

Sends a Read and Clear Event Latches command in binary protocol to the I/O unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16 bits of latch data, one bit for each of the first 16 latches. The return data field of DDDD = 0003 indicates that latches 01 and 00 are set. All others are cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND OPTIONALLY CLEAR EVENT LATCH — COMMAND zA

DESCRIPTION:

This command returns event latch data and the current event entry enable and interrupt enable status for the entry specified by data field EE. Data field FF indicates whether or not to clear the latch.

VERSIONS:

Digital Version 1.08 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	2	2	2	2 or 4	1
>	AA	zA	EE	FF	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
zA = ASCII Characters zA (7A41 Hex)
EE = Entry Number
FF = Clear Flag
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2 or 4	1
A	DD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DD = Event Latch Data and Status
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	2	1	1	1 or 2
AA	LEN	7A41 hex	EE	FF	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field
7A41 Hex = Command zA EE = Entry Number (00 to FF Hex)
FF = Clear Flag DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (06 or 07 Hex) EC = Error Code = 00 Hex (Zero)
DD = Event Latch Data and Status DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Data field FF in the command specifies whether the event latch is to be cleared. If FF is not equal to 00, the event latch will be cleared after it is read. If FF is equal to 00, the event latch will not be cleared.

The data byte returned by this command (DD) contains only 3 bits that are of any significance to the user.

Bit 0: if 1, the event latch is set; the event has occurred.
if 0, the event latch is cleared; the event has not occurred since the last time this latch was read and cleared.

Bit 6: if 1, the event interrupt is enabled.
if 0, the event interrupt is disabled.

Bit 7: if 1, the event entry is enabled and will be scanned.
if 0, the event entry is disabled and will not be scanned.

DESCRIPTION:

This command returns the event/reaction Table Entry for the entry number specified by data field EE. Fourteen (14) bytes of data are returned for the binary protocol and 28 Hex bytes for the ASCII protocol.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	O	EE	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
O = ASCII Character O (4F Hex)
EE = Event Table Entry Number
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2	2	2	2	2	8	8	2 or 4	1
A	CB	CC	RC	RCC	ED	RES	NNNNNNNN	TTTTTTTT	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CB = Control Byte
CC = Compare Channel
RC = Reaction Command Number
RCC = Reaction Channel Number
ED = Enable/Disable Flag, 0=disable,
1 = enable
RES = Reserved for Future Use
NNNNNNNN = Compare Must ON/Must OFF mask or Counter Value
TTTTTTTT = Reaction Must ON/Must OFF mask or Time Delay Value
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

The Control Byte contains flag bits which are defined as follows:

Bit 7: Event Scan Enable Flag. This bit is set if the event is enabled for scanning.
Bit 6: Interrupt Enable Flag. This bit is set if the event interrupt is enabled.
Bit 5: Last Entry Flag. This bit is set if this event is the last valid entry.
Bit 4, 3: Event Type
- 0, 0 = Must On - Must Off Compare
- 0, 1 = Communications Link Watchdog Monitor
- 1, 0 = Counter Channel <= Setpoint Compare
- 1, 1 = Counter Channel >= Setpoint Compare

Bit 2: Valid Entry Flag. If this bit is clear, this entry is considered a null entry and is not being scanned. All other data is therefore meaningless.
Bit 1: Match Latch. This bit is set when an event condition is matched and cleared when that condition no longer exists.
Bit 0: Event Latch. This bit is set by an event condition match and can only be cleared by the host.

Success Response Message Frame (cont.)

The compare channel is valid only for events which must monitor data of a certain channel.

The reaction command number is the command which will be executed when this event occurs.

The reaction channel number is only valid for reaction commands which require a channel or event number.

The enable/disable flag is only valid for reaction commands which require an enable or disable flag. A zero byte represents a disable flag, and a nonzero byte an enable flag.

The compare value field will contain the compare data for this event. If the event is monitoring a must ON/must OFF condition, the most significant word of this field will contain the must OFF mask and the least significant word will contain the must ON mask. If the event is monitoring a counter value, this field will contain the compare value. The most significant byte is first and the least significant byte is last. This field is meaningless for events which monitor the communications watchdog time-out condition.

The reaction data field is valid only for those reaction commands which require data. If the reaction command is a 01, "Set Output Module State-Group" command, the most significant 16-bit word of this field will contain the mask of modules to be deactivated, (must OFF). The least significant
16-bit word will hold the mask of modules to be activated, (must ON). If the reaction command is a 02 (Start ON Pulse) or 03 (Start OFF Pulse) this field will contain the delay value in 100 microsecond units.

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	4F hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
4F Hex = Command O (4F Hex) EE = Event Table Entry Number
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1	1	1	1	1	1	4	4	1 or 2
LEN	EC=00	CB	CC	RC	RN	ED	RES	NNNNNNNN	TTTTTTTT	DVF

Where: LEN = Length Field (10 or 11 Hex) EC = Error Code = 00 Hex (Zero)
CB = Control Byte CC = Compare Channel
RC = Reaction Command Number RN = Reaction Channel Number
ED = Enable/Disable Flag, 0=disable, 1=enable
RES = Reserved for Future Use
NNNNNNNN = Compare Must ON/Must OFF mask or Counter Value
TTTTTTTT = Reaction Must ON/Must OFF mask or Delay Value
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

See the examples below for the format of the returned data.

EXAMPLES:

ASCII Protocol:

Command	> 64 0 16 20 cr
Response	A A7 00 01 00 00 00 003000C0 0000C000 C3 cr

Sends a Read Event Table Entry command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 16 Hex specifies that event/reaction Table Entry number 16 Hex is to be returned. Data verification is 8-bit checksum.

The structure of the return data field is as follows:

A	=	Acknowledge (Success) for Command, A = 41 Hex
A7	=	Control Byte (A7 Hex = 10100111 binary)
00	=	Compare Channel Number
01	=	Reaction Command Number = 01 Hex 'Set Output Module State'
00	=	Reaction Channel Number = 00 Hex
00	=	Enable/Disable Flag = 00 Hex
00	=	Reserved for Future Use
003000C0	=	Compare Must ON/Must OFF Mask, NNNNNNNN = 003000C0 Hex
0000C000	=	Reaction Must ON/Must OFF Mask, TTTTTTTT = 0000C000 Hex
C3	=	Data Verification Field = C3 Hex (8-bit checksum).
cr	=	Carriage Return (0D Hex)

Control Byte = 10100111 binary is interpreted as follows:

Bit 7	=	1	This Event/Reaction Table Entry is enabled. When the event occurs, the reaction command will be executed.
Bit 6	=	0	The event interrupt is disabled.
Bit 5	=	1	This is the last event table entry. (command K).
Bit 3	=	0	This event is not monitoring the watchdog timer.
Bit 2	=	1	This is a valid Event/Reaction Table Entry.
Bit 1	=	1	The event condition is currently matched (Modules 5 and 4 are OFF and modules 7 and 6 are ON).
Bit 0	=	1	The event has occurred. The event latch is set. Must be cleared by host.

The Compare Channel is CC=00. Data is not valid for this event table entry.

The Reaction Command Number is RC=01, 'Set Output Module State'. The modules which will be set or cleared upon an event match are given by the field TTTTTTTT = 0000C000. There are no output modules to be turned OFF because the most significant 16-bit word is 0000. Modules 15 and 14 are to be turned ON because the least significant 16-bit word is C000.

The Reaction Channel Number is RN=00. Data is not valid for this event table entry.

The enable/disable Flag is ED=00. Data is not valid for this event table entry.

The field RES=00. This field is reserved for future use. Data is not valid.

The Compare Data Field is NNNNNNNN = 003000C0. The most significant 16-bit word = 0030 is the must OFF mask. The least significant 16-bit word = 00C0 is the must ON mask. When modules 5 and 4 are OFF and modules 7 and 6 are ON, the event has occurred and the reaction command 01 will be executed, since the event table entry is enabled (bit 7 = 1 in the control byte).

The Reaction Data Field is TTTTTTTT = 0000C000. This field gives the modules which will be set or cleared upon an event match. There are no output modules to be turned OFF because the most significant 16-bit word is 0000. Modules 15 and 14 are to be turned ON because the least significant 16-bit word is C000.

The Data Verification Field is DVF=C3. The data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 4F 0E 05 EB
Response	11 00 B7 05 02 0F 01 00 00 00 00 12
	00 09 27 C0 59 9C

Sends a Read Event Table Entry command in binary protocol to the I/O unit at address 64 Hex. Data field EE = 0E Hex specifies that Event/Reaction Table Entry number 0E Hex is to be returned. Data verification is 16-bit CRC (DVF = 05EB for command and DVF = 599C for the response). Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.

The structure of the return data frame is as follows:

11	=	Length Field. LEN = 11 Hex (17 decimal).
00	=	Error Code EC = 00 Hex (no error).
B7	=	Control Byte CB = B7 Hex.
05	=	Compare Channel Number CC=05.
02	=	Reaction Command Number = 02 Hex 'Start ON Pulse'.
0F	=	Reaction Channel Number = 0F Hex.
01	=	Enable/Disable Flag = 01 Hex.
00	=	Reserved for Future Use.
00000012	=	Compare Counter Value = 00000012 Hex.
000927C0	=	Reaction Time Delay Value = 000927C0 Hex (120 seconds).
599C	=	Data Verification Field (16-bit CRC) DVF = 599C Hex.

The Length Field is LEN=11 Hex (17 decimal). It indicates the number of bytes to follow.

The Error Code is EC=00. It indicates that there are no errors.

The Control Byte is CB=B7 Hex (10110111 binary) is interpreted as follows:

Bit 7	= 1	This Event/Reaction Table Entry is enabled. When the event occurs, the reaction command will be executed.
Bit 6	= 0	The event interrupt is disabled
Bit 5	= 1	This is the last event table entry.
Bit 4	= 1	The event entry is 'Set Event on Counter /Timer >=' (see command L).
Bit 3	= 0	This event is not monitoring the watchdog timer.
Bit 2	= 1	This is a valid Event/Reaction Table Entry
Bit 1	= 1	The event condition is currently matched (Count is >= 00000012 Hex).
Bit 0	= 1	The event has occurred. The event latch is set. Must be cleared by host.

The Compare Channel is CC=05. The counter associated with channel 05 is being monitored.

The Reaction Command Number is RC=02, 'Start ON Pulse'.

The Reaction Channel Number is RN=0F Hex (15 decimal). This is the channel number for the ON pulse. This channel must be configured as a standard output.

The Enable/Disable Flag is ED=01. Data is not valid for this event table entry.

The field RES=00. This field is reserved for future use. Data is not valid.

The Compare Data Field is NNNNNNNN = 00000012. This is the compare count value. When the counter associated with channel 05 equals or exceeds 00000012 Hex, the event has occurred.

The Reaction Data Field is TTTTTTTT = 000927C0 Hex (120 seconds). This field gives the time delay in units of 100 microseconds for the ON pulse on channel 15.

SET EVENT INTERRUPT STATUS — COMMAND I

DESCRIPTION:

This command is used to enable or to disable the interrupt output function of an entry in the event/reaction Table Entry specified by data field EE. A zero value in data field SS specifies that the interrupt is to be disabled. A non-zero value specifies that the interrupt is to be enabled.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	I	EE	SS	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
I = ASCII Character I (49 Hex) EE = Event Entry Number
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	49 hex	EE	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
49 Hex = Command I (49 Hex) EE = Event Entry Number
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Before issuing this command, you must first issue a K, L, y or } command to place an entry in the event/reaction Table. The Global Event Interrupt Enable Bit (bit 7) in the Option Control Byte must be set to a '1' before interrupts will be sent to the host computer. An interrupt is generated whenever there is an event match for the event/reaction Table Entry whose interrupt status is a '1'.

Interrupts are sent to the host computer by placing a logic zero on the INT line. An interrupt will be generated in the host computer if the communications line is setup to generate an interrupt whenever the INT line goes low (logic zero).

EXAMPLES:

ASCII Protocol:

Command	> 64 I 02 FF A1 cr
Response	A41cr

Sends an Set Event Interrupt Status command in ASCII protocol to the I/O unit at address 64 Hex. Data field EE = 02 specifies that the event/reaction Table Entry number is 02 Hex. The enable status field SS = FF Hex specifies that interrupts are to be enabled for this table entry. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 05 49 01 01 82 60
Response	03 00 F0 00

Sends an Set Event Interrupt Status command in binary protocol to the I/O unit at address 64 Hex. Data field EE = 01 specifies that the Event/Reaction Table Entry number is 01 Hex. The enable status field SS = 01 Hex specifies that interrupts are to be enabled for this table entry. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET EVENT ON COMM LINK WATCHDOG TIME-OUT — COMMAND y

DESCRIPTION:

This command places an event entry in the event/reaction Table. The entry number is specified by data field EE. The event occurs when the watchdog timer times out.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	y	EE	DVF	CR

Where:					
>	=	ASCII Character > (3E Hex)	AA	=	Address
y	=	ASCII Character y (79 Hex)	EE	=	Event Table Entry Group
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:					
A	=	ASCII Character A (41 Hex)	DVF	=	Data Verification Field
CR	=	Carriage Return (0D Hex)			

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:					
N	=	ASCII Character N (4E Hex)	EC	=	Error Code
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	79 hex	EE	DVF

Where: AA = Address (00 to FF Hex)
79 Hex = Command y (79 Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
EE = Event Table Entry Group

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

In order for this event to be effective, you must (1) enable the watchdog timer (see command D) and (2) enable the event/reaction Table Entry (see command N). The channels specified by the reaction command must be configured properly.

EXAMPLES:

ASCII Protocol:

Command	> 64 y 10 44 cr
Response	A41cr

Sends a Set Event on Comm Link Watchdog Time-out command in ASCII protocol to the I/O unit at address 64. The Event Table Entry Number, EE = 10 Hex (16 decimal).

This command enters the specified event in the event/reaction Table at entry number 10 Hex. The event occurs when the watchdog timer times out.

Binary Protocol:

Command	64 04 79 10 AD 7C
Response	03 00 F0 00

Sends a Set Event on Comm Link Watchdog Time-out command in binary protocol to the I/O unit at address 64. The Event Table Entry Number, EE = 10 Hex (16 decimal).

This command enters the specified event in the event/reaction Table at entry number 10 Hex. The event occurs when the watchdog timer times out.

Data verification method is 16-bit CRC. Data Verification Field is AD7C Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command places an event entry in the Event/Reaction Table at the entry number specified by data field EE. The event specification defines a counter/timer input channel CC and a 32-bit count value NNNNNNNN. The event occurs when the count on input channel CC equals or exceeds NNNNNNNN.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	8	2 or 4	1
>	AA	L	EE	CC	NNNNNNNN	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- L = ASCII Character L (4C Hex)
- EE = Event Table Entry Number
- CC = Compare Counter/Timer Channel
- NNNNNNNN = Compare Count/Time Value
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	49 hex	EE	SS	DVF

Where: AA = Address | LEN = Length Field (08 or 09 Hex)
4C Hex = Command L (4C Hex) | EE = Event Table Entry Number
CC = Compare Counter/Timer Channel
NNNNNNNN = Compare Count/Time Value DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) | EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) | EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The channel specified may be configured as either a quadrature or standard counter, pulse duration or ON/OFF time totalizer. Command 'G' or command 'a' may be used. A counter channel should be enabled with command 'H' or command 'b' and the event table entry should be enabled using command 'N' before this command will be effective.

When the count of a quadrature or standard counter is less than NNNNNNN, then becomes greater than or equal to NNNNNNN, the event has occurred and the reaction command will be executed. Quadrature counter values are interpreted as signed numbers (i.e., FFFFFFFE Hex = -2 decimal). Standard counter values are unsigned.

EXAMPLES:

ASCII Protocol:

Command	>64 L 02 04 00000020 FE cr
Response	A41cr

Sends a Set Event on Counter >= command in ASCII protocol to the I/O unit at address 64.

The fields for the command frame are as follows:

>	=	Start of command frame character
64	=	Address field AA = 64 Hex
L	=	Command L = Set Event/Reaction on Counter >=
02	=	Event Table Entry Number, EE = 02 Hex (02 decimal)
04	=	Compare Counter Channel, CC = 04 Hex (04 decimal)
00000020	=	Counter Compare Value, NNNNNNNN = 00000020 Hex
FE	=	Data Verification Field (8-bit checksum). DVF = FE Hex
cr	=	Carriage return (0D Hex)

This command enters the specified event entry in the event/reaction Table as entry number 02 Hex. Command L is the "Set Event/Reaction on Counter >=" command. When the count on input counter channel 04 reaches 00000020 Hex (32 decimal), the event has occurred.

Data verification method is 8-bit checksum. Checksum is FE Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	64 09 4C 02 04 00 00 00 20 96 26
Response	03 00 F0 00

Sends a Set Event on Counter >= command in binary protocol to the I/O unit at address 64. The fields for the command frame are as follows:

64	=	Address field AA = 64 Hex
0A	=	Length Field, LEN = 0A Hex (10 decimal)
4C	=	Command L (4C Hex) = Set Event/Reaction on Counter >=
02	=	Event Table Entry Number, EE = 02 Hex (2 decimal)
04	=	Compare Counter Channel, CC = 04 Hex (4 decimal)
00000020	=	Counter Compare Value, NNNNNNNN = 00000020 Hex
9626	=	Data Verification Field (16-bit CRC). DVF = 9626 Hex

This command enters the specified event/reaction in the event/reaction Table as entry number 02 Hex. Command L is the "Set Event/Reaction on Counter >=" command. When the count on input counter channel 04 reaches 00000020 Hex (32 decimal), the event has occurred. If the event table entry 02 has been enabled, then the reaction command will be executed.

Data verification method is 16-bit CRC. Data Verification Field is 9626 Hex for the command and F000 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command places an event entry in the event/reaction Table at the entry number specified by data field EE. The event specification defines a counter/timer input channel CC and a 32 bit count value NNNNNNNN. The event occurs when the count on input channel CC becomes less than or equal to NNNNNNNN.

VERSIONS:

Digital - Firmware 1.04 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	8	2 or 4	1
>	AA	}	EE	CC	NNNNNNNN	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- } = ASCII Character } (7D Hex)
- EE = Event Table Entry Number
- CC = Compare Counter/Timer Channel
- NNNNNNNN = Compare Count/Time Value
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	4	1 or 2
AA	LEN	7D hex	EE	CC	NNNNNNNN	DVF

Where: AA = Address LEN = Length Field (08 or 09 Hex)
7D Hex = Command } (7D Hex) EE = Event Table Entry Number
CC = Compare Counter/Timer Channel
NNNNNNNN = Compare Count/Time Value DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified may be configured as either a quadrature or standard counter, pulse duration or ON/OFF time totalizer. Command 'G' or command 'a' may be used. A counter channel should be enabled with command 'H' or command 'b' and the event table entry should be enabled using command 'N' before this command will be effective.

When the count of a quadrature or standard counter is higher that NNNNNNNN, then becomes less than or equal to NNNNNNN, the event has occurred and the reaction command will be executed. Quadrature counter values are interpreted as signed numbers (i.e., FFFFFFFE Hex = -2 decimal). Standard counter values are unsigned.

Examples:

ASCII Protocol:

Command	> 07 } 04 0A 000003E8 59 cr
Response	A41cr

Sends a Set Event on Counter <= command in ASCII protocol to the I/O unit at address 07. The fields for the command frame are as follows:

>	=	Start of command frame character
07	=	Address Field AA = 07 Hex (7 decimal)
}	=	Command } = Set Event on Counter <= command
04	=	Event Table Entry Number, EE = 04 Hex (04 decimal)
0A	=	Compare Counter Channel, CC = 0A Hex (10 decimal)
000003E8	=	Counter Compare Value, NNNNNNNN = 000003E8 Hex (1000 decimal)
59	=	Data Verification Field, DVF = 59 Hex (8-bit checksum)
cr	=	Carriage return (0D Hex)

This command enters the specified event entry in the event/reaction Table as entry number 4. When the count on input counter channel 10 becomes less than or equal to 3E8 Hex (1000 decimal), the event has occurred and it is associated latch will be set. If a reaction command has been specified, it will be executed.

Binary Protocol:

Command	07 09 7D 04 0A 000003E8 66D2
Response	03 00 F000

Sends a Set Event on Counter <= command in binary protocol to the I/O unit at address 7. The fields for the cimmand frame are as follows:

07	=	Address Field AA = 07 Hex (7 decimal)
09	=	Length Field, LEN = 09 Hex (9 decimal)
7D	=	Command } (7D Hex) = Set Event on Counter <= command
04	=	Event Table Entry Number, EE = 04 Hex (04 decimal)
0A	=	Compare Counter Channel, CC = 0A Hex (10 decimal)
000003E8	=	Counter Compare Value, NNNNNNNN = 000003E8 Hex (1000 decimal)
66D2	=	Data Verification Field, DVF = 66D2 Hex (166 bit CRC)

This command will do the exact same function as the ASCII protocol command explained above.

Data verification method is 16-bit CRC. Data Verification Field is 66D2 Hex for the command and F000 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command places an event entry in the event/reaction Table at the entry number specified by the data field EE. The event specification field MMMM specifies the modules that must be ON and field NNNN specifies the modules that must be OFF for an event to occur. Modules not specified are 'don't care'.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	4	2 or 4	1
>	AA	K	EE	MMMM	NNNN	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
K = ASCII Character K (4B Hex) EE = Event Table Entry Number
MMMM = Compare Must ON Mask NNNN = Compare Must OFF Mask
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	2	1 or 2
AA	LEN	4B hex	EE	MMMM	NNNN	DVF

Where: AA = Address LEN = Length Field (07 or 08 Hex)
4B Hex = Command K (4B Hex) EE = Event Table Entry Number
MMMM = Compare Must ON Mask NNNN = Compare Must OFF Mask
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channels specified in the MOMO field can be inputs or outputs. The event table entry EE must be enabled using command 'N' before the reaction command will be executed. An error will occur if the channels specified by the reaction command are not configured properly.

When the channels specified by field MMMM are ON and the channels specified by field NNNN are OFF, the event has occurred. Note however, that the reaction will be executed only once (if the entry is enabled) even though the MOMO match persists. The event must change from a non-matching to a matching condition before the reaction will be executed again. The MOMO match must change to a non-matching condition first, then a matching condition must occur again before this can happen.

EXAMPLES:

ASCII Protocol:

Command	> 64 K 16 00C0 0030 B2 cr
Response	A41cr

Sends a Set event/reaction on MOMO Match command in ASCII protocol to the I/O unit at address 64. The fields for the command frame are as follows:

>	=	Start of command frame character
64	=	Address field AA = 64 Hex
K	=	Command K = Set Event/Reaction on MOMO Match
16	=	Event Table Entry Number, EE = 16 Hex (22 decimal)
00C0	=	Compare Must ON Mask, MMMM = 00C0 Hex
0030	=	Compare Must OFF Mask, NNNN = 0030 Hex
B2	=	Data Verification Field (8-bit checksum). DVF = B2 Hex
cr	=	Carriage Return (0D Hex)

This command enters the specified event in the event/reaction Table as entry number 16 Hex. Command K is the "Set Event/Reaction on MOMO Match" command. When the channels specified by field MMMM = 00C0 (ch. 7 and ch. 6) are ON and the channels specified by field NNNN = 0030 (ch. 5 and ch. 4) are OFF, the event has occurred. If the event table entry 16 has been enabled, then the reaction command will be executed.

Data verification method is 8-bit checksum. Checksum is B2 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	64 08 4B 16 00 C0 00 30 4B C8
Response	03 00 F0 00

Sends a Set event/reaction on MOMO Match command in binary protocol to the I/O unit at address 64. The fields for the command frame are as follows:

64	=	Address field AA = 64 Hex
08	=	Length Field, LEN = 08 Hex (8 decimal)
4B	=	Command K (4B Hex) = Set Event/Reaction on MOMO Match
16	=	Event Table Entry Number, EE = 16 Hex (22 decimal)
00C0	=	Compare Must ON Mask, MMMM = 00C0 Hex
0030	=	Compare Must OFF Mask, NNNN = 0030 Hex
4BC8	=	Data Verification Field (16-bit CRC). DVF = 4BC8 Hex

This command enters the specified event in the event/reaction Table as entry number 16 Hex. Command K is the "Set Event/Reaction on MOMO Match" command. When the channels specified by field MMMM = 00C0 (ch. 7 and ch. 6) are ON and the channels specified by field NNNN = 0030 (ch. 5 and ch. 4) are OFF, the event has occurred. If the event table entry 16 Hex has been enabled, then the reaction command will be executed.

Data verification method is 16-bit CRC. Data Verification Field is 4B C8 Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command places a reaction command entry in the event/reaction Table. The event table entry contains two parts (1) an event specification and (2) a reaction specification. The event specification is entered into the event/reaction Table by commands K, L, y or }. The reaction command can be selected from a list of nine commands. See "Remarks" for a list of reaction commands that can be used.

VERSIONS:

Digital

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	1	2 or more	2 or 4	1
>	AA	M	EE	[REACTION COMMAND]	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
M = ASCII Character M (4D Hex) EE = Event Table Entry Number
REACTION COMMAND = Command to be entered into Event/Reaction Table
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *See Remarks for a list of reaction commands that can be used.*

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or more	1 or 2
AA	LEN	4D hex	EE	[REACTION COMMAND]	DVF

Where: AA = Address LEN = Length Field
4D Hex = Command M (4D Hex) EE = Event Table Entry Number
REACTION COMMAND = Command to entered into Event/Reaction Table
DVF = Data Verification Field

Note: *See Remarks for a list of reaction commands that can be used.*

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

In order for an event/reaction to be executed, you must (1) enter an event specification into the event/reaction Table using commands K, L or y, (2) enter a reaction command into the event/reaction Table using command M and (3) enable the table entry using command N. In addition to the above, you must remember to configure and/or enable the channels specified by the reaction command. Refer to Chapter 3 for detailed description of event/reaction processing.

The following reaction commands are allowed:

Reaction Commands Allowed For Command M

Command Name	Command Format		
NULL REACTION (DO NOTHING)	00		
SET OUTPUT MODULE STATE-GROUP	01	MMMM	NNNN
START ON PULSE	02	CC	TTTTTTTT
START OFF PULSE	03	CC	TTTTTTTT
ENABLE/DISABLE COUNTER	04	CC	SS
CLEAR COUNTER	05	CC	
ENABLE/DISABLE EVENT TABLE ENTRY	06	EE	SS
ENABLE/DISABLE EVENT ENTRY GROUP	07	GG	MMMM NNNN
READ AND HOLD COUNTER/TIMER VALUE	08	CC	

DESCRIPTION OF REACTION COMMANDS:

NULL REACTION: 00

No command is executed. The event latch is set upon event occurrence.

SET OUTPUT MODULE STATE: 01 MMMM NNNN

Where MMMM = Modules to go ON
NNNN = Modules to go OFF

See command J, page 7-8.

START ON PULSE: 02 CC TTTTTTTT

Where CC = Channel Number to go ON (must be standard output)
TTTTTTTT = Delay Time in 100 uS units.

See command f, page 9-14.

START OFF PULSE: 03 CC TTTTTTTT

Where CC = Channel Number to go OFF (must be standard output)
TTTTTTTT = Delay Time in 100 uS units.

See command g, page 9-12.

ENABLE/DISABLE COUNTER: 04 CC SS

Where CC = Channel Number of Counter to enable or disable
SS = Enable Setting, 0 = Disable, 1 = Enable.

See command b, page 8-6.

CLEAR COUNTER: 05 CC

Where CC = Channel Number of Counter to be Cleared

See command c, page 8-1.

ENABLE/DISABLE EVENT TABLE ENTRY: 06 EE SS

Where EE = Event Table Entry Number
SS = Enable Setting, 0 = Disable, 1 = Enable.

See command N, page 11-8.

ENABLE/DISABLE EVENT ENTRY GROUP: 07 GG MMMM NNNN

Where GG = Event Entry Group (A group of 16 events)
MMMM = Bitmask for Entries to be Enabled.
NNNN = Bitmask for Entries to be Disabled.

Setting the same bit in MMMM and NNNN will result in an error being returned and the command is not executed. If GG (Event Entry Group Number) is set to FF Hex, then ALL event table entries are specified and bitmask MMMM is checked for enable/disable status. If MMMM is equal to zero (0), then all entries are disabled. If MMMM is not equal to zero (0), then all entries are enabled. In either case, bitmask NNNN *must* be sent, but is ignored. See command { chapter 11. Applies to firmware revision 1.02 or later.

READ AND HOLD COUNTER VALUE: 08 CC

Where CC = Channel Number of Counter to be Read

Upon event occurrence, this reaction command reads the current 32 bit counter value for the counter input channel specified by data field CC. Only one reading per event table entry is allowed. Each time the reaction command is executed, the new counter value replaces the old reading. Use command "|" (7C Hex) to read the counter value from the holding register. The input channel specified by CC must be configured as a counter. Applies to firmware revision 1.02 or later.

