M-7003

User Manual

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notification. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademarks

Names are used for identification purposes only and may be registered trademarks of their respective companies.

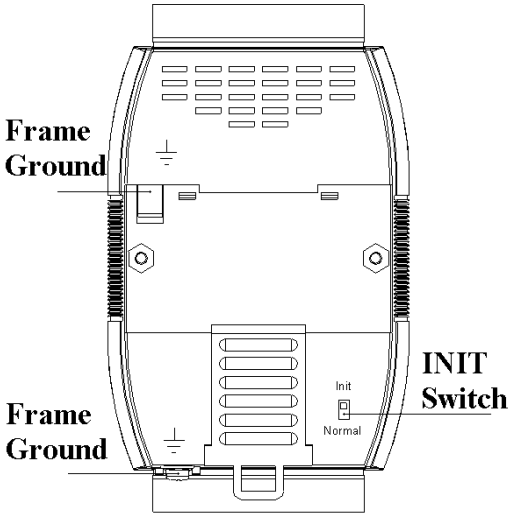
Date: 2015/06/09

Table of Contents

1. Introduction

The M-7000 series is a family of network data acquisition and control modules that provide Analog-to-Digital, Digital-to-Analog, Digital Input/Output, Timer/Counter and other functions. The modules can be remotely controlled using a set of commands called the DCON protocol. Communication between the module and the host is in ASCII format via an RS-485 bi-directional serial bus standard. Baud Rates are software programmable and transmission speeds of up to 115.2 Kbps can be selected.

A number of M-7000 modules feature a new design for the frame ground and INIT switch, as shown in the figure (rear view). The frame ground provides enhanced static (ESD) protection abilities and ensures that the module is more reliable. The INIT switch allows easier access to INIT mode. Refer to Sections 5.1 and 5.3 for more details.



The common features of the M-7000 series are as follows:

1. Voltage or Current Input
2. +/-120 VDC Overvoltage Protection
3. High Resolution: 16-bit
4. 3500 V_{DC} Intra-module Isolation
5. Support for Relay Outputs
6. DIN-Rail Mountable
7. Dual Watchdog
8. Wide Operating Temperature Range: -25 to +75°C

1.1 Applications

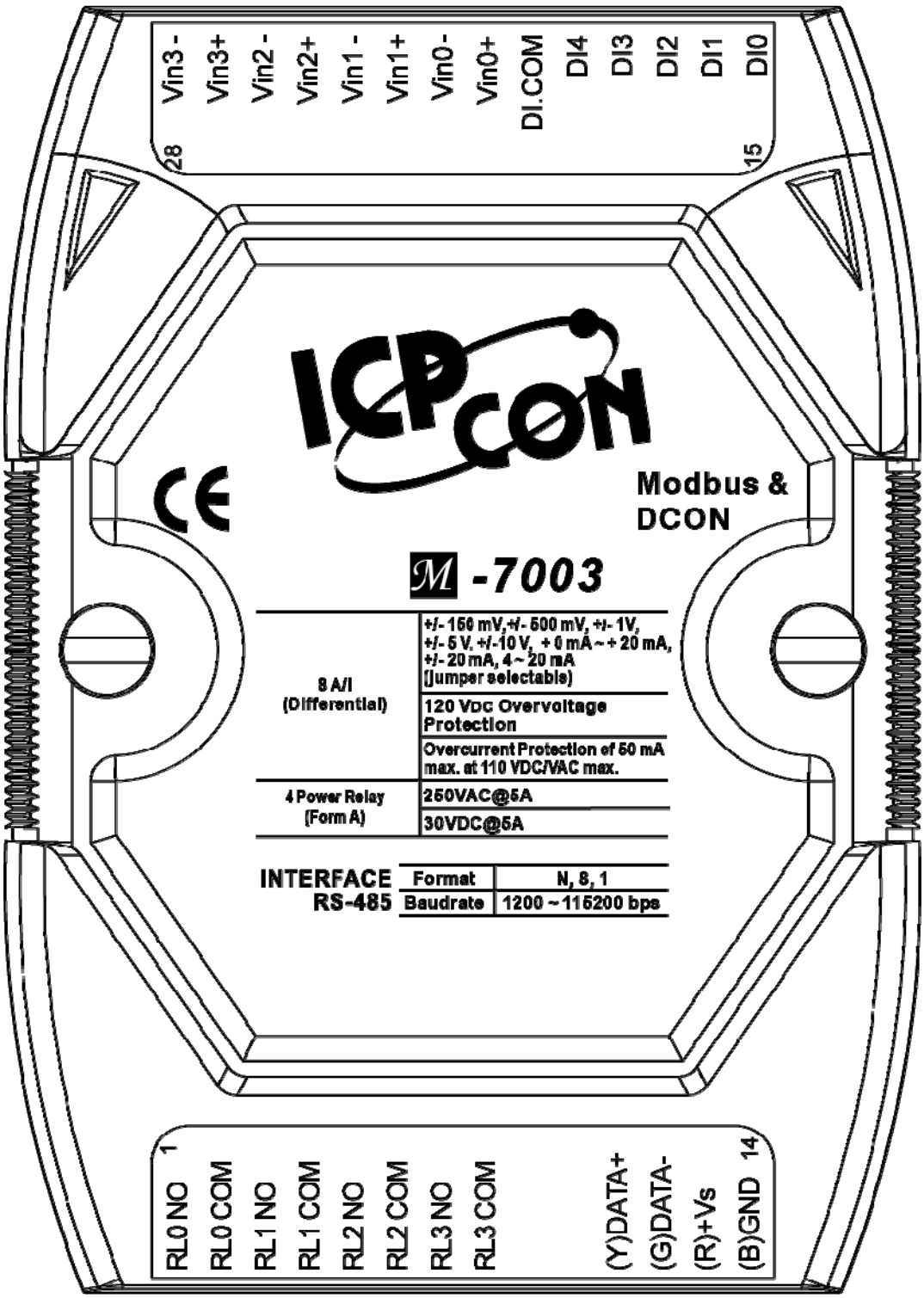
1. Building Automation
2. Factory Automation
3. Machine Automation
4. Remote Maintenance
5. Remote Diagnosis
6. Testing Equipment

1.2 More Information

Refer to Chapter 1 of the M-7000 Bus Converter User Manual" for more information regarding the following:

1.1.	I-7000 Overview
1.2.	I-7000 Related Documentation
1.3.	I-7000 Common Features
1.4.	I-7000 System Network Configuration
1.5.	I-7000 Dimensions

1.3 Pin Assignments



1.4 Specifications

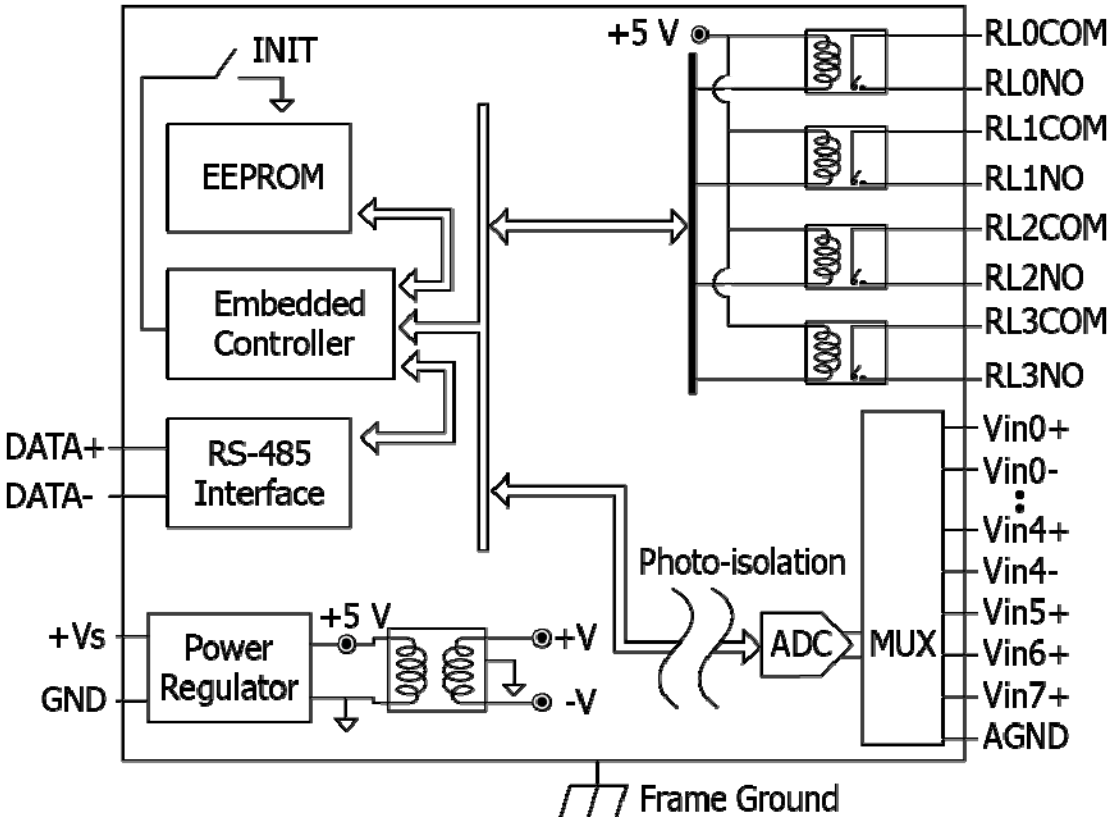
1.4.1 System Specifications

Communication	
Interface	RS-485
Format	N, 8, 1
Baud Rate	1200 to 115200 bps
Protocol	DCON/Modbus RTU
Dual Watchdog	Yes, Module (1.6 Seconds), Communication (Programmable)
LED Indicator/Display	
System LED Indicator	Yes, 1 LED as Power/Communication Indicator
I/O LED Indicator	-
7-Segment LED Display	-
Isolation	
Intra-Module Isolation, Field-to-Logic	3500 V_{DC}
EMS Protection	
ESD (IEC 61000-4-2)	+/-4 kV
EFT (IEC 61000-4-4)	+/-4 kV
Surge (IEC 61000-4-5)	+/-3 kV
Power	
Reverse Polarity Protection	Yes
Input Voltage Range	+10 to +30 V_{DC}
Consumption	1.25 W
Mechanical	
Dimensions (W x L x H)	72 mm x 123 mm x 35 mm
Installation	DIN-Rail or Wall Mounting
Environment	
Operating Temperature	-25 to +75°C
Storage Temperature	-40 to +85°C
Humidity	10 to 95% RH, Non-condensing

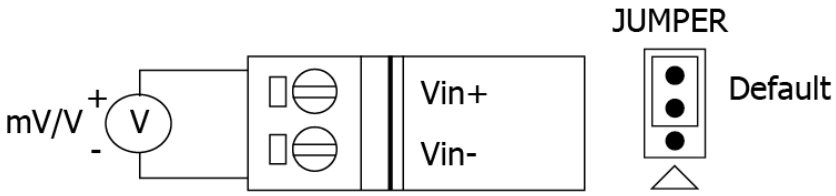
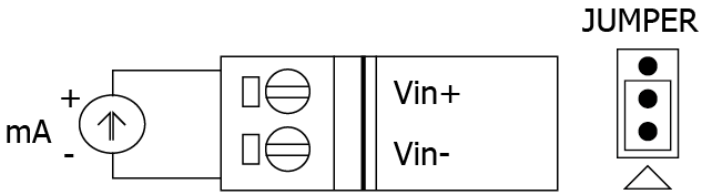
1.4.2 I/O Specifications