EXAMPLES:

ASCII Protocol:

Command > 64 M 02 00 79 cr
Response A41cr

Sends a Set Event Reaction command in ASCII protocol to the I/O unit at address 64. The fields for the command frame are as follows:

- > = Start of command frame character
- 64 = Address field AA = 64 Hex
- M = Command M = Set Event Reaction
- 02 = Event Table Entry Number, EE = 02 Hex (02 decimal)
- 00 = Reaction Command 00 = NULL REACTION
- 79 = Data Verification Field (8-bit checksum). DVF = 79 Hex
- cr = Carriage Return (0D Hex)

This command enters the specified reaction command in the event/reaction Table. The reaction command is Command 00 = 'NULL REACTION'. When the event occurs (set by commands K, L or y), only the latch will be set. No other action takes place.

ASCII Protocol:

Command	> 64 M 16 01 C000 0000 12 cr
Response	A41cr

Sends a Set Event Reaction command in ASCII protocol to the I/O unit at address 64. The fields for the command frame are as follows:

>	=	Start of command frame character
64	=	Address field AA = 64 Hex
M	=	Command M = Set Event Reaction
16	=	Event Table Entry Number, EE = 16 Hex (20 decimal)
01	=	Reaction Command 01 = Set Output Module State
C000	=	Modules to go ON, MMMM = C000 Hex
0000	=	Modules to go OFF NNNN = 0000 Hex
12	=	Data Verification Field (8-bit checksum). DVF = 12 Hex
cr	=	Carriage Return (0D Hex)

This command enters the specified event/reaction in the event/reaction Table as entry number 16 Hex. Command 01 is the "Set Output Module State" reaction command. C000 Hex (modules 15 and 14) are the modules to be turned ON when the event occurs and 0000 are the modules to be turned OFF. Data verification method is 8-bit checksum. Checksum is 12 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	64 0A 4D 16 02 0E 00 09 27 C0 22 2D
Response	03 00 F0 00

Sends a Set Event Reaction command in binary protocol to the I/O unit at address 64. The fields for the command frame are as follows:

64	=	Address field AA = 64 Hex
0A	=	Length Field, LEN = 0A Hex (10 decimal)
4D	=	Command M (4D Hex) = Set Event Reaction
16	=	Event Table Entry Number, EE = 16 Hex (20 decimal)
02	=	Reaction Command 02 'Start ON Pulse'
0E	=	Channel Number for Output Pulse (0E Hex = 14 decimal).
000927C0	=	Delay Time in 100 uS units, TTTTTTTT = 000927C0 Hex (120 sec.)
222D	=	Data Verification Field (16-bit CRC). DVF = 222D Hex

This command enters the specified event reaction command in the event/reaction Table as entry number 16 Hex. Command M is the "Set Event Reaction" command. The reaction command is Command 02 = "Start ON Pulse". Data field CC = 0E Hex (14 decimal) specifies that output channel (module) number 14 is to start an ON pulse. Data field TTTTTTTT = 000927C0 Hex (1,200,000 decimal) specifies that the delay time is to be 120 seconds.

Data verification method is 16-bit CRC. Data Verification Field is 22 2D Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

CHAPTER 12

Analog Setup/System Commands

IDENTIFY TYPE — COMMAND F

DESCRIPTION:

This command causes the addressed unit to send back a response to the host that identifies the type of I/O unit.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	F	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- F = ASCII Character F (46 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDD = Brain Board Type - 4 Byte Data Field
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	46 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
46 Hex = Command F (46 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (04 or 05 Hex) DD = B4 Type - 2 Byte Data Field
EC = Error Code = 00 Hex (Zero) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Response contains a four (4) byte (ASCII) or a two (2) byte (Binary) data field that identifies the brainboard type as follows:

0010 = Remote 16 Point Digital Multifunction I/O unit
0012 = Remote 16 Point Analog Multifunction I/O unit
0013 = Remote 8 Point Analog Multifunction I/O unit
0020 = Local 16 Point Digital Multifunction I/O unit
0021 = Local 16 Point Digital Non-multifunction I/O unit
0022 = Local 16 Point Analog Multifunction I/O unit
0023 = Local 8 Point Analog Multifunction I/O unit
0030 = B3000 (digital address) Multifunction I/O unit
0032 = B3000 (analog address) Multifunction I/O unit

EXAMPLES:

ASCII Protocol:

Command	>25FADcr
Response	A001204cr

Sends an Identify Type command in ASCII protocol to the I/O unit at address 25 Hex. Data verification method is 8-bit checksum. Command checksum is AD Hex. Response checksum is 04 Hex. Response of DDDD = 0012 Hex indicates that the addressed unit is a Remote 16 Point Analog Multifunction I/O unit.

Binary Protocol:

Command	C9 03 46 3C 51
Response	05 00 00 10 00 01

Sends an Identify Type command in Binary protocol to the I/O unit at address C9. Data verification method is 16-bit CRC. Response of DDDD = 0010 Hex indicates that the addressed unit is a Remote 16-Point Digital Multifunction I/O unit.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

POWER UP CLEAR — COMMAND A

DESCRIPTION:

Prevents the I/O unit from returning a Power Up Clear Expected error message in response to instructions following application of power or the Reset command.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	A	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1 or 2
AA	LEN	41 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
41 Hex = Command A (41 Hex) DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 0 (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

This command must be sent after power-up, a Reset command or any other reset condition has occurred (i.e. watchdog timeout, low voltage, etc.). If any other command is sent first, a Power Up Clear Expected error is returned, and the command is NOT executed. A Power Up Clear Expected error will be returned after every command until the Power Up Clear command is executed.

The Power Up Clear Expected error provides an indication to the host that there has been a power failure and that the I/O unit has been reset to the power-up configuration stored in EEPROM. When the host receives a Power Up Clear Expected error, it must send a Power Up Clear command. For analog modules, the host should (1) set ADC module offset and gain, if you want something different than that restored from EEPROM, (2) set Engineering unit scaling parameters, if you want something different than that restored from EEPROM, (3) set totalization sample rate, and (4) set all Event/Reaction table entries 0 to 255, if needed.

EXAMPLES:

ASCII Protocol:

Command	>79AB1cr
Response	A41cr

Sends a Power Up Clear command in ASCII protocol to the I/O unit at address 79. Data verification method is 8-bit checksum.

Binary Protocol:

Command	79 03 41 19 11
Response	03 00 F0 00

Sends a Power Up Clear command in Binary protocol to the I/O unit at address 79. Data verification method is 16-bit CRC.

REPEAT LAST RESPONSE

DESCRIPTION:

This command causes the addressed brainboard unit to repeat the response to the previous command.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	^	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
^ = ASCII Character ^ (5E Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 or more	2 or 4	1
A	RDF	DVF	CR

Where:
A = ASCII Character A (41 Hex)
RDF = Response Data Field
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1 or 2
AA	LEN	5E hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
5E Hex = Command ^ (5E Hex) DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	0 or more	1 or 2
LEN	EC=00	RDF	DVF

Where: LEN = Length Field RDF = Response Data Fields
EC = Error Code = 00 Hex (Zero) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The addressed I/O unit repeats its previous response. Note that this response is not necessarily the last response that was received by the host computer. The last response to the host computer could have come from another I/O unit on the same communication link.

EXAMPLES:

ASCII Protocol:

Command	>50^C3cr
Response	A000001cr

Sends a Repeat Last Response command in ASCII protocol to the I/O unit at address 50 Hex. Data verification method is 8-bit checksum. Response is the response to to the last command sent (in this case, the last response was to a Set System Options command - Command C).

Binary Protocol:

Command	50 03 5E 19 81
Response	05 00 00 00 CC 00

Sends a Repeat Last Response command in Binary protocol to the I/O unit at address 50 Hex. Data verification method is 16-bit CRC. Response is the response to to the last command sent (in this case, the last response was to a Set System Options command - Command C).

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command forces a hardware reset. The I/O unit is restored to the configuration stored in EEPROM. The factory default configuration is used if nothing was previously saved to EEPROM.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	B	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
B = ASCII Character B (42 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	42 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
42 Hex = Command B (42 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

A delay of 1 second minimum is required before issuing another command. The first command sent after this command must be the POWER UP CLEAR command. The I/O unit will issue an acknowledgment to the host of the RESET command, prior to performing the hardware reset. A RESET command issued to an Analog I/O unit will initiates the following actions:

ANALOG:

Initial DAC output settings are restored from EEPROM.
I/O Module Configuration Type is restored from EEPROM.
The Option Control Byte is restored from EEPROM.
COM Link Watchdog Time Delay setting is restored from EEPROM.
COM Link Watchdog DAC Values are restored from EEPROM.
Response Delay Setting is restored from EEPROM.
PID Loop Parameters are restored from EEPROM.
Event/Reaction table entries 0 to 63 are restored from EEPROM.
Event/Reaction table entries 64 to 255 are cleared.
Offset,, Gain and Engineering Scaling unit parameters are restored from EEPROM.
Global Event Interrupt is disabled.

EXAMPLES:

ASCII Protocol:

Command	>22BA6cr
Response	A41cr

Sends a Reset command in ASCII protocol to the I/O unit at address 22 Hex and restores it to the configuration that was last saved in EEPROM. See command E. Data verification method is 8-bit checksum.

Binary Protocol:

Command	22 03 42 CB 20
Response	03 00 F0 00

Sends a Reset command in Binary protocol to the I/O unit at address 22 Hex and restores it to the configuration that was last saved in EEPROM. See command E. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command will cause all modules, PID and event parameters to be reset to the factory default conditions. See Remarks for a list of the conditions.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	J	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
J = ASCII Character J (4A Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	4A hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
4A Hex = Command J (4A Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command will cause all modules, PID and event parameters to be reset to the following factory default conditions:

All modules are configured as generic input, Type 00.
All module offsets are cleared to zero.
All module gain coefficients are set to 1.0
For all modules, Engineering unit scaling is set for 0 to 4095.
All totalization and averaging is halted.
All PID loops are stopped.
All PID data is cleared.
The Event/Reaction Table is cleared.
Global event interrupt is disabled.
Communications link watchdog is disabled.
Response Delay is set to zero.

EXAMPLES:

ASCII Protocol:

Command	>22JAEcr
Response	A41cr

Sends a Reset All Parameters To Default command in ASCII protocol to the I/O unit at address 22 Hex. The addressed I/O unit is restored to the factory default configuration. Data verification method is 8-bit checksum.

Binary Protocol:

Command	22 03 4A 0D 21
Response	03 00 F0 00

Sends a Reset All Parameters To Default command in Binary protocol to the I/O unit at address 22 Hex. The addressed I/O unit is restored to the factory default configuration. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets the communications line watchdog timeout period for the addressed I/O unit. When enabled, if a command (or an ">" character for ASCII protocol) is not received after the time delay specified by data field TTTT, the watchdog timer will time out. Upon time out, the analog output modules which have been instructed to output a specified value (by command H) will do so. After a watchdog timer has timed out, the next command sent to the I/O unit must be a Power Up Clear Command (Command A) otherwise an error will be returned. A delay of zero (0) disables the watchdog function.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	2 or 4	1
>	AA	D	TTTT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
D = ASCII Character D (44 Hex)
TTTT = Time Delay x 10 ms. (Minimum delay is 200 milliseconds, TTTT = 0014 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1 or 2
AA	LEN	44 hex	TTTT	DVF

Where: AA = Address
LEN = Length Field (04 or 05 Hex)
44 Hex = Command D (44 Hex)
TTTT = Time Delay x 10 ms. (Minimum delay is 200 milliseconds. TTTT = 0014 Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command is intended to be used in conjunction with command H.

Upon timeout, DACs are set to previously specified values, then a complete scan of the Event/Reaction Table is done starting at 0.

Before this command can set output I/O modules (DAC) to specified values, you must first configure the appropriate I/O modules for outputs using commands 'G' or 'a' to set the configuration.

The minimum delay for the watchdog timer is 200 milliseconds (TTTT = 0014 Hex).

If the watchdog timer for a particular I/O unit times out, the next instruction sent to the I/O unit will not be executed. Instead an error code will be sent back to the host computer. This error code is sent as a warning to let the host know that a timeout occurred. Subsequent commands will be executed in a normal manner, provided the time interval between commands is shorter than the watchdog timer delay time.

On power up this parameter is restored from EEPROM memory.

EXAMPLES:

ASCII Protocol:

Command	>F2D03E89Ccr
Response	A41cr

Commands the I/O unit at address F2 to set the watchdog time delay to 10 seconds (03E8 Hex = 1000). Upon time out, the analog output modules which have been instructed to output a specified value (by command H) will do so. Protocol is ASCII. Data verification method is 8-bit checksum. Command checksum is 9C Hex and response checksum is 41 Hex.

Binary Protocol:

Command	F2 05 44 00 00 0C 79
Response	03 00 F0 00

Commands the I/O unit at address F2 to disable the watchdog timer. Data field TTTT = 0000. Protocol is Binary. Data verification method is 16-bit CRC. Command data verification field is 0C79 Hex and response data verification field is F000 Hex. The Length Field for the command is 05 indicating that 5 bytes are to follow the lengthfield.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the desired output state for the specified analog modules upon a communications link watchdog timeout condition. A one (1) in data field MMMM indicates that the specified output module will have its output changed to the value indicated by the corresponding field DDDDDDDD. A zero (0) in data field MMMM indicates that the value for the specified output module will not be changed. For each bit in MMMM that is set to a one (1), there must be a corresponding 8-byte (ASCII) or 4-byte (Binary) data field DDDDDDDD. The value in field DDDDDDDD is in Engineering units. It specifies the value to which the output module will be set.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	0 or more	2 or 4	1
>	AA	H	MMMM	DDDDDDDD	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- H = ASCII Character H (48 Hex)
- MMMM = Modules to be changed upontimeout
- DDDDDDDD = Value, in Engineering units, to which the specified output module will be set upon watchdog timer timeout. For each bit set to a 1 in MMMM, there must be a corresponding 8-byte data field DDDDDDDD.
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex) DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	0 or more	1 or 2
AA	LEN	48 hex	MMMM	DDDD	DVF

Where: AA = Address
LEN = Length Field
48 Hex = Command H (48 Hex)
MMMM = Modules to be changed upon timeout
DDDD = Value, in Engineering units, to which the specified output module will be set upon watchdog timer timeout. For each bit set to a 1 in MMMM, there must be a corresponding 4-byte data field DDDD.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command is intended to be used in conjunction with command D. You can use this command to instruct the I/O unit to set the analog output modules, specified by data field MMMM, to the value specified by data field DDDDDDDD or to do nothing upon watchdog timer timeout.

Data field MMMM consists of 16-bits. Each bit corresponds to a specific output module. The most significant bit corresponds to module #15 and the least significant bit corresponds to module #00.

A module is selected by setting a '1' in the corresponding data field MMMM. The correspondence between the bits in field MMMM and the module numbers is illustrated by the following example:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
Hex Data	0	0	3	0

In the above example, bits are set for modules 15, 13, 09, 07, 06, and 02. These (DAC) output modules will have their output settings changed to the new values given by data fields DDDDDDDD. Six (6) modules have been specified, therefore six (6) data fields DDDDDDDD must be sent following data field MMMM.

Bits in field MMMM which are zero (0) correspond to modules that will not have their outputs changed upon a watchdog timer timeout. In the above example, modules 14, 12, 11, 10, 08, 05, 04, 03, 01 and 00 will not be changed when the watchdog timer times out.

Data fields DDDDDDDD are sent in the same order that the bits appear in data field MMMM. Data for the most significant bit set is sent first and data for the least significant bit set is sent last. In the above example, the first data field DDDDDDDD following MMMM is the data for module 15. the second data field DDDDDDDD following MMMM is for module 13, etc. The last data field DDDDDDDD is for module 02.

Data fields DDDDDDDD are 32-bits in length and represents a signed integer value in Engineering units. Except for output module type 80, the scaling parameters for Engineering units are determined automatically when you specify the output module type (by command 'G' or command 'a'). For example, if you specify that module 15 is an OPTO 22, DA4 analog output module (0 to 5 VDC in Engineering units), then the data field DDDDDDDD represents volts in increments of 1/65,536 of a volt. The range is 0 to 5 VDC. If DDDDDDD exceeds 00050000 Hex = 5 VDC it will be clipped to 00050000 Hex = 5 VDC. If DDDDDDDD is less than 00000000, it will be set to 00000000.

Engineering units are real-world units of measurement such as pressure, temperature, PSIG, DC volts, etc. For a 12-bit OPTO 22 analog module, the numbers that you read from or write to an analog module are normally in the range of 0 to 4,095 (some modules allow some under-range and over-range). You can scale the inputs and/or outputs from the analog modules to represent Engineering units. For example, you may wish to scale the input so that the readings 0 to 4,095 correspond to 4 to 40 milliamperes. Command 'f' allows you to set the Engineering Scaling units for anything that you wish.

Even though the Engineering units are specified in increments of 1/65,536 of an Engineering unit, the resolution of the analog module is still only 12-bits.

For output module type 80 Hex, you can define your own Engineering units by using command 'f'. The default will be 0 to 4,095 counts unless you use command 'f' to set different scaling parameters. For example, assume that you have used command 'a' to specify that module 13 is generic type 80 Hex. Further assume that you want the Engineering units to be 0 to 3000 pounds per sq. in. pressure. You would use command 'f' to set the proper scaling parameters. Thereafter DDDDDDD would represent pounds pressure in increments of 1/65536 of a pound. The maximum value that you could use for DDDDDDDD would be 0BB80000 Hex = 3000 PSI and the minimum value would be 00000000 = 0 PSI.

Before this command can change the output for the specified output modules, you must first configure the appropriate I/O modules for outputs. Commands 'G' or 'a' can be used to set the configuration.

The minimum delay for the watchdog timer is 200 milliseconds (TTTT = 0014 Hex).

If the watchdog timer for a particular I/O unit times out, the next instruction sent to the I/O unit will not be executed. Instead an error code will be sent back to the host computer. This error code is sent as a warning to let the host know that a timeout occurred. Subsequent commands will be executed in a normal manner, provided the time interval between commands is shorter than the watchdog timer delay time.

The parameters in data fields MMMM and DDDDDDDD can be saved in EEPROM by command E. On powerup the parameters are restored from EEPROM memory.

EXAMPLES:

ASCII Protocol:

Command	>F2H80080BB800000002800C8cr
Response	A41cr

Instructs the I/O unit at address F2 to set output module 15 to 0BB80000 Hex and to set the module at position 3 to 2.5 VDC. Assuming that module 15 is a type 80 (generic analog output module) and that command 'f' was used to re-scale the Engineering units to 0 to 3,000 PSI, then the first data field DDDDDDDD = 0BB80000 Hex = 3000 would represent 3,000 PSI. Assuming that module 03 is an OPTO 22 type 84, DA5 analog output module (Engineering units to 0 to 5 VDC), then the second data field DDDDDDDD = 00028000 Hex = 2.5 would represent 2.5 VDC. Upon watchdog timer timeout, modules 15 and 03 would be set to the above values. All other modules would remain unchanged.

Protocol is ASCII. Data verification method is 8-bit checksum. The command checksum is C8 Hex. The response checksum is 41 Hex. The leading character in the response data frame is an ASCII 'A', indicating that there were no errors.

Binary Protocol:

Command	F2 0D 48 80 08 0B B8 00 00 00 02 80 00 51 0C
Response	03 00 F0 00

Instructs the I/O unit at address F2 to set output module 15 to 0BB80000 Hex. Assuming that module 15 is a type 80 (generic analog output module) and that command 'f' was used to re-scale the Engineering units to 0 to 3000 PSI, then the first data field DDDDDDDD = 0BB80000 Hex = 3000 would represent 3,000 PSI. Assuming that module 03 is an OPTO 22 type 84, DA5 analog output module (Engineering units to 0 to 5 VDC), then the second data field DDDDDDDD = 00028000 Hex = 2.5 would represent 2.5 VDC. Upon watchdog timer timeout, modules 15 and 03 would be set to the above values. All other modules would remain unchanged.

Protocol is Binary. Data verification method is 16-bit CRC. Data verification field is 510C. The Length Field is 0D Hex = 13 indicating that 13 bytes are to follow the length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET RESPONSE DELAY — COMMAND ~

DESCRIPTION:

This command is used to set a delay time for command responses. Data field DD specifies the delay time in units of 10 milliseconds.

VERSIONS:

Analog Version 1.09 or later.

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	~	DD	DVF	CR

Where:

>	=	ASCII Character > (3E Hex)	AA	=	Address
~	=	ASCII Character ~ (7E Hex)	DD	=	Response Delay
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:

A	=	ASCII Character A (41 Hex)	DVF	=	Data Verification Field
CR	=	Carriage Return (0D Hex)			

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:

N	=	ASCII Character N (4E Hex)	EC	=	Error Code
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

1	1	1	1	1 or 2
AA	LEN	7E hex	DD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
7E Hex = Command ~ (7E Hex) DD = Response Delay
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

After acknowledging the 'Set Response Delay' command, for all subsequent commands, the I/O unit will wait DD x 10 milliseconds before responding. Note that any particular command is executed immediately and that only the acknowledge or data response is delayed.

The default response delay setting is zero unless a different value has been set and saved in EEPROM with a 'Store System Configuration' command.

EXAMPLES:

ASCII Protocol:

Command	>38~054Ecr
Response	A41cr

Sends a Set Response Delay command in ASCII protocol to the I/O unit at address 38 Hex. The response delay will be set to 50 milliseconds. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 7E 05 02 AD
Response	03 00 F0 00

Sends a Set Response Delay command in Binary protocol to the I/O unit at address 38 Hex. The response delay will be set to 50 milliseconds. Data verification method is 16-bit CRC.

Note: Blank spaces in the above example are *added* for clarity only. They do not actually appear in the command or in the response.

DESCRIPTION:

This command is used to set (SS) or clear (CC) the bits in the Option Control Byte. The Option Control Byte is used to select certain system options. The system options and the controlling bits are as follows:

Bit 0 = Temperature conversion unit: 0 = Celsius (Centigrade), 1 = Fahrenheit
Bit 1 = Reserved. Not used.
Bit 2 = Reserved. Not used.
Bit 3 = Reserved. Not used.
Bit 4 = CRC initialization value: 0 = 0000, 1 = FFFF.
Bit 5 = CRC method select: 0 = reverse, 1 = classical.
Bit 6 = CRC polynomial select: 0 = CRC16, 1 = CCITT.
Bit 7 = Global event interrupt enable: 0 = disabled, 1 = enabled.

The Option Control Byte is stored in EEPROM when command E is executed and is restored upon power up or when the RESET command B is executed. The factory default value for the Option Control Byte is 00.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	C	SS	CC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- C = ASCII Character C (43 Hex)
- SS = Bits to set in Option Control Byte
- CC = Bits to clear in Option Control Byte
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2	2 or 4	1
A	00	DD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- 00 = Reserved. Not used.
- DD = Option Control Byte
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	43 hex	SS	CC	DVF

Where: AA = Address LEN = Length Field (04 or 05 Hex)
43 Hex = Command C (43 Hex) SS = Bits to set in Option Control Byte
CC = Bits to clear in Option Control Byte
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1	1	1 or 2
LEN	EC=00	00	DD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
00 = Reserved. Not used. DD = Option Control Byte
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

You can use this command to set (SS), clear (CC) or do nothing to the bits in the Option Control Byte. You can use the Store System Configuration (Command E) to save the Option Control Byte in EEPROM. The Option Control Byte is always restored from EEPROM upon power up or when the RESET (Command B) is executed.

The response from this command is a 16-bit word. The current value of the Option Control Byte is returned as the least significant byte of the 16-bit word. The most significant byte can be ignored. It is for future expansion.

EXAMPLES:

ASCII Protocol:

Command	>50C00FF94cr
Response	A000001cr

Instructs the I/O unit at address 50 to clear all bits of the Option Control Byte. This is the factory default condition. Protocol is ASCII. Data verification method is 8-bit checksum.

Binary Protocol:

Command	50 05 43 00 FF 54 71
Response	05 00 00 00 CC 00

Instructs the I/O unit at address 50 to clear all bits of the Option Control Byte. This is the factory default condition. Protocol is Binary. Data verification method is 16-bit CRC. The Length Field for the command is 05 indicating that 5 bytes are to follow the length field. The Length Field for the response is also 05.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Analog I/O Configuration Commands

CALCULATE AND SET ADC MODULE OFFSET — COMMAND d

DESCRIPTION:

This command instructs the addressed I/O unit to use the current input value as the zero point input value. The current reading is multiplied by minus one (-1) and is used as an offset. The offset value is returned to the host computer. This command is intended to be used for calibration purposes.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	d	CC	DVF	CR

Where:

> = ASCII Character > (3E Hex)
d = ASCII Character d (64 Hex)
DVF = Data Verification Field
AA = Address
CC = Channel Number (00 to 0F Hex)
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	2 or 4	1
A	OOOO	DVF	CR

Where: A = ASCII Character A (41 Hex)
OOOO = Offset in counts, signed 16-bit integer
DVF = Data Verification Field   CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex)   EC = Error Code
DVF = Data Verification Field   CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	64 hex	CC	DVF

Where: AA = Address (00 to FF Hex)   LEN = Length Field (04 or 05 Hex)
64 Hex = Command d (64 Hex)   CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	OOOO	DVF

Where: LEN = Length Field (04 or 05 Hex)   EC = Error Code = 00 Hex (Zero)
OOOO = Offset in counts, signed 16-bit integer
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

WhereLEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command is intended to be used, along with command e, to calibrate ADC input modules for proper offset and gain.

EXAMPLES:

ASCII Protocol:

Command	>4Cd023Dcr
Response	A000001cr

Sends a Calculate and Set ADC Module Offset command in ASCII protocol to the I/O unit at address 4C Hex. The current input to the ADC module is used as the zero reference point for all further inputs. The response data field OOOO = 0000 indicates that the offset is zero. Data verification method is 8-bit checksum.

Binary Protocol:

Command	70 04 64 02 00 F0
Response	05 00 00 00 CC 00

Sends a Calculate and Set ADC Module Offset command in Binary protocol to the I/O unit at address 70 Hex. The current input to the ADC module is used as the zero reference point for all further inputs. The response data field OOOO = 0000 indicates that the offset is zero. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command instructs the addressed I/O unit to use the current input value as the full scale input value. A gain coefficient is calculated and used for all subsequent readings. The coefficient is returned to the host computer. This command is intended to be used for calibration purposes

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	e	CC	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
e = ASCII Character e (65 Hex) CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	GGGG	DVF	CR

Where: A = ASCII Character A (41 Hex)
GGGG = Gain coefficient, unsigned 16-bit integer scaled by a factor of 4096.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	65 hex	CC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
65 Hex = Command e (65 Hex) CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	GGGG	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
GGGG = Gain coefficient unsigned 16-bit integer scaled by a factor of 4096
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

This command is intended to be used, along with command d, to calibrate ADC input modules for proper offset and gain.

EXAMPLES:

ASCII Protocol:

Command	>4Ce023Ecr
Response	A0F5B2Ecr

Sends a Calculate and Set ADC Module Gain command in ASCII protocol to the I/O unit at address 4C Hex. The current input to the ADC module is used as the full scale point for all further inputs. The response data field GGGG = 0F5B Hex (3931) indicates that the gain coefficient is 3931/4096 = 0.960. Data verification method is 8-bit checksum.

Binary Protocol:

Command	70 04 65 02 90 F1
Response	05 00 0F 5B C7 44

Sends a Calculate and Set ADC Module Gain command in Binary protocol to the I/O unit at address 70 Hex. The current input to the ADC module is used as the full scale point for all further inputs. The response data field GGGG = 0F5B Hex (3931) indicates that the gain coefficient is 3931/4096 = 0.960. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command causes the addressed I/O unit to send back a response to the host that identifies the module configuration type for each of the 16 channels. Data is returned as 2 ASCII Hex digits for each of the 16 channel for ASCII protocol. For Binary protocol, data is returned as 1 data byte for each channel. See Remarks for possible configuration types.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	Y	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
Y = ASCII Character Y (59 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	32	2 or 4	1
A	TT	DVF	CR

Where A = ASCII Character A (41 Hex) TT = Module Type (Configuration)
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Note: *Two ASCII characters (representing Hex data) are returned for each channel.*

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1 or 2
AA	LEN	59 hex	DVF

Where AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
59 Hex = Command Y (59 Hex) DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	16	1 or 2
LEN	EC=00	TT	DVF

Where: LEN = Length Field (12 or 13 Hex) EC = Error Code = 00 Hex (Zero)
TT = Module Type (Configuration) DVF = Data Verification Field

Note: *One data byte is returned for each channel.*

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

For ASCII protocol, the Return Data Field (TT) will contain 32 ASCII (Hex) characters. For Binary protocol, the Return Data Field (TT) will contain 16 binary data bytes, one for each channel. The data is returned in a specific order from the highest channel number to the lowest. Data for channel 15 is returned first and data for channel 00 is returned last.

The returned data is the Configuration Type for each channel (module). The data is interpreted as follows:

INPUTS - Bit 7 = 0

00 = Generic Input Module
01 = Reserved
02 = Reserved
03 = G4AD3 4 to 20 mA
04 = G4AD4 ICTD
05 = G4AD5 Type J Thermocouple
06 = G4AD6 0 to 5 VDC
07 = G4AD7 0 to 10 VDC
08 = G4AD8 Type K Thermocouple
09 = G4AD9 0 to 50 mV
0A = G4AD10 100 Ohm RTD
0B = G4AD11 -5 to +5 VDC
0C = G4AD12 -10 to +10 VDC

INPUTS - Bit 7 = 0 (Cont.)

0D = G4AD13 0 to 100 mV
10 = G4AD16 0 to 5 Amperes
11 = G4AD17 Type R Thermocouple
12 = G4AD18 Type T Thermocouple
13 = G4AD19 Type E Thermocouple
14 = G4AD20 0 to 4095 Hz.
16 = G4AD22 0 to 1 VDC
17 = G4AD17 Type S Thermocouple
18 = G4AD24 Type B Thermocouple
19 = G4AD25 0 to 100 VAC/VDC

OUTPUTS - Bit 7 = 1

80 = Generic Output Module
81 = Reserved
82 = Reserved
83 = G4DA3 4 to 20 mA
84 = G4DA4 0 to 5 VDC
85 = G4DA5 0 to 10 VDC
86 = G4DA6 -5 to +5 VDC
87 = G4DA7 -10 to +10 VDC
88 = G4DA8 0 to 20 mA
89 = G4DA9 Time Proportional Output

EXAMPLES:

ASCII Protocol:

Command	>4CYD0cr
Response	A 84 00 00 00 00 00 00 00 00 00 00 00 84 06 04 04 67 cr

Sends a Read Module Configuration command in ASCII protocol to the I/O unit at address 4C Hex. Data verification method is 8-bit checksum. Data verification field DVF = 67 Hex for the response and DVF = D0 Hex for the command. Response indicates that channel 15 is configured as a DA4 (0 to 5 VDC) analog output (84 Hex), channels 14 thru 04 are configured as generic input modules, channel 03 is configured as a DA4 (0 to 5 VDC) analog output module (84 Hex), channel 02 is configured as an AD6 (0 to 5 VDC) analog input module (06 Hex), and channels 01 and 00 are configured as AD4 (ICTD) temperature detector inputs (04). Unless the configuration is changed by commands 'G' or 'a', the modules are set by default to generic analog input modules, type 00.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Binary Protocol:

Command	52 03 59 1B 61
Response	13 00 84 00 00 00 00 00 00 00 00 00 00 00 84 06 04 04 10 5C

Sends a Read Module Configuration command in Binary protocol to the I/O unit at address 52 Hex. Data verification method is 16-bit CRC. Data verification field DVF = 105C Hex for the response and DVF = 1B61 Hex for the command. Response indicates that channel 15 is configured as a DA4 (0 to 5 VDC) analog output (84 Hex), channels 14 thru 04 are configured as generic input modules, channel 03 is configured as a DA4 (0 to 5 VDC) analog output (84 Hex), channel 02 is configured as an AD6 (0 to 5 VDC) analog input (06 Hex), and channels 01 and 00 are configured as AD4 (ICTD) temperature detector inputs (04). Unless the configuration is changed by commands 'G' or 'a', the modules are set by default to generic analog input modules, type 00. The length field of 13 Hex (19 decimal) for the response indicates that 19 data bytes will follow the length field. Length field for the command is 03 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets an offset which is added to the raw count data before any engineering unit scaling is done. Units are counts. Data field CC specifies the analog input module.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	2 or 4	1
>	AA	b	CC	OOOO	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
b = ASCII Character b (62 Hex) CC = Channel Number (00 to 0F Hex)
OOOO = Offset in counts (0 to 4095) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	1 or 2
AA	LEN	62 hex	CC	OOOO	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
62 Hex = Command b (62 Hex) CC = Channel Number (00 to 0F Hex)
OOOO = Offset in counts (0 to 4095) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command sets the offset in counts for the analog input module specified by data field CC. An offset of zero is the factory default setting. If command E is used to store the system configuration, this parameter will be restored upon power up or Reset.

EXAMPLES:

ASCII Protocol:

Command	>4Cb020000FBcr
Response	A41cr

Sends a Set ADC Module Offset command in ASCII protocol to the I/O unit at address 4C Hex. Data field CC = 02 specifies that the analog input channel is 02. Data field OOOO = 0000 specifies an offset of 0000 counts. Data verification method is 8-bit checksum. The command checksum is FB Hex. Response checksum is 41 Hex.

Binary Protocol:

Command	91 06 62 02 00 00 F9 2A
Response	03 00 F0 00

Sends a Set ADC Module Offset command in Binary protocol to the I/O unit at address 91Hex. Data field CC = 02 specifies that the analog input channel is 02. Data field OOOO = 0000 specifies an offset of 0000 counts. Data verification method is 16-bit CRC. The command data verification field DVF = F92A Hex. Response data verification field DVF = F000 Hex. Length field LEN = 06 Hex for the command indicated that 6 bytes will follow the length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET ADC MODULE GAIN — COMMAND c

DESCRIPTION:

This command is used to set the gain coefficient for the analog input channel specified by data field CC. The gain coefficient is specified by data field GGGG. The gain coefficient is an unsigned 16-bit integer scaled by a factor of 4096.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	2 or 4	1
>	AA	c	CC	GGGG	DVF	CR

Where:
> = ASCII Character > (3E Hex)
c = ASCII Character c (63 Hex)
GGGG = Gain Coefficient
CR = Carriage Return (0D Hex)
AA = Address
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	2	1 or 2
AA	LEN	63 hex	CC	GGGG	DVF

Where: AA = Address (00 to FF Hex)
63 Hex = Command c (63 Hex)
GGGG = Gain Coefficient
LEN = Length Field (05 or 06 Hex)
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

Remarks:

This command sets the gain for the analog input module specified by data field CC. A count of 4096 is the factory default setting and corresponds to a gain coefficient of 1.0. Setting the gain coefficient to the value specified by data field GGGG has the same effect as making this the new full scale value. If command E is used to store the system configuration, this parameter will be restored upon power up or Reset.

EXAMPLES:

ASCII Protocol:

Command	>4Cc021000FDcr
Response	A41cr

Sends a Set ADC Module Gain command in ASCII protocol to the I/O unit at address 4C Hex. Data field CC = 02 specifies that the analog input channel is 02. Data field GGGG = 1000 (4096) specifies a gain coefficient of 4096/4096 = 1.00. Data verification method is 8-bit checksum. The command checksum is FD Hex. Response checksum is 41 Hex.

Binary Protocol:

Command	91 06 63 02 10 00 C5 26
Response	03 00 F0 00

Sends a Set ADC Module Gain command in Binary protocol to the I/O unit at address 91 Hex. Data field CC = 02 specifies that the analog input channel is 02. Data field GGGG = 1000 (4096) specifies a gain coefficient of 4096/4096 = 1.00. Data verification method is 16-bit CRC. The command data verification field DVF = C526 and the response data verification field DVF = F000 Hex. The length data field LEN = 06 Hex for the command indicates that 6 bytes will follow the length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command initiates digital filtering on the channel specified by data field CC. Averaging is permitted for input channels only. The averaging algorithm is:

New Average = ((Current Reading - Old Average) / DDDD) + Old Average

Data field DDDD specifies the sample weight. A value of DDDD = 0000 specifies that averaging is to be discontinued.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	2 or 4	1
>	AA	h	CC	DDDD	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
h = ASCII Character h (68 Hex) CC = Channel Number (00 to 0F Hex)
DDDD = Sample Weight DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	1 or 2
AA	LEN	68 hex	CC	DDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
68 Hex = C ommand h (68 Hex) CC = Channel Number (00 to 0F Hex)
DDDD = Sample Weight DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command initiates averaging on the specified channel. When a Read command (commands R, r, S or s) is issued for data type 01 (Average Counts) or data type 11 (Average Engineering units), the specified data will be returned. If this command is not issued *first*, then any read command that specifies an average will return a reading of -32768 to indicate an error condition.

The averaging algorithm

New Average = ((Current Reading - Old Average) / DDDD) + Old Average

used by this command is basically not an averaging algorithm but is an implementation of a first order lag digital filter. Ref. "Programming Microprocessor Interfaces for Control and Instrumentation" by Michael Andrews page 273. The sample weight DDDD is related to the sampling period = TS and the time constant = T1 by the following equation:

DDDD = (TS + T1)/TS

The sampling period TS is fixed at TS = 100 mS. The above equation can be used to compute DDDD if the time constant T1 is known.

If averaging is active, the peak and valley detection is derived from the average reading and is compared each time the average is calculated (every 100 milliseconds) to see if a new peak or valley has occurred.

EXAMPLES:

ASCII Protocol:

Command	>4Ch02001002cr
Response	A41cr

Sends a Set Averaging Sample Weight (Digital Filtering) command in ASCII protocol to the I/O unit at address 4C. Commands channel 02 to initiate averaging. Data field DDDD = 0010 specifies that the sample weight is 0010 Hex (16 decimal). The time constant associated with the first order lag filter is T1 = 1500 mS. Data verification method is 8-bit checksum. Note that the peak and valley detection is affected by this command. When averaging is active, the peaks and valleys apply to the average readings and not to the raw input data.

Binary Protocol:

Command	35 06 68 02 10 00 C5 3D
Response	03 00 F0 00

Sends a Set Averaging Sample Weight (Digital Filtering) command in Binary protocol to the I/O unit at address 4C. Commands channel 02 (CC = 02) to initiate averaging. Data field DDDD = 0010 specifies that the sample weight is 0010 Hex (16 decimal). The time constant associated with the first order lag filter is T1 = 1500 mS. Data verification method is 16-bit CRC. Note that the peak and valley detection is affected by this command. When averaging is active, the peaks and valleys apply to the average readings and not to the raw input data.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to configure the channel specified by data field CC to the configuration type specified by data field TT. See Remarks for the configuration types.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	a	CC	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
a = ASCII Character a (61 Hex)
CC = Channel Number (00 to 0F Hex)
TT = Configuration Type
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	61 hex	CC	TT	DVF

Where:
AA = Address (00 to FF Hex)
61 Hex = Command a (61 Hex)
TT = Configuration Type
LEN = Length Field (04 or 05 Hex)
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where:
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where:
LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

The channel number specified by CC is in the range 00 to 0F. The Configuration Type specified by data field TT must be one of the following:

INPUTS - Bit 7 = 0

00 = Generic Input Module
01 = Reserved
02 = Reserved
03 = G4AD3 4 to 20 mA
04 = G4AD4 ICTD
05 = G4AD5 Type J Thermocouple
06 = G4AD6 0 to 5 VDC
07 = G4AD7 0 to 10 VDC
08 = G4AD8 Type K Thermocouple
09 = G4AD9 0 to 50 mV
0A = G4AD10 100 Ohm RTD
0B = G4AD11 -5 to +5 VDC
0C = G4AD12 -10 to +10 VDC
0D = G4AD13 0 to 100 mV
10 = G4AD16 0 to 5 Amperes
11 = G4AD17 Type R Thermocouple
12 = G4AD18 Type T Thermocouple
13 = G4AD19 Type E Thermocouple
14 = G4AD20 0 to 4095 Hz.
16 = G4AD22 0 to 1 VDC
17 = G4AD17 Type S Thermocouple
18 = G4AD24 Type B Thermocouple
19 = G4AD25 0 to 100 VAC/VDC
1A = G4AD26 -5 V/s to +5 V/s dV/dt
1E = Type N Thermocouple*
1F = Type G Thermocouple*
20 = Type C Thermocouple*
21 = Type D Thermocouple*

* These types only supported by the High Density analog bricks.

OUTPUTS - Bit 7 = 1

80 = Generic Output Module
81 = Reserved
82 = Reserved
83 = G4DA3 4 to 20 mA
84 = G4DA4 0 to 5 VDC
85 = G4DA5 0 to 10 VDC
86 = G4DA6 -5 to +5 VDC
87 = G4DA7 -10 to +10 VDC
88 = G4DA8 0 to 20 mA
89 = G4DA9 Time Proportional Output

For all Configuration Types except 00 and 80, the Engineering Scaling units are set according to the Configuration Type specified.

EXAMPLES:

ASCII Protocol:

Command	>F2a0F80B7cr
Response	A41cr

Sends a Set Channel Configuration command in ASCII protocol to the I/O unit at address F2 Hex. Configures channel 0F Hex (15 decimal) as a generic analog output module (TT = 80 Hex). Engineering units default to counts (0 to 4095) unless command 'f' is used to specify a different Engineering units scaling parameter.

Binary Protocol:

Command	F2 05 61 0F 84 54 68
Response	03 00 F0 00

Sends a Set Channel Configuration command in Binary protocol to the I/O unit at address F2 Hex. Configures channel 0F Hex (15 decimal) as a type 84 DA5 0 to 5 VDC analog output module (Data field TT = 84 Hex). Engineering units will be 0 to 5 VDC. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the Engineering units Scaling Parameters specified by data fields HHHHHHHH and LLLLLLLL for the channel specified by data field CC. Data field HHHHHHHH specifies the Engineering units (in increments of 1/65,536 of an Engineering unit) corresponding to full scale (4095 counts). Data field LLLLLLLL specifies the Engineering units (in increments of 1/65,536 of an Engineering unit) corresponding to zero scale (0000 counts).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	8	2 or 4	1
>	AA	f	CC	HHHHHHHH	LLLLLLLL	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- f = ASCII Character f (66 Hex)
- CC = Channel Number (00 to 0F Hex)
- HHHHHHHH = Engineering units corresponding to full scale (4095 counts) in units of 1/65,536 of an Engineering unit.
- LLLLLLLL = Engineering units corresponding to zero scale (0000 counts) in units of 1/65,536 of an Engineering unit.
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	8	8	2 or 4
AA	LEN	66 hex	CC	HHHHHHHH	LLLLLLLL	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (0B or 0C Hex)
66 = Command f (66 Hex) CC = Channel Number (00 to 0F Hex)
HHHHHHHH = Engineering units corresponding to full scale (4095 counts) in increments of 1/65,536 of an Engineering unit.
LLLLLLLL = Engineering units corresponding to zero scale (0000 counts) in increments of 1/65536 of an Engineering unit.
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by data field CC must be configured as a type 00 generic input module or as a type 80 generic output module before issuing this command or else an error is returned.

Basically this command is used to scale an analog input or output module for the desired Engineering units. Suppose for example that you have a pressure transducer connected to an OPTO 22 AD3 analog input module. The AD3 module accepts 4 to 20 mA input signals from the external pressure transducer. Assume that the transducer is calibrated such that 4 mA corresponds to zero 0 PSIG and that 20 mA corresponds to 10,000 PSIG. You would first send command 'a' to configure the analog input module for Configuration Type 00 = generic analog input. Then you would want to send command 'f' with LLLLLLLL = 00000000 and HHHHHHHH = 27100000 Hex (10,000 decimal). Note that HHHHHHHH is in increments (units) of 1/65,536 PSIG. The upper 16-bits of HHHHHHHH represents the integral part of the data and the lower 16-bits represents the fractional part of the data.

Note that this command does not change the *actual* input or *actual* output for the specified module, only the scaling factor. For example, if you have an OPTO 22 DA8 0 to 20 mA module (Configuration Type 88) and you rescale it for 0 to 20 amperes Engineering units, you must still *only* apply 0 to 20 mA input to the module.

EXAMPLES:

ASCII Protocol:

Command	>F2f0F0BB8000000000000A0cr
Response	A41cr

Sends a Set Engineering unit Scaling Parameters command in ASCII protocol to the I/O unit at address F2 Hex. Specifies that channel 0F Hex is to be rescaled for 0 to 3000 RPM. Data field HHHHHHHH = 0BB80000 (3000 decimal) specifies the full scale Engineering units in increments of 1/65,536 RPM. Data field LLLLLLLL = 00000000 (0000 decimal) specifies the zero scale Engineering units in increments of 1/65,536 RPM. Data verification method is 8-bit checksum. Checksum is A0 Hex.

Binary Protocol:

Command	35 0C 66 0E 00 05 00 00 00 00 00 00 DD 89
Response	03 00 F0 00

Sends a Set Engineering unit Scaling Parameters command in Binary protocol to the I/O unit at address 35 Hex. Specifies that channel 0E Hex is to be rescaled for 0 to 5 hp. Data field HHHHHHHH = 00050000 (5 decimal) specifies the full scale Engineering units in increments of 1/65,536 hp. Data field LLLLLLLL = 00000000 (0000 decimal) specifies the zero scale Engineering units in increments of 1/65,536 hp. Data verification method is 16-bit CRC. Data Verification field is DD89 Hex for the command and F000 for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to configure the channels specified by data field MMMM to the configuration type specified by data field TT. Data field MMMM has 16-bits. Each bit in data field MMMM that is set to a 1 corresponds to an analog point that is to be configured to the type specified by the corresponding data field TT. For each bit in MMMM that is set to a one (1), there must be a corresponding data field TT. The highest order bit (bit 15) corresponds to module 15 and the lowest order bit (bit 00) corresponds to module 00. See Remarks for configuration types.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	0 to 32	2 or 4	1
>	AA	G	MMMM	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
G = ASCII Character G (47 Hex)
MMMM = Channels to be Configured
TT = Configuration Type (One field TT for each bit in MMMM that is set to a 1)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	2	0 to 16	1 or 2
AA	LEN	47 hex	MMMM	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field
47 Hex = Command G (47 Hex) MMMM = Channels to be Configured
TT = Configuration Type (One field TT for each bit in MMMM that is set to a 1)
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

A channel (module) is selected by setting a '1' in the corresponding 16-bit Command Data Field (MMMM). The correspondence is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 1 1 1	0 0 0 0	0 0 0 0	0 0 0 1
Hex Data	F	0	0	1

In the above example, data bits are set for channels (modules) 15, 14, 13, 12, and 00. For the ASCII protocol, the Command Data Field (CDF) will contain the four ASCII characters 'F001'. For the Binary protocol, the Command Data Field (CDF) will contain the two Hex data bytes 'F0' and '01'.

For each bit set to a one (1) in data field MMMM, there must be a corresponding data field TT that specifies the configuration type. In the above example MMMM = F001, therefore five (5) data fields TT are required. Data fields TT are sent in the same order as the one (1) bits in MMMM, most significant first and least significant last. In the above example, the first data field, TT, corresponds to the most significant bit in MMMM that is set to a one (1) (Channel 15) and the last data field, TT, corresponds to the least significant bit in MMMM that is set to a one (1) (Channel 00).

The Configuration Type specified by data field TT must be one of the follows:

INPUTS - Bit 7 = 0

00 = Generic Input Module
01 = Reserved
02 = Reserved
03 = G4AD3 4 to 20 mA
04 = G4AD4 ICTD
05 = G4AD5 Type J Thermocouple
06 = G4AD6 0 to 5 VDC
07 = G4AD7 0 to 10 VDC
08 = G4AD8 Type J Thermocouple
09 = G4AD9 0 to 50 mV
0A = G4AD10 100 Ohm RTD
0B = G4AD11 -5 to +5 VDC
0C = G4AD12 -10 to +10 VDC
0D = G4AD13 0 to 100 mV
10 = G4AD16 0 to 5 Amperes
11 = G4AD17 Type R Thermocouple
12 = G4AD18 Type T Thermocouple
13 = G4AD19 Type E Thermocouple
14 = G4AD20 0 to 4095 Hz.
16 = G4AS22 0 to 1 VDC
17 = G4AD17 Type S Thermocouple
18 = G4AD24 Type B Thermocouple
19 = G4AD25 0 to 100 VAC/VDC
1A = G4AD26 -5 V/s to +5 V/s dV/dt
1E = Type N Thermocouple*
1F = Type G Thermocouple*
20 = Type C Thermocouple*
21 = Type D Thermocouple*

* These types only supported by the High Density analog bricks.

OUTPUTS - Bit 7 = 1

80 = Generic Output Module
81 = Reserved
82 = Reserved
83 = G4DA3 4 to 20 mA
84 = G4DA4 0 to 5 VDC
85 = G4DA5 0 to 10 VDC
86 = G4DA6 -5 to +5 VDC
87 = G4DA7 -10 to +10 VDC
88 = G4DA8 0 to 20 mA
89 = G4DA9 Time Proportional Output

For all configuration types except 00 and 80, the Engineering Scaling units are set according to the configuration type specified. Default configuration type is 00.

This parameter is saved to EEPROM by command E. This parameter is restored from EEPROM upon power up or Reset.

EXAMPLES:

ASCII Protocol:

Command	>8F G 000F 84 06 04 04 35 cr
Response	A41cr

Sends a Set I/O Configuration - Group command in ASCII protocol to the I/O unit at address 8F Hex. Channels 03, 02, 01 and 00 are selected (MMMM = 000F) to be configured as follows:

Channel 03 = 84 Hex (DA4 0 to 5 VDC analog output)
Channel 02 = 06 Hex (AD6 0 to 5 VDC analog input)
Channel 01 = 04 Hex (AD4 ICTD analog input)
Channel 00 = 04 Hex (AD4 ICTD analog input)

Data verification method is 8-bit checksum.

Binary Protocol:

Command	8F 09 47 000F 84 06 04 04 AB48
Response	03 00 F0 00

Sends a Set I/O Configuration - Group command in Binary protocol to the I/O unit at address 8F Hex. Channels 03, 02, 01 and 00 are selected (MMMM = 000F) to be configured as follows:

Channel 03 = 84 Hex (DA4 0 to 5 VDC analog output)
Channel 02 = 06 Hex (AD6 0 to 5 VDC analog input)
Channel 01 = 04 Hex (AD4 ICTD analog input)
Channel 00 = 04 Hex (AD4 ICTD analog input)

Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET TOTALIZATION SAMPLE RATE — COMMAND g

DESCRIPTION:

This command initiates totalization on the specified channel. Totalization is allowed for both input and output analog channels. Totalization is also allowed for both filtered and unfiltered readings. Data field CC specifies the channel and data field DDDD specifies the sample rate in 100 mS units. DDDD is a 16-bit unsigned integer. Bit 15 is used as a flag to indicate whether totalization will be for filtered or unfiltered readings. If bit 15 is zero, the unfiltered readings will be totalized. If bit 15 is a one, the filtered readings will be totalized. A sample rate of DDDD = 0000 will discontinue totalization for the specified channel. See remarks for further information on this command.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	2 or 4	1
>	AA	g	CC	DDDD	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
g = ASCII Character g (67 Hex) CC = Channel Number (00 to 0F Hex)
DDDD = Sampling Rate for Totalization in units of 100 mS.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	1 or 2
AA	LEN	67 hex	CC	DDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
67 Hex = Command g (67 Hex) CC = Channel Number (00 to 0F Hex)
DDDD = Sampling Rate for Totalization in units of 100 mS.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command initiates totalization on the specified channel. When a Read command (commands R, r, S or s) is issued for data type 04 (Totalized Counts) or data type 14 (Totalized Engineering units), the specified data will be returned. If this command is not issued *first*, then any read command that specifies a totalized value will return an error. Note that while this command must be issued first, it needs to be issued only once. It does not have to be issued before *each* command R, r, S or s.

The sampling period DDDD is in units of 100 mS. DDDD is a 16-bit unsigned integer. Bit 15 is used as a flag to indicate whether totalization will be for filtered or unfiltered readings. If bit 15 is zero, the unfiltered readings will be totalized. If bit 15 is a one, the filtered readings will be totalized. A sample rate of DDDD = 0000 will discontinue totalization for the specified channel. The maximum totalization sample rate is 32767 x 100 milliseconds = 3,276.7 seconds.

If the module is an input and does not have the filtering function active, the total is set to -32768 (8000 Hex) and totalization is halted.

If the specified channel is an output, bit 15 of data field DDDD is ignored.

If the module is an input which is too far underscale to give a reading, or has stopped functioning for any reason, the total is set to -32768 (8000 Hex) as an error indicator, and totalization is halted. The totalization function may be restarted by issuing the "SET TOTALIZATION RATE" command again, or the "READ AND CLEAR I/O MODULE DATA" command.

A sampling period of 0000 will discontinue totalization.

The maximum totalized value is 65,536 Engineering units or 2^{32} counts.

EXAMPLES:

ASCII Protocol:

Command	>4C g 02 0002 02 cr
Response	A41cr

Sends a Set Totalization Sample Rate command in ASCII protocol to the I/O unit at address 4C. Commands channel 02 to initiate totalization. Data field DDDD = 0002 specifies that the sampling rate weight is 0002 Hex (2 decimal) x 100 mS = 200 mS. Data verification method is 8-bit checksum.

Binary Protocol:

Command	35 06 67 02 00 02 D0 B2
Response	03 00 F0 00

Sends a Set Totalization Sample Rate command in Binary protocol to the I/O unit at address 35 Hex. Commands channel 02 to initiate totalization. Data field DDDD = 0002 specifies that the sampling rate weight is 0002 Hex (2 decimal) x 100 mS = 200 mS. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the resolution of a time proportional output module (G4DA9). See remarks for resolution settings.

VERSIONS:

Analog - Firmware 1.06 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	]	CC	SS	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
-] = ASCII Character a (5D Hex)
- SS = Resolution Setting
- CR = Carriage Return (0D Hex)
- AA = Address
- CC = Channel Number (00 to 0F Hex)
- DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- CR = Carriage Return (0D Hex)
- DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

*** Time Proportional Output**

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	5D hex	CC	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
5D Hex = Command] (5D Hex) CC = Channel Number (00 to 0F Hex)
SS = Resolution Setting DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel number specified by CC is in the range 00 to 0F. The resolution setting specified by data field SS may be any number between 00 and FF. The resolution setting is in 500 microsecond units and affects the total TPO period according to the equation below:

TPO Period = Resolution Setting * 0.5 milliseconds * 4,096

Some typical resolution settings and resulting TPO periods:

Resolution Setting (Data Field SS)	**Time Per Bit** (ms)	**Total TPO Period** (sec)
0, 1	0.5	2.048
2	1	4.096
3	1.5	3.144
4	2	8.192
5	2.5	10.24
0A hex	5	20.48
1E hex	15	61.44
3B hex	29.5	120.8 (2.01 min)
75 hex	58.5	239.6 (3.99 min)
FF hex	127.5	522.2 (8.7 min)

The default resolution of the G4DA9 output module is 1 millisecond. The TPO period will be 4.096 seconds.

If the Store System Configuration command is executed, all TPO resolution settings will be saved and automatically restored upon power-up.

EXAMPLES:

ASCII Protocol:

Command	>07]04058Dcr
Response	A41cr

Sends a Set TPO Resolution command in ASCII protocol to the I/O unit at address 07. This command will set the resolution of a time proportional output module in channel 4 to 2.5 milliseconds per bit. This will result in a total TPO period of 10.24 seconds.

Binary Protocol:

Command	07 05 5D 0A 1E 7C 46
Response	03 00 F0 00

Sends a Set TPO Resolution command in Binary protocol to the I/O unit at address 7C Hex. This command will set the resolution of a time proportional output module in channel 10 to 15 milliseconds per bit. This will result in a total TPO period of 61.44 seconds.

STORE SYSTEM CONFIGURATION — COMMAND E

DESCRIPTION:

This command saves the current system parameters to EEPROM. The parameters saved to EEPROM are:

(1) I/O module configuration type.
(2) Initial DAC analog output settings.
(3) TPO Resolution Settings.
(4) Communications link watchdog DAC values.
(5) Option Control Byte.
(6) Input offset and gain settings.
(7) Engineering units scaling parameters.
(8) PID loop parameters.
(9) Communications link watchdog time.
(10) Response Delay Setting.

This command requires one (1) second to execute. The command response is sent after the command has finished executing.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	E	DVF	CR

Where:
> = ASCII Character > (3E Hex)   AA = Address
E = ASCII Character E (45 Hex)   DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)   DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	45 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
45 Hex = Command E (45 Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The parameters stored in EEPROM are restored upon power up or if a RESET (Command B) is issued. This command requires one (1) second to execute. The command response is sent after the command has finished executing.

The current DAC output settings at the time of execution of the 'E' command become the initial DAC output values which are restored after RESET.

Examples:

ASCII Protocol:

Command	>4CEBCcr
Response	A41cr

Sends a Store System Configuration command in ASCII protocol to the I/O unit at address 4C Hex. Data verification method is 8-bit checksum.