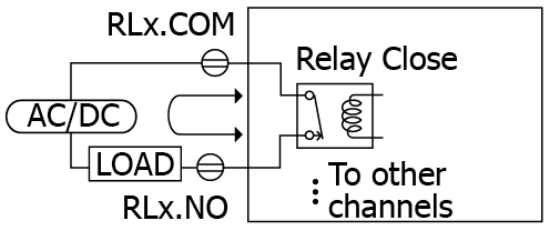
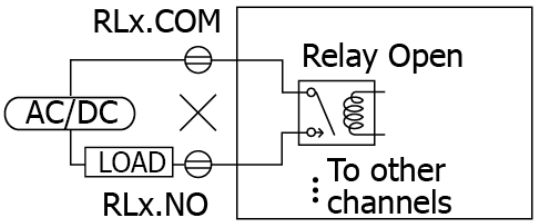
Analog Input		
Channels		8
Wiring		5-channel differential and 3-channel single-ended
Input Range		±150 mV, ±500 mV, ±1 V, ±5 V, ±10 V ±20 mA , 0 to 20 mA, 4 to 20 mA (Jumper Selectable)
Resolution		16-bit
Accuracy	Normal Mode	±0.1%
	Fast Mode	±0.5%
Sampling Rate	Normal Mode	10 Hz
	Fast Mode	60 Hz
Input Impedance	Voltage	2 MΩ
	Current	140 Ω
Individual Channel Configuration		Yes
Overvoltage Protection		+/-120 VDC
Power Relay		
Channels		4
Type		Form A (SPST N.O)
Contact Rating		5 A @ 250 VAC/ 24 VDC (Resistive Load)
Min. Contact Load		10 mA @ 5 V
Operate Time		10 ms (max.)
Release Time		5 ms (max.)
Mechanical Endurance		2 x 10^7 ops.
Electrical Endurance		10^5 ops.
Power-on Value		Yes, Programmable
Safe Value		Yes, Programmable

1.5 Block Diagram



1.6 Application Wiring

Voltage Input Wire Connection

Current Input Wire Connection


Power Relay	
Relay Output	
ON State Readback as 1	
OFF State Readback as 0	

1.7 Default Settings

The default settings for the M-7003 are:

- ☐ Module Address: 01
- ☐ Analog Input Type: Type 08, -10 V to +10 V
- ☐ Protocol: Modbus RTU
- ☐ Baud Rate: 9600 bps
- ☐ Checksum disabled
- ☐ Engineering Units format
- ☐ Filter set at 60 Hz rejection

1.8 Calibration

***Warning**: It is not recommended that calibration be performed until the process is fully understood.*

1.8.1 Analog Input

The Analog Input calibration procedure is as follows:

1. Warm up the module for 30 minutes.
2. Set the Type Code to the type you wish to calibrate. Refer to Section 2.12 for details.
3. Enable calibration. Refer to Section 2.33 for details.
4. Apply the zero calibration voltage/current.
5. Send the Zero Calibration command. Refer to Section 2.6 for details.
6. Apply the span calibration voltage/current.
7. Send the Span Calibration command. Refer to Section 2.5 for details.
8. Repeat steps 3 to 7 three times.

Notes:

1. The calibration voltage/current source should be connected to channel 0.
2. When calibrating Type Code 0D, the jumper for channel 0 should be set to the “current input” position.
3. Calibration voltages and currents are shown below.

Calibration Voltage/Current:

Type Code	08	09	0A	0B	0C	0D
Zero Input	0 V	0 V	0 V	0 mV	0 mV	0 mA
Span Input	+10 V	+5 V	+1 V	+500 mV	+150 mV	+20 mA

1.9 Configuration

1.9.1 Baud Rate Settings (CC)

Bits 5:0

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

Bits 7:6
00: No Parity, 8 Data Bit, 1 Stop Bit
01: No Parity, 8 Data Bit, 2 Stop Bits
10: Even Parity, 8 Data Bit, 1 Stop Bit
11: Odd Parity, 8 Data Bit, 1 Stop Bit

1.9.2 Data Format Settings (FF)

7	6	5	4	3	2	1	0
FS	CS	MS	Reserved			DF	

Key	Description
DF	Data Format 00: Engineering Units 01: % of FSR (full scale range) 10: 2's complement hexadecimal
MS	Mode Settings 0: Normal Mode (16 bits) 1: Fast Mode (12 bits)
CS	Checksum Settings 0: Disabled 1: Enabled
FS	Filter Settings 0: 60Hz Rejection 1: 50Hz Rejection

Note:
Reserved bits should be zero.

1.9.3 Analog Input Type Codes and Data Format

Type Code	Input Type	Data Format	+F.S	-F.S.
07	+4 to +20 mA	Engineering Units	+20.000	+04.000
		% of FSR	+100.00	+000.00
		2's Complement Hexadecimal	FFFF	0000
08	-10 to +10 V	Engineering Units	+10.000	-10.000
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
09	-5 to +5 V	Engineering Units	+5.0000	-5.0000
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
0A	-1 to +1 V	Engineering Units	+1.0000	-1.0000
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
0B	-500 to +500 mV	Engineering Units	+500.00	-500.00
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
0C	-150 to +150 mV	Engineering Units	+150.00	-150.00
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
0D	-20 to +20 mA	Engineering Units	+20.000	-20.000
		% of FSR	+100.00	-100.00
		2's Complement Hexadecimal	7FFF	8000
1A	0 to +20 mA	Engineering Units	+20.000	+00.000
		% of FSR	+100.00	+000.00
		2's Complement Hexadecimal	FFFF	0000

1.10 M-7000 Notes

The main difference between the I-7000 and M-7000 series is that the M-7000 series has additional support for the Modbus RTU communication protocol, which is the default protocol of the M-7000 series. The communication Baud Rates for the Modbus RTU protocol can be in the range of 1200 bps to 115200 bps, and the parity, data and stop bits are fixed as no parity, 8 data bits and 1 stop bit.

Modbus functions supported by the module are described in Chapter 3.

1.10.1 Protocol Switching

To switch to the DCON protocol:

1. Uses sub-function 06h of the function 46h and set byte 8 to a value of 1. See Section 3.3.4 for details.
2. After a power-on reset, the communication protocol will be changed to DCON.

To switch to the Modbus RTU protocol:

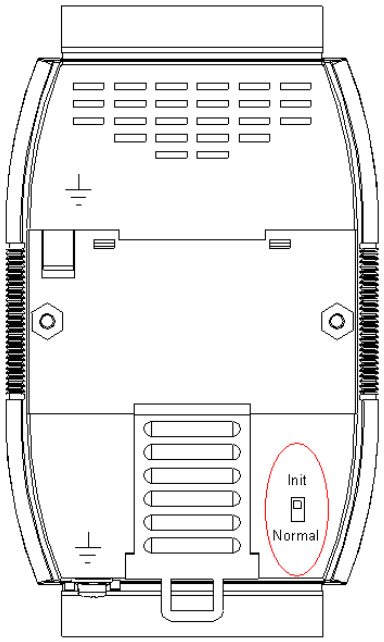
1. Sends the $AAPN command and set N to a value of 1. Note that the slide switch on the rear side of the module should be set to INIT position, see the figure on the next page. See Section 2.20 for details.
2. After a power-on reset, the communication protocol will be changed to Modbus RTU protocol.

1.10.2 INIT Mode

When the module is powered on, with the rear slide switch set to INIT position as shown in the figure below, the module is in INIT mode (Section 5.1), and the communication settings are as follows:

1. Address: 00
2. Baud Rate: 9600 bps
3. No checksum
4. Protocol: DCON

If communication with the module is not possible, set the module to INIT mode and use the above settings to communicate with the module. To read the current settings, send the commands $AA2 (Section 2.7), and $AAPN (Section 2.20). The new communication settings will be effective after the next power-on reset.



INIT Switch

2. DCON Protocol

All communication with the M-7003 module consists of commands generated by the Host and responses transmitted by the module. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The ID is set to 01 by default and can be changed by sending the appropriate user command. All commands to the modules contain the ID number as the address, meaning that only the addressed module will respond. There are two exceptions to this, however: the #** command (Section 2.2) and the ~** command (Section 2.22), which is sent to all modules, but, in these cases, the modules do not respond to the command.

Command Format:

Delimiter Character	Module Address	Command	[CHKSUM]	CR

Response Format:

Delimiter Character	Module Address	Data	[CHKSUM]	CR

CHKSUM A 2-character checksum that is present when the checksum setting is enabled. See Sections 2.1 and 5.1 for details.
CR End of command character, carriage return (0x0D)

Calculating the Checksum:

1. Sum the ASCII codes of all the characters contained in the command/response string, except for the carriage return character (CR).
2. The checksum is equal to the sum value masked by 0FFh.

Example:
Command $012(CR)

1. The sum of the string = “$” + “0” + “1” + “2” = 24h+30h+31h+32h = B7h
2. Therefore the checksum is B7h, and so CHKSUM = “B7”
3. The DCON command string with the checksum = $012B7(CR)

Response !01200600(CR)

1. The sum of the string = “!” + “0” + “1” + “2” + “0” + “0” + “6” + “0” + “0” = 21h+30h+31h+32h+30h+30h+36h+30h+30h = 1AAh
2. Therefore the checksum is AAh, and so CHKSUM = “AA”
3. The DCON response string with the checksum = !01200600AA(CR)

Note:
All characters should be expressed in upper case.