Binary Protocol:

Command	52 03 45 D2 60
Response	03 00 F0 00

Sends a Store System Configuration command in Binary protocol to the I/O unit at address 52 Hex. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

Analog Read, Write, Output Commands

RAMP DAC OUTPUT TO ENDPOINT — COMMAND Z

DESCRIPTION:

This command is used to ramp an analog output DAC module from its present setting to a specified ramp endpoint at a specified ramp rate. Data field EEEEEEEE specifies the ramp endpoint in Engineering units. Data field SSSSSSSS specifies the ramp slope in Engineering units per second. Data field CC specifies the output channel.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	8	2 or 4	1
>	AA	Z	CC	EEEEEEEE	SSSSSSSS	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
Z = ASCII Character Z (5A Hex)
CC = Channel Number
EEEEEEEE = Ramp Endpoint in Engineering units
SSSSSSSS = Ramp Slope in Engineering units per Second
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	4	1 or 2
AA	LEN	5A hex	CC	EEEEEEEE	SSSSSSSS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (0B or 0C Hex)
5A = Command Z (5A Hex) CC = Channel Number (00 to 0F Hex)
EEEEEEEE = Ramp Endpoint in Engineering units
SSSSSSSS = Ramp Slope in Engineering units per Second
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

WhereLEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by data field CC must be configured as an analog output before issuing this command. Command 'G' or command 'a' may be used. The ramp slope is an unsigned integer. The units are Engineering units per second in increments of 1/65536 of an Engineering unit per second. A slope of zero (SSSSSSSS = 00000000) is not allowed and will return an error. The ramp endpoint is a signed integer. The units are Engineering units in increments of 1/65536 of an Engineering unit. New data is sent to the specified DAC analog output module every 50 milliseconds.

EXAMPLES:

ASCII Protocol:

Command	>4C Z 03 00050000 00004000 3D cr
Response	A41cr

Sends a Ramp DAC Output to Endpoint command in ASCII protocol to the I/O unit at address 4C Hex. The DAC analog module at channel 03 is a configuration type 84 DA4 0 to 5 VDC analog output module. This command instructs channel CC = 03 Hex to ramp from its present setting to the endpoint specified by EEEEEEEE = 00050000 Hex (5.0 VDC). The ramp rate is specified to be SSSSSSSS = 00004000 Hex (1/4 volt per second). The ramp rate is calculated as follows: SSSSSSSS = 00004000 Hex = 16,384 decimal. Since this is in increments of 1/65,536 volts per second, we have 16,384 x (1/65,536) volts per second = .25 volts per second. Data verification method is 8-bit checksum. Checksum is 3D Hex.

Binary Protocol:

Command	69 0C 5A 03 00 05 00 00 00 00 40 00 9D F0
Response	03 00 F0 00

Sends a Ramp DAC Output to Endpoint command in Binary protocol to the I/O unit at address 69 Hex. The DAC analog module at channel 03 Hex is a configuration type 84 DA4 0 to 5 VDC analog output module. This command instructs channel CC = 03 Hex to ramp from its present setting to the endpoint specified by EEEEEEEE = 00050000 Hex (5.0 VDC). The ramp rate is specified to be SSSSSSSS = 00004000 Hex (1/4 volt per second). The ramp rate is calculated as follows: SSSSSSSS = 00004000 Hex = 16,384 decimal. Since this is in increments of 1/65,536 volts per second, we have 16,384 x (1/65,536 volts per second) = .25 volts per second. Data verification method is 16-bit CRC. Data Verification field is 9DF0 Hex for the command and F000 for the response. The Length field LEN = 0C Hex indicates that 0C Hex = 12 bytes will follow the Length field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command first reads then clears the analog channel (module) specified by data field CC. Data field TT specifies the type of data to be returned. See Remarks for data types. If the data type is Current (00, 10, 20 or 30) or Average (01, 11, 21 or 31), it will be read but will not be cleared. If the data type being requested is in counts (TT = 00, 01, 02, 03 or 04), then the return data field DDDD is 16-bits long (4 Hex characters in ASCII protocol). If the data type being requested is in *square root of counts* (TT = 20, 21, 22, 23 or 24), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol). If the data type being requested is in Engineering units (TT = 10, 11, 12, 13, 14, 30, 31, 32, 33 or 34), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	s	CC	TT	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- s = ASCII Character s (73 Hex)
- TT = Data Type. See Remarks
- CR = Carriage Return (0D Hex)
- AA = Address (00 to FF Hex)
- CC = Channel Number (00 to 0F Hex)
- DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 8	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDD = Magnitude Data in Counts (4 chars.), Square Root of Counts (8 chars.) or Engineering units (8 chars.)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	73 hex	CC	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
73 Hex = Command s (73 Hex) CC = Channel Number (00 to 0F Hex)
TT = Data Type. See Remarks DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2 or 4	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 to 07 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Magnitude Data in Counts (2 bytes), Square Root of Counts (4 bytes) or Engineering units (4 bytes)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Output modules can be read but will *not* be cleared by this command. Only current or totalized values can be read from *output* modules. If average, peak or lowest value is specified as the data type TT for an output module, then the current value will be returned instead.

The data type specified in data field TT must be one of the following types:

COUNTS

00 = Current Counts
01 = Average Counts
02 = Peak Counts
03 = Lowest Counts
04 = Totalized Counts
20 = Square Root of Current Counts
21 = Square Root of Average Counts
22 = Square Root of Peak Counts
23 = Square Root of Lowest Counts
24 = Square Root of Totalized Counts

ENGINEERING UNITS

10 = Current Engineering units
11 = Average Engineering units
12 = Peak Engineering units
13 = Lowest Engineering units
14 = Totalized Engineering units
30 = Square Root of Current Engineering units
31 = Square Root of Average Engineering units
32 = Square Root of Peak Engineering units
33 = Square Root of Lowest Engineering units
34 = Square Root of Totalized Engineering units

The specified analog I/O module is read and the data returned *before* it is cleared. Totalized values are cleared to zero (0). Peak values are cleared to 8000 Hex (-32,768) Engineering units or counts. Lowest values are cleared to 7FFF (32,767) Engineering units or counts. Square roots of values are computed from the base values each time that a square root value is requested, therefore square roots are not cleared. If the square root is requested and the value read from the I/O module is negative, then the value returned in data field DDDD will be the square root of the absolute value.

Channels without a module installed or thermocouple modules that have an open thermocouple will return a value of 8000 Hex (-32768) as an indication of error. Attempts to read an Average before Command 'h' is issued will return the same error value of 8000 Hex.

If averaging is active, the peak and valley detection is derived from the average reading and is compared each time the average is calculated (every 100 milliseconds) to see if a new peak or valley has occurred.

EXAMPLES:

ASCII Protocol:

Command	>4Cs0210ADcr
Response	A 0005 3570 D5cr

Sends a Read and Clear I/O Module Data command in ASCII protocol to the I/O unit at address 4C Hex. Data field CC = 02 Hex specifies that channel 02 is to be read and then cleared. Data field TT = 10 Hex specifies that the data is to be in Current Engineering units. Thirty two (32)-bits of data will be returned. For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. Therefore, the data will be in increments of 1/65,536 of DC Volt. The return data field DDDD = 00053570 (341,360 decimal). The value in Engineering units is 341,360 x 1/65,536 VDC = 5.209 VDC. Data verification method is 8-bit checksum. The command checksum is AD Hex.

Binary Protocol:

Command	69 05 73 02 11 72 6C
Response	07 00 00 05 35 70 02 07

Sends a Read and Clear I/O Module Data command in Decimal protocol to the I/O unit at address 69 Hex. Data field CC = 02 Hex specifies that channel 02 is to be read and then cleared. Data field TT = 11 Hex specifies that the data is to be in Average Engineering units. Thirty two (32)-bits of data will be returned. For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. Therefore, the data will be in increments of 1/65,536 of DC Volt. The return data field DDDD = 00053570 (341,360 decimal). The value in Engineering units is 341,360 x 1/65,536 VDC = 5.209 VDC. Data verification method is 16-bit CRC. The Data Verification Field DVF = 726C Hex for the command and DVF = 0207 Hex for the response. The Length data field LEN = 07 for the response indicates that 7 bytes of data follow the LEN field.

DESCRIPTION:

This command first reads, then clears the analog channels (modules) specified by data field MMMM. Data field TT specifies the type of data to be returned. See Remarks for data types. If the data type is Current (00, 10, 20 or 30) or Average (01, 11, 21 or 31), it will be read but will not be cleared. Output modules can be read but will not be cleared by this command. The number of data fields DDDD returned by this command will be equal to the number of 1's in MMMM. If the data type being requested is in counts (TT = 00, 01, 02, 03 or 04), then the return data field DDDD is 16-bits long (4 Hex characters in ASCII protocol). If the data type being requested is in *square root of counts* (TT = 20, 21, 22, 23 or 24), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol). If the data type being requested is in Engineering units (TT = 10, 11, 12, 13, 14, 30, 31, 32, 33 or 34), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	S	MMMM	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
S = ASCII Character S (53 Hex)
MMMM = Modules for Which Data is to be Returned.
TT = Data Type. See Remarks DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDD = Magnitude Data in Counts (4 chars. for each channel specified), Square Root of Counts (8 chars. for each channel specified) or Engineering units (8 chars. for each channel specified). One data field DDDD will be returned for each bit that is set to a 1 in MMMM (the specified channels).
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1	1 or 2
AA	LEN	53 hex	MMMM	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
53 Hex = Command S (53 Hex)
MMMM = Modules for Which Data is to be Returned.
TT = Data Type. See Remarks DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Magnitude Data in Counts (4 chars. for each channel specified), Square Root of Counts (8 chars. for each channel specified) or Engineering units (8 chars. for each channel specified). One data field DDDD will be returned for each bit that is set to a 1 in MMMM (the specified channels).
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

Output modules can be read but will *not* be cleared by this command. Only current or totalized values can be read from output modules. If average, peak or lowest value is specified as the data type TT for an output module, then the current value will be returned instead.

The data type specified in data field TT must be one of the following types:

COUNTS

00 = Current Counts
01 = Average Counts
02 = Peak Counts
03 = Lowest Counts
04 = Totalized Counts
20 = Square Root of Current Counts
21 = Square Root of Average Counts
22 = Square Root of Peak Counts
23 = Square Root of Lowest Counts
24 = Square Root of Totalized Counts

ENGINEERING UNITS

10 = Current Engineering units
11 = Average Engineering units
12 = Peak Engineering units
13 = Lowest Engineering units
14 = Totalized Engineering units
30 = Square Root of Current Engineering units
31 = Square Root of Average Engineering units
32 = Square Root of Peak Engineering units
33 = Square Root of Lowest Engineering units
34 = Square Root of Totalized Engineering units

The specified analog I/O module is read and the data returned *before* it is cleared. Totalized values are cleared to zero (0). Peak values are cleared to 8000 Hex (-32,768) Engineering units or counts. Lowest values are cleared to 7FFF (32,767) Engineering units or counts. Square roots of values are computed from the base values each time that a square root value is requested, therefore square roots are not cleared.

If the square root is requested and the value read from the I/O module is negative, then the value returned in data field DDDD will be the square root of the absolute value.

If averaging is active, the peak and valley detection is derived from the average reading and is compared each time the average is calculated (every 100 milliseconds) to see if a new peak or valley has occurred.

Channels without a module installed or thermocouple modules that have an open thermocouple will return a value of 8000 Hex (-32768) as an indication of error. Attempts to read an Average before Command 'h' is issued will return the same error value of 8000 Hex.

The channels to be read (and cleared) are specified by setting bits to a '1' in data field MMMM. A '0' in the data field MMMM indicates that no action is to be taken for the corresponding channel. Each bit in the 16-bit data field MMMM corresponds to an analog I/O channel. The correspondence between the bits in the data field MMMM and the channel numbers is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 1
Hex Data	0	0	0	7

In the above example, channels 2, 1, and 0 are to be read and cleared. No action is taken for the other channels.

EXAMPLES:

ASCII Protocol:

Command	> 4C S 0007 02 F1 cr
Response	A 0C8A 0A2D 09F7 FA cr

Sends a Read and Clear I/O Module Data - Group command in ASCII protocol to the I/O unit at address 4C Hex. Data field MMMM = 0007 Hex specifies that channels 02, 01 and 00 are to be read and then cleared. Data field TT = 02 Hex specifies that the data is to be in Peak Counts. Sixteen (16)-bits of data will be returned for each of the 3 channels specified.

For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. The data will be in counts. Zero counts corresponds to 0 VDC. A count of 4095 corresponds to 5 VDC. The return data field DDDD = 0C8A (3210 decimal) counts. This represents 3210/4095 x 5.0 VDC = 3.92 VDC.

The module plugged into channel 01 is an AD4 IC Temp. Sensor. The temperature range is -188.4 °C to +150 °C. Data will be in counts. Counts are converted to temperature in degrees Celsius by the equation: TEMP = (0.08262 * COUNTS) - 188.4. The return data field is DDDD = 0A2D Hex (2605 decimal) counts. This represents a temperature of 26.8 degrees Celsius.

The module plugged into channel 00 is an AD4 IC Temp. Sensor. The temperature range is -188.4 °C to +150 °C. Data will be in counts. Counts are converted to temperature in degrees Celsius by the equation: TEMP = (0.08262 * COUNTS) - 188.4. The return data field is DDDD = 09F7 Hex (2551 decimal) counts. This represents a temperature of 22.4 degrees Celsius.

Data verification method is 8-bit checksum. The command checksum is F3 Hex.

Binary Protocol:

Command	8F 06 53 00 07 11 47 44
Response	0F 00 00 03 EA 80 00 0E 4C 00 00 0E 04 00 34 C7

Sends a Read and Clear I/O Module Data - Group command in Decimal protocol to the I/O unit at address 8F Hex. Data field MMMM = 0007 Hex specifies that channels 02, 01 and 00 are to be read and then cleared. Data field TT = 11 Hex specifies that the data is to be in Average Engineering units. Thirty two (32)-bits of data will be returned for each channel specified. In this case 3 channels.

For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. The returned data in field DDDD will be in Engineering units in increments of 1/65536 of a VDC. The return data field DDDD = 0003EA80 (256,640 decimal). This represents 256,640/65,536 VDC = 3.92 VDC.

The module plugged into channel 01 is an AD4 IC Temp. Sensor. The temperature range is -188.4 °C to +150 °C. The returned data in field DDDD will be in Engineering units in increments of 1/65536 of a degree Celsius. The return data field DDDD = 000E4C00 (936,960 decimal). This represents 936,960/65,536 VDC = 14.3 degrees Celsius.

Data verification method is 16-bit CRC. The Data Verification Field DVF = 4744 Hex for the command and DVF = 34C7 Hex for the response.

The Length data field LEN = 0F Hex (15 decimal) for the response indicates that 15 bytes of data follow the LEN field.

Note: *Blank spaces in the above examples are added for clarity only. They do not actually appear in the command or in the response.*

READ I/O MODULE MAGNITUDE — COMMAND r

Description:

This command reads the analog channel (module) specified by data field CC. Data field TT specifies the type of data to be returned. See Remarks for data types. If the data type being requested is in counts (TT = 00, 01, 02, 03 or 04), then the return data field DDDD is 16-bits long (4 Hex characters in ASCII protocol). If the data type being requested is in *square root of counts* (TT = 20, 21, 22, 23 or 24), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol). If the data type being requested is in Engineering units (TT = 10, 11, 12, 13, 14, 30, 31, 32, 33 or 34), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol).

The channel specified by data field CC may be an analog input or an analog output I/O module. If the specified channel (module) is an analog *output* module, then only current or totalized values will be returned. If average, peak or lowest value is specified as the data type for an *output* module, then the current value will be returned instead.

Versions:

Analog

ASCII Command Format:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	r	CC	TT	DVF	CR

Where:
> = ASCII Character > (3E Hex)
r = ASCII Character r (72 Hex)
TT = Data Type. See Remarks
CR = Carriage Return (0D Hex)
AA = Address (00 to FF Hex)
CC = Channel Number (00 to 0F Hex)
DVF = Data Verification Field

ASCII Response Format

Success Response Message Frame

number of bytes →

1	4 or 8	2 or 4	1
A	DDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDD = Magnitude Data in Counts (4 chars.), Square Root of Counts (8 chars.) or Engineering units (8 chars.)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	72 hex	CC	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
72 Hex = Command r (72 Hex) CC = Channel Number (00 to 0F Hex)
TT = Data Type. See Remarks DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2 or 4	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 to 07 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Magnitude Data in Counts (2 bytes), Square Root of Counts (4 bytes) or Engineering units (4 bytes)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The specified analog I/O module is read and the data returned. If the square root is requested and the value read from the I/O module is negative, then the value returned in data field DDDD will be the square root of the absolute value.

Channels without a module installed or thermocouple modules that have an open thermocouple will return a value of 8000 Hex (-32768) as an indication of error. Attempts to read an Average before Command 'h' is issued will also return an error value of 8000 Hex.

If averaging is active, the peak and valley detection is derived from the average reading and is compared each time the average is calculated (every 100 milliseconds) to see if a new peak or valley has occurred.

The data type specified in data field TT must be one of the following types:

COUNTS

00 = Current Counts
01 = Average Counts
02 = Peak Counts
03 = Lowest Counts
04 = Totalized Counts
20 = Square Root of Current Counts
21 = Square Root of Average Counts
22 = Square Root of Peak Counts
23 = Square Root of Lowest Counts
24 = Square Root of Totalized Counts

ENGINEERING UNITS

10 = Current Engineering units
11 = Average Engineering units
12 = Peak Engineering units
13 = Lowest Engineering units
14 = Totalized Engineering units
30 = Square Root of Current Engineering units
31 = Square Root of Average Engineering units
32 = Square Root of Peak Engineering units
33 = Square Root of Lowest Engineering units
34 = Square Root of Totalized Engineering units

EXAMPLES:

ASCII Protocol:

Command	>4Cr0210ACcr
Response	A 0002 E9A0 F2 cr

Sends a Read I/O Module Magnitude command in ASCII protocol to the I/O unit at address 4C Hex. Data field CC = 02 Hex specifies that channel 02 is to be read. Data field TT = 10 Hex specifies that the data is to be in Current Engineering units. Thirty two (32)-bits of data will be returned. For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. Therefore, the data will be in increments of 1/65,536 of VDC. The return data field DDDD = 0002E9A0 (190,880 decimal). The value in Engineering units is 190,880 x 1/65,536 VDC = 2.91 VDC.

Data verification method is 8-bit checksum. The command checksum is AC Hex.

Binary Protocol:

Command	91 05 72 02 10 A6 5D
Response	07 00 00 05 35 70 02 07

Sends a Read I/O Module Magnitude command in Binary protocol to the I/O unit at address 91 Hex. Data field CC = 02 Hex specifies that channel 02 is to be read. Data field TT = 10 Hex specifies that the data is to be in Current Engineering units. Thirty two (32)-bits of data will be returned. For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module. Therefore, the data will be in increments of 1/65,536 of a VDC. The return data field DDDD = 00053570 (341,360 decimal). The value in Engineering units is 341,360 x 1/65,536 VDC = 5.209 VDC.

Data verification method is 16-bit CRC. The Data Verification Field DVF = A65D Hex for the command and DVF = 0207 Hex for the response.

The Length data field LEN = 07 for the response indicates that 7 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command reads the analog channels (modules) specified by data field MMMM. Data field TT specifies the type of data to be returned. See Remarks for data types. The number of data fields, DDDD, returned by this command will be equal to the number of 1's in MMMM. If the data type being requested is in counts (TT = 00, 01, 02, 03 or 04), then the return data field DDDD is 16-bits long (4 Hex characters in ASCII protocol). If the data type being requested is in *square root of counts* (TT = 20, 21, 22, 23 or 24), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol). If the data type being requested is in Engineering units (TT = 10, 11, 12, 13, 14, 30, 31, 32, 33 or 34), then the return data field DDDD is 32-bits long (8 Hex characters in ASCII protocol).

The channels specified by data field MMMM may be analog input or analog output I/O modules. If the specified channels (or channel) are analog *output* modules, then only current or totalized values will be returned. If average, peak or lowest value is specified as the data type for an *output* module, then the current value will be returned instead.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	R	MMMM	TT	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
R = ASCII Character R (52 Hex)
MMMM = Modules for Which Data is to be Returned.
TT = Data Type. See Remarks DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	0 to 128	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex)
DDDD = Magnitude Data in Counts (4 chars. for each channel specified), Square Root of Counts (8 chars. for each channel specified) or Engineering units (8 chars. for each channel specified). One data field DDDD will be returned for each bit that is set to a 1 in MMMM (the specified channels).
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)Description:

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	1	1 or 2
AA	LEN	52 hex	MMMM	TT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
52 Hex = Command R (52 Hex)
MMMM = Modules for Which Data is to be Returned.
TT = Data Type. See Remarks DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	0 to 64	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Magnitude Data in Counts (4 chars. for each channel specified), Square Root of Counts (8 chars. for each channel specified) or Engineering Units (8 chars. for each channel specified). One data field DDDD will be returned for each bit that is set to a 1 in MMMM (the specified channels).
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The specified analog I/O modules are read and the data returned. If a square root is requested and the value read from the I/O module is negative, then the value returned in data field DDDD will be the square root of the absolute value.

Channels without a module installed or thermocouple modules that have an open thermocouple will return a value of 8000 Hex (-32768) as an indication of error. Attempts to read an Average before Command 'h' is issued will also return an error value of 8000 Hex.

If averaging is active, the peak and valley detection is derived from the average reading and is compared each time the average is calculated (every 100 milliseconds) to see if a new peak or valley has occurred.

The data type specified in data field TT must be one of the following types:

COUNTS

00 = Current Counts
01 = Average Counts
02 = Peak Counts
03 = Lowest Counts
04 = Totalized Counts
20 = Square Root of Current Counts
21 = Square Root of Average Counts
22 = Square Root of Peak Counts
23 = Square Root of Lowest Counts
24 = Square Root of Totalized Counts

ENGINEERING UNITS

10 = Current Engineering units
11 = Average Engineering units
12 = Peak Engineering units
13 = Lowest Engineering units
14 = Totalized Engineering units
30 = Square Root of Current Engineering units
31 = Square Root of Average Engineering units
32 = Square Root of Peak Engineering units
33 = Square Root of Lowest Engineering units
34 = Square Root of Totalized Engineering units

The channels to be read are specified by setting bits to a '1' in data field MMMM. A '0' in the data field MMMM indicates that no action is to be taken for the corresponding channel. Each bit in the 16-bit data field MMMM corresponds to an analog I/O channel. The correspondence between the bits in the data field MMMM and the channel numbers is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 1 0 0	0 0 0 0	0 0 0 0	1 1 0 0
Hex Data	4	0	0	C

In the above example, channels 14, 03, and 02 are to be read. No action is taken for the other channels.

EXAMPLES:

ASCII Protocol:

Command	> 4C R 0004 00 ED cr
Response	A 0C8A 11 cr

Sends a Read I/O Module Magnitude command in ASCII protocol to the I/O unit at address 4C Hex. Data field MMMM = 0004 Hex specifies that channel 02 is to be read. Data field TT = 00 Hex specifies that the data is to be in Current Counts. Sixteen (16)-bits of data will be returned.

For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module.
The data will be in counts. Zero counts corresponds to 0 VDC. A count of 4095 corresponds to 5 VDC. The return data field DDDD = 0C8A (3210 decimal) counts. This represents 3210/4095 x 5.0 VDC = 3.92 VDC.

Data verification method is 8-bit checksum. The command checksum is ED Hex.

Binary Protocol:

Command	35 06 52 00 04 01 1D 5E
Response	05 00 10 AB B3 4C

Sends a Read I/O Module Magnitude command in Binary protocol to the I/O unit at address 35 Hex. Data field MMMM = 0004 Hex specifies that channel 02 is to be read. Data field TT = 01 Hex specifies that the data type is to be Average Counts. Sixteen (16)-bits of data will be returned.

For this example, the module plugged into channel 02 is an AD6 0 to 5 VDC analog input module.
The data will be in counts. Zero counts corresponds to 0 VDC. A count of 4095 corresponds to 5 VDC. The return data field DDDD = 10AB (4267 decimal) counts. This represents 4267/4095 x 5.0 VDC = 5.21 VDC. Input is over range.

Data verification method is 16-bit CRC. The Data Verification Field DVF = 1D5E Hex for the command and DVF = B34C Hex for the response.

The Length data field LEN = 05 Hex (15 decimal) for the response indicates that 5 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the output of the analog I/O module specified by data field CC to the value in Engineering units specified by data field DDDDDDDD. Data field DDDDDDDD is a 32-bit signed integer. units are in increments of 1/65536 of an Engineering unit. The most significant 16-bits represents the integer part and the least significant 16-bits represents the fractional part of the data. Analog output data is updated every 50 milliseconds.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	w	CC	DDDDDDDD	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
w = ASCII Character w (77 Hex) CC = Channel Number
DDDDDDDD = Analog output value in Engineering units. Units are in increments of 1/65536 of an Engineering unit.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	77 hex	CC	DDDDDDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
77 = Command w (77 Hex) CC = Channel Number
DDDDDDDD = Analog output value in Engineering units. Units are in increments of 1/65536 of an Engineering unit.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by data field CC must be configured as an analog output before issuing this command. Command 'G' or command 'a' may be used.

Data field DDDDDDDD is a 32-bit signed integer. Units are in increments of 1/65536 of an Engineering unit. The most significant 16-bits represents the integer part and the least significant 16-bits represents the fractional part of the data. To convert a value in Engineering units to a signed integer DDDDDDDD, first multiply the Engineering units by 65,536, drop the fractional part and then convert to Hex. Some examples are:

2.75 x 65536 = 180,224 decimal = 2C000 Hex
472.893 x 65536 = 30,991,515 decimal = 1D8E496 Hex
-22.915 x 65536 = -1,501,757 decimal = FFE915C3 Hex

Notice that negative numbers are represented in two's complement format.

Analog output data is updated every 50 milliseconds.

EXAMPLES:

ASCII Protocol:

Command	>4C w 03 00020000 D3 cr
Response	A41cr

Sends a Set DAC Module Magnitude, Engineering units command in ASCII protocol to the I/O unit at address 4C. This command specifies that channel CC = 03 Hex is to be set to DDDDDDDD = 00020000 Hex (2.00 decimal) Engineering units.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. The data will be in Engineering units of DC Volts in increments of 1/65536 of a VDC. We want to set the output to 2.00 VDC. Multiply by 65536 and convert to Hex. 2.00 x 65536 = 131,072 decimal = 20000 Hex = DDDDDDDD.

Data verification method is 8-bit checksum. Checksum is D3 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	0F 08 77 03 00 02 00 00 C7 26
Response	03 00 F0 00

Sends a Set DAC Module Magnitude, Engineering units command in Binary protocol to the I/O unit at address 4C. This command specifies that channel CC = 03 Hex is to be set to an analog output of DDDDDDDD = 00020000 Hex (2.00 decimal) Engineering units.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. The data will be in Engineering units of DC Volts in increments of 1/65536 of a VDC. We want to set the output to 2.00 VDC. Multiply by 65536 and convert to Hex. 2.00 x 65536 = 131,072 decimal = 20000 Hex = DDDDDDDD.

Data verification method is 16-bit CRC. Data Verification field is C726 Hex for the command.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets the analog output channels (modules) specified by data field MMMM to the value in Engineering units specified by data field DDDDDDDD. The number of data fields, DDDDDDDD, required by this command will be equal to the number of 1's in MMMM. Data field DDDDDDDD is a 32-bit signed integer (8 Hex characters for ASCII and 4 data bytes for Binary). Units are in increments of 1/65536 of an Engineering unit. The most significant 16-bits represents the integer part and the least significant 16-bits represents the fractional part of the data.

Analog output data is updated every 50 milliseconds.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	0 to 128	2 or 4	1
>	AA	W	MMMM	DDDDDDDD	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
W = ASCII Character W (57 Hex)
MMMM = Analog Output Modules to be Set to the Value Specified by Data Field DDDDDDDD in Engineering units.
DDDDDDDD = Analog Output Value in Engineering units. Units are in Increments of 1/65536 of an Engineering unit.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	0 to 64	1 or 2
AA	LEN	57 hex	MMMM	DDDDDDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field
57 = Command W (57 Hex)
MMMM = Analog Output Modules to be Set to the Value Specified by Data Field DDDDDDDD in Engineering units.
DDDDDDDD = Analog Output Value in Engineering units. Units are in Increments of 1/65536 of an Engineering unit.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The channels specified by data field MMMM must be configured as analog output channels before issuing this command. Command 'G' or command 'a' may be used.

Data field DDDDDDDD is a 32-bit signed integer. Units are in increments of 1/65536 of an Engineering unit. The most significant 16-bits represents the integer part and the least significant 16-bits represents the fractional part of the data. To convert a value in Engineering units to a signed integer DDDDDDDD, first multiply the Engineering units by 65,536, drop the fractional part and then convert to Hex. Some examples are:

2.75 x 65536 = 180,224 decimal = 2C000 Hex
472.893 x 65536 = 30,991,515 decimal = 1D8E496 Hex
-22.915 x 65536 = -1,501,757 decimal = FFE915C3 Hex

Notice that negative numbers are represented in two's complement format.

There is must be a 32-bit data field DDDDDDDD for each bit in MMMM that is set to a '1'. The data fields DDDDDDDD are sent in the order from the highest numbered channel to the lowest numbered channel.