General Command Set			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the Configuration of the Module	2.1
$AA2	!AANNTTCCFF	Reads the Configuration of the Module	2.7
$AA5	!AAS	Reads the Reset Status of the Module	2.9
$AAC	!AA	Clears the Status of the Latched Digital Input/Output Channels	2.14
$AAF	!AA(Data)	Reads the Firmware Version Information for the Module	2.15
$AAI	!AAS	Reads the Status of the INIT Switch	2.16
$AALS	!(Data)	Reads the Status of the Latched Digital Input and Digital Output Channels	2.17
$AAM	!AA(Data)	Reads the Name of the Module	2.18
$AAP	!AASC	Reads the Communication Protocol currently being used by the Module	2.19
$AAPN	!AA	Sets the Communication Protocol to be used by the Module	2.20
~AAD	!AAVV	Reads the Miscellaneous Settings for the Module	2.31
~AADVV	!AA	Sets the Miscellaneous Settings for the Module	2.32
~AAI	!AA	Enables the Software INIT Modification Function	2.34
~AAO(Data)	!AA	Sets the Name of the Module	2.35
~AAR	!AATT	Reads the Reset Time	2.36
~AARTT	!AA	Sets the Reset Time	2.37
~AARD	!AATT	Reads the Response Delay Time for the Module	2.38
~AARDTT	!AA	Sets the Response Delay Time	2.39

		for the Module	
~AATnn	!AA	Sets the Software INIT Timeout Value for the Module	2.40
@AADI	!AAOOII	Reads the Status of all Digital Input and Digital Output Channels	2.48
@AADODD	!AA	Sets the Status of the Digital Output Channels	2.49

Analog Input Command Set			
Command	**Response**	**Description**	**Section**
#**	No Response	Sends the Synchronized Sampling Command	2.2
#AA	>(Data)	Reads the Analog Input Data from all Channels	2.3
#AAN	>(Data)	Reads the Analog Input Data from a Specific Channel	2.4
$AA0	!AA	Performs an Analog Input Span Calibration on the Module	2.5
$AA1	!AA	Performs an Analog Input Zero Calibration on the Module	2.6
$AA4	>AAS(Data)	Reads the previously stored Synchronized Sampling Data	2.8
$AA5VV	!AA	Enables or Disables Specific Analog Input Channels	2.10
$AA6	!AAVV	Reads whether each Analog Input Channels is Enabled or Disabled	2.11
$AA7CiRrr	!AA	Sets the Type Code for a Specific Analog Input Channel	2.12
$AA8Ci	!AACiRrr	Reads the Type Code for a Specific Analog Input Channel	2.13
$AAS1	!AA	Reloads the Default Calibration Parameters	2.21
~ACT	!AAVV	Reads the Threshold of AI Type 4~20mA	2.29
~ACTEVV	!AA	Sets the Threshold of AI Type 4~20mA	2.30

~AAEV	!AA	Enables or Disables Analog Input Calibration for the Module	2.33
@AACH	!AA	Clears the High Latch Values for all Analog Input Channels	2.41
@AACHCi	!AA	Clears the Status of the High Alarm for a Specific Analog Input Channel	2.42
@AACHi	!AA	Clears the High Latch Value for a Specific Analog Input Channel	2.43
@AACL	!AA	Clears the Low Latch Values for all Analog Input Channels	2.44
@AACLCi	!AA	Clears the Status of the Low Alarm for a Specific Analog Input Channel	2.45
@AACLi	!AA	Clears the Low Latch Value for a Specific Analog Input Channel	2.46
@AADA	!AA	Disables the Alarm	2.47
@AAEAT	!AA	Sets the Alarm Mode	2.50
@AAHI(Data)Ci	!AA	Sets the High Alarm Value for a Specific Analog Input Channel	2.51
@AALO(Data)Ci	!AA	Sets the Low Alarm Value for a Specific Analog Input Channel	2.52
@AARAO	!AAHHLL	Reads the Status of the Analog Input Alarm for all Analog Input Channels	2.53
@AARH	!AA(Data)	Reads the High Latch Values for all Analog Input Channels	2.54
@AARHCi	!AA(Data)	Reads the High Alarm Limit for a Specific Analog Input Channel	2.55
@AARHi	!AA(Data)	Reads the High Latch Value for a Specific Analog Input Channel	2.56
@AARL	!AA(Data)	Reads the Low Latch Values for all Analog Input Channels	2.57
@AARLCi	!AA(Data)	Reads the Low Alarm Limit for a Specific Analog Input Channel	2.58
@AARLi	!AA(Data)	Reads the Low Latch Value for a Specific Analog Input Channel	2.59

Host Watchdog Command Sets			
Command	**Response**	**Description**	**Section**
~**	No Response	The command to inform all modules that the Host is OK	2.22
~AA0	!AASS	Reads the Status of the Host Watchdog	2.23
~AA1	!AA	Resets the Status of the Host Watchdog Timeout	2.24
~AA2	!AAEVV	Reads the Timeout Settings for the Host Watchdog	2.25
~AA3ETT	!AA	Enables or disables the Host Watchdog and sets the Host Watchdog Timeout Value	2.26
~AA4	!AAPPSS	Reads the Digital Output Power-on Value and Digital Output Safe Value for the Module	2.27
~AA5PPSS	!AA	Sets the Digital Output Power-on Value and the Digital Output Safe Value for the Module	2.28

2.1 %AANNTTCCFF

Description:
This command is used to set the configuration for a specific module.

Syntax:
%AANNTTCCFF[CHKSUM](CR)

%	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
NN	The new address of the module in hexadecimal format (00 to FF)
TT	Not used by the M-7003 and should be set to 00.
CC	The new Baud Rate code, see Section 1.9.1 for details. To change the Baud Rate, the module should first be switched to INIT* mode.
FF	The command used to set the data format, checksum, and filter settings. See Section 1.9.2 for details of the data format. To change the checksum settings, the module should first be switched to INIT* mode.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command (Note that if the Baud Rate or checksum settings are changed without first switching to INIT* mode, the module will return a response indicating that the command was invalid.)
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
Changes to the address, Type Code, Data Format and Filter settings take effect immediately after a valid command is received. Changes to the Baud Rate and checksum settings take effect at the next power-on reset.

Examples:
Command: %0102000600 Response: !02
Changes the address of module 01 to 02 and returns a response indicating that the command was successful.

Command: %0202000602 Response: !02
Sets the data format for module 02 to type 2 (2's complement hexadecimal). The module returns a response indicating that the command was successful.

Command: %0101000A00 Response: ?01

Attempts to change the Baud Rate of module 01 to 115200 bps, but returns a response indicating that the command was unsuccessful because the module was not switched to INIT* mode before sending the command.

Command: %0101000A00 Response: !01

Changes the Baud Rate of module 01 to 115200 bps and the module is in INIT* mode. The module returns a response indicating that the command was successful.

Related Commands:

Section 2.7 $AA2, Section 2.34 ~AAI

Related Topics:

Section 1.9 Configuration
Section 5.1 INIT Mode

2.2 #**

Description:
This command instructs every Analog Input module to read data from every Analog Input channel and store the data in the buffer for later retrieval.

Syntax:
#[CHKSUM](CR)**
Delimiter character
****** The synchronized sampling command

Response:
There is no response to this command. To access the data, another command, $AA4, must be sent, see Section 2.8 for details.

Examples:
Command: #** No response
Sends the synchronized sampling command to all Analog Input modules.

Command: $014 Response:
>011+025.12+020.45+012.78+018.97+000.00+000.00+000.00+000.00
Sends the command to read the synchronized sampling data from module 01. The module returns a response indicating that the command was successful, containing the data (in Engineering Units format) that was stored when the synchronized sampling command was last issued. The status byte of the response is 1, which means that it is the first time the synchronized sampling data has been read since the previous #** command was received.

Command: $014 Response:
>010+025.12+020.45+012.78+018.97+000.00+000.00+000.00+000.00
Sends the command to read the synchronized sampling data from module 01. The module returns a response indicating that the command was successful, containing the data (in Engineering Units format) that was stored when the synchronized sampling command was last issued. The status byte of the response is 0, which means that it is NOT the first time the synchronized sampling data has been read since the previous #** command was received.

Related Commands:
Section 2.8 $AA4

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.3 #AA

Description:
This command is used to read data from all Analog Input channels of a specified module.

Syntax:
#AA[CHKSUM](CR)
Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)

Response:
Valid Command: **>(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
> Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The data from all Analog Input channels. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: #01 Response: >+025.12+020.45+012.78+018.97+000.00+000.00+000.00+000.00

Reads data from the Analog Input channels of module 01 and returns a response indicating that the command was successful, with the data from all Analog Input channels in Engineering Units format.

Command: #02 Response: >4C532628E2D683A20000000000000000

Reads data from the Analog Input channels of module 02 and returns a response indicating that the command was successful, with the data from all Analog Input channels in hexadecimal format.

Command: #03 Response: >-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9

Reads data from the Analog Input channels of module 03, but returns a response indicating that although the command was successful, the data is not within the valid range.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.4 #AAN, Section 2.7 $AA2, Section 2.12 $AA7CiRrr

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.4 #AAN

Description:
This command is used to read data from Analog Input channel N of a specified module.

Syntax:
#AAN[CHKSUM](CR)
Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
N The Analog Input channel to be read, zero based (0-7)

Response:
Valid Command: **>(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
> Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The data from the specified Analog Input channel. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: #032 Response: >+025.13
Reads data from Analog Input channel 2 of module 03 and returns a response indicating that the command was successful, and that the Analog Input value is +025.13 (+25.13 mV).

Command: #02F Response: ?02
Attempts to read data from Analog Input channel 15 of module 02, but returns a response indicating that the command was unsuccessful because channel 15 does not exist.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.3 #AA, Section 2.7 $AA2, Section 2.12 $AA7CiRrr

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.5 $AA0

Description:
This command is used to perform an Analog Input span calibration on a specified module.

Syntax:
$AA0[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be calibrated in hexadecimal format (00 to FF)
0 The command to perform the Analog Input span calibration

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
The “enable calibration” command, ~AAEV (see Section 2.33), must be sent before this command is used. See Section 1.8.1 for details.

Examples:
Command: $010 Response: ?01
Attempts to perform an Analog Input span calibration on module 01, but a response indicating that the command was unsuccessful is returned because the “Enable Calibration” command (~AAEV, see Section 2.33) was not sent in advance.

Command: ~01E1 Response: !01
Enables calibration on module 01 and returns a response indicating that the command was successful.

Command: $010 Response: !01
Performs an Analog Input span calibration on module 01 and returns a response indicating that the command was successful.

Related Commands:
Section 2.6 $AA1, Section 2.21 $AAS1, Section 2.33 ~AAEV

Related Topics:
Section 1.8.1 Analog Input Calibration

2.6 $AA1

Description:
This command is used to perform an Analog Input zero calibration on a specified module.

Syntax:
$AA1[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be calibrated in hexadecimal format (00 to FF)
1 The command to perform the Analog Input zero calibration

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
The “enable calibration” command, ~AAEV (see Section 2.33), must be sent before this command is used. See Section 1.8.1 for details.

Examples:
Command: $011 Response: ?01
Attempts to perform an Analog Input zero calibration on module 01, but a response indicating that the command was unsuccessful is returned because the “Enable Calibration” command (~AAEV, see Section 2.33) was not sent in advance.

Command: ~01E1 Response: !01
Enables calibration on module 01 and returns a response indicating that the command was successful.

Command: $011 Response: !01
Performs an Analog Input zero calibration on module 01 and returns a response indicating that the command was successful.

Related Commands:
Section 2.5 $AA0, Section 2.21 $AAS1, Section 2.33 ~AAEV

Related Topics:

Section 1.8.1 Analog Input Calibration

2.7 $AA2

Description:
This command is used to read the configuration of a specified module.

Syntax:
$AA2[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
2 The command to read the configuration of the module

Response:
Valid Command: **!AATTCCFF[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
TT Not used by the M-7003 and should be 00
CC The Baud Rate code for the module. See Section 1.9.1 for details of the data format.
FF The data format, checksum and filter settings for the module. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $012 Response: !01000A00
Reads the configuration of module 01. The response indicates that the command was successful and shows that the address is 0x01, the Baud Rate is 0A (115200 bps), the filter settings are set to 60Hz rejection, data format is Engineering Units and the checksum is disabled.

Command: $022 Response: !02000602
Reads the configuration of module 02. The response indicates that the command was successful and shows that the address is 0x02, the Baud Rate is 06 (9600 bps), the filter settings are set to 60Hz rejection, data format is hexadecimal units and the checksum is disabled.