Each bit in the 16-bit data field MMMM corresponds to an analog I/O channel. The correspondence between the bits in the data field MMMM and the channel numbers is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
Hex Data	8	0	0	8

In the above example, channels 15, and 03 are to be set to the Engineering units specified by data field DDDDDDDD. No action is taken for the other channels. The data field MMMM = 8008 Hex.

Analog output data is updated every 50 milliseconds.

EXAMPLES:

ASCII Protocol:

Command	>4C W 8008 00038000 00004000 AD cr
Response	A41cr

Sends a Set DAC Module Magnitude, Engineering units - Group command in ASCII protocol to the I/O unit at address 4C Hex. The data field MMMM = 8008 Hex, specifies channels 15 and 03. Data for channel 15 is DDDDDDDD = 00038000 Hex (3.50 decimal) Engineering units and data for channel 03 is DDDDDDDD = 00004000 Hex (0.25 decimal) Engineering units. Data for channel 15 will be sent first.

For this example, the module plugged into channel 15 is a DA8 0 to 20 mA analog output module. The data will be in Engineering units of milliamperes in increments of 1/65536 of a milliampere. We want to set the output to 3.50 mA. Multiply by 65536 and convert to Hex. 3.50 x 65536 = 229,376 decimal = 38000 Hex = DDDDDDDD.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. The data will be in Engineering units of DC Volts in increments of 1/65536 of a VDC. We want to set the output to 0.25 VDC. Multiply by 65536 and convert to Hex. 0.25 x 65536 = 16,384 decimal = 4000 Hex = DDDDDDDD.

Data verification method is 8-bit checksum. Checksum is AD Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	0F 0D 57 80 08 00 03 80 00 00 00 40 00 81 9F
Response	03 00 F0 00

Sends a Set DAC Module Magnitude, Engineering units - Group command in Binary protocol to the I/O unit at address 0F Hex. The data field MMMM = 8008 Hex, specifies channels 15 and 03. Data for channel 15 is DDDDDDDD = 00038000 Hex (3.50 decimal) Engineering units and data for channel 03 is DDDDDDDD = 00004000 Hex (0.25 decimal) Engineering units. Data for channel 15 will be sent first.

For this example, the module plugged into channel 15 is a DA8 0 to 20 mA analog output module. The data will be in Engineering units of milliamperes in increments of 1/65536 of a milliampere. We want to set the output to 3.50 mA. Multiply by 65536 and convert to Hex. 3.50 x 65536 = 229,376 decimal = 38000 Hex = DDDDDDDD.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. The data will be in Engineering units of DC Volts in increments of 1/65536 of a VDC. We want to set the output to 0.25 VDC. Multiply by 65536 and convert to Hex. 0.25 x 65536 = 16,384 decimal = 4000 Hex = DDDDDDDD.

Data verification method is 16-bit CRC. Data Verification field is 819F Hex for the command and F000 for the response.

The Length data field LEN = 0D Hex (13 decimal) for the command indicates that 13 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET DAC MODULE MAGNITUDE, COUNTS — COMMAND x

DESCRIPTION:

This command is used to set the output of the analog I/O module specified by data field CC to the value in Counts specified by data field DDDD. Data field DDDD is a 16-bit signed integer in the range of 0 to 4095. Values less than 0 will be set to 0. Values greater than 4095 will be clipped to 4095. Analog output data is updated every 50 milliseconds.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	2 or 4	1
>	AA	x	CC	DDDD	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
x = ASCII Character x (78 Hex) CC = Channel Number (00 to 0F)
DDDD = Analog Output Value in Counts (0 to 4095).
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	2	1 or 2
AA	LEN	78 hex	CC	DDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
78 = Command x (78 Hex) CC = Channel Number (00 to 0F Hex)
DDDD = Analog Output Value in Counts (0 to 4095).
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channel specified by data field CC must be configured as an analog output before issuing this command. Command 'G' or command 'a' may be used.

Data field DDDD is a 16-bit signed integer. Values less than 0 will be set to 0. Values greater than 4095 will be set to 4095.

Analog output data is updated every 50 milliseconds.

EXAMPLES:

ASCII Protocol:

Command	>0F x 03 0200 13 cr
Response	A41cr

Sends a Set DAC Module Magnitude, Counts command in ASCII protocol to the I/O unit at address 0F Hex. This command specifies that channel CC = 03 Hex is to be set to DDDD = 0200 Hex (512 decimal) Counts.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. Zero counts represents 0 VDC and 4095 counts represents 5 VDC. The data field DDDD = 0200 Hex = 512 decimal represents 0.625 VDC. (512/4095) x 5 VDC = 0.625 VDC.

Data verification method is 8-bit checksum. Checksum is 13 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	0F 06 78 03 02 00 3F 61
Response	03 00 F0 00

Sends a Set DAC Module Magnitude, Counts command in Binary protocol to the I/O unit at address 0F Hex. This command specifies that channel CC = 03 Hex is to be set to DDDD = 0200 Hex (512 decimal) Counts.

For this example, the module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. Zero counts represents 0 VDC and 4095 counts represents 5 VDC. The data field DDDD = 0200 Hex = 512 decimal represents 0.625 VDC. (512/4095) x 5 VDC = 0.625 VDC.

Data verification method is 16-bit CRC. Data Verification field is 3F61 Hex for the command.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets the analog output channels (modules) specified by data field MMMM to the value in Counts specified by data field DDDD. The number of data fields, DDDD, required by this command will be equal to the number of 1's in MMMM. Data field DDDD is a 16-bits signed integer (4 Hex characters for ASCII and 2 data bytes for Binary). Analog output data is updated every 50 milliseconds.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	4	0 to 64	2 or 4	1
>	AA	X	MMMM	DDDD	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
X = ASCII Character X (58 Hex)
MMMM = Analog Output Modules to be Set to the Count Value Specified by Data Field DDDD.
DDDD = Analog Output Value in Counts.DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	2	0 to 32	1 or 2
AA	LEN	58 hex	MMMM	DDDD	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field
58 = Command X (58 Hex)
MMMM = Analog Output Modules to be Set to the Count Value Specified by Data Field DDDD.
DDDD = Analog Output Value in Counts.DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The channels specified by data field MMMM must be configured as analog output channels before issuing this command. Command 'G' or command 'a' may be used.

Data field DDDD is a 16-bit signed integer. Units are in increments of counts. Negative numbers are represented in two's complement format.

There is must be a 16-bit data field DDDD for each bit in MMMM that is set to a '1'. The data fields DDDD are sent in the order from the highest numbered channel to the lowest numbered channel.

Each bit in the 16-bit data field MMMM corresponds to an analog I/O channel. The correspondence between the bits in the data field MMMM and the channel numbers is as follows:

Data Field	M	M	M	M
Channel No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	1 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
Hex Data	8	0	0	8

In the above example, channels 15, and 03 are to be set to the Engineering units specified by data field DDDDDDDD. No action is taken for the other channels. The data field MMMM = 8008 Hex.

Analog output data is updated every 50 milliseconds.

EXAMPLES:

ASCII Protocol:

Command	>0F X 8008 0400 0800 2A cr
Response	A41cr

Sends a Set DAC Module Magnitude, Counts - Group command in ASCII protocol to the I/O unit at address 0F Hex. The data field MMMM = 8008 Hex, specifies channels 15 and 03. Data for channel 15 is DDDD = 0400 Hex (1024 decimal) Counts and data for channel 03 is DDDD = 0800 Hex (2048 decimal) Counts. Data for channel 15 will be sent first.

For this example, the module plugged into channel 15 is a DA8 0 to 20 mA analog output module. The data will be in Counts DDDD = 0400 Hex (1024 decimal) corresponds to (1024/ 4095) x 20 mA = 5.0 milliamperes. The module plugged into channel 03 is a DA4 0 to 5 VDC analog output module. The data will be in Counts DDDD = 0800 Hex (2048 decimal) corresponds to (2048/ 4095) x 20 mA = 10.0 milliamperes.

Data verification method is 8-bit checksum. Checksum is 2A Hex for the command and 41 Hex for the response.

Analog Event/Reaction Commands

CLEAR EVENT/REACTION TABLE — COMMAND _

DESCRIPTION:

This command is used to clear the entire Event/Reaction Table. All event latches and interrupts are also cleared.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	_	DVF	CR

Where:
> = ASCII Character > (3E Hex)
_ = ASCII Character _ (5F Hex)
CR = Carriage Return (0D Hex)
AA = Address
DVF = Data Verification Field

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
CR = Carriage Return (0D Hex)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1 or 2
AA	LEN	5F hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
5F Hex = Command _ (5F Hex) DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The first 64 entries in the Event/Reaction table are restored from EEPROM upon power up or execution of a 'Reset' Command.

EXAMPLES:

ASCII Protocol:

Command	>4E_D8cr
Response	A41cr

Sends a Clear Event/Reaction Table Entry command in ASCII protocol to the I/O Unit at address 4E Hex. Clears the Event/Reaction Table for the addressed unit. Event latches and interrupts are also cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	4E 03 5F DF 20
Response	03 00 F0 00

Sends a Clear Event/Reaction Table Entry command in Binary protocol to the I/O Unit at address 4E Hex. Clears the Event/Reaction Table for the addressed unit. Event latches and interrupts are also cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

CLEAR EVENT TABLE ENTRY — COMMAND \

DESCRIPTION:

This command is used to clear a single entry in the Event/Reaction table. Data field EE specifies the entry to be cleared.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	\	EE	DVF	CR

Where:

>	=	ASCII Character > (3E Hex)	AA	=	Address
\	=	ASCII Character \ (5C Hex)	EE	=	Event Table Entry Number (0 to FF Hex)
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:

A	=	ASCII Character A (41 Hex)	DVF	=	Data Verification Field
CR	=	Carriage Return (0D Hex)			

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:

N	=	ASCII Character N (4E Hex)	EC	=	Error Code
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	5C hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
5C = Command \ (5C Hex) EE = Event Table Entry Number (0 to FF Hex)
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The command data field EE specifies the Event/Reaction table entry to be cleared. The Event/ Reaction table can have as many as 256 entries. The first 64 entries will restored from EEPROM upon power up or reset.

EXAMPLES:

ASCII Protocol:

Command	>38\1D3Ccr
Response	A41cr

Sends a Clear Event Table Entry command in ASCII protocol to the I/O Unit at address 38 Hex. Specifies that entry number 1D Hex (29 decimal) is to be cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	38 04 5C 1D A8 B5
Response	03 00 F0 00

Sends a Clear Event Table Entry command in Binary protocol to the I/O Unit at address 38 Hex. Specifies that entry number 1D Hex (29 decimal) is to be cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to clear the Event/Reaction interrupt output. Event latches are not affected.

VERSIONS:

Analog Version 1.11 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2 or 4	1
>	AA	zB	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
zB = ASCII Character zB (7A42 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1 or 2
AA	LEN	7A42 hex	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (02 or 03 Hex)
7A42 Hex = Command zB (7A42 Hex) DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

This command will only clear the interrupt output from the I/O unit. It will not clear any event latch or return data.

DESCRIPTION:

This command is used to enable and/or disable a selected group of 16 event table entries in the event/ reaction table. Data field GG specifies which group of entries is to be enabled (or disabled). The most significant nibble (the upper 4-bits) of GG is used to determine which group of 16 entries will be enabled (or disabled). The lower nibble is ignored. See Remarks for examples. Data field MMMM is a bitmask representing entries to be enabled. Data field NNNN is a bitmask representing entries to be disabled. Setting the same bit in both enable and disable bitmasks will result in an error being returned and the command is not executed. An entry which is not enabled is not checked for it's event occurrence by the CPU.

VERSIONS:

Analog - FIrmware 1.03 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	4	2 or 4	1
>	AA	{	GG	MMMM	NNNN	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address (00 to FF Hex)
- { = ASCII Character { (7B Hex)
- GG = Event Entry Group Number
- MMMM = Bitmask Representing Entries to be Enabled
- NNNN = Bitmask Representing Entries to be Disabled
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1	1 or 2
AA	LEN	7B hex	GG	MM	NN	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (05 or 06 Hex)
7B Hex = Command { (7B Hex) GG = Event Entry Group Number
MM = Bitmask Representing Entries to be Enabled
NN = Bitmask Representing Entries to be Disabled
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Once the Event/Reaction Table entries are enabled, the CPU will begin comparing for an event match. An entry which is not enabled is not checked for it's event occurrence by the CPU, and the event latch will not be set.

If GG (Event Entry Group Number) is set to FF Hex, then ALL event table entries are specified and bitmask MMMM is checked for enable/disable status. If MMMM is equal to zero (0), then all entries are disabled. If MMMM is not equal to zero (0), then all entries are enabled. In either case, bitmask NNNN *must* be sent, but is ignored.

Data field GG specifies which group of entries is to be enabled (or disabled). If GG is not equal to FF Hex, then the most significant nibble (the upper 4-bits) of GG is used to determine which group of 16 entries will be enabled (or disabled). The lower nibble is ignored. For example, if GG = 23 Hex, MMMM = 0006 Hex and NNNN = C300 Hex, then the 2 Event/Reaction Table entries 21 Hex and 22 Hex will be enabled and the 4 Event/Reaction Table entries 2F Hex, 2E Hex, 29 Hex and 28 Hex will be disabled. If GG = FF Hex and if MMMM = 0001 Hex, then all entries will be enabled.

The correspondence between the bits in the data field MMMM and the entry offset is as follows:

Data Field	M	M	M	M
Entry Offset	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Offset Hex	0F 0E 0D 0C	0B 0A 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for entry offset numbers 02 Hex and 01 Hex. To get the actual Event/Reaction Table entry number rather than just the entry offset, you must add the group number given by the upper nibble of GG to the entry offset value in Hex. For example, if the group number is 20 Hex, then the actual Event/Reaction Table entry numbers for this example would be 20 + 01 = 21 Hex and 20 + 02 = 22 Hex.

The correspondence between the bits in the data field NNNN and the entry offset is as follows:

Data Field	N	N	N	N
Entry Offset	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Offset Hex	0F 0E 0D 0C	0B 0A 09 08	07 06 05 04	03 02 01 00
Example	1 1 0 0	0 0 1 1	0 0 0 0	0 0 0 0
Hex Data	C	3	0	0

In the above example, data bits are set for entry offset numbers 0F Hex, 0E Hex, 09 Hex and 08 Hex. To get the actual Event/Reaction Table entry number rather than just the entry offset, you must add the group number given by the upper nibble of GG to the entry offset value in Hex. For example, if the group number is 20 Hex, then the actual Event/Reaction Table entry numbers for this example would be 2F Hex, 2E Hex, 29 Hex and 28 Hex.

EXAMPLES:

ASCII Protocol:

Command	> 8F { 23 0006 C300 FA cr
Response	A41cr

Sends an Enable/Disable Event Entry Group command in ASCII protocol to the I/O Unit at address 8F Hex. Data field GG = 23 Hex, specifies that the 16 Event/Reaction Table entries 20 Hex to 2F Hex will be selected. Data field MMMM = 0006 specifies that the Event/Reaction table entries 22 Hex and 21 Hex shall be enabled. Data field NNNN = C300 specifies that the Event/Reaction table entries 2F Hex, 2E Hex, 29 Hex and 28 Hex shall be disabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	8F 08 7B AF 0001 1000 B3 43
Response	03 00 F0 00

Sends an Enable/Disable Event Entry Group command in Binary protocol to the I/O Unit at address 8F Hex. Data field GG = AF Hex, specifies that the 16 Event/Reaction Table entries A0 Hex to AF Hex will be selected. Data field MMMM = 0001 specifies that the Event/Reaction table entry A0 Hex shall be enabled. Data field NNNN = 1000 specifies that the Event/Reaction table entry AF Hex shall be disabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to enable or to disable an event table entry specified by data field EE. A zero value in data field SS specifies that the entry is to be disabled. A non-zero value specifies that the entry is to be enabled.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	N	EE	SS	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
N = ASCII Character N (4E Hex) EE = Event Entry Number (0 to FF Hex)
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	4E hex	EE	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
4E Hex = Command N (4E Hex) EE = Event Entry Number (0 to FF Hex)
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

Before issuing this command, you must first issue a K, L or y command to place an entry in the Event/Reaction table, otherwise an error is returned.

Once enabled, the CPU will begin comparing for an event match. If disabled, the entry is skipped and no action is taken upon an event match, and the event latch will not be set.

This command can be used as a reaction command for an event. This will allow an Event/Reaction to be suspended until after a different event has occurred.

EXAMPLES:

ASCII Protocol:

Command	> 64 N 16 01 80 cr
Response	A41cr

Sends an Enable/Disable Event Table Entry command in ASCII protocol to the I/O Unit at address 64 Hex. Event/reaction table entry number is 16 Hex (22 decimal). The enable status field SS = 01 specifies that the table entry is to be enabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 05 4E 16 01 B3 DE
Response	03 00 F0 00

Sends an Enable/Disable Event Table Entry command in Binary protocol to the I/O Unit at address 64 Hex. Event/Reaction table entry number is 16 Hex. The enable status field SS = 01 specifies that the table entry is to be enabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND CLEAR EVENT LATCHES — COMMAND Q

DESCRIPTION:

This command is used to read and then clear the event latches. If command data field EE = FF Hex, then all 256 latch bits (one for each latch) are returned, otherwise only 16-bits are returned. The most significant 4-bits of data field EE determine which group of 16 latch bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	Q	EE	DVF	CR

Where:

> = ASCII Character > (3E Hex)	AA = Address
Q = ASCII Character Q (51 Hex)	EE = Latch Group (00 to F0 Hex, or FF Hex)
DVF = Data Verification Field	CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where:

A = ASCII Character A (41 Hex)	DDDD = Latch Bits (16 Bits or 256 Bits)
DVF = Data Verification Field	CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:

N = ASCII Character N (4E Hex)	EC = Error Code
DVF = Data Verification Field	CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	51 hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
51 = Command Q (51 Hex) EE = Latch Group (00 to F0 Hex or FF Hex)
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Latch Bits (16 or 256) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

There are 256 event latches, one for each event/reaction table entry. The 256 latches are grouped into 16 groups of 16 latches each. Each bit of the return data field represents one latch. If the latch is set the data bit will be a '1'. If the latch is cleared, the data bit will be a '0'. Latches are set each time an event changes from a non-matching to a matching condition. If command data field EE = FF then all 256 latch bits are returned otherwise only one group of latch bits (16-bits) are returned. The upper nibble (most significant 4-bits) of command data field EE determines which one of the 16 groups of latch bits will be returned. The lower nibble (least significant 4-bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the relative latch number is as follows:

Data Field	D	D	D	D
Latch No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative latch numbers 02 and 01. To get the actual latch number rather than just relative latch number, you must add an offset value to the relative latch number. First multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the relative latch number. For example assume that the group number is 5 and the relative latch number is 2. The actual latch number is (5x16)+2=82.

The latches are cleared after they are read. If EE = FF Hex all 256 latches are cleared otherwise only the 16 latches for the specified group are cleared.

EXAMPLES:

ASCII Protocol:

Command	>64Q001Bcr
Response	A000607cr

Sends a Read and Clear Event Latches command in ASCII protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16-bits of latch data, one-bit for each of the first 16 latches. The return data field of DDDD = 0006 indicates that latches 02 and 01 are set. All others are cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 51 00 61 63
Response	05 00 00 03 CD 40

Sends a Read and Clear Event Latches command in Binary protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16-bits of latch data, one-bit for each of the first 16 latches. The return data field of DDDD = 0003 indicates that latches 01 and 00 are set. All others are cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to read data which has been previously read and held by the execution of analog reaction command 08. The returned data will be a 32-bit number. See Remarks.

VERSIONS:

Analog - Firmware 1.03 and later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	|	EE	DVF	CR

Where:
> = ASCII Character > (3E Hex)
| = ASCII Character | (7C Hex)
DVF = Data Verification Field
AA = Address
EE = Event Entry Number
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	8	2 or 4	1
A	DDDDDDDD	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DDDDDDDD = 32 Bit Data Field
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	7C hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
7C Hex = Command | (7C Hex) EE = Event Entry Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	4	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (06 or 07 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = 32 Bit Data Field DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Data DDDDDDDD will only be valid if the analog reaction command 08 has been *executed*. See Analog Event/Reaction Command M. Data field DDDDDDDD is 32-bits. The most significant (upper) 16-bits represents the whole number part and the least significant (lower) 16-bits represent the fractional part in increments of 1/65536 of a unit. If the data type is counts, only the whole number part is used. The least significant (lower) 16-bits will always be zeros. If the data type is Engineering units, then all 32-bits are used.

EXAMPLES:

ASCII Protocol:

Command	>64|0248cr
Response	A00454000CEcr

Sends a Read Event Data Holding Buffer command in ASCII protocol to the I/O Unit at address 64 Hex. Data field EE = 02 specifies that the Event/Reaction table entry number is 02 Hex. Data verification method is 8-bit checksum. The 8 checksum is 48 for the command. The response of A indicates a successful command. The returned data DDDDDDDD = 00 45 40 00 Hex (69.25 decimal) is in Engineering units. The checksum for the response is CE Hex.

Binary Protocol:

Command	60 04 7C 00 01 7F
Response	07 00 0A AA 00 00 4F 22

Sends a Read Event Data Holding Buffer command in Binary protocol to the I/O Unit at address 60 Hex. Data field EE = 00 specifies that the Event/Reaction table entry number is 00 Hex. Data verification method is 16-bit CRC. The 16-bit CRC for the command is 01 7F Hex. The response of 07 indicates that 07 Hex bytes are to follow. A error code of EC=00 indicates a successful command. The returned data DDDD = 0A AA 00 00 Hex (2730 decimal) is in counts. The last 16-bits are not used. The 16-bit CRC for the response is 4F22 Hex.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command returns 16 status bits or 256 status bits. Each status bit represents one event table entry. The status bit will be a '1' for event table entries which are enabled and a '0' for entries which are disabled. If data field EE = FF Hex, then 256 status bits are returned, otherwise only 16 status bits are returned. The most significant 4-bits of data field EE determine which group of 16 event table entry status bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	v	EE	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
v = ASCII Character v (76 Hex) EE = Event Table Entry Group
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = Status Bits (16 Bits or 256 Bits)
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	76 hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
76 = Command v (76 Hex) EE = Event Table Entry Group
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length FieldEC = Error Code = 00 Hex (Zero)
DDDD = Status Bits (16 or 256) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

There are 256 event table entries. They can be grouped into 16 groups of 16 event table entries each. Each status bit in the return data field represents one event table entry. If the event table entry is enabled the data bit will be a '1'. If the event table entry is disabled, the data bit will be a '0'. If command data field EE = FF then all 256 status bits are returned otherwise only one group of status bits (16-bits) are returned. The upper nibble (most significant 4-bits) of command data field EE determines which one of the 16 groups of event table entries status bits will be returned. The lower nibble (least significant 4-bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the *relative event table entry number* is as follows:

Data Field	D	D	D	D
Event No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative event table entry numbers 02 and 01. To get the actual event table entry numbers rather than just relative event table entry numbers, you must add an offset value to the relative event table entry number. First multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the relative event table entry number. For example assume that the group number is 5 and the relative event table entry number is 2. The actual event table entry number is (5x16)+2=82.

Examples:

ASCII Protocol:

Command	>64v0040cr
Response	A000203cr

Sends a Read Event Entry Enable/Disable Status command in ASCII protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the status bits for the first group of 16 event table entries are to be returned. The return data field of DDDD = 0002 indicates that event table entry 01 is enabled. All others in the group are disabled. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 76 00 91 78
Response	05 00 00 03 CD 40

Sends a Read Event Entry Enable/Disable Status command in Binary protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the status bits for the first group of 16 event table entries are to be returned. The return data field of DDDD = 0003 indicates that event table entries 01 and 00 are enabled. All others in the group are disabled. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ EVENT LATCHES — COMMAND P

DESCRIPTION:

This command is used to read the event latches. If command data field EE = FF Hex, then all 256 latch bits (one for each latch) are returned, otherwise only 16-bits are returned. The most significant 4-bits of data field EE determine which group of 16 latch bits will be returned. The least significant bits of data field EE are ignored (unless EE = FF Hex).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	P	EE	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- P = ASCII Character P (50 Hex)
- EE = Latch Group (00 to F0 Hex, or FF Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 64	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDD = Latch Bits (16 Bits or 256 Bits)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	50	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
50 = Command P (50 Hex) EE = Latch Group (00 to F0 Hex or FF Hex)
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2 or 32	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = Latch Bits (16 or 256) DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

There are 256 event latches, one for each Event/Reaction table entry. The 256 latches are grouped into 16 groups of 16 latches each. Each bit of the return data field represents one latch. If the latch is set the data bit will be a '1'. If the latch is cleared, the data bit will be a '0'. Latches are set each time an event changes from a non-matching to a matching condition. If command data field EE = FF then all 256 latch bits are returned otherwise only one group of latch bits (16-bits) are returned. The upper nibble (most significant 4-bits) of command data field EE determines which one of the 16 groups of latch bits will be returned. The lower nibble (least significant 4-bits) of command data field EE is ignored unless EE = FF Hex. The correspondence between the bits in the return data field DDDD and the relative latch number is as follows:

Data Field	D	D	D	D
Latch No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0
Hex Data	0	0	0	6

In the above example, data bits are set for relative latch numbers 02 and 01. To get the actual latch number rather than just relative latch number, you must add an offset value to the relative latch number. First multiply the group number (the upper nibble of EE) by 16 (to get the offset) then add that to the relative latch number. For example assume that the group number is 5 and the relative latch number is 2. The actual latch number is (5x16)+2=82.

EXAMPLES:

ASCII Protocol:

Command	>64P001Acr
Response	A000607cr

Sends a Read Event Latches command in ASCII protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16-bits of latch data, one-bit for each of the first 16 latches. The return data field of DDDD = 0006 indicates that latches 02 and 01 are set. All others are cleared. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 04 50 00 F1 62
Response	05 00 00 03 CD 40

Sends a Read and Clear Event Latches command in Binary protocol to the I/O Unit at address 64 Hex. Data field EE = 00 Hex specifies that the first group of 16 latches are to be read and then reset. Returns 16-bits of latch data, one-bit for each of the first 16 latches. The return data field of DDDD = 0003 indicates that latches 01 and 00 are set. All others are cleared. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

READ AND OPTIONALLY CLEAR EVENT LATCH — COMMAND zA

DESCRIPTION:

This command returns event latch data and the current event entry enable and interrupt enable status for the entry specified by data field EE. Data field FF indicates whether or not to clear the latch.

VERSIONS:

Analog Version 1.11 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	2	2	2	2 or 4	1
>	AA	zA	EE	FF	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- zA = ASCII Characters zA (7A41 Hex)
- EE = Entry Number
- FF = Clear Flag
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2 or 4	1
A	DD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DD = Event Latch Data and Status
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	7A41 hex	EE	FF	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field
7A41 Hex = Command zA EE = Entry Number (00 to FF Hex)
FF = Clear Flag DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DD	DVF

Where: LEN = Length Field (06 or 07 Hex) EC = Error Code = 00 Hex (Zero)
DD = Event Latch Data and Status DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

Data field FF in the command specifies whether the event latch is to be cleared. If FF is not equal to 00, the event latch will be cleared after it is read. If FF is equal to 00, the event latch will not be cleared.

The data byte returned by this command (DD) contains only 3-bits that are of any significance to the user.

Bit 0: if 1, the event latch is set; the event has occurred.
if 0, the event latch is cleared; the event has not occurred since the last time this latch was read and cleared.

Bit 6: if 1, the event interrupt is enabled.
if 0, the event interrupt is disabled.

Bit 7: if 1, the event entry is enabled and will be scanned.
if 0, the event entry is disabled and will not be scanned.

READ EVENT TABLE ENTRY — COMMAND O

DESCRIPTION:

This command returns the Event/Reaction table entry for the entry number specified by data field EE. Fourteen (14) bytes of data are returned for the Binary protocol and 28 Hex bytes for the ASCII protocol.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	O	EE	DVF	CR

Where:

>	= ASCII Character > (3E Hex)	AA	= Address
O	= ASCII Character O (4F Hex)	EE	= Event Table Entry Number
DVF	= Data Verification Field	CR	= Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2	2	2	2	4	8	8	2 or 4	1
A	CB	CC	RC	RN	SSSS	NNNNNNNN	ZZZZZZZZ	DVF	CR

Where:

- A = ASCII Character A (41 Hex)
- CB = Control Byte. See Remarks
- CC = Compare Channel Number (Most significant 4-bits) and Compare Data Type (Least significant 4-bits). See Remarks for code.
- RC = Reaction Command Number
- RN = Reaction Channel Number
- SSSS = Reaction DAC Output Value or Ramp Endpoint in Counts (See Remarks)
- NNNNNNNN = Compare Data (Setpoint) in Engineering Units
- ZZZZZZZZ = Reaction Ramp Slope (See Remarks)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	4F hex	EE	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (03 or 04 Hex)
4F = Command O (4F Hex) EE = Event Table Entry Number
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1	1	1	1	1	4	4	1 or 2
LEN	EC=00	CB	CC	RC	RN	SSSS	NNNNNNNN	ZZZZZZZZ	DVF

Where: LEN = Length Field (0F or 10 Hex)
EC = Error Code = 00 Hex (Zero)
CB = Control Byte (See Remarks)
CC = Compare Channel Number (Most significant 4-bits) and Compare Data Type (Least significant 4-bits). See Remarks for code.
RC = Reaction Command Number
RN = Reaction Channel Number
SSSS = Reaction DAC Output Value or Ramp Endpoint in Counts (See Remarks)
NNNNNNNN = Compare Data (Setpoint) in Engineering Units
ZZZZZZZZ = Reaction Ramp Slope (See Remarks)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The Control Byte contains flag bits which are defined as follows:

Bit 7: Event Scan Enable Flag. This bit is set if the event is enabled for scanning.
Bit 6: Interrupt Enable Flag. This bit is set if the event interrupt is enabled.
Bit 5: Last Entry Flag. This bit is set if this event is the last valid entry.
Bit 4: Compare >= or Compare <= flag.
If bit this is set (bit = 1), then the analog input is compared for >= the setpoint.
If bit this is cleared (bit = 0), then the analog input is compared for <= the setpoint.
Bit 3: Communications Link Watchdog Event. If this bit is set, this event is monitoring the communications link watchdog flags for a timeout condition. Bit 4 is ignored.
Bit 2: Valid Entry Flag. If this bit is clear, this entry is considered a null entry and is not being scanned. All other data is therefore meaningless.
Bit 1: Match Latch. This bit is set when an event condition is matched and cleared when that condition no longer exists.
Bit 0: Event Latch. This bit is set by an event condition match and can only be cleared by the host.

For return data in the Success Response Message Frame, the following is true:

The Compare Channel CC data field is valid only for events which must monitor data of a certain channel. The upper 4-bits (nibble) is the compare channel and the lower four-bits (least significant nibble) is the data type code defined as follows:

Code	Description Data Type (See Commands K and L)
00 Hex	Current Counts
01 Hex	Current Eng. Units
02 Hex	Totalized Counts
03 Hex	Totalized Eng. Units
04 Hex	Average Counts
05 Hex	Average Eng. Units
06 Hex	Peak Counts
07 Hex	Lowest Counts
08 Hex	Peak Eng. Units
09 Hex	Lowest Eng. Units

The Reaction Command Number RC is the reaction command which will be executed when this event occurs. See Command 'M'.

The Reaction Channel Number RN is only valid for reaction commands which require a channel or event number.

The Compare Value field NNNNNNNN will contain the compare data (setpoint) for this event. The most significant byte is first and the least significant byte is last. This field is meaningless for events which monitor the communications watchdog timeout condition.

The Reaction DAC Output Value or Ramp Endpoint data field SSSS is valid only for those reaction commands which require data (Reaction commands 01, 02 or 03).

Data Field SSSS will contain the Reaction DAC Output Value in counts or the Ramp Slope in counts. If the original data was specified in Engineering Units, the data is converted to counts. See Reaction commands 01, 02 and 03.

The Reaction Ramp Slope ZZZZZZZZ is the *incremental* value that is added to the DAC output every 50 mS. It is a pre-computed value used by the CPU upon execution of a reaction command requiring a ramp function. It is only valid for Reaction Command 03. To calculate this value take the slope = S in Engineering Units = EU per second (EU/sec.) and divide by 20 to get Engineering Units per 50 milliseconds (EU/50 mS). Next calculate the counts = C per Engineering Unit (C/EU). To do this take the total range in counts = 4095 and divide it by the total range in Engineering Units = EU range. To get counts per 50 mS, just multiply the Engineering Units per 50 milliseconds = EU/50 mS by the counts per Engineering Unit. The total range in Engineering Units is the Engineering Units corresponding to full scale minus the Engineering Units corresponding to zero scale. For example assume the Engineering Units are 0 to 5 VDC. The Engineering Units corresponding to full scale is 5 VDC and the Engineering Units corresponding to zero scale is 0 VDC. The range is 5 VDC - 0 VDC = 5 VDC. In this case the counts per Engineering Unit is equal to 4096/5 = 819 counts per volt. Also let us assume that the slope is 0.125 volts per second, then the slope in volts per 50 mS = 0.125/20 = 0.00625 volts per 50 mS. To get the counts per 50 mS, you multiply the counts per volt by the volts per 50 mS. Thus slope = 819 x .00625 = 5.11875 counts per 50 mS. To convert this into units of 1/65536 of a count per 50 mS, we must multiply by 65,536. Thus ZZZZZZZZ = 5.11875 x 65,536 = 335,462 decimal (00051E66 Hex).

The above can be stated as an equation, as follows:

ZZZZZZZZ = (S/20) x (C/EU range) x 65,536

Where

S = Slope in Engineering Units per second.
C = Total range in counts = 4095.
EU Range = Engineering Units corresponding to full scale minus the Engineering Units corresponding to zero scale.

EXAMPLES:

ASCII Protocol:

Command	> 8F O 20 2F cr
Response	A B4 21 03 03 0FFF 00028000 00051EB8 21 cr

Sends a Read Event Table Entry command in ASCII protocol to the I/O Unit at address 8F Hex. Data field EE = 20 Hex specifies that data for the Event/Reaction Table entry number 20 Hex is to be returned. Data verification is 8-bit checksum.

The structure of the return data field is as follows:

A	=	Acknowledge (Success) for Command, A = 41 Hex
B4	=	Control Byte (B4 Hex = 10110100 binary)
21	=	CC = 21: Compare Channel Number = 2, Compare Data Type = 1
03	=	Reaction Command Number = 03 Hex 'Ramp DAC Output to Endpoint'
03	=	Reaction Channel Number = 03 Hex
0FFF	=	SSSS = Reaction DAC Endpoint in Counts (See Note below)
00028000	=	NNNNNNNN = 00028000 Hex = Compare Data Field (Setpoint)
00051EB8	=	ZZZZZZZZ = 00051EB8 Hex = Reaction Ramp Slope
24	=	Data Verification Field = 24 Hex (8-bit checksum).
cr	=	Carriage Return (0D Hex)

Note: *The module connected to Channel 03 is an OPTO 22 #DA4, 0 to 5 VDC module. The endpoint specified by command M (issued previously) was 5.0 VDC. This corresponds to a count of SSSS = 0FFF Hex = 4095 decimal.*

Control Byte = 10110100 binary is interpreted as follows:

Bit 7 = 1 This event/reaction table entry is enabled. When the event occurs, the reaction command will be executed.

Bit 6 = 0 The event interrupt is disabled.

Bit 5 = 1 This is the last event table entry.

Bit 4 = 1 The event entry is 'Set Event on I/O >= Setpoint ' (see command K).

Bit 3 = 0 This event is *not* monitoring the watchdog timer.

Bit 2 = 1 This is a valid Event/Reaction table entry.

Bit 1 = 0 The event condition is currently *no*t matched

Bit 0 = 0 The event has not occurred. The event latch is clear.

The Compare Channel is CC=21. The upper nibble = 2 specifies that the compare channel is channel 02. The data type code is the lower nibble = 1. This indicates that the data type is Current Engineering Units (10 Hex).

The Reaction Command Number is RC=03, 'Ramp DAC Output to Endpoint'.

The Reaction Channel Number is RN=03. This is the analog output channel that will be used for ramping.

The field SSSS=0FFF Hex = 4095 Hex is the ramp endpoint in counts. This endpoint was specified (by a previous Command 'M') as 5.00 VDC. The module connected to channel 03 is an OPTO 22 #DA4 (0 to 5 VDC). The endpoint of 4095 counts corresponds to 5 VDC in Engineering Units. All OPTO 22 modules are scaled for 0 to 4095 counts (12-bit DACs and ADCs).

The Compare Data Field is NNNNNNNN = 00028000 is the comparison value (setpoint) for the Event Command 'Set Event on I/O >= Setpoint' (Command 'K')

The Reaction Ramp Slope is ZZZZZZZZ = 00051EB8 is the incremental value that is added to the analog output of the Reaction Channel Number each 50 milliseconds. The units are in 1/65536 of a count per 50 mS. The slope was given as 0.125 volts per second by a previous command (Command 'M').

The Data Verification Field is DVF=21. The data verification method is 8-bit checksum.

Binary Protocol:

Command	8F 04 4F 0F 211F
Response	11 00 A6 21 02 03 0999 00014000 00030000 E119

Sends a Read Event Table Entry command in Binary protocol to the I/O Unit at address 8F Hex. Data field EE = 0F Hex specifies that data for Event/Reaction table entry number 0F Hex is to be returned. Data verification is 16-bit CRC (DVF = 211F for command and DVF = E119 for the response).

The structure of the return data frame is as follows:

11 = Length field. LEN = 11 Hex (17 decimal).
00 = Error Code EC = 00 Hex (no error).
A6 = Control Byte CB = A6 Hex = 10100110 binary.
21 = Compare Channel Number CC=05.
02 = Reaction Command Number = 02 Hex.
03 = Reaction Channel Number = 0F Hex.
0999 = SSSS = Reaction DAC Output Value in Counts
00014000 = NNNNNNNN = 00014000 Hex.
00030000 = ZZZZZZZZ = 00030000 Hex.
E119 = Data Verification Field (16-bit CRC) DVF = E119 Hex.

The Length Field is LEN=11 Hex (17 decimal). It indicates the number of bytes to follow.

The Error Code is EC=00. It indicates that there are no errors.

The Control Byte is CB=A6 Hex (10100110 binary) is interpreted as follows:

Bit 7 = 1 This Event/Reaction table entry is enabled. When the event occurs, the reaction command will be executed.
Bit 6 = 0 The event interrupt is disabled
Bit 5 = 1 This is the last event table entry.
Bit 4 = 0 The event entry is 'Set Event on <= Setpoint ' (see command L).
Bit 3 = 0 This event is not monitoring the watchdog timer.
Bit 2 = 1 This is a valid Event/Reaction table entry
Bit 1 = 1 The event condition is currently matched.
Bit 0 = 0 The event has not occurred. The event latch is clear.

The Compare Channel Data Field is CC=21. The upper nibble = 2 specifies that the compare channel is channel 02. The data type code is the lower nibble = 1. This indicates that the data type is Current Engineering Units (10 Hex).

The Reaction Command Number is RC=02, 'Set DAC Module Magnitude, Eng. Units'.

The Reaction Channel Number is RN=03. This is the analog output channel that will be used for ramping.

The field SSSS=0999 Hex = 2457 is the Reaction DAC Output Value in counts. This value was specified (by a previous Command 'M') as 3.00 VDC. The module connected to channel 03 is an OPTO 22 G4DA4 (0 to 5 VDC). The output value of 2457 counts corresponds to 3.0 VDC in Engineering Units. All OPTO 22 modules are scaled for 0 to 4095 counts (12-bit DACs and ADCs).

The Compare Data Field is NNNNNNNN = 00014000 is the comparison value (setpoint) for the Event Command 'Set Event on I/O <='. The Reaction Ramp Slope is ZZZZZZZZ = 0003000 has no meaning for this example.

Blank spaces in the above example are *added* for clarity only. They do not actually appear in the command or in the response.

DESCRIPTION:

This command is used to enable or to disable the interrupt output function of an entry in the Event/Reaction table entry specified by data field EE. A zero value in data field SS specifies that the interrupt is to be disabled. A non-zero value specifies that the interrupt is to be enabled.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	I	EE	SS	DVF	CR

Where:
> = ASCII Character > (3E Hex) AA = Address
I = ASCII Character I (49 Hex) EE = Event Entry Number
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	49 hex	EE	SS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
49 Hex = Command I (49 Hex) EE = Event Entry Number
SS = Enable Status, zero to disable, non-zero to enable
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

Before issuing this command, you must first issue a K, L or y command to place an entry in the Event/Reaction table. The Global Event Interrupt Enable Bit (bit 7) in the Option Control Byte must be set to a '1' before interrupts will be sent to the host computer. An interrupt is generated whenever there is an event match for the Event/Reaction table entry whose interrupt status is a '1'.

EXAMPLES:

ASCII Protocol:

Command	>64I02FFA1cr
Response	A41cr

Sends an Set Event Interrupt Status command in ASCII protocol to the I/O Unit at address 64 Hex. Data field EE = 02 specifies that the Event/Reaction table entry number is 02 Hex. The enable status field SS = FF Hex specifies that interrupts are to be enabled for this table entry. Data verification method is 8-bit checksum.

Binary Protocol:

Command	64 05 49 01 01 82 60
Response	03 00 F0 00

Sends an Set Event Interrupt Status command in Binary protocol to the I/O Unit at address 64 Hex. Data field EE = 01 specifies that the Event/Reaction table entry number is 01 Hex. The enable status field SS = 01 Hex specifies that interrupts are to be enabled for this table entry. Data verification method is 16-bit CRC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command places the *event* entry in the event/reaction table. The entry number is specified by data field EE. The event occurs when the watchdog timer times out.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	y	EE	DVF	CR

Where:
> = ASCII Character > (3E Hex)
AA = Address
y = ASCII Character y (79 Hex)
EE = Event Table Entry Group
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
EC = Error Code
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	79 hex	EE	DVF

Where: AA = Address (00 to FF Hex)
79 = Command y (79 Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
EE = Event Table Entry Group

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code = 00 Hex (Zero)

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)
DVF = Data Verification Field
EC = Error Code (Non Zero)

REMARKS:

In order for this event to be effective, you must (1) enable the watchdog timer (see command D) and (2) enable the Event/Reaction table entry (see command N) and (3) use command 'M' to define the reaction. The channels specified by the reaction command must be configured properly.

EXAMPLES:

ASCII Protocol:

Command	>64y1044cr
Response	A41cr

Sends a Set Event on Comm Link Watchdog Timeout command in ASCII protocol to the I/O Unit at address 64. The Event Table Entry Number, EE = 10 Hex (16 decimal).

This command enters the specified event in the Event/Reaction Table at entry number 10 Hex. The event occurs when the watchdog timer times out.

Binary Protocol:

Command	64 04 79 10 AD 7C
Response	03 00 F0 00

Sends a Set Event on Comm Link Watchdog Timeout command in Binary protocol to the I/O Unit at address 64. The Event Table Entry Number, EE = 10 Hex (16 decimal).

This command enters the specified event in the Event/Reaction Table at entry number 10 Hex. The event occurs when the watchdog timer times out.

Data verification method is 16-bit CRC. Data Verification field is AD7C Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET EVENT ON I/O >= SETPOINT — COMMAND K

DESCRIPTION:

This command places an event entry in the Event/Reaction table at the entry number specified by data field EE. The event specification defines an analog input or output channel CC and a 16-bit or 32-bit compare setpoint DDDD[DDDD]. Data field TT specifies the data type. See Remarks for data types. The event occurs when the value of the analog input on channel CC equals or exceeds the compare setpoint DDDD[DDDD].

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2	4 or 8	2 or 4	1
>	AA	K	EE	CC	TT	DDDD[DDDD]	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address
- K = ASCII Character K (4B Hex)
- EE = Event Table Entry Number
- CC = I/O Compare Channel Number
- TT = Comparison Data Type
- DDDD[DDDD] = Compare setpoint. 16-bits for counts. 32-bits for Engineering Units
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1	4	1 or 2
AA	LEN	4B hex	EE	CC	TT	DDDD[DDDD]	DVF

Where: AA = Address LEN = Length Field (09 or 0A)
4B = ASCII Character K (4B Hex) EE = Event Table Entry Number
CC = I/O Compare Channel Number TT = Comparison Data Type
DDDD[DDDD] = Compare setpoint. 16-bits for counts. 32-bits for Engineering Units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

When the analog input on channel CC equals or exceeds the setpoint specified by data field DDDD[DDDD], the event has occurred. Note however that the reaction will be executed only once (if the entry is enabled) even though the analog input remains above the setpoint. The event must change from a non-matching to a matching condition before the reaction will be executed again. The analog input must go below the setpoint DDDD[DDDD] before a non-matching condition exists again.

The channel specified may be configured as an input or an output. Data type indicators are in Hexadecimal. Allowed data types are:

00 = Current Counts	10 = Current Engineering Units
01 = Average Counts	11 = Average Engineering Units
02 = Peak Counts	12 = Peak Engineering Units
03 = Lowest Counts	13 = Lowest Engineering Units
04 = Totalized Counts	14 = Totalized Engineering Units

The event table entry must be enabled using command 'N' before this command will be effective. See Chapter 3 for a detailed of explanation of Event/Reaction processing.

EXAMPLES:

ASCII Protocol:

Command	> 0F K 00 02 10 00030000 67cr
Response	A41cr

Sends a Set Event/Reaction on I/O >= Setpoint command in ASCII protocol to the I/O Unit at address 0F Hex.

The fields for the command frame are as follows:

>	=	Start of command frame character
0F	=	Address field AA = 0F Hex
K	=	Command K = Set Event/Reaction on >= Setpoint
00	=	Event Table Entry Number, EE = 00 Hex (0 decimal)
02	=	Analog Compare Channel, CC = 02 Hex (2 decimal)
10	=	Comparison Data Type TT = 10 Hex (Current Engineering Units)
00030000	=	Analog I/O Compare Value, DDDD[DDDD] = 00030000 Hex (3.0 decimal)
67	=	Data verification field (8-bit checksum). DVF = 67 Hex
cr	=	Carriage return (0D Hex)

This command enters the specified event/reaction in the Event/Reaction Table as entry number EE = 00 Hex. Command K is the 'Set Event/Reaction on >= Setpoint' command. When the analog input on channel CC = 02 Hex equals or exceeds 00030000 Hex (3.00 decimal), the event has occurred. If the event table entry 00 has been enabled, then the reaction command will be executed.

Data verification method is 8-bit checksum. Checksum is 67 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	4C 0A 4B FE 02 11 00 02 80 00 E6 DB
Response	03 00 F0 00

Sends a Set Event/Reaction on Counter >= command in Binary protocol to the I/O Unit at address 4C. The fields for the command frame are as follows:

4C	=	Address field AA = 4C Hex
0A	=	Length Field, LEN = 0A Hex (10 decimal)
4B	=	Command K (4B Hex) = Set Event/Reaction on >= Setpoint
FE	=	Event Table Entry Number, EE = FE Hex (254 decimal)
02	=	Analog Compare Channel, CC = 02 Hex (2 decimal)
11	=	Comparison Data Type TT = 11 Hex (Average Engineering Units)
00028000	=	Counter Compare Value, DDDD[DDDD] = 00028000 Hex (2.5 decimal)
E6DB	=	Data verification field (16-bit CRC). DVF = E6DB Hex

This command enters the specified event/reaction in the Event/Reaction Table as entry number FE Hex. Command K is the 'Set Event/Reaction on >= Setpoint' command. When the analog input on channel 02 Hex equals or exceeds 00028000 Hex (2.50 decimal), the event has occurred. If the event table entry FE has been enabled, then the reaction command will be executed.

Data verification method is 16-bit CRC. Data Verification field is E6 DB Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET EVENT ON I/O <= SETPOINT — COMMAND L

DESCRIPTION:

This command places an event entry in the Event/Reaction table at the entry number specified by data field EE. The event specification defines an analog input or output channel CC and a 16-bit or 32-bit compare setpoint DDDD[DDDD]. Data field TT specifies the data type. See Remarks for data types. The event occurs when the value of the analog input or output on channel CC is equal to or less than the compare setpoint DDDD[DDDD].

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2	4 or 8	2 or 4	1
>	AA	L	EE	CC	TT	DDDD[DDDD]	DVF	CR

Where:
> = ASCII Character > (3E Hex)   AA = Address
L = ASCII Character L (4C Hex)   EE = Event Table Entry Number
CC = I/O Compare Channel Number   TT = Comparison Data Type
DDDD[DDDD] = Compare setpoint. 16-bits for counts. 32-bits for Engineering Units
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)   DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1	4 or 8	1 or 2
AA	LEN	4C hex	EE	CC	TT	DDDD[DDDD]	DVF

Where: AA = Address LEN = Length Field (09 or 0A)
4C = ASCII Character L (4C Hex) EE = Event Table Entry Number
CC = I/O Compare Channel Number TT = Comparison Data Type
DDDD[DDDD] = Compare setpoint. 16-bits for counts. 32-bits for Engineering Units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

When the analog input on channel CC equals or exceeds the setpoint specified by data field DDDD[DDDD], the event has occurred. Note however that the reaction will be executed only once (if the entry is enabled) even though the analog input remains above the setpoint. The event must change from a non-matching to a matching condition before the reaction will be executed again. The analog input must go below the setpoint DDDD[DDDD] before a non-matching condition exists again.

The channel specified may be configured as an input or an output. Data type indicators are in Hexadecimal. Allowed data types are:

00 = Current Counts	10 = Current Engineering Units
01 = Average Counts	11 = Average Engineering Units
02 = Peak Counts	12 = Peak Engineering Units
03 = Lowest Counts	13 = Lowest Engineering Units
04 = Totalized Counts	14 = Totalized Engineering Units

The event table entry must be enabled using command 'N' before this command will be effective. See Chapter 3 for a detailed of explanation of Event/Reaction processing.

EXAMPLES:

ASCII Protocol:

Command >0F L 00 02 10 00028000 67cr
Response A41cr

Sends a 'Set Event/Reaction on I/O <='. The fields for the command frame are as follows:

>	=	Start of command frame character
0F	=	Address field AA = 0F Hex
L	=	Command L = Set Event/Reaction on <= Setpoint
00	=	Event Table Entry Number, EE = 00 Hex (0 decimal)
02	=	Analog Compare Channel, CC = 02 Hex (2 decimal)
10	=	Comparison Data Type TT = 10 Hex (Current Engineering Units)
00028000	=	Analog I/O Compare Value, DDDD[DDDD] = 00028000 Hex (2.5 decimal)
67	=	Data verification field (8-bit checksum). DVF = 67 Hex
cr	=	Carriage return (0D Hex)

This command enters the specified event/reaction in the Event/Reaction Table as entry number EE = 00 Hex. Command L is the 'Set Event/Reaction on <='. Data verification method is 8-bit checksum. Checksum is 67 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	4C 0A 4C 20 02 11 00 02 80 00 0D A4
Response	03 00 F0 00

Sends a Set Event/Reaction on Counter >= command in Binary protocol to the I/O Unit at address 4C. The fields for the command frame are as follows:

4C	=	Address field AA = 4C Hex
0A	=	Length Field, LEN = 0A Hex (10 decimal)
4C	=	Command L (4C Hex) = Set Event/Reaction on <= Setpoint
20	=	Event Table Entry Number, EE = 20 Hex (32 decimal)
02	=	Analog Compare Channel, CC = 02 Hex (2 decimal)
11	=	Comparison Data Type TT = 11 Hex (Average Engineering Units)
00028000	=	Counter Compare Value, DDDD[DDDD] = 00028000 Hex (2.5 decimal)
0DA4	=	Data verification field (16-bit CRC). DVF = 0DA4 Hex

This command enters the specified event/reaction in the Event/Reaction Table as entry number 20 Hex. Command L is the 'Set Event/Reaction on <='. Data verification method is 16-bit CRC. Data Verification field is 0D A4 Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET EVENT/REACTION COMMAND — COMMAND M

DESCRIPTION:

This command places a reaction command entry in the Event/Reaction table. The event table entry contains two parts (1) an event specification and (2) a reaction specification. The event specification is entered into the Event/Reaction table by commands y, K, and L. The reaction command can be selected from a list of ten commands. See 'Remarks' for a list of reaction commands that can be used.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or more	2 or 4	1
>	AA	M	EE	[REACTION COMMAND]	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address
M = ASCII Character M (4D Hex) EE = Event Table Entry Number
REACTION COMMAND = Command to be entered into Event/Reaction Table
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Note: *See Remarks for a list of reaction commands that can be used.*

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1 or more	1 or 2
AA	LEN	4D hex	EE	[REACTION COMMAND]	DVF

Where: AA = Address LEN = Length Field
4D = Command M (4D Hex) EE = Event Table Entry Number
REACTION COMMAND = Command to entered into Event/Reaction Table
DVF = Data Verification Field

Note: *See Remarks for a list of reaction commands that can be used.*

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

In order for an event/reaction to be executed, you must (1) enter an event specification into the Event/Reaction Table using commands K, L or y, (2) enter a reaction command into the Event/Reaction Table using command M and (3) enable the table entry using command N. In addition to the above, you must remember to configure and/or enable the channels specified by the reaction command.

The following reaction commands are allowed:

Description of Reaction Commands Allowed For Command M:

NULL REACTION (NO ACTION): 00

No command is executed. The event latch is set upon event occurrence.

SET DAC MODULE MAGNITUDE COUNTS: 01 CC DDDD

Where CC = Channel Number (00 to 0F Hex)
DDDD = Output Magnitude in Counts (0 to 4095)

See Command x, chapter 14.

SET DAC MODULE MAGNITUDE, ENG. UNITS: 02 CC DDDDDDDD

Where CC = Analog Output Channel Number for Ramping (00 to 0F Hex)
DDDDDDDD = Output Magnitude in Eng. Units in increments of 1/65,536 of an Engineering Unit

See command w, chapter 14.

RAMP DAC OUTPUT TO ENDPOINT: 03 CC EEEEEEEE SSSSSSSS

Where CC = Analog Output Channel Number for Ramping (00 to 0F Hex)
EEEEEEEE = Ramp Endpoint in Engineering Units
SSSSSSSS = Ramp Slope in Engineering Units per Second

See command Z, chapter 14.

ENABLE/DISABLE PID LOOP: 04 LL SS

Where LL = PID Loop Number
SS = ENABLE/DISABLE Status. 0 = Disable, 1 = Enable.

See command i and command j, chapter 16.

SET PID LOOP SETPOINT: 05 LL SSSSSSSS

Where LL = PID Loop Number
SSSSSSSS = PID Loop Setpoint in Engineering Units.

See command k, chapter 16.

ENABLE/DISABLE EVENT TABLE ENTRY: 06 EE SS

Where EE = Event Table Entry Number
SS = Enable Setting, 0 = Disable, 1 = Enable.

See command N, chapter 15.

ENABLE/DISABLE EVENT ENTRY GROUP: 07 GG MMMM NNNN

Where GG = Event Entry Group (A group of 16 events)
MMMM = Bitmask for Entries to be Enabled.
NNNN = Bitmask for Entries to be Disabled.

Setting the same bit in MMMM and NNNN will result in an error being returned and the command is not executed. If GG (Event Entry Group Number) is set to FF Hex, then ALL event table entries are specified and bitmask MMMM is checked for enable/disable status. If MMMM is equal to zero (0), then all entries are disabled. If MMMM is not equal to zero (0), then all entries are enabled. In either case, bitmask NNNN *must* be sent, but is ignored.

See command { chapter 15. Applies to firmware revision 1.03 or later.

READ AND HOLD I/O DATA 08 CC TT

Where CC = I/O Channel Number
TT = Data Type. See command 'R' for a list of data types

This reaction command is used to store an I/O value upon an event occurrence for the host to read at a later time. Only one reading per event table entry is buffered and each time the reaction command is executed a new reading will overwrite the old. Command '|' (07C Hex) is used to read the value from the holding register. Square root data types are not supported by this function.

Applies to firmware revision 1.03 or later.

SET PID LOOP MIN-MAX OUTPUT LIMITS 09 LL HHHHHHHH LLLLLLLL

Where LL = PID Loop Number
HHHHHHHH = Output High Limit in Engineering Units.
LLLLLLLL = Output Low Limit in Engineering Units.

Upon event occurrence, this reaction command is used to set the upper and lower limits for the output of a PID loop. The loop output will be clamped at the upper limit if the calculated output value exceeds the upper limit. The loop output will be clamped at the lower limit if the calculated output value is less than the lower limit. Data field LL specifies the PID loop number. Data field HHHHHHHH specifies the output high limit in Engineering Units. Data field LLLLLLLL specifies the output low limit in Engineering Units.

See command p, chapter 16. Applies to firmware revision 1.03 or later.

EXAMPLES:

ASCII Protocol:

Command	> 0F M 00 03 03 00050000 00002000 F0 cr
Response	A 41 cr

Sends a Set Event Reaction command in ASCII protocol to the I/O Unit at address 0F. The fields for the command frame are as follows:

>	=	Start of command frame character
0F	=	Address field AA = 0F Hex
M	=	Command M = Set Event Reaction
00	=	Event Table Entry Number, EE = 02 Hex (02 decimal)
03	=	Reaction Command 03 = Ramp DAC Output to Endpoint
03	=	Analog Output Channel Number (00 to 0F Hex), CC = 03
00050000	=	Ramp Endpoint in Engineering Units, EEEEEEEE = 00050000
00002000	=	Ramp Slope in Engineering Units per Second, SSSSSSSS = 00002000
F0	=	DVF = F0 Hex
cr	=	Carriage return (0D Hex)

This command enters the specified reaction command in the Event/Reaction Table. The reaction command is 03 = Ramp DAC Output to Endpoint. When the event occurs (set by commands K, L or y), the analog output channel specified by data field CC will be ramped from it's current output to the output specified by data field EEEEEEEE, the Ramp Endpoint in Engineering Units. The rate at which the output changes is specified by data field SSSSSSSS, the Ramp Slope in Engineering Units per Second.