Related Commands:
Section 2.1 %AANNTTCCFF

Related Topics:
Section 1.9 Configuration

Section 5.1 INIT Mode

2.8 $AA4

Description:
This command is used to read the synchronized sampling data that was stored by a specified module when the last #** command (Section 2.2) was issued.

Syntax:
$AA4[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
4 The command to read the synchronized sampling data

Response:
Valid Command: **!AAS(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
S The status of the synchronized sampling data
0: This is the first time the data has been read
1: This is NOT the first time the data has been read
(Data) The synchronized sampling data. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: #** No response
Sends the synchronized sampling command instructing every Analog Input module to read data from every input channel and store the data for later retrieval.

Command: $014 Response:
>011+00.000+00.100+01.000+10.000+00.000+00.000+00.000+00.000
Sends the command to read the synchronized sampling data from module 01. The module returns a response indicating that the command was successful, and containing the data (in Engineering Units format) that was stored when the synchronized sampling command was last issued. The status byte of the response is 1, which means that it is the first time the synchronized sampling data has been read since the previous #** command was received.

Command: $014 Response:
>010+00.000+00.100+01.000+10.000+00.000+0

0.000+00.000+00.000

Sends the command to read the synchronized sampling data from module 01. The module returns a response indicating that the command was successful, and containing the data (in Engineering Units format) that was stored when the synchronized sampling command was last issued. The status byte of the response is 0, which means that it is **NOT** the first time the synchronized sampling data has been read since the previous #** command was received.

Related Commands:
Section 2.2 #**

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.9 $AA5

Description:
This command is used to read the reset status of a specified module.

Syntax:
$AA5[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
5 The command to read the reset status

Response:
Valid Command: **!AAS[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
S The reset status of the module:
0: This is **NOT** the first time the command has been sent since the module was powered on, which denotes that there has been no module reset since the last $AA5 command was sent.
1: This is the first time the command has been sent since the module was powered on.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $015 Response: !011
Reads the reset status of module 01. The module returns a response indicating that the command was successful and that it is the first time the $AA5 command has been sent since the module was powered on.

Command: $015 Response: !010
Reads the reset status of module 01. The module returns a response indicating that the command was successful and that there has been no module reset since the last $AA5 command was sent.

Related Commands:
None

2.10 $AA5VV

Description:
This command is used to specify the Analog Input channels to be enabled on a specified module.

Syntax:
$AA5VV[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
5 The command to set the Analog Input channel(s) to enabled
VV A two-digit hexadecimal value representing the Analog Input channel, where bit 0 corresponds to channel 0, and bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is to be disabled and 1 denotes that the channel is to be enabled.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command.
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $0150A Response: !01
Enables Analog Input channels 1 and 3 on module 01 and disables all other Analog Input channels. The module returns a response indicating that the command was successful.

Command: $016 Response: !010A
Reads the status of the Analog input channels on module 01 and returns a response indicating that the command was successful, with a value of 0A, which denotes that Analog Input channels 1 and 3 are enabled and all other Analog Input channels are disabled.

Related Commands:
Section 2.11 $AA6

2.11 $AA6

Description:
This command is used to read whether each Analog Input channel of a specified module is either enabled or disabled.

Syntax:
$AA6[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
6 The command to read the status of the Analog Input channels

Response:
Valid Command: **!AAVV[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
VV A two-digit hexadecimal value representing the Analog Input channel, where bit 0 corresponds to channel 0, and bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is disabled, and 1 denotes that the channel is enabled.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $0150A Response: !01
Enables Analog Input channels 1 and 3 on module 01 and disables all other Analog Input channels. The module returns a response indicating that the command was successful.

Command: $016 Response: !010A
Reads the status of the Analog Input channels on module 01 and returns a response indicating that the command was successful, with a value of 0A, which denotes that Analog Input channels 1 and 3 are enabled and all other Analog Input channels are disabled.

Related Commands:
Section 2.10 $AA5VV

2.12 $AA7CiRrr

Description:
This command is used to set the Type Code for a specific Analog Input channel of a specified module.

Syntax:
$AA7CiRrr[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
7	The command to set the channel Type Code
Ci	i specifies the Analog Input channel to be set, zero based (0-7)
Rrr	rr represents the Type Code to be set for the Analog Input channel. See Section 1.9.3 for details of the data format.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command or an invalid type code
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $017C0R08 Response: !01
Sets the Type Code for Analog Input channel 0 of module 01 to 08 (-10 to +10 V) and the module returns a response indicating that the command was successful.

Command: $018C0 Response: !01C0R08
Reads the Type Code information for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of 08 denoting that the input range is -10 to +10 V.

Command: $037C1RFF Response: ?03
Attempts to set the Type Code for Analog Input channel 1 of module 03 to FF. The module returns a response indicating that the command was unsuccessful because the Type Code is incorrect.

Related Commands:
Section 2.13 $AA8Ci

Related Topics:

Section 1.9.3 Analog Input Type Codes and Data Format

2.13 $AA8Ci

Description:
This command is used to read the Type Code information for a specific Analog Input channel of a specified module.

Syntax:
$AA8Ci[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
8 The command to read the Type Code information for the Analog Input channel
Ci i specifies which Analog Input channel to access for the Type Code information, zero based (0-7)

Response:
Valid Command: **!AACiRrr[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
Ci i specifies which Analog Input channel the Type Code information relates to zero based (0-7)
Rrr rr represents the Type Code used for the specified Analog Input channel. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $017C0R08 Response: !01
Sets the Type Code for Analog Input channel 0 of module 01 to 08 (-10 to +10 V) and the module returns a response indicating that the command was successful.

Command: $018C0 Response: !01C0R08
Reads the Type Code information for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of 08 denoting that the input range is -10 to +10 V.

Command: $018CF Response: ?01
Attempts to read the Type Code information for Analog Input channel 15 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:

Section 2.3 #AA, Section 2.4 #AAN, Section 2.12 $AA7CiRrr

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.14 $AAC

Description:
This command is used to clear the Digital Input and Digital Output latch data for a specified module.

Syntax:
$AAC[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
C The command to clear the Digital Input and Digital Output latch data

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01L1 Response: !030000
Reads the low latch data for module 01 and returns a response indicating that the command was successful, with a value of 030000 showing that Digital Output channels 0 and 1 are latched low.

Command: $01C Response: !01
Clears the latch data for all channels of module 01 and returns a response indicating that the command was successful.

Command: $01L1 Response: !000000
Reads the low latch data for module 01 and returns a response indicating that the command was successful, with a value of 000000 showing that all latched Digital Output and Digital Input channels have been cleared.

Related Commands:
Section 2.17 $AALS

2.15 $AAF

Description:
This command is used to read the firmware version information for a specified module.

Syntax:
$AAF[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
F The command to read the firmware version information

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The firmware version information for the module as a string value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01F Response: !01A1.0
Reads the firmware version information for module 01 and returns a response indicating that the command was successful and showing that the current firmware is version A1.0.

Related Commands:
None

2.16 $AAI

Description:
This command is used to read the status of the INIT switch on a specified module.

Syntax:
$AAI[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
I The command to read the status of the INIT switch on the module

Response:
Valid Command: **!AAS[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
S The status of the INIT switch on the module
0: The INIT switch is currently in the INIT position
1: The INIT switch is currently in the Normal position

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01I Response: !010
Reads the status of the INIT switch on module 01 and returns a response indicating that the command was successful and showing that the switch is currently in the INIT position.

Related Commands:
None

Related Topics:
Section 5.1 INIT Mode

2.17 $AALS

Description:
This command is used to read the status of the latched Digital Input and Digital Output channels of a specified module.

Syntax:
$AALS[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
L The command to read the status of the latched Digital Input and Digital Output channels
S The status to be read
0: Reads the status of the low latched Digital Input and Digital Output channels
1: Reads the status of the high latched Digital Input and Digital Output channels

Response:
Valid Command: **!(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The status of the latched Digital Output and Digital Input channels represented by a four digit hexadecimal value followed by 00.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01L1 Response: !020000
Reads the status of the high latched Digital Output and Digital Input channels of module 01 and returns a response indicating that the command was successful, with a value of 020000 showing that Digital Output channel 1 is latched high.

Command: $01C Response: !01
Clears the status of the latched Digital Output and Digital Input channels of module 01 and returns a response indicating that the command was successful.

Command: $01L1 Response: !000000
Reads the status of the high latched Digital Output and Digital Input channels of module 01 and returns a response indicating that the command was successful, with a value of 000000 showing that all high

latched Digital Output and Digital Input channels have been cleared.

Command: $01L2 Response: ?01

Attempts to read the status of the high latched Digital Output and Digital Input channels of module 01, but returns a response indicating that the command was unsuccessful because the status byte parameter was incorrect.

Related Commands:

Section 2.14 $AAC, Section 2.49 @AADODD

2.18 $AAM

Description:
This command is used to read the name of a specified module.

Syntax:
$AAM[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
M The command to read the name of the module

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The name of the module as a string value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01O7003 Response: !01
Sets the name of module 01 to “7003” and returns a response indicating that the command was successful.

Command: $01M Response: !017003
Reads the name of module 01 and returns a response indicating that the command was successful, and that the name is “7003”.

Related Commands:
Section 2.35 ~AAO(Data)

2.19 $AAP

Description:
This command is used to read which communication protocol is supported and currently being used by a specified module.

Syntax:
$AAP[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
P The command to read the communication protocol

Response:
Valid Command: **!AASC[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
S Indicates which protocol is supported
0: Only the DCON protocol is supported
1: Both the DCON and Modbus RTU protocols are supported
C Indicates which protocol is currently being used
0: The protocol set in the EEPROM is DCON
1: The protocol set in the EEPROM is Modbus RTU

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01P1 Response: !01
Sets the communication protocol for module 01 to Modbus RTU and returns a response indicating that the command was successful.

Command: $01P Response: !0111
Reads which communication protocol is being used by module 01 and returns a response indicating that the command was successful with a value of 11, meaning that the module supports both the DCON and Modbus RTU protocols, and that the protocol which will be used at the next power-on reset is Modbus RTU.

Related Commands:
Section 2.20 $AAPN

2.20 $AAPN

Description:
This command is used to set the communication protocol to be used by a specified module.

Syntax:
$AAPN[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
P The command to set the communication protocol
N The protocol to be used
0: DCON
1: Modbus RTU

Note:
Before using this command, the INIT switch must be in the INIT position, see Section 5.1 for details. The settings for the new protocol are saved in the EEPROM and will become effective after the next power-on reset.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01P1 Response: ?01
Attempts to set the communication protocol for module 01 to Modbus RTU, but returns a response indicating that the command was unsuccessful, because the INIT switch is not in INIT position.

Command: $01P1 Response: !01
Sets the communication protocol for module 01 to Modbus RTU and returns a response indicating that the command was successful. The new protocol will become effective after the next power-on reset.

Command: $01P Response: !0111
Reads which communication protocol is being used by module 01 and returns a response indicating that the command was successful, with a value of 11 meaning that the module supports both the DCON and Modbus RTU protocols, and that the protocol which will be used at the next power-on reset is Modbus RTU.

Related Commands:
Section 2.19 $AAP

Related Topics:
Section 5.1 INIT Mode

2.21 $AAS1

Description:
This command is used to reload the factory default calibration parameters for a specified module, including the internal calibration parameters.