Binary Protocol:

Command	4C 0E 4D 00 03 03 00050000 00002000 D11F
Response	03 00 F0 00

Sends a Set Event Reaction command in BINARY protocol to the I/O Unit at address 4C. The fields for the command frame are as follows:

4C	=	Address field AA = 64 Hex
0E	=	Length Field, LEN = 0E Hex (14 decimal)
4D	=	Command M (4D Hex) = Set Event Reaction
00	=	Event Table Entry Number, EE = 00 Hex (0 decimal)
03	=	Reaction Command 03 = Ramp DAC Output to Endpoint
03	=	Analog Output Channel Number (00 to 0F Hex), CC = 03
00050000	=	Ramp Endpoint in Engineering Units, EEEEEEEE = 00050000
00002000	=	Ramp Slope in Engineering Units per Second, SSSSSSSS = 00002000
D11F	=	Data verification field (16-bit CRC). DVF = D11F Hex

This command enters the specified reaction command in the Event/Reaction Table. The reaction command is 03 = Ramp DAC Output to Endpoint. When the event occurs (set by commands K, L or y), the analog output channel specified by data field CC will be ramped from it's current output to the output specified by data field EEEEEEEE, the Ramp Endpoint in Engineering Units. The rate at which the output changes is specified by data field SSSSSSSS, the Ramp Slope in Engineering Units per Second. Data verification method is 16-bit CRC. Data Verification field is D11F Hex for the command and F0 00 Hex for the response.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

CHAPTER 16

Analog PID Loop Commands

INITIALIZE PID LOOP — COMMAND i

DESCRIPTION:

This command is used to initialize the PID loop number specified by data field LL. Data field II specifies the input channel, data field SS specifies the setpoint channel and data field OO specifies the output channel for the PID loop. Data field TTTT is used to specify the scan rate in units of 100 mS. Eight PID loops
are available (00 to 07).

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2	2	4	2 or 4	1
>	AA	i	LL	II	SS	OO	TTTT	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- i = ASCII Character i (69 Hex)
- LL = PID Loop Number (00 to 07 Hex)
- II = Input Channel Number (00-0F)
- SS = Setpoint Channel (00 to 0F Hex)
- OO = Output Channel Number (00-0F)
- TTTT = Scan Rate in 100 millisecond units
- DVF = Data Verification Field
- AA = Address
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	1	1	1	2	1 or 2
AA	LEN	69 hex	LL	II	SS	OO	TTTT	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (08 or 09 Hex)
69 Hex = Command i (69 Hex) LL = PID Loop Number (0-7)
II = Input Channel Number (00-0F) SS = Setpoint Channel (00-0F)
OO = Output Channel Number (00-0F)
TTTT = Scan Rate in 100 millisecond units
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

This command initializes the specified PID loop as follows:

Loop control and flags are cleared to zero
Loop gain is set to 1.0
Loop integral repeats per mintute is set to zero
Loop derivative rate is set to zero
Loop setpoint is set to zero
Loop output is set to zero
Loop minimum, maximum setpoint limits are set to zero
Loop upper (max.) output limit is set to full scale
Loop lower (min.) output limit is set to zero scale

Although the setpoint and output is set to zero, these settings will have no effect until the control word is set to allow the loop calculations to start. See Command 'j'.

All input, output and setpoint channels of a PID loop must belong to the same 8 point or 16 point analog I/O unit. If the setpoint is supplied (input) from a separate setpoint channel, then the setpoint channel must have the same scaling (Eng. units range) as the input channel. Alternately, the setpoint may be supplied by the host computer using Command 'k'. In this case, any dummy channel may be specified as the setpoint channel and the dummy channel is *not* required to have the same scaling (Eng. units range) as the input channel. However, the setpoint supplied by the host computer must be in the same Engineering units as the input channel for the PID loop.

Instead of sending the output data to a PID loop output channel, the output data may be read by the host and transferred to any output channel elsewhere in the system. In the case of an off-board output, any dummy channel may be specified as the loop output. The loop output should be set as 'disabled' so the PID loop controller will not affect the normal operation of the dummy channel. Specifying an input channel will have no affect on that channel's normal operation.

The PID loop scan rate is specified in units of 100 milliseconds. This is the rate at which the controller will read the loop inputs, do the calculations, and take control action on the output if it is enabled. A scan rate of zero will default to 1 (100 msec.).

EXAMPLES:

ASCII Protocol:

Command	> FF i 00 06 06 07 0001 49 cr
Response	A41cr

Sends an Initalize PID Loop command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 specifies that loop 00 is to be initalized. Data field II = 06 Hex specifies that the PID loop input channel is 06 Hex. Data field SS = 06 Hex specifies that the PID loop setpoint channel is 06 Hex. This can be any dummy channel number if the setpoint is supplied by the host computer. Data field OO = 07 Hex specifies that the PID loop output channel is 07 Hex. The scan rate is specified by data field TTTT = 0001 to be 100 milliseconds. Data verification method is 8 bit checksum. The checksum is 49 Hex for the command and 41 Hex for the response.

Binary Protocol:

Command	77 09 69 00 06 00 07 0001 9F82
Response	03 00 F000

EXAMPLES:

Sends an Initalize PID Loop command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 specifies that loop 00 is to be initalized. Data field II = 06 Hex specifies that the PID loop input channel is 06 Hex. Data field SS = 00 Hex specifies that the PID loop setpoint channel is 00 Hex. This can be any dummy channel number if the setpoint is supplied by the host computer. Data field OO = 07 Hex specifies that the PID loop output channel is 07 Hex. The scan rate is specified by data field TTTT = 0001 to be 100 milliseconds. Data Verification Field is 9F82 Hex for the command and F000 for the response. The Length Field LEN = 09 Hex indicates that 9 bytes will follow the Length Field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command reads all PID loop parameters. Data field LL specifies the PID loop number (00-07). All PID loop parameters are read and returned. If only one parameter is needed, Command 't' is more efficient.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2 or 4	1
>	AA	T	LL	DVF	CR

Where:

>	=	ASCII Character > (3E Hex)	AA	=	Address (00 to FF Hex)
T	=	ASCII Character T (54 Hex)	LL	=	PID Loop Number (00 to 07 Hex)
DVF	=	Data Verification Field	CR	=	Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4	4	2	2	2	2	8
A	CCCC	RRRR	II	SS	ZZ	OO	PPPPPPPP

number of bytes →

8	8	8	8	8
SSSSSSSS	GGGGGGGG	IIIIIIII	DDDDDDDD	HHHHHHHH

number of bytes →

8	4	4	8	8
LLLLLLLL	ZZZZ	YYYY	OOOOOOOO	AAAAAAAA

number of bytes →

8	8	8	8	1 or 2
BBBBBBBB	QQQQQQQQ	FFFFFFFF	KKKKKKKK	DVF

Where:
A = ASCII Character A (41 Hex).
CCCC = Loop Control Word.
RRRR = Scan Rate in 100 mS units.
II = Input Channel Number (00-0F).
SS = Setpoint Channel Number (00-0F).
ZZ = Reserved for Future Use.
OO = Output Channel Number (00-0F).
PPPPPPPP = Loop Input in Engineering units (Process Variable).
SSSSSSSS = Loop Setpoint in Engineering units.
GGGGGGGG = Loop Gain Constant.
IIIIIIII = Loop Reset Rate in Repeats per Minute (RPM).
DDDDDDDD = Loop Derivative Term in Minutes.
HHHHHHHH = Loop Maximum Setpoint Limit.
LLLLLLLL = Loop Minimum Setpoint Limit.
ZZZZ = Reserved for Future Use.
YYYY = Loop Output in Counts (Controlled Variable).
OOOOOOOO = Loop Output Setting in Engineering units of Output Channel.
AAAAAAAA = Loop Output Maximum Limit in Engineering units.
BBBBBBBB = Loop Output Minimum Limit in Engineering units.
QQQQQQQQ = Loop Output Maximum Change Per Scan
FFFFFFFF = Loop Input Full Scale in Engineering units.
KKKKKKKK = Loop Input Zero Scale in Engineering units.
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)
DVF = Data Verification Field
EC = Error Code
CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1 or 2
AA	LEN	54 hex	LL	DVF

Where: AA = Address (00 to FF Hex)
54 Hex = Command T (54 Hex)
DVF = Data Verification Field
LEN = Length Field (03 or 04 Hex)
LL = PID Loop Number (00 to 07 Hex)

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	2	1	1	1	1	4
LEN	EC=00	CCCC	RRRR	II	SS	ZZ	OO	PPPPPPPP

number of bytes →

4	4	4	4	4
SSSSSSSS	GGGGGGGG	IIIIIIII	DDDDDDDD	HHHHHHHH

number of bytes →

4	2	2	4
LLLLLLLL	ZZZZ	YYYY	OOOOOOOO

number of bytes →

4	4	4	4	4	1 or 2
AAAAAAAA	BBBBBBBB	QQQQQQQQ	FFFFFFFF	KKKKKKKK	DVF

Where: LEN = Length Field (2E or 2F Hex).
EC = Error Code = 00 Hex (Zero).
CCCC = Loop Control Word.
RRRR = Scan Rate in 100 mS units.
II = Input Channel Number (00-0F).
SS = Setpoint Channel Number (00-0F).
ZZ = Reserved for Future Use.
OO = Output Channel Number (00-0F).
PPPPPPPP = Loop Input in Engineering units (Process Variable).

SSSSSSSS = Loop Setpoint in Engineering units.
GGGGGGGG = Loop Gain Constant.
IIIIIIII = Loop Reset Rate in Repeats per Minute (RPM).
DDDDDDDD = Loop Derivative Term in Minutes.
HHHHHHHH = Loop Maximum Setpoint Limit.
LLLLLLLL = Loop Minimum Setpoint Limit.
ZZZZ = Reserved for Future Use.
YYYY = Loop Output in Counts (Controlled Variable).
OOOOOOOO = Loop Output Setting in Engineering units of Output Channel.
AAAAAAAA = Loop Output Maximum Limit in Engineering units.
BBBBBBBB = Loop Output Minimum Limit in Engineering units.
QQQQQQQQ = Loop Output Maximum Change Per Scan.
FFFFFFFF = Loop Input Full Scale in Engineering units.
KKKKKKKK = Loop Input Zero Scale in Engineering units.
DVF = Data Verification Field.

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

If you need only one PID loop parameter, Command 't' is more efficient.

EXAMPLES:

ASCII Protocol:

Command >FF T 00 40 cr
Response A 00E0 0001 06 00 00 07 00024770 00024000
00028000 00078000 00000266 0004C000
00014000 0000 0727 00047860 6A cr

Sends a Read All PID Loop Parameters command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies PID Loop number 00. All PID loop parameters are returned.

The fields for the success response frame are as follows:

A = ASCII Character A = Start of response frame character.
00E0 = Loop Control Word, CCCC = 00E0. Flags = 00 Hex. Switches = E0 Hex.
0001 = Scan Rate in 100 mS units. RRRR = 0001 (100 mS).
06 = Analog Input Channel Number (00 to 0F), II = 06 Hex.
00 = Setpoint Channel Number (00 to 0F Hex), SS = 00 Hex.
00 = Reserved for Future Use (00 to 0F Hex), ZZ = 00 Hex.

07 = Analog Output Channel Number (00 to 0F Hex), OO = 07 Hex.
00024770 = Loop Input in Engineering units, PPPPPPPP = 00024770 Hex.
00024000 = Loop Setpoint in Engineering units, SSSSSSSS = 00024000 Hex.
00028000 = Loop Gain Constant (Dimensionless units), GGGGGGGG = 00028000.
00078000 = Loop Reset Rate in Repeats per Minute (RPM), IIIIIII = 00078000 Hex.
00000266 = Loop Derivative Term in Minutes, DDDDDDDD = 00000266 Hex.
0004C000 = Loop Maximum Setpoint Limit, HHHHHHHH = 0004C000 Hex.
00014000 = Loop Minimum Setpoint Limit, LLLLLLLL = 00014000 Hex.
0000 = Reserved for Future Use, ZZZZ = 0000 Hex.
0727 = Loop Output in Counts (Controlled Variable), YYYY = 0727 Hex counts.
00047860 = Loop Output Setting in Eng. units, OOOOOOOO = 00047860 Hex.
00044000 = Loop Output Maximum in Eng. units, AAAAAAAA = 00044000 Hex.
00008000 = Loop Output Minimum in Eng. units, BBBBBBBB = 00008000 Hex.
7A = Data Verification Field, DVF = 6A Hex.
cr = Carriage Return (0D Hex).

EXAMPLES:

The Control Word CCCC = 00E0 is interpreted as follows: The Flags Byte = 00 indicates that no flags are set. The Switch Byte = E0 means; (bit 7=1) the PID loop is active; (bit 6=1) the PIP loop is in the 'Auto' mode; (bit 5=1) the output is enabled; (bit 4= 0) not applicable when bit 6=1; (bit 3=0) not applicable when bit 6=1; (bit 2=0) setpoint min-max clamping is disabled; (bit 1=0) the setpoint is being supplied by the host computer; and (bit 0=0) the input type is current reading. See Command 'j' for a description of the Control Word, the Flags Byte and the Switch Byte.

Binary Protocol:

Command 77 04 54 00 B564
Response 2F 00 00E0 0001 06 00 00 07 0002FF80
00030000 00028000 00078000 00000266
00050000 00000000 0000 08D9 000587A0
00044000 00008000 3FF1

Sends a Read All PID Loop Parameters command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies PID Loop number 00. All PID loop parameters are returned.

The fields for the success response frame are as follows:

2F = Length Field, LEN= 2F Hex (47 Decimal). Number of Bytes to Follow.
00 = Error Code, EC = 00. Always 00 If There Are No Errors.
00E0 = Loop Control Word, CCCC = 00E0. Flags = 00 Hex. Switches = E0 Hex.
0001 = Scan Rate in 100 mS units. RRRR = 0001 (100 mS.).
06 = Analog Input Channel Number (00 to 0F), II = 06 Hex.
00 = Setpoint Channel Number (00 to 0F Hex), SS = 00 Hex.
00 = Reserved for Future Use (00 to 0F Hex), ZZ = 00 Hex.
07 = Analog Output Channel Number (00 to 0F Hex), OO = 07 Hex.
0002FF80 = Loop Input in Engineering units, PPPPPPPP = 0002FF80 Hex.
00030000 = Loop Setpoint in Engineering units, SSSSSSSS = 00030000 Hex.
00028000 = Loop Gain Constant (Dimensionless units), GGGGGGGG = 00028000.
00078000 = Loop Reset Rate in Repeats per Minute (RPM), IIIIIII = 00078000 Hex.
00000266 = Loop Derivative Term in Minutes, DDDDDDDD = 00000266 Hex.

00050000	=	Loop Maximum Setpoint Limit, HHHHHHHH = 00050000 Hex.
00000000	=	Loop Minimum Setpoint Limit, LLLLLLLL = 00000000 Hex.
0000	=	Reserved for Future Use, ZZZZ = 0000 Hex.
08D9	=	Loop Output in Counts (Controlled Variable), YYYY = 08D9 Hex counts.
000587A0	=	Loop Output Setting in Eng. units, OOOOOOOO = 000587A0 Hex.
00044000	=	Loop Output Maximum in Eng. units, AAAAAAAA = 00044000 Hex.
00008000	=	Loop Output Minimum in Eng. units, BBBBBBBB = 00008000 Hex.
3FF1	=	Data Verification Field, DVF = 3FF1 Hex (16 bit CRC).

The Control Word CCCC = 00E0 is interpreted as follows. The Flags Byte = 00 indicates that no flags are set. The Switch Byte = E0 means; (bit 7=1) the PID loop is active; (bit 6=1) the PIP loop is in the 'Auto' mode; (bit 5=1) the output is enabled; (bit 4= 0) not applicable when bit 6=1; (bit 3=0) not applicable when bit 6=1; (bit 2=0) setpoint min-max clamping is disabled; (bit 1=0) the setpoint is being supplied by the host computer; and (bit 0=0) the input type is current reading. See Command 'j' for a description of the Control Word, the Flags Byte and the Switch Byte.

The loop output in counts YYYY = 08D9 Hex = 2265 decimal corresponds to (2265/4096)x10 VDC = 5.53 VDC. The loop output in Engineering units OOOOOOOO = 000587A0 Hex = 362,400 decimal corresponds to 362,400/65,536 VDC = 5.53 VDC. The analog output module for channel 07 is an OPTO 22 P/N DA5 (0 to 10 VDC).

Data verification method is 16 bit CRC. The Data Verification Field DVF = B564 Hex for the command and DVF = 3FF1 Hex for the response.

The Length data field LEN = 2F Hex (47 decimal) for the response indicates that 47 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command reads a PID loop parameter. Data field LL specifies the PID loop number (00-07) and data field PP specifies the PID loop parameter number to be read. See 'Remarks' for a list of the parameters that can be read. The length of the return data field DDDD will depend upon the specific parameter requested.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	2	2 or 4	1
>	AA	t	LL	PP	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
t = ASCII Character t (74 Hex) LL = PID Loop Number (0 to 7 Hex)
PP = PID Loop Parameter Number. (00 to FF Hex) See Remarks
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 8	2 or 4	1
A	DDDD	DVF	CR

Where: A = ASCII Character A (41 Hex) DDDD = PID Loop Parameter. See Remarks.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	1	1 or 2
AA	LEN	74 hex	LL	PP	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (04 or 05 Hex)
74 Hex = Command t (74 Hex) LL = PID Loop Number (0 to 7 Hex)
PP = PID Loop Parameter Number. See Remarks
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2 or 4	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field EC = Error Code = 00 Hex (Zero)
DDDD = PID Loop Parameter. See Remarks.
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The specified PID loop parameter is read and the data returned in data field DDDD. The parameter number, the parameters which can be read and the length of the return data field DDDD is as follows:

Parameter Number	Parameter	Length of DDDD ASCII Protocol Characters	Length of DDDD Binary Protocol Characters
00	Control Word	4	2
01	Scan Rate Word	4	2
02	Loop Output Counts (Word)	4	2
03	Loop Input, Setpoint, Output Channels (Long)	8	4
04	Loop Input Value (Process Variable)	8	4
05	Loop Setpoint Value	8	4
06	Loop Output Value	8	4
07	Loop Gain Term	8	4
08	Loop Integral Term (RPM)	8	4
09	Loop Derivative Term	8	4
0A	Loop Setpoint Maximum Limit	8	4
0B	Loop Setpoint Minimum Limit	8	4
0C	Loop Output Maximum Limit	8	4
0D	Loop Output Minimum Limit	8	4
0E	Loop Output Maxiimum Change per Scan	8	4
0F	Loop Output Counts (Long)	8	4
10	Loop Input Full Scale in Engineering units	8	4
11	Loop Input Zero Scale in Engineering units	8	4
12 to FF	Reserved for Future Use	N/A	N/A

A single parameter is returned *except* when parameter number 03 is specified. Then, data field DDDD corresponds to the following *four* PID loop parameters:

DDDD = II SS ZZ OO

Where: II = Input Channel Number (00-0F)SS = Setpoint Channel Number (00-0F)
ZZ = Reserved for Future Use OO = Output Channel Number (00-0F)

EXAMPLES:

ASCII Protocol:

Command	> FF t 00 00 C0 cr
Response	A 01E0 17 cr

Sends a Read PID Loop Parameter command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies PID Loop number 00. Data field PP = 00 Hex specifies that the parameter to be read is 'Control Word'. Sixteen (16) bits of data will be returned. The returned data is DDDD = 01E0 Hex. Data verification method is 8 bit checksum. The command checksum is C0 Hex.

ASCII Protocol:

Command	> FF t 00 02 C2 cr
Response	A 070F 1E cr

Sends a Read PID Loop Parameter command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies PID Loop number 00. Data field PP = 02 Hex specifies that the parameter to be read is 'Loop Output Counts (Word)'. Sixteen (16) bits of data will be returned. The returned data is DDDD = 070F Hex. Data verification method is 8 bit checksum. The command checksum is C2 Hex.

Binary Protocol:

Command	77 05 74 00 01 1D75
Response	05 00 0000 CC00

Sends a Read PID Loop Parameter command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies PID Loop number 00. Data field PP = 01 Hex specifies that the parameter to be read is 'Scan Rate Word'. Sixteen (16) bits of data will be returned. The returned data is DDDD = 0000 Hex. Data verification method is 16 bit CRC. The Data Verification Field DVF = 1D75 Hex for the command and DVF = CC00 Hex for the response.

The Length data field LEN = 05 for the response indicates that 5 bytes of data follow the LEN field.

Binary Protocol:

Command	77 05 74 00 03 DCF4
Response	07 00 06000007 FD40

Sends a Read PID Loop Parameter command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies PID Loop number 00. Data field PP = 03 Hex specifies that the parameter to be read is 'Loop Input Value (Process Variable)'. Thirty two (32) bits of data will be returned. The returned data is DDDD = 06000007 Hex is interpreted as follows; 06 = II is the PID loop input channel number; 00 = SS is the PID loop setpoint channel number; 00 = ZZ is reserved for future use and ; 07 is the PID loop output channel number. Data verification method is 16 bit CRC.
The Data Verification Field DVF = DCF4 Hex for the command and DVF = FD40 Hex for the response.

The Length data field LEN = 07 for the response indicates that 7 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP CONTROL OPTIONS — COMMAND j

DESCRIPTION:

This command is used to set the PID loop control options. Data field LL specifies the PID Loop number (00 - 07). Data field SSSS specifies bits to be set in the Control Word. Data field CCCC specifies bits to be cleared in the Control Word. See 'Remarks' for a definition of the Control Word.

VERSIONS:

Analog Version 1.12 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	4	4	2 or 4	1
>	AA	j	LL	SSSS	CCCC	DVF	CR

Where:
- > = ASCII Character > (3E Hex)
- AA = Address (00 to FF Hex)
- j = ASCII Character j (6A Hex)
- LL = PID Loop Number (00 to 07 Hex)
- SSSS = Mask of Control Bits to Set
- CCCC = Mask of Control Bits to Clear
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	4 or 8	2 or 4	1
A	DDDD	DVF	CR

Where:
- A = ASCII Character A (41 Hex)
- DDDD = Control Word. See Remarks.
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
- N = ASCII Character N (4E Hex)
- EC = Error Code
- DVF = Data Verification Field
- CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	2	2	1 or 2
AA	LEN	6A hex	LL	SSSS	CCCC	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
6A Hex = Command j (6A Hex) LL = PID Loop Number (00-07)
SSSS = Mask of Control Bits to Set CCCC= Mask of Control Bits to Clear
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	2	1 or 2
LEN	EC=00	DDDD	DVF

Where: LEN = Length Field (04 or 05 Hex) EC = Error Code = 00 Hex (Zero)
DDDD = Control Word. See Remarks.DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

Remarks:

The control word is defined as follows: The most significant byte (MSB) of the control word is called the Flags Byte. The least significant byte (LSB) of the control word is called the Switch Byte. Normally you would not set the bits in the Flags Byte using this command *unless* you wanted to do so for purpose of debugging your program. However, once they are set, they must be cleared by the host computer using this command. The bits in the Control Word are defined as follows:

Byte 1: PID Loop Flags Byte:

Bit 7 - Bit 4: **Reserved for Future Use.**

Bit 3: **1 = Use Square Root value from input channel**.
0 = Use actual input value.

Bit 2: **Maximim Setpoint Exceeded**
This bit is set by the controller when the loop setpoint value exceeds the limit preset by the host. This bit is only affected when the loop is in automatic mode (Switch Byte, bit 6 = 1) *and* when the setpoint min-max clamping is enabled (Switch Byte, bit 2 = 1). This flag bit is a latch. Once set, it will remain set until cleared by the host.

Bit 1: **Minimum Setpoint Exceeded**
This bit is set by the controller when the loop setpoint value falls below the limit preset by the host. This bit is only affected when the loop is in automatic mode (Switch Byte, bit 6 = 1) *and* when the setpoint min-max clamping is enabled (Switch Byte, bit 2 = 1). This flag bit is a latch. Once set, it will remain set until cleared by the host.

Bit 0: **Input Module Inactive (Under-Range)**
This bit is set by the controller when the specified loop input channel is not functioning when data is needed (ie. when the input channel data = 8000 Hex counts). This flag bit is a latch. Once set, it will remain set until cleared by the host.

Byte 0: PID Loop Switch Byte

Bit 7: **1 = Active 0 = Reset**
When this bit is zero, the loop is in reset mode; no calculations are done, all process errors are set to zero, and the scan rate timer is reset. The output will remain unchanged. When this bit is one, processing occurs in accordance with the settings of the other switches.

Bit 6: **1 = Auto 0 = Manual**
When this bit is zero, the loop controller is in manual mode. In manual mode, the controller will check control switch bits 3 and 4 and perform the selected functions. When this bit is one, the loop controller will perform the PID calculation and take control action upon the output.

Bit 5: **1 = Output Enabled 0 = Output Disabled**
When this bit is zero, the loop controller will not move it's output data to the output channel. Any computed or tracking output value is stored in the PID Loop Data Table, but is not moved to the output channel's data table for outputting. When this bit is one, the output channel specified is set to the value determined by the PID loop controller.

Bit 4: **1 = Output Track Input in Manual 0 = No Action**
In manual mode, when this bit is one, the loop output will proportionally track the input. This can be used to create a signal converter (4-20 mA to 0-10 VDC, for example).

Bit 3: **1 = Setpoint Track Input in Manual 0 = No Action**

In manual mode, when this bit is one the loop setpoint will be set equal to the current loop input value. This can be used to permit a bumpless transfer upon return to auto mode.

Bit 2: **1 = Process Variable From Host 0 = Proc. Var. From Channel**

In automatic mode, when this bit is zero, the loop controller will read the specified input channel for the value to be used as the process variable in the PID calculations. When this bit is one, the controller will use the current process variable value from the loops data table. This value may be set by command from the host.

Bit 1: **1 = Setpoint From Channel 0 = Setpoint From Host**

In automatic mode, when this bit is one, the loop controller will read the specified setpoint channel for the value to be used as the setpoint in the PID calculations. When this bit is zero, the controller will use the current setpoint value from the loops data table. This value may be set by command from the host.

Bit 0: **1 = Average Reading 0 = Current Reading**

This bit determines the input type: When this bit is zero, the controller will use the current reading of the specified input channel as the loop input. When this bit is one, and has an averging (filter) function active (Command 'h'), the controller will use the averaged reading of the specified input channel as the loop input value.

A '1' in data field SSSS specifies that the corresponding bit in the Control Word is to be *set*. A '0' in the data field SSSS specifies that no action is to be taken for the corresponding bit position. If it is already a '0', it will remain a zero. If it is already a '1', it will remain a one.

The correspondence between the bits in the data field SSSS and the bit numbers is as follows:

Data Field	S	S	S	S
Bit No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Hex Data	0	0	0	0

In the above example, bits 07, 06, and 05 are to be set. No action is taken for the other bits.

A '1' in data field CCCC specifies that the corresponding bit in the Control Word is to be *cleared*. A '0' in the data field CCCC indicates that no action is to be taken for the corresponding bit position
in the Control Word. If it is already a '0', it will remain a zero. If it is already a '1', it will remain a one.

The correspondance between the bits in the data field CCCC and the bit numbers is as follows:

Data Field	C	C	C	C
Bit No.	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
Example	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Hex Data	0	0	0	0

In the above example, no bits are to be cleared.

EXAMPLES:

ASCII Protocol:

Command > FF j 00 00E0 0000 EB cr
Response A 00E0 16 cr

Sends a Set PID Loop Control Options command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID Loop Number as 00. Data field SSSS = 00E0 specifies the bits in the Control Word that are to be set. Bits 07, 06, and 05 of the Switch Byte in the Control Word are to be set. Data field CCCC = 0000 specifies the bits in the Control Word that are to be cleared. No bits are to be cleared for this example. The Control Word (16 bits) is returned in data field DDDD. Data verification method is 8 bit checksum. The command checksum is EB Hex.

Binary Protocol:

Command 77 08 6A 00 00E0 0000 FEC7
Response 05 00 00E0 4401

Sends a Set PID Loop Control Options command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID Loop Number (00 to 0F). Data field SSSS = 00E0 specifies that bits 07, 06, and 05 of the Switch Byte in the Control Word are to be set. Data field CCCC = 0000 specifies that no bits in the Control Word that are to be cleared. The Control Word (16 bits) is returned in data field DDDD.

Data verification method is 16 bit CRC. The Data Verification Field DVF = FEC7 Hex for the command and DVF = 4401 Hex for the response.