Syntax:
$AAS1[CHKSUM](CR)
$ Delimiter character
AA The address of the module where the default calibration parameters are to be reloaded in hexadecimal format (00 to FF)
S1 The command to reload the factory default calibration parameters

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
If the accuracy of either the Analog Input or the Analog Output is not within the specifications, the factory default calibration parameters must be reloaded. After sending the $AAS1 command, the parameters will be changed directly without needing to reboot the module.

Examples:
Command: $01S1 Response: !01
Sends a command to reload the factory default calibration parameters for module 01 and returns a response indicating that the command was successful.

Command: $01S0 Response: ?01
Attempts to send a command to reload the factory default calibration parameters for module 01, but returns a response indicating that the command was unsuccessful because the command was incorrect.

Related Commands:
Section 2.5 $AA0, Section 2.6 $AA1, Section 2.33 ~AAEV

Related Topics:
Section 1.8 Calibration

2.22 ~**

Description:
This command is used to inform all modules on the network that the host is OK.

Syntax:
~[CHKSUM](CR)**
~ Delimiter character
****** The “Host OK” command

Response:
There is no response to this command.

Examples:
Command: ~** No response
Sends a “Host OK” command to all modules on the network.

Related Commands:
Section 2.23 ~AA0, Section 2.24 ~AA1, Section 2.25 ~AA2, Section 2.26 ~AA3ETT

Related Topics:
Section 5.2 Dual Watchdog Operation

2.23 ~AA0

Description:
This command is used to read the status of the Host Watchdog for a specified module.

Syntax:
~AA0[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
0 The command to read the status of the Host Watchdog

Response:
Valid Command: **!AASS[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
SS A two-digit hexadecimal value that represents the status of the Host Watchdog, where:
Bit 2: 0 indicates that no Host Watchdog timeout has occurred, and 1 indicates that a Host Watchdog timeout has occurred.
Bit 7: 0 indicates that the Host Watchdog is disabled, and 1 indicates that the Host Watchdog is enabled.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
The status information for the Host Watchdog is stored in the EEPROM and can only be reset using the ~AA1 command. See Section 2.24 for details.

Examples:
Command: ~010 Response: !0100
Reads the status of the Host Watchdog for module 01 and returns a response indicating that the command was successful, with a value of 00, meaning that the Host Watchdog is disabled and no Host Watchdog timeout has occurred.

Command: ~020 Response: !0204
Reads the status of the Host Watchdog for module 02 and returns a response indicating that the command was successful, with a value of 04, meaning that a Host Watchdog timeout has occurred.

Related Commands:

Section 2.22 ~**, Section 2.24 ~AA1, Section 2.25 ~AA2, Section 2.26 ~AA3ETT

Related Topics:
Section 5.2 Dual Watchdog Operation

2.24 ~AA1

Description:
This command is used to reset the status of the Host Watchdog timeout for a specified module.

Syntax:
~AA1[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be reset in hexadecimal format (00 to FF)
1 The command to reset the status of the Host Watchdog timeout

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~010 Response: !0104
Reads the status of the Host Watchdog for module 01 and returns a response indicating that the command was successful, and that a Host Watchdog timeout has occurred.

Command: ~011 Response: !01
Resets the status of the Host Watchdog timeout for module 01 and returns a response indicating that the command was successful.

Command: ~010 Response: !0100
Reads the status of the Host Watchdog for module 01 and returns a response indicating that the command was successful, and showing that no Host Watchdog timeout has occurred.

Related Commands:
Section 2.22 ~**, Section 2.23 ~AA0, Section 2.25 ~AA2, Section 2.26 ~AA3ETT

Related Topics:
Section 5.2 Dual Watchdog Operation

2.25 ~AA2

Description:
This command is used to read the Host Watchdog timeout value for a specified module.

Syntax:
~AA2[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
2 The command to read the Host Watchdog timeout value

Response:
Valid Command: **!AAEVV[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
E The status of the Host Watchdog
0: The Host Watchdog is disabled
1: The Host Watchdog is enabled
VV A two-digit hexadecimal value that represents the timeout value in tenths of a second. For example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~013164 Response: !01
Enables the Host Watchdog for module 01 and sets the Host Watchdog timeout value to 64 (10.0 seconds). The module returns a response indicating that the command was successful.

Command: ~012 Response: !01164
Reads the Host Watchdog timeout value for module 01 and returns a response indicating that the command was successful, with a value of 1FF, which denotes that the Host Watchdog is enabled and that the Host Watchdog timeout value is 10.0 seconds (64).

Related Commands:
Section 2.22 ~**, Section 2.23 ~AA0, Section 2.24 ~AA1, Section 2.26 ~AA3ETT

Related Topics:

Section 5.2 Dual Watchdog Operation

2.26 ~AA3ETT

Description:
This command is used to enable or disable the Host Watchdog for a specified module and sets the Host Watchdog timeout value.

Syntax:

~AA3ETT[CHKSUM](CR)

~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
3 The command to set the Host Watchdog
E The command to set the Host Watchdog
0: Disables the Host Watchdog
1: Enables the Host Watchdog
TT A two-digit hexadecimal value to represent the Host Watchdog timeout value in tenths of a second. For example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~013164 Response: !01
Enables the Host Watchdog for module 01 and sets the Host Watchdog timeout value to 64 (10.0 seconds). The module returns a response indicating that the command was successful.

Command: ~012 Response: !01164
Reads the Host Watchdog timeout value for module 01 and returns a response indicating that the command was successful, with a value of 164, which denotes that the Host Watchdog is enabled and that the Host Watchdog timeout value is 10.0 seconds (64).

Related Commands:
Section 2.22 ~**, Section 2.23 ~AA0, Section 2.24 ~AA1, Section 2.25 ~AA2

Related Topics:
Section 5.2 Dual Watchdog Operation

Note:
When a Host Watchdog timeout occurs, the Host Watchdog will be disabled. In this case the ~AA3ETT command should be sent again to re-enable the Host Watchdog.

2.27 ~AA4

Description:
This command is used to read whether Digital Output power-on value and the Digital Output safe value for a specified module is either active or inactive.

Syntax:
~AA4[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
4 The command to read the Digital Output power-on value and the Digital Output safe value

Response:
Valid Command: **!AAPPSS[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
PP A two-digit hexadecimal value representing the status of the Digital Output power-on value, where bit 0 corresponds to Digital Output channel 0, and bit 1 corresponds to Digital Output channel 1, etc. When the bit is 0, it denotes that the Digital Output power-on is inactive, and 1 denotes that the Digital Output power-on is active.
SS A two-digit hexadecimal value representing the status of the Digital Output safe value, where bit 0 corresponds to Digital Output channel 0, and bit 1 corresponds to Digital Output channel 1, etc. When the bit is 0, it denotes that the Digital Output safe value is inactive, and 1 denotes that the Digital Output safe value is active.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address

Note:
Neither the Digital Output power-on value nor the Digital Output safe value has any effect on Digital Output channels that are associated with alarm outputs.

Examples:
Command: ~0150102 Response: !01
Sets the Digital Output power-on value for module 01 to 01 indicating that the power-on value for Digital Output channel 0 is active and is inactive for all others channels and sets the Digital Output safe value to 02 indicating that the safe value for Digital Output channel 1 is active and is inactive for all others channels, and returns a response indicating that the command was successful.

Command: ~014 Response: !010102

Reads the Digital Output power-on value and the Digital Output safe value for module 01 and returns a response indicating that the command was successful, with a value of 0102, which denotes that the power-on value for Digital Output channel 0 is active and is inactive for all other channels, and that the safe value for Digital Output channel 1 is active and is inactive for all other channels.

Related Commands:

Section 2.28 ~AA5PPSS

2.28 ~AA5PPSS

Description:

This command is used to set the Digital Output power-on value and the Digital Output safe value for a specified module to either active or inactive.

Syntax:

~AA5PPSS[CHKSUM](CR)

~ Delimiter character

AA The address of the module to be set in hexadecimal format (00 to FF)

5 The command to set the Digital Output power-on value and the Digital Output safe value

PP A two-digit hexadecimal value representing the status of the Digital Output power-on value, where bit 0 corresponds to Digital Output channel 0, and bit 1 corresponds to Digital Output channel 1, etc. When the bit is 0, it denotes that the Digital Output power-on is inactive, and 1 denotes that the Digital Output power-on is active.

SS A two-digit hexadecimal value representing the status of the Digital Output safe value, where bit 0 corresponds to Digital Output channel 0, and bit 1 corresponds to Digital Output channel 1, etc. When the bit is 0, it denotes that the Digital Output safe value is inactive, and 1 denotes that the Digital Output safe value is active.

Response:

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate a valid command

? Delimiter character to indicate an invalid command

AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:

Neither the Digital Output power-on value nor the Digital Output safe value has any effect on Digital Output channels that are associated with alarm outputs.

Examples:

Command: ~0150102 Response: !01

Sets the Digital Output power-on value for module 01 to 01 indicating that the power-on value for Digital Output channel 0 is active and is inactive for all others channels, and sets the Digital Output safe value to 02 indicating that the safe value for Digital Output channel 1 is active and is inactive for all others channels, and returns a response indicating that the command was successful.

Command: ~014 Response: !010102

Reads the Digital Output power-on value and the Digital Output safe value for module 01 and returns a response indicating that the command was successful, with a value of 0102, which denotes that the power-on value for Digital Output channel 0 is active and is inactive for all other channels, and that the safe value for Digital Output channel 1 is active and is inactive for all other channels.

Related Commands:

Section 2.27 ~AA4

2.29 ~AACT

Description:
This command is used to read the threshold value for AI type 4~20mA(Type 7).

Syntax:

~AACT[CHKSUM](CR)

~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
CT The command to read the threshold value

Response:

Valid Command: **!AAEVV[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
E The threshold status
0: Disables the threshold function
1: Enables the threshold function
VV A two-digit hexadecimal value to represent the threshold value in 0.1mA. For example, 01 denotes 0.1mA and 28 denotes 4mA. (00 to 28)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~01CT11A Response: !01
Enables the threshold function for module 01 and sets the threshold value to 1A (2.6mA). The module returns a response indicating that the command was successful.

Command: ~01CT Response: !0111A
Reads the threshold value for module 01 and returns a response indicating that the command was successful, with a value of 11A, which denotes that the threshold function is enabled and that the threshold value is 2.6mA (1A).

Related Commands:

Section 2.12 $AA7CiRrr, Section 2.30 ~AACTEVV

2.30 ~AACTEVV

Description:

This command is used to enable or disable the threshold function for a specified module and sets the threshold value.

Syntax:

~AACTEVV[CHKSUM](CR)

~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
CT The command to set the threshold function
E The threshold status
0: Disables the threshold function
1: Enables the threshold function
VV A two-digit hexadecimal value to represent the threshold value in 0.1mA. For example, 01 denotes 0.1mA and 28 denotes 4mA. (00 to 28)

Response:

Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~01CT11A Response: !01
Enables the threshold function for module 01 and sets the threshold value to 1A (2.6mA). The module returns a response indicating that the command was successful.

Command: ~01CT Response: !0111A
Reads the threshold value for module 01 and returns a response indicating that the command was successful, with a value of 11A, which denotes that the threshold function is enabled and that the threshold value is 2.6mA (1A).

Command: ~01CT130 Response: ?01
Enables the threshold function for module 01 and sets the threshold value to 30 (4.8mA). The module returns a response indicating that the command was unsuccessful, because the value is longer than 28.

Related Commands:

Section 2.12 $AA7CiRrr, Section 2.29 ~AACT

2.31 ~AAD

Description:
This command is used to read the Digital Input and Digital Output configuration for a specified module.

Syntax:
~AAD[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
D The command to read the Digital Input and Digital Output configuration

Response:
Valid Command: **!AAVV[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
VV A two-digit hexadecimal value representing the miscellaneous settings, as indicated in the tables below:

7	6	5	4	3	2	1	0
Reserved						OA	Reserved

Key	Description
OA	Bit 1: Specifies the active state of the Digital Output signal 0: An output value of 0 indicates that the output is inactive An output value of 1 indicates that the output is active 1: An output value of 0 indicates that the output is active An output value of 1 indicates that the output is inactive

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~$01D02 Response: !01
Sets the miscellaneous settings for module 01 to 02, meaning that the Digital Output channels are in inactive mode, and returns a response indicating that the command was successful.

Command: ~$01D Response: !0102
Reads the miscellaneous settings for module 01 and returns a response indicating that the command was successful, with a value of 02, which

denotes that the Digital Output channels are in inactive mode.

Related Commands:
Section 2.32 ~AADVV, Section 2.48 @AADI, Section 2.49 @AADODD

2.32 ~AADVV

Description:
This command is used to set the Digital Input and Digital Output configuration for a specified module.

Syntax:
~AADVV[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
D The command to set the Digital Input and Digital Output configuration
VV A two-digit hexadecimal value representing the miscellaneous settings, as indicated in the tables below:

7	6	5	4	3	2	1	0
Reserved						OA	Reserved

Key	Description
OA	Bit 1: Specifies the active state of the Digital Output signal 0: An output value of 0 indicates that the output is inactive An output value of 1 indicates that the output is active 1: An output value of 0 indicates that the output is active An output value of 1 indicates that the output is inactive

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~$01D02 Response: !01
Sets the miscellaneous settings for module 01 to 02, meaning that the Digital Output channels are in inactive mode, and returns a response indicating that the command was successful.

Command: ~$01D Response: !0102
Reads the miscellaneous settings for module 01 and returns a response indicating that the command was successful, with a value of 02, which denotes that the Digital Output channels are in inactive mode.

Related Commands:
Section 2.31 ~AAD, Section 2.48 @AADI, Section 2.49 @AADODD

2.33 ~AAEV

Description:
This command is used to enable or disable calibration for a specified module.

Syntax:
~AAEV[CHKSUM](CR)
~ Delimiter character
AA The address of the module where calibration is to be enabled or disabled in hexadecimal format (00 to FF)
E The command to set the calibration
V The command to enable or disable calibration
0: Disables calibration
1: Enables calibration

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
This command must be sent before any other calibration command can be used.

Examples:
Command: $010 Response: ?01
Attempts to send the command to perform a span calibration on module 01, but returns a response indicating that the command was unsuccessful because the “Enable Calibration” command (~AAEV) has not yet been sent.

Command: ~01E1 Response: !01
Enables calibration on module 01 and returns a response indicating that the command was successful.

Command: $010 Response: !01
Sends the command to perform a span calibration on module 01 and returns a response indicating that the command was successful.

Related Commands:
Section 2.5 $AA0, Section 2.6$AA1, Section 2.21AAS1

Related Topics:
Section 1.8 Calibration

2.34 ~AAI

Description:
This command is used to enable modification of the Baud Rate and checksum settings for a specified module using the software INIT function only.

Syntax:
~AAI[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
I The command to enable the software INIT function

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01T10 Response: !01
Sets the timeout value for the software INIT function on module 01 to 10 (16 seconds) and returns a response indicating that the command was successful.

Command: ~01I Response: !01
Enables the software INIT function on module 01 and returns a response indicating that the command was successful.

Command: %0101000600 Response: !01
Sets the Baud Rate for module 01 to 9600 bps and returns a response indicating that the command was successful.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.34 ~AAI, Section 2.40 ~AATnn

Related Topics:
Section 1.9 Configuration
Section 5.1 INIT Mode

2.35 ~AAO(Data)

Description:
This command is used to set the name of a specified module

Syntax:
~AAO(Data)[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
O The command to set the name of the module
(Data) The new name of the module (Max. 12 characters)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01O7003 Response: !01
Sets the name of module 01 to “7003” and returns a response indicating that the command was successful.

Command: $01M Response: !017003
Reads the name of module 01 and returns a response indicating that the command was successful, with the name “7003”.

Command: ~01O123456789ABCDEF Response: ?01
Attempts to set the name of module 01 to “123456789ABCDEF”, but returns a response indicating that the command was unsuccessful, because the name is longer than 12 characters.

Related Commands:
Section 2.18 $AAM

2.36 ~AAR

Description:
This command is used to read the reset time for a specified module. The time will be restart when received a command. When time up, the module will be reset automatically.

Syntax:
~AAR[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
R The command to read the reset time

Response:
Valid Command: **!AATT[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
TT A two-digit hexadecimal value representing the response delay time value in seconds. For example, 00 denotes disabled reset time and 05 denotes 5 seconds. (00, 05 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01R10 Response: !01
Sets the reset time for module 01 to 10 (16 seconds) and returns a response indicating that the command was successful.

Command: ~01R Response: !0110
Reads the reset time for module 01 and returns a response indicating that the command was successful, with a value of 10 indicating 16 seconds, meaning that will reset the module without any command and response after 16 seconds.

Related Commands:
Section 2.37 ~AARTT

2.37 ~AARTT

Description:

This command is used to set the reset time for a specified module. The time will be restart when received a command. When time up, the module will be reset automatically.

Syntax:

~AARDTT[CHKSUM](CR)

~ Delimiter character

AA The address of the module to be set in hexadecimal format (00 to FF)

R The command to set the reset time

TT A two-digit hexadecimal value representing the response delay time value in seconds. For example, 00 denotes disabled reset time and 05 denotes 5 seconds. (00, 05 to FF)

Response:

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate a valid command

? Delimiter character to indicate an invalid command

AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~01R10 Response: !01

Sets the reset time for module 01 to 10 (16 seconds) and returns a response indicating that the command was successful.

Command: ~01R Response: !0110

Reads the reset time for module 01 and returns a response indicating that the command was successful, with a value of 10 indicating 16 seconds, meaning that will reset the module without any command and response after 16 seconds.

Command: ~01R04 Response: ?01

Sets the reset time for module 01 to 04 (4 seconds) and returns a response indicating that the command was unsuccessful, because the value is less than 05.

Related Commands:

Section 2.36 ~AAR

2.38 ~AARD

Description:
This command is used to read the response delay time for a specified module.

Syntax:
~AARD[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RD The command to read the response delay time

Response:
Valid Command: **!AATT[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
TT A two-digit hexadecimal value representing the response delay time value in milliseconds. For example, 01 denotes 1 millisecond and 1A denotes 26 milliseconds. The value must be less than or equal to 1E.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01RD10 Response: !01
Sets the response delay time for module 01 to 10 (16 milliseconds) and returns a response indicating that the command was successful.

Command: ~01RD Response: !0110
Reads the response delay time for module 01 and returns a response indicating that the command was successful, with a value of 10 indicating 16 milliseconds, meaning that all responses will be sent after 16 milliseconds have elapsed.

Related Commands:
Section 2.39 ~AARDTT

2.39 ~AARDTT

Description:
This command is used to set the response delay time for a specified module.

Syntax:
~AARDTT[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
RD The command to set the response delay time
TT A two-digit hexadecimal value representing the response time value in milliseconds. For example, 01 denotes 1 millisecond and 1A denotes 26 milliseconds. The value must be less than or equal to 1E.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01RD10 Response: !01
Sets the response delay time for module 01 to 10 (16 milliseconds) and returns a response indicating that the command was successful.

Command: ~01RD Response: !0110
Reads the response delay time for module 01 and returns a response indicating that the command was successful, with a value of 10 indicating 16 milliseconds, meaning that all responses will be sent after 16 milliseconds have elapsed.

Command: ~01RD1F Response: ?01
Attempts to set the response delay time for module 01 to 1F (31 milliseconds), but returns a response indicating that the command was unsuccessful because the duration was not within the valid range.

Related Commands:
Section 2.38 ~AARD

2.40 ~AATnn

Description:
This command is used to set the timeout value for the software INIT function on a specified module.

Syntax:
~AATnn[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
T The command to set the timeout value for the software INIT function
nn A two-digit hexadecimal value representing the timeout value for the software INIT function in seconds. For example, 01 denotes 1 second and 1A denotes 26 seconds. The value must be less than or equal to 3C.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~01T10 Response: !01
Sets the timeout value for the software INIT function on module 01 to 16 seconds and returns a response indicating that the command was successful.

Command: ~01I Response: !01
Enables the software INIT function on module 01 and returns a response indicating that the command was successful.

Command: ~01TFF Response: ?01
Attempts to set the timeout value for the software INIT function on module 01 to FF (255 seconds), but returns a response indicating that the command was unsuccessful because the duration is greater than the permitted value (3C).

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.34 ~AAI

2.41 @AACH

Description:
This command is used to clear the high latch values for all Analog Input channels of a specified module.

Syntax:
@AACH[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CH The command to clear the high latch values

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RH0 Response: !01+05.000
Reads the high latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +05.000 (+5.0 V).

Command: @01CH Response: !01
Clears the high latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful.

Command: @01RH0 Response: !01+00.000
Reads the high latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the high latch has been cleared.

Related Commands:
Section 2.43 @AACHi, Section 2.54 @AARH, Section 2.56 @AARHi

2.42 @AACHCi

Description:
This command is used to clear the status of the high alarm for a specific Analog Input channel of a specified module.

Syntax:
@AACHCi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CH The command to clear the status of the high alarm for the Analog Input channel
Ci i specifies the Analog Input channel to be cleared, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RAO Response: !018000
Reads the current status of the alarms associated with module 01 and returns a response indicating that the command was successful, and that a high alarm has occurred on Analog Input channel 7.

Command: @01CHC0 Response: !01
Clears the status of the high alarm for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful.

Command: @01RAO Response: !010000
Reads the current status of the alarms associated with module 01 and returns a response indicating that the command was successful, and that no alarms have occurred.

Command: @01CHCF Response: !01
Attempts to clear the status of the high alarm for Analog Input channel 15 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.47 @AADA, Section 2.48 @AADI, Section 2.50 @AAEAT, Section

2.51 @AAHI(Data)Ci, Section 2.53 @AARAO, Section 2.55 @AARHCi

2.43 @AACHi

Description:
This command is used to clear the high latch value for a specific Analog Input channel of a specified module.

Syntax:
@AACHi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CH The command to clear the high latch value for the Analog Input channel
i The Analog Input channel to be cleared, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RH1 Response: !01+06.000
Reads the high latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful, with a value of +06.000 (6.0 V).

Command: @01CH1 Response: !01
Clears the high latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful.

Command: @01RH1 Response: !01+00.000
Reads the high latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the high latch value has been cleared.