The Length data field LEN = 05 Hex (5 decimal) for the response indicates that 5 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP DERIVATIVE RATE — COMMAND n

DESCRIPTION:

This command sets the PID loop derivative rate constant specified by data field DDDDDDDD for the PID loop number specified by data field LL. Units are in minutes in increments of 1/65,536 of a minute.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	n	LL	DDDDDDDD	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
n = ASCII Character n (6E Hex) LL = PID Loop Number (00 to 07 Hex)
DDDDDDDD = Derivative Rate Constant in Minutes.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	6E hex	LL	DDDDDDDD	DVF

Where: AA = Address (00 to FF Hex)   LEN = Length Field (07 or 08 Hex)
6E Hex = Command n (6E Hex)   LL = PID Loop Number (00 to 07 Hex)
DDDDDDDD = Derivative Rate Constant in Minutes.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex)   EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands 'a' or 'G'.

(2) If you are not using standard scaling parameters, use Command 'f' to set the special Engineering unit Scaling Parameters.

(3) Use Command 'i' to initialize the PID loop.

(4) Use Command 'k' to to set the PID loop setpoint.

(5) Use Command 'l' to set the PID loop gain.

(6) Use Command 'm' to set the PID integral reset rate.

(7) Use Command 'n' to set the PID loop derivative rate constant.

(8) Use Command 'u' to set the PID loop maximum rate of change per scan.

(9) Use Command 'j' to set the PID loop control options. Command 'j' starts the PID loop operating (Switch Byte bit 7 = 1).

Data field DDDDDDDD specifies the PID loop derivative rate constant in minutes. The units are in increments of 1/65,536 of a minute. The data values for DDDDDDDD are in Hex.

EXAMPLES:

ASCII Protocol:

Command	> FF n 00 00000266 E8 cr
Response	A 41 cr

Sends a Set PID Loop Derivative Rate Constant command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The rate constant is specified by data field DDDDDDDD = 00000266 Hex (614 decimal) in units of 1/65536 of a minute. Thus the Derivative Rate Constant is 614/65536 = 0.009369 minutes.

Data verification method is 8 bit checksum. The command checksum is E8 Hex.

Binary Protocol:

Command	77 08 6E 00 00000266 C646
Response	03 00 F000

Sends a Set PID Loop Derivative Rate Constant command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The rate constant is specified by data field DDDDDDDD = 00000266 Hex (614 decimal) in units of 1/65536 of a minute. Thus the Derivative Rate Constant is 614/65536 = 0.009369 minutes.

Data verification method is 16 bit CRC. The Data Verification Field DVF = C646 Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 08 for the command indicates that 8 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP GAIN

DESCRIPTION:

This command sets the PID gain constant specified by data field GGGGGGGG for the PID loop number specified by data field LL. A negative loop gain is used to effect a reverse control action.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	l	LL	GGGGGGGG	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
l = ASCII Character l (6C Hex) LL = PID Loop Number (00 to 07 Hex)
GGGGGGGG = Loop Gain Constant (no dimensions).
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	6C hex	LL	GGGGGGGG	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
6C Hex = Command I (6C Hex) LL = PID Loop Number (00 to 07 Hex)
GGGGGGGG = Loop Gain Constant (no dimensions).
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands ‘a’ or ‘G’.

(2) If you are not using standard scaling parameters, use Command ‘f’ to set the special Engineering unit Scaling Parameters.

(3) Use Command ‘i’ to initialize the PID loop.

(4) Use Command ‘k’ to to set the PID loop setpoint.

(5) Use Command ‘l’ to set the PID loop gain.

(6) Use Command ‘m’ to set the PID integral reset rate.

(7) Use Command ‘n’ to set the PID loop derivative rate constant.

(8) Use Command 'u' to set the PID loop maximum rate of change per scan.

(9) Use Command ‘j’ to set the PID loop control options. Command ‘j’ starts the PID loop operating (Switch Byte bit 7 = 1).

Data field GGGGGGGG specifies the PID loop gain constant in dimensionless units. It is a 32 bit signed integer scaled by a factor of 65536. A negative loop gain is used to effect a reverse control action. The values for GGGGGGGG are in Hex.

EXAMPLES:

ASCII Protocol:

Command	> FF l 00 00028000 E2 cr
Response	A 41 cr

Sends a Set PID Loop Gain command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The gain is specified by data field GGGGGGGG = 00028000 Hex (163,840 decimal) in units of 1/65536 of a unit. Thus the Gain is 163840/65536 = 2.50.

Data verification method is 8 bit checksum. The command checksum is E2 Hex.

Binary Protocol:

Command	77 08 6C 00 00028000 6E06
Response	03 00 F000

Sends a Set PID Loop Gain command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The gain is specified by data field GGGGGGGG = 00028000 Hex (163,840 decimal) in units of 1/65536 of a unit. Thus the Gain is 163840/65536 = 2.50.

Data verification method is 16 bit CRC. The Data Verification Field DVF = 6E06 Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 08 for the command indicates that 8 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP INTEGRAL RESET RATE — COMMAND m

Description:

This command sets the PID integral reset rate constant specified by data field IIIIIIII for the PID loop specified by data field LL.

Versions:

Analog

ASCII Command Format:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	m	LL	IIIIIIII	DVF	CR

Where:
> = ASCII Character > (3E Hex)   AA = Address (00 to FF Hex)
m = ASCII Character m (6D Hex)   LL = PID Loop Number (00 to 07 Hex)
IIIIIIII = PID Loop Integral Reset Rate.DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII Response Format

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where:
A = ASCII Character A (41 Hex)   DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where:
N = ASCII Character N (4E Hex)   EC = Error Code
DVF = Data Verification Field   CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	6D hex	LL	IIIIIIII	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
6D Hex = Command m (6D Hex) LL = PID Loop Number (00 to 07 Hex)
IIIIIIII = PID Loop Integral Reset Rate. DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands ‘a’ or ‘G’.

(2) If you are not using standard scaling parameters, use Command ‘f’ to set the special Engineering unit Scaling Parameters.

(3) Use Command ‘i’ to initialize the PID loop.

(4) Use Command ‘k’ to to set the PID loop setpoint.

(5) Use Command ‘l’ to set the PID loop gain.

(6) Use Command ‘m’ to set the PID integral reset rate.

(7) Use Command ‘n’ to set the PID loop derivative rate constant.

(8) Use Command 'u' to set the PID loop maximum rate of change per scan.

(9) Use Command ‘j’ to set the PID loop control options. Command ‘j’ starts the PID loop operating (Switch Byte, bit 7 = 1).

Data field IIIIIIII specifies the PID integral reset rate in repeats per minute (RPM). It is a 32 bit unsigned integer scaled by a factor of 65,536.

EXAMPLES:

ASCII Protocol:

Command	> FF m 00 00078000 E8 cr
Response	A 41 cr

Sends a Set PID Loop Integral Reset Rate command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The integral reset rate is specified by data field IIIIIIII = 00078000 Hex (491,520 decimal) in units of 1/65536 of repeats per minute. Thus the integral reset rate is 491,520/65,536 = 7.50 RPM.

Data verification method is 8 bit checksum. The command checksum is E8 Hex.

Binary Protocol:

Command	77 08 6D 00 00078000 BE17
Response	03 00 F000

Sends a Set PID Loop Integral Reset Rate command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The integral reset rate is specified by data field IIIIIIII = 00078000 Hex (491,520 decimal) in units of 1/65536 of repeats per minute (RPM). Thus the integral reset rate is 491,520/65,536 = 7.50 RPM.

Data verification method is 16 bit CRC. The Data Verification Field DVF = BE17 Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 08 for the command indicates that 8 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP MAXIMUM RATE OF CHANGE — COMMAND u

DESCRIPTION:

This command sets the PID loop maximum rate of change specified by data field RRRRRRRR for the PID loop specified by data field LL.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	u	LL	RRRRRRRR	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
u = ASCII Character u (75 Hex) LL = PID Loop Number (00 to 07 Hex)
RRRRRRRR = PID Loop Maximum Rate of Change per Scan.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	75 hex	LL	RRRRRRRR	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
75 Hex = Command u (75 Hex) LL = PID Loop Number (00 to 07 Hex)
RRRRRRRR = PID Loop Maximum Rate of Change per Scan
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands ‘a’ or ‘G’.

(2) If you are not using standard scaling parameters, use Command ‘f’ to set the special Engineering unit Scaling Parameters.

(3) Use Command ‘i’ to initialize the PID loop.

(4) Use Command ‘k’ to to set the PID loop setpoint.

(5) Use Command ‘l’ to set the PID loop gain.

(6) Use Command ‘m’ to set the PID integral reset rate.

(7) Use Command ‘n’ to set the PID loop derivative rate constant.

(8) Use Command ‘u’ to set the PID loop maximum rate of change per scan.

(9) Use Command ‘j’ to set the PID loop control options. Command ‘j’ starts the PID loop operating (Switch Byte, bit 7 = 1).

Data field RRRRRRRR specifies the PID maximum rate of change per scan. It is a 32 bit unsigned integer scaled by a factor of 65,536.

EXAMPLES:

ASCII Protocol:

Command	> FF u 00 00078000 E8 cr
Response	A 41 cr

Sends a Set PID Loop Maximum Rate of Change command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The integral reset rate is specified by data field RRRRRRRR = 00078000 Hex (491,520 decimal) in units of 1/65536. Thus the integral maximum rate of change per scan is 491,520/65,536 = 7.50.

Data verification method is 8 bit checksum. The command checksum is E8 Hex.

Binary Protocol:

Command	77 08 75 00 00078000 6614
Response	03 00 F000

Sends a Set PID Loop Maximum Rate of Change command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The integral reset rate is specified by data field RRRRRRRR = 00078000 Hex (491,520 decimal) in units of 1/65536. Thus the integral maximum rate of change per scan is 491,520/65,536 = 7.50.

Data verification method is 16 bit CRC. The Data Verification Field DVF = 6614 Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 08 for the command indicates that 8 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

SET PID LOOP MIN-MAX OUTPUT LIMITS — COMMAND p

DESCRIPTION:

This command is used to set the upper and lower limits for the output of a PID loop. The loop output will be clamped at the upper limit if the calculated output value exceeds the upper limit. The loop output will be clamped at the lower limit if the calculated output value is less than the lower limit. Data field LL specifies the PID loop number. Data field HHHHHHHH specifies the output high limit in Engineering units. Data field LLLLLLLL specifies the output low limit in Engineering units.

VERSIONS:

Analog - Firmware Revision 1.03 or Later.

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	8	2 or 4	1
>	AA	p	LL	HHHHHHHH	LLLLLLLL	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
p = ASCII Character p (70 Hex) LL = PID Loop Number (00 to 07 Hex)
HHHHHHHH = Upper Output Limit in Engineering units.
LLLLLLLL = Lower Output Limit in Engineering units.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	4	1 or 2
AA	LEN	70 hex	LL	HHHHHHHH	LLLLLLLL	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (0B or 0C Hex)
p = ASCII Character p (70 Hex) LL = PID Loop Number (00 to 0F Hex)
HHHHHHHH = Upper Output Limit in Engineering units.
LLLLLLLL = Lower Output Limit in Engineering units.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The 'Initialize PID Loop' command 'i' sets the defualt output limits to full scale for the high limit and to zero scale for the lower limit. An error is returned if an attempt is made to set a high limit which is less than the low limit. An error is returned if the PID loop output channel is not configured as an output.

EXAMPLES:

ASCII Protocol:

Command	> 6F p 01 00044000 0001C000 69 cr
Response	A 41 cr

Sends a Set PID Loop Min-Max Output Limits command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 specifies PID Loop Number 00. Data field HHHHHHHH = 00044000 (4.25 decimal) specifies the upper output limit in Engineering units. Data field LLLLLLLL = 0001C000 (1.75) specifies the lower output limit in Engineering units.

Data verification method is 8 bit checksum. The command checksum is 69 Hex.

Binary Protocol:

Command	77 0C 70 04 00050000 00000000 0FEE
Response	03 00 F000

Sends a Set PID Loop Min-Max Output Limits command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 04 specifies PID Loop Number 04. Data field HHHHHHHH = 00050000 (5.0 decimal) specifies the upper output limit in Engineering units. Data field LLLLLLLL = 00000000 (0.0 decimal) specifies the lower output limit in Engineering units.

Data verification method is 16 bit CRC. The Data Verification Field DVF = 0FEE Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 0C Hex (12 decimal) for the response indicates that 12 bytes of data follow the LEN field.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command is used to set the minimum and maximum allowable setpoints for the specified PID loop. Data field LL specifies the PID loop number. Data field HHHHHHHH specifies the upper setpoint limit in Engineering units. Data field LLLLLLLL specifies the lower setpoint limit in Engineering units.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	8	2 or 4	1
>	AA	o	LL	HHHHHHHH	LLLLLLLL	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
o = ASCII Character o (6F Hex) LL = PID Loop Number (00 to 07 Hex)
HHHHHHHH = Upper Setpoint Limit in Engineering units.
LLLLLLLL = Lower Setpoint Limit in Engineering units.
DVF = Data Verification Field CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	2	1	2	4	4	2 or 4
AA	LEN	6F hex	LL	HHHHHHHH	LLLLLLLL	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (0B or 0C Hex)
6F Hex = ASCII Character o (6F Hex) LL = PID Loop Number (00 to 0F Hex)
HHHHHHHH = Upper Setpoint Limit in Engineering units.
LLLLLLLL = Lower Setpoint Limit in Engineering units.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

The 'Initialize PID Loop' command 'i' sets the default setpoint limits to full scale for the high limit and to zero scale for the lower limit. An error is returned if an attempt is made to set a high limit which is less than the low limit.

If the setpoint exceeds the upper limit HHHHHHHH, the setpoint will be clamped to the upper limit HHHHHHHH and flag bit 2 in the Flags Byte will be set.

If the setpoint is less than the lower limit LLLLLLLL, the setpoint will be clamped to the lower limit LLLLLLLL and flag bit 1 in the Flags Byte will be set.

The units must be the same as those specified for the input.

EXAMPLES:

ASCII Protocol:

Command	> FF o 00 00048000 00014000 6C cr
Response	A 41 cr

Sends a Set PID Loop Min-Max Setpoint Limits command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 specifies PID Loop Number 00. Data field HHHHHHHH = 00048000 specifies the upper setpoint limit in Engineering units. Data field LLLLLLLL = 00014000 specifies the lower setpoint limit in Engineering units.

Data verification method is 8 bit checksum. The command checksum is 6C Hex.

Binary Protocol:

Command	77 0C 6F 00 0004C000 00010000 6ABD
Response	03 00 F000

Sends a Set PID Loop Min-Max Setpoint Limits command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 specifies PID Loop Number 00. Data field HHHHHHHH = 0004C000 (311,296 decimal) specifies the upper setpoint limit in Engineering units. The upper setpoint limit is 311,296/65,536 = 4.75 Volts DC. Data field LLLLLLLL = 00014000 (81,920 decimal) specifies the lower setpoint limit in Engineering units. The upper setpoint limit is 81,920/65,536 = 1.25 Volts DC.

Data verification method is 16 bit CRC. The Data Verification Field DVF = 6ABD Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 0C Hex (12 decimal) for the response indicates that 12 bytes of data follow the LEN field.

For these examples, the input module was scaled for 0 to 5 VDC. Therefore, the upper and lower setpoint limits must be specified in VDC.

Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

DESCRIPTION:

This command sets the PID loop process variable as specified by data field SSSSSSSS for the PID loop number specified by data field LL. For this command to be effective, bit 2 of the Switch Byte must be set to 1. See Command 'j' for a description of the Switch Byte. Engineering units for SSSSSSSS must be the same as those for the setpoint.

VERSIONS:

Analog Version 1.12 or later

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	q	LL	SSSSSSSS	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
q = ASCII Character q (71 Hex) LL = PID Loop Number (00 to 07 Hex)
SSSSSSSS = PID Loop Process Variable in Engineering units.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

Binary Command Format:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	71 hex	LL	SSSSSSSS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
71 Hex = Command q (71 Hex) LL = PID Loop Number (00 to 0F Hex)
SSSSSSSS = PID Loop Process Variable in Engineering units.
DVF = Data Verification Field

Binary Response Format

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands ‘a’ or ‘G’.

(2) If you are not using standard scaling parameters, use Command ‘f’ to set the special Engineering unit Scaling Parameters.

(3) Use Command ‘i’ to initialize the PID loop.

(4) Use Command ‘k’ to to set the PID loop setpoint.

(5) Use Command ‘l’ to set the PID loop gain.

(6) Use Command ‘m’ to set the PID integral reset rate.

(7) Use Command ‘n’ to set the PID loop derivative rate constant.

(8) Use Command 'u' to set the PID loop maximum rate of change per scan.

(9) Use Command ‘j’ to set the PID loop control options. Command ‘j’ starts the PID loop operating (Switch Byte bit 7 = 1).

Data field SSSSSSSS specifies the PID loop setpoint. Units are Engineering units in increments of 1/65,536 of an Engineering unit. The Engineering units for the process variable must be the same as those for the setpoint used for the PID loop.

DESCRIPTION:

This command sets the PID loop setpoint as specified by data field SSSSSSSS for the PID loop number specified by data field LL. For this command to be effective, bit 1 of the Switch Byte must be set to zero. See Command 'j' for a description of the Switch Byte. Engineering units for SSSSSSSS must be the same as those for the analog input.

VERSIONS:

Analog

ASCII COMMAND FORMAT:

number of bytes →

1	2	1	2	8	2 or 4	1
>	AA	k	LL	SSSSSSSS	DVF	CR

Where: > = ASCII Character > (3E Hex) AA = Address (00 to FF Hex)
k = ASCII Character k (6B Hex) LL = PID Loop Number (00 to 07 Hex)
SSSSSSSS = PID Loop Setpoint in Engineering units.
DVF = Data Verification Field
CR = Carriage Return (0D Hex)

ASCII RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	2 or 4	1
A	DVF	CR

Where: A = ASCII Character A (41 Hex) DVF = Data Verification Field
CR = Carriage Return (0D Hex)

Error Response Message Frame

number of bytes →

1	2	2 or 4	1
N	EC	DVF	CR

Where: N = ASCII Character N (4E Hex) EC = Error Code
DVF = Data Verification Field CR = Carriage Return (0D Hex)

BINARY COMMAND FORMAT:

number of bytes →

1	1	1	1	4	1 or 2
AA	LEN	6B hex	LL	SSSSSSSS	DVF

Where: AA = Address (00 to FF Hex) LEN = Length Field (07 or 08 Hex)
6B Hex = Command k (6B Hex) LL = PID Loop Number (00 to 0F Hex)
SSSSSSSS = PID Loop Setpoint in Engineering units.
DVF = Data Verification Field

BINARY RESPONSE FORMAT

Success Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC=00	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code = 00 Hex (Zero)
DVF = Data Verification Field

Error Response Message Frame

number of bytes →

1	1	1 or 2
LEN	EC	DVF

Where: LEN = Length Field (02 or 03 Hex) EC = Error Code (Non Zero)
DVF = Data Verification Field

REMARKS:

To set up a PID loop, you would normally perform the following steps in the order shown:

(1) Set the channel configuration for the analog input, analog output and the analog setpoint channels using commands 'a' or 'G'.

(2) If you are not using standard scaling parameters, use Command 'f' to set the special Engineering unit Scaling Parameters.

(3) Use Command 'i' to initialize the PID loop.

(4) Use Command 'k' to to set the PID loop setpoint.

(5) Use Command 'l' to set the PID loop gain.

(6) Use Command 'm' to set the PID integral reset rate.

(7) Use Command 'n' to set the PID loop derivative rate constant.

(8) Use Command 'u' to set the PID loop maximum rate of change per scan.

(9) Use Command 'j' to set the PID loop control options. Command 'j' starts the PID loop operating (Switch Byte bit 7 = 1).

Data field SSSSSSSS specifies the PID loop setpoint. Units are Engineering units in increments of 1/65,536 of an Engineering unit. The Engineering units for the setpoint must be the same as those for the analog input channel used for the PID loop.

EXAMPLES:

ASCII Protocol:

Command	> FF k 00 00024000 DD cr
Response	A 41 cr

Sends a Set PID Loop Setpoint command in ASCII protocol to the I/O unit at address FF Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The setpoint is specified by data field SSSSSSSS = 00024000 Hex (147,456 decimal) in Engineering units of 1/65536 of Volts DC. The setpoint is calculated as 147,456/65,536 = 2.25 VDC.

Data verification method is 8 bit checksum. The command checksum is DD Hex.

Binary Protocol:

Command	77 08 6B 00 00030000 D937
Response	03 00 F000

Sends a Set PID Loop Setpoint command in Binary protocol to the I/O unit at address 77 Hex. Data field LL = 00 Hex specifies the PID loop number as 00. The setpoint is specified by data field SSSSSSSS = 00030000 Hex (196,608 decimal) in Engineering units of 1/65536 of Volts DC. The setpoint is calculated as 196,608/65,536 = 3.00 VDC.

Data verification method is 16 bit CRC. The Data Verification Field DVF = D937 Hex for the command and DVF = F000 Hex for the response.

The Length data field LEN = 08 for the command indicates that 8 bytes of data follow the LEN field.

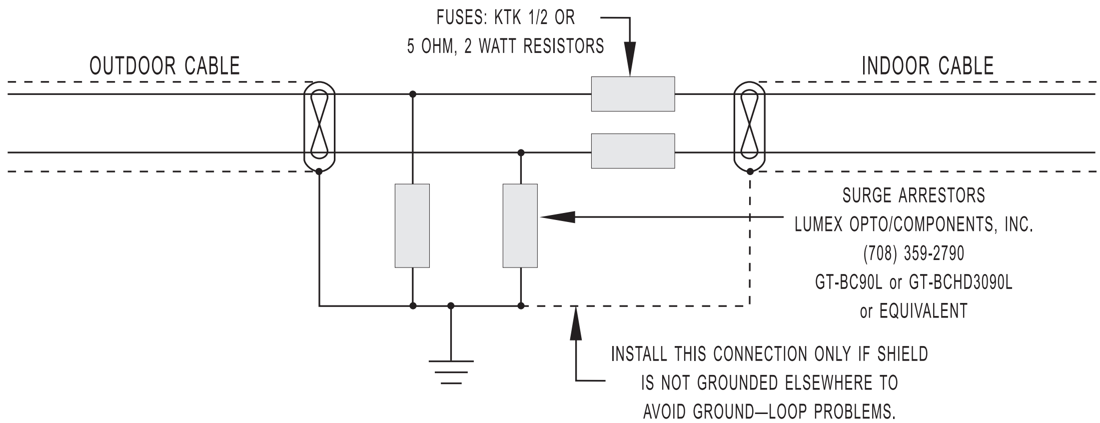
Note: *Blank spaces in the above example are added for clarity only. They do not actually appear in the command or in the response.*

ASCII Character Table

HEX	CHAR.	HEX	CHAR.	HEX	CHAR.	HEX	CHAR.
0	Ctrl-@	20	Space	40	@	60	
1	Ctrl-A	21	!	41	A	61	a
2	Ctrl-B	22		42	B	62	b
3	Ctrl-C	23	#	43	C	63	c
4	Ctrl-D	24	$	44	D	64	d
5	Ctrl-E	25	%	45	E	65	e
6	Ctrl-F	26	&	46	F	66	f
7	Ctrl-G	27		47	G	67	g
8	Ctrl-H	28	(	48	H	68	h
9	Ctrl-I	29	)	49	I	69	i
0A	Ctrl-J	2A	*	4A	J	6A	j
0B	Ctrl-K	2B	+	4B	K	6B	k
0C	Ctrl-L	2C	,	4C	L	6C	l
0D	Ctrl-M	2D	-	4D	M	6D	m
0E	Ctrl-N	2E	.	4E	N	6E	n
0F	Ctrl-O	2F	/	4F	O	6F	o
10	Ctrl-P	30	0	50	P	70	p
11	Ctrl-Q	31	1	51	Q	71	q
12	Ctrl-R	32	2	52	R	72	r
13	Ctrl-S	33	3	53	S	73	s
14	Ctrl-T	34	4	54	T	74	t
15	Ctrl-U	35	5	55	U	75	u
16	Ctrl-V	36	6	56	V	76	v
17	Ctrl-W	37	7	57	W	77	w
18	Ctrl-X	38	8	58	X	78	x
19	Ctrl-Y	39	9	59	Y	79	y
1A	Ctrl-Z	3A	:	5A	Z	7A	z
1B	Ctrl-[	3B	;	5B	[	7B	{
1C	Ctrl-\	3C	<	5C	\	7C	|
1D	Ctrl-]	3D	=	5D	]	7D	}
1E	Ctrl-^	3E	>	5E	^	7E	~
1F	Ctrl-_	3F	?	5F	_	7F	DEL
Ctrl+@ = Null Ctrl+J = Line Feed (LF)				Ctrl+L = Form Feed (FF) Ctrl+M = Carriage Return (CR)			

Surge Protection For RS–422/485 Communication Links

The following is a recommended method of installing surge arrestors on RS-422/485 communications links that are routed outdoors. The surge arrestors help to protect equipment from possible destruction due to lightning strikes. However, this method does not guarantee that damage will not occur, especially in the event of a direct strike.



Notes: *Duplicate as needed for additional pairs. Locate as close as possible to building penetration. Effect of the 5-Ohm resistors is about the same as an additional 200 feet of wire.*

Troubleshooting

Q: I send a command to turn on output 5 of an I/O unit at address 255 and output 4 goes on.

A: Module positions are numbered from 0 to 15, therefore the 5th bit in the bitmask is for position labeled 4. Following are the bitmask values for each position:

Position	Bit Mask
0	0001
1	0002
2	0004
3	0008
4	0010
5	0020
6	0040
7	0080
8	0100
9	0200
10	0400
11	0800
12	1000
13	2000
14	4000
15	8000

Q: I send a command to activate several outputs, but the I/O unit responds with an error, and none of my outputs come on.

A: If an output does not turn on, check the following:

1. Make sure you configured the outputs of the I/O units correctly using command "G" or command "a." On power-up the configuration is restored from EEPROM memory. Factory-default is that all channels are configured as counters (inputs).

2. If power was lost at any time and returned, the I/O unit will have lost its configuration unless it was saved in EEPROM memory with command "E." The I/O unit will have reported a N04 error to all commands sent after power up until a Power-Up Clear command is received.

3. If using a digital I/O unit, make sure you are using output modules with a 5 VDC logic voltage (G4OAC5, G4ODC5, etc.). Sometimes modules with a 15 VDC or 24 VDC logic voltage (G4ODC15, G4OAC24, etc.) are inserted by mistake. The LED may turn on or appear dim with the 15 VDC and 24 VDC modules, but there will be no output on the field side.

4. If the output LED is on, but the load does not turn on, check the field voltage and wiring. Also, the load may not draw enough current to meet the minimum current requirements of G4OAC5 modules (20 mA).

Q: I send a command message to the I/O remote unit at address 0, and I get no response. However, the I/O unit's REC light flashes.

A: The receive lights on all the I/O remote units that are wired in a multidrop mode should light whenever a command is sent. Only the I/O unit at the address that matches the command message address should respond. When an I/O remote unit responds, the XMT (transmit) light will flash. At high baud rates, and for messages with only a short response ("A41cr"only), the flash will be so brief that it may be missed visually.

If the host receives no response, and no flash of the XMT light is visible at the slowest baud rate (300 baud) then check the following:

1. Make sure the address in the command message matches the I/O remote unit's address.Check the jumpers. A common mistake is a reversal of jumpers. Installed address jumpers correspond to a "1," and removed address jumpers removed correspond to a "0." Jumpers are labeled "ADDRESS."

2. Make sure the baud rate is correct between the host and the I/O remote unit. All I/O units on the same link should be configured to the same baud rate.

3. Make sure you have a solid +24 volts powering the I/O remote unit. If the voltage is too low, the RCV light may flash but the I/O remote unit will not be able to respond. Measure the voltage at the I/O unit power terminals, not at the power supply.

4. Check to make sure the communications link is wired with the correct polarity.

Q: When I read position 3 of a I/O unit analog board, I receive an 8,000 Hex, which converts to a decimal -32,768 value.

A: A -32,768 decimal reading may indicate one of the following:

1. You are reading a module position that has no input module installed.
2. You are reading a thermocouple module that has no thermocouple installed or the thermocouple probe is open.
3. An ICTD module that has the ICTD wired in reverse.
4. A 4-20 mA module is wired with reverse polarity.
5. Input is too far underscale. For example, 0 mA into a 4-20 mA module or -1.5 VDC into a 0-5 VDC module.
6. The field connections are made to the wrong terminals. Field connections vary with each module. Refer to *Mistic 200 Family Data Book*, Form 532 for module hook-up.

Q: I receive a large number of checksum errors (N02) when I send commands to a I/O unit.

A: Make sure that you are using twisted pair cable. The RS422/485 network is only reliable when the communications cable is twisted (+ and - lines of EACH pair twisted together), with at least half a twist per inch. Sometimes, the wire is twisted but one of the connections of a pair is actually used as the mate to the opposite pair (- of one pair used as - of opposite pair). This cross-twist condition is usually due to jacketed twisted pair cables that have a wire of each pair with the same color code. In this case, strip the jacket back far enough to properly identify the individual pairs.