Command: @01CHF Response: !01
Attempts to clear the high latch value for Analog Input channel 15 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:

Section 2.41 @AACH, Section 2.54 @AARH, Section 2.56 @AARHi

2.44 @AACL

Description:
This command is used to clear the low latch value for all Analog Input channels of a specified module.

Syntax:
@AACL[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CL The command to clear the low latch values for all Analog Input channels

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RL0 Response: !01-05.000
Reads the low latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of -05.000 (-5.0 V).

Command: @01CL Response: !01
Clears the low latch value for all Analog Input channels of module 01 and returns a response indicating that the command was successful.

Command: @01RL0 Response: !01+00.000
Reads the low latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the low latch value has been cleared.

Related Commands:
Section 2.46 @AACLi, Section 2.57 @AARL, Section 2.59 @AARLi

2.45 @AACLCi

Description:
This command is used to clear the status of the low alarm for a specific Analog Input channel of a specified module.

Syntax:
@AACLCi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CL The command to clear the status of the low alarm for the Analog Input channel
Ci i specifies the Analog Input channel to be cleared, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RAO Response: !010020
Reads the current status of the alarms associated with module 01 and returns a response indicating that the command was successful, and that a low alarm has occurred on Analog Input channel 5.

Command: @01CLC5 Response: !01
Clears the status of the low alarm for Analog Input channel 5 of module 01 and returns a response indicating that the command was successful.

Command: @01RAO Response: !010000
Reads the current status of the alarms associated with module 01 and returns a response indicating that the command was successful, and that no alarms have occurred.

Command: @01CLCF Response: !01
Attempts to clear the status of the low alarm for Analog Input channel 15 of module 01 and returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.47 @AADA, Section 2.48 @AADI, Section 2.50 @AAEAT, Section

2.52 @AALO(Data)Ci, Section 2.53 @AARAO, Section 2.58 @AARLCi

2.46 @AACLi

Description:
This command is used to clear the low latch value for a specific Analog Input channel of specified module.

Syntax:
@AACLi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be cleared in hexadecimal format (00 to FF)
CL The command to clear the low latch value for the Analog Input channel
i The Analog Input channel to be cleared, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RL1 Response: !01-06.000
Reads the low latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful, with a value of -06.000 (-6.0 V).

Command: @01CL1 Response: !01
Clears the low latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful.

Command: @01RL1 Response: !01+00.000
Reads the low latch value for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the low latch value has been cleared.

Command: @01CLF Response: ?01
Attempts to clear the low latch value for Analog Input channel 15 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:

Section 2.44 @AACL, Section 2.57 @AARL, Section 2.59 @AARLi

2.47 @AADA

Description:
This command is used to disable the high and low alarms for all Analog Inputs of a specified module.

Syntax:
@AADA[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
DA The command to disable the high and low alarms for all Analog Inputs

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01DI Response: !0110000
Reads the Alarm Type for all Analog Inputs of module 01 and returns a response indicating that the command was successful and that the Alarm Type is Momentary.

Command: @01DA Response: !01
Disables the high and low alarms for all Analog Inputs of module 01 and returns a response indicating that the command was successful.

Command: @01DI Response: !0100000
Reads the Alarm Type for all Analog Inputs of module 01 and returns a response indicating that the command was successful and that the alarm is disabled.

Related Commands:
Section 2.42 @AACHCi, Section 2.45 @AACLCi, Section 2.48 @AADI, Section 2.50 @AAEAT, Section 2.51 @AAHI(Data)Ci, Section 2.52 @AALO(Data)Ci, Section 2.53 @AARAO, Section 2.55 @AARHCi, Section 2.58 @AARLCi

2.48 @AADI

Description:
This command is used to read the status of the Digital Input and Digital Output channels of a specified module.

Syntax:
@AADI[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
DI The command to read the status of the Digital Input and Digital Output channels

Response:
Valid Command: **!AATOOII[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
T Analog Input alarm type
0: Disabled
1: Momentary Type
2: Latch Type
OO A two-digit hexadecimal value to denote the status of the Digital Output, where bit 0 corresponds to Digital Output channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the Digital Output channel is inactive, and 1 denotes that the channel is active.
II A two-digit hexadecimal value to denote the status of the Digital Input, where bit 0 corresponds to Digital Input channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the Digital Input channel is inactive, and 1 denotes that the channel is active.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01DO01 Response: !01
Sets Digital Output channel 0 of module 01 to ON, and sets the other Digital Output channels of module 01 to OFF and returns a response indicating that the command was successful.

Command: @01DI Response: !0110100
Reads the status of the Digital Input and Digital Output channels of module 01 and returns a response indicating that the command was successful, with a value of 10100 denoting that the Digital Output is active

on channel 0 and Analog Input alarm type is momentary.

Related Commands:
Section 2.49 @AADODD, Section 2.47 @AADA, Section 2.50 @AAEAT

2.49 @AADODD

Description:
This command is used to set the Digital Output channels of a specified module to active or inactive.

Syntax:
@AADODD[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
DO The command to set the Digital Output channels
DD A two-digit hexadecimal value representing the status of the Digital Output channels, where bit 0 corresponds to Digital Output channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the Digital Output channel will be set to inactive, and 1 denotes that the channel will be set to active.

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Notes:

1. If the Digital Output channel is already set as an alarm output, then the value written to the channel will be ignored.

2. If a Host Watchdog timeout occurs, the module will return a response indicating that the command was invalid and the Digital Output value that was sent will be ignored.

Examples:
Command: @01DO01 Response: !01
Sets Digital Output channel 0 of module 01 to active, and sets the other Digital Output channels to inactive and returns a response indicating that the command was successful.

Command: @01DI Response: !0100100
Reads the status of the Digital Input and Digital Output for all channels of module 01 and returns a response indicating that the command was successful, and showing that Digital Output is active on channel 0.

Related Commands:
Section 2.48 @AADI

2.50 @AAEAT

Description:
This command is used to enable the alarm function and set the Alarm Type for all Analog Inputs of a specified module.

Syntax:
@AAEAT[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be set in hexadecimal format (00 to FF)
EA The command to enable the alarm function for the Analog Input
T The Alarm Type for the Analog Input channel
M: Momentary
L: Latched

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01EAM Response: !01
Enables the alarm function for all Analog Inputs of module 01 and sets the Alarm Type to Momentary and returns a response indicating that the command was successful.

Command: @01DI Response: !0110000
Reads the status of the alarm for Analog Input of module 01 and returns a response indicating that the command was successful and that the alarm is enabled and the Alarm Type is Momentary.

Related Commands:
Section 2.42 @AACHCi, Section 2.45 @AACLCi, Section 2.48 @AADI, Section 2.47 @AADA, Section 2.51 @AAHI(Data)Ci, Section 2.52 @AALO(Data)Ci, Section 2.53 @AARAO, Section 2.55 @AARHCi, Section 2.58 @AARLCi

2.51 @AAHI(Data)Ci

Description:
This command is used to enable the high alarm for a specific Analog Input channel of a specified module and set the high alarm limit.

Syntax:
@AAHI(Data)Ci[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
HI	The command to enable the high alarm for the Analog Input channel
(Data)	The high alarm limit for the Analog Input channel, which should be consistent with the Engineering Units format. Refer to Section 1.9.4 for details of the data format.
Ci	i specifies the Analog Input channel to be set, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01HI+09.000C0 Response: !01
Enables the high alarm for Analog Input channel 0 of module 01 and sets the high alarm limit to +09.000 (+9.0 V), and returns a response indicating that the command was successful.

Command: @01RHC0 Response: !01+09.000
Reads the high alarm limit for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +09.000 (9.0 V).

Command: @01HI+09.000CF Response: ?01
Attempts to enable the high alarm for Analog Input channel 15 of module 01 and set the high alarm limit to +09.000 (9.0 V), but returns an response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.42 @AACHCi, Section 2.47 @AADA, Section 2.50 @AAEAT, Section 2.53 @AARAO, Section 2.55 @AARHCi

Related Topics:

Section 1.9.3 Analog Input Type Codes and Data Format

2.52 @AALO(Data)Ci

Description:
This command is used to enable the low alarm for a specific Analog Input channel of a specified module and set the low alarm limit.

Syntax:
@AALO(Data)Ci[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LO	The command to enable the low alarm for the Analog Input channel
(Data)	The low alarm limit for the Analog Input channel, which should be consistent with the Engineering Units format. Refer to Section 1.9.4 for details of the data format.
Ci	i specifies the Analog Input channel to be set, zero based (0-7)

Response:
Valid Command: **!AA[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01LO-03.000C1 Response: !01
Enables the low alarm for Analog Input channel 1 of module 01 and sets the low alarm limit to -03.000 (-3.0 V), and returns a response indicating that the command was successful.

Command: @01RLC1 Response: !010-03.000
Reads the low alarm limit for Analog Input channel 1 of module 01 and returns a response indicating that the command was successful, with a value -03.000 (-3.0 V).

Command: @01LO-03.000CF Response: ?01
Attempts to enable the low alarm for channel 15 of module 01 and set the low alarm limit to -03.000 (-3.0 V), but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.45 @AACLCi, Section 2.47 @AADA, Section 2.50 @AAEAT, Section 2.53 @AARAO, Section 2.58 @AARLCi

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.53 @AARAO

Description:
This command is used to read which currently activated alarms are associated with a specified module.

Syntax:
@AARAO[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RAO The command to read the currently activated alarms associated with the module

Response:
Valid Command: **!AAHHLL[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
HH A two-digit hexadecimal value representing the currently activated high alarms associated with the module, where bit 0 corresponds to Analog Input channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that there are no activated high alarms associated with the channel. When the bit is 1, it denotes that there is an activated high alarm associated with the channel.
LL A two-digit hexadecimal value representing the currently activated low alarms associated with the module, where bit 0 corresponds to Analog Input channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that there are no activated low alarms associated with the channel. When the bit is 1, it denotes that there is an activated low alarm associated with the channel.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Note:
Digital Output channel 0 will be active if a currently activated high or low alarm is associated with Analog Input channel 0, and Digital Output channel 1 is associated with Analog Input channel 1 and so on.

Examples:
Command: @01RAO Response: !010100
Reads the currently activated alarms associated with module 01. The module returns a response indicating that the command was successful, with a value of 0100, which denotes that there is currently an activated high alarm associated with Analog Input channel 0.

Command: @01CHC0 Response: !01

Clears the status of the high alarms for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful.

Command: @01RAO Response: !010000

Reads the currently activated alarms associated with module 01 and returns a response indicating that the command was successful, and that no alarms have occurred.

Related Commands:

Section 2.42 @AACHCi, Section 2.45 @AACLCi, Section 2.47 @AADA, Section 2.51 @AAHI(Data)Ci, Section 2.52 @AALO(Data)Ci, Section 2.55 @AARHCi, Section 2.58 @AARLCi

2.54 @AARH

Description:
This command is used to read the high latch values for all Analog Input channels of a specified module.

Syntax:
@AARH[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RH The command to read the high latch values for all Analog Input channels

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The high latch values for all Analog Input channels. The data format will be the same as that set by the %AANNTTCCFF command (Section 2.1). See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RH Response:
!01+08.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the high latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful, with the data in Engineering Units format.

Command: @01CH Response: !01
Clears the high latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful.

Command: @01RH Response:
!01+00.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the high latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful, with the data in Engineering Units format.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.41 @AACH, Section 2.43 @AACHi, Section 2.56 @AARHi

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.55 @AARHCi

Description:
This command is used to read the high alarm limit for a specific Analog Input channel of a specified module.

Syntax:
@AARHCi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RH The command to read the high alarm limit for the Analog Input channel
Ci The Analog Input channel to be read, zero based (0-7)

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The high alarm limit for the specified Analog Input channel in Engineering Units format. See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01HI+08.000C0 Response: !01
Sets the high alarm limit for Analog Input channel 0 of module 01 to +08.000 (+8.0 V) and returns a response indicating that the command was successful.

Command: @01RHC0 Response: !01+08.000
Reads the high alarm limit for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, and that the high alarm limit is +08.000 (+8.0 V).

Command: @01RHCF Response: ?01
Attempts to read the high alarm limit for Analog Input channel 15 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.42 @AACHCi. Section 2.47 @AADA, Section 2.50 @AAEAT, Section 2.51 @AAHI(Data)Ci, Section 2.53 @AARAO

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.56 @AARHi

Description:
This command is used to read the high latch value for a specific Analog Input channel of a specified module.

Syntax:
@AARHi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RH The command to read the high latch value for the Analog Input channel
i The Analog Input channel to be read, zero based (0-7)

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The high latch value for the specified Analog Input channel. The data format will be the same as that set by the %AANNTTCCFF command (Section 2.1). See Section 1.9.4 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RH0 Response: !01+08.000
Reads the high latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +08.000 (8.0 V) in Engineering Units format.

Command: @01CH Response: !01
Clears the high latch value for all Analog Input channels of module 01 and returns a response indicating that the command was successful.

Command: @01RH0 Response: !01+00.000
Reads the high latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) in Engineering Units format.

Command: @01RHF Response: ?01
Attempts to read the high latch value for Analog Input channel 15 of module 01, but returns an response indicating that the command was unsuccessful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.41 @AACH, Section 2.43 @AACHi, Section 2.54 @AARH

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.57 @AARL

Description:
This command is used to read the low latch values for all Analog Input channels of a specified module.

Syntax:
@AARL[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RL The command to read the low latch values for all Analog Input channels

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The low latch values for all Analog Input channels. The data format will be the same as that set by the %AANNTTCCFF command (Section 2.1). See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RL Response:
!01-02.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000

Reads the low latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful, with the data in Engineering Units format.

Command: @01CL Response: !01

Clears the low latch values for all Analog Input channels of module 01 and returns a response indicating that the command was successful.

Command: @01RH Response:
!01+00.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000

Reads the low latch values from all Analog Input channels of module 01 and returns a response indicating that the command was successful, with the data in Engineering Units format signifying that the low latch values have been cleared.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.44 @AACL, Section 2.46 @AACLi, Section 2.59 @AARLi

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

2.58 @AARLCi

Description:
This command is used to read the low alarm limit for a specific Analog Input channel of a specified module.

Syntax:
@AARLCi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RL The command to read the low alarm limit for the Analog Input channel
Ci The Analog Input channel to be read, zero based (0-7)

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The low alarm limit for the specified Analog Input channel in Engineering Units format. See Section 1.9.4 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01LO-03.000C0 Response: !01
Sets the low alarm limit for Analog Input channel 0 of module 01 to -03.000 (-3.0 V) and returns a response indicating that the command was successful.

Command: @01RLC0 Response: !01-03.000
Reads the low alarm limit for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, and that the low alarm limit is -03.000 (-3.0 V).

Command: @01RLCF Response: ?01
Attempts to read the low alarm limit for Analog Input channel 15 of module 01, but returns a response indicating that the command was successful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.45 @AACLCi, Section 2.47 @AADA, Section 2.50 @AAEAT, Section 2.52 @AALO(Data)Ci, Section 2.53 @AARAO

Related Topics:

Section 1.9.3 Analog Input Type Codes and Data Format

2.59 @AARLi

Description:
This command is used to read the low latch value for a specific Analog Input channel of a specified module.

Syntax:
@AARLi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read in hexadecimal format (00 to FF)
RL The command to read the low latch value for the Analog Input channel
i The Analog Input channel to be read, zero based (0-7)

Response:
Valid Command: **!AA(Data)[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character to indicate a valid command
? Delimiter character to indicate an invalid command
AA The address of the responding module in hexadecimal format (00 to FF)
(Data) The low latch value for the specified Analog Input channel. The data format will be the same as that set by the %AANNTTCCFF command (Section 2.1). See Section 1.9.3 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: @01RL0 Response: !01-02.000
Reads the low latch value for Analog input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of -02.000 (-2.0 V) in Engineering Units format.

Command: @01CL0 Response: !01
Clears the low latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful.

Command: @01RL0 Response: !01+00.000
Reads the low latch value for Analog Input channel 0 of module 01 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) in Engineering Units format.

Command: @01RLF Response: ?01
Attempts to read the low latch value for Analog Input channel 15 of module 01, but returns an response indicating that the command was successful because Analog Input channel 15 does not exist.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.44 @AACL, Section 2.46 @AACLi, Section 2.57 @AARL

Related Topics:
Section 1.9.3 Analog Input Type Codes and Data Format

3. Modbus RTU Protocol

The Modbus protocol was developed by Modicon Inc., and was originally designed for Modicon controllers. Detailed information can be found at http://www2.schneider-electric.com/sites/corporate/en/products-services/automation-control/automation-control.page. You can also visit http://www.modbus.org for more valuable information.

Function Code	Description	Section
04 (0x04)	Reads the Analog Input channels	3.1
05 (0x05)	Writes to a single Digital Output channel	3.2
70 (0x46)	Reads/writes the module settings	3.3

Error Responses

Number	Description	Length	Value
00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	Function code + 0x80
02	Exception Code	1 Byte	01

Note:
If a CRC mismatch occurs, the module will not respond.

3.1 Function 04 (0x04) - Read the Analog Input Channels

This function code is used to read from contiguous Analog Input channels.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x04
02 - 03	Starting Channel	2 Bytes	0 to 3
04 - 05	Number of Input Channels (N)	2 Bytes	N, 1 to 8 (starting channel + N)

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x04
02	Byte Count	1 Byte	2 x N
03 -	Data from Input Channels	2 x N Bytes	Data is in either 2's complement hexadecimal format or Engineering Units format.

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x84
02	Exception Code	1 Byte	03: the (starting channel + the number of input channels) is out of range, or an incorrect number of bytes were received.

3.2 Function 05 (0x05) - Write to a Single Digital Output Channel

This function code is used to write to contiguous Digital Output channels.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x05
02 - 03	Starting Channel	2 Bytes	0 to 3
04 - 05	Data	2 Bytes	FF00h for ON and 0000h for OFF

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x04
02 - 03	Byte Count	1 Byte	0 to 3
04 - 05	Data	2 Bytes	FF00h for ON and 0000h for OFF

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x85
02	Exception Code	1 Byte	03: the (starting channel + the number of input channels) is out of range, or an incorrect number of bytes were received.

3.3 Function 70 (0x46) - Read/Write the Module Settings

This function code is used to either read or change the configuration settings for the module. The following sub-function codes are supported.

Sub-function Code	Description	Section
00 (0x00)	Reads the Name of the Module	3.3.1
04 (0x04)	Sets the Address of the Module	3.3.2
05 (0x05)	Reads the Communication Protocol Settings	3.3.3
06 (0x06)	Sets the Communication Protocol Settings	3.3.4
07 (0x07)	Reads the Type Code	3.3.5
08 (0x08)	Sets the Type Code	3.3.6
32 (0x20)	Reads the Firmware Version Information	3.3.7
37 (0x25)	Reads whether a Specific Channel is Enabled or Disabled	3.3.8
38 (0x26)	Sets a Specific Channel to either Enabled or Disabled	3.3.9
41 (0x29)	Reads the Miscellaneous Settings	3.3.10
42 (0x2A)	Writes the Miscellaneous Settings	3.3.11

If the sub-function code specified in the message is not supported, then the module will respond with an error code as per the table below:

Error Response

Number	Description	Length	Value
00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	02: Indicates an invalid sub-function code

3.3.1 Sub-function 00 (0x00) - Read the Name of the Module

This sub-function code is used to read the name of the module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x00

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x00
03 - 06	Module Name	4 Bytes	0x00 0x70 0x03 0x00

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: An incorrect number of bytes were received

3.3.2 Sub-function 04 (0x04) - Set the Address of the Module

This sub-function code is used to set the address of a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x04
02	New Address	1 Byte	1 to 247
04 - 06	Reserved	3 Bytes	0x00 0x00 0x00

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x04
03	The address that was set	1 Byte	0: OK Others: Error
04 - 06	Reserved	3 Bytes	0x00 0x00 0x00

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: The new address is out of range, reserved bits should be filled with zero, or an incorrect number of bytes were received

3.3.3 Sub-function 05 (0x05) - Read the Communication Protocol

This sub-function code is used to read the communication protocol currently being used by the module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x05
03	Reserved	1 Byte	0x00

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x05
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	The Baud Rate code, see Section 1.9.1 for details.
05	Reserved	1 Bytes	0x00
06	Parity	1 Byte	The parity code, see Section 1.9.1 for details.
07	Reserved	1 Bytes	0x00
08	Protocol	1 Byte	0: DCON 1: Modbus RTU
09 - 10	Reserved	2 Bytes	0x00 0x00

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: Reserved bits should filled with zero, or an incorrect number of bytes were received

3.3.4 Sub-function 06 (0x06) - Set the Communication Protocol

This sub-function code is used to set the communication protocol to be used by the module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	The Baud Rate code, see Section 1.9.1 for details.
05	Reserved	1 Bytes	0x00
06	Parity	1 Byte	The parity code, see Section 1.9.1 for details.
07	Reserved	1 Bytes	0x00
08	Protocol	1 Byte	0: DCON 1: Modbus RTU
09 - 10	Reserved	2 Bytes	0x00 0x00
11	Reserved	1 Bytes	0x00

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0: OK Others: Error
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0: OK Others: Error
07	Reserved	1 Byte	0x00
08	Protocol	1 Byte	0: OK Others: Error
09 - 10	Reserved	2 Bytes	0x00 0x00

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: The Baud Rate or the module address is out of range, reserved bits should be filled with zero, or an incorrect number of bytes were received

3.3.5 Sub-function 07 (0x07) - Read the Analog Input Type Code

This sub-function code is used to read the Type Code information for a specific Analog Input channel of a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x07
03	Reserved	1 Byte	0x00
04	Channel Number	1 Byte	0x00 to 0x07

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x07
03	Type Code	1 Byte	The Type Code, see Section 1.9.3 for details.

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: The channel number is out of range reserved bits should be filled with zero, or an incorrect number of bytes were received

3.3.6 Sub-function 08 (0x08) - Set the Analog Input Type Code

This sub-function code is used to set the Type Code for a specific Analog Input channel of a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x08
03	Reserved	1 Byte	0x00
04	Channel Number	1 Byte	0x00 to 0x07
05	Type Code	1 Byte	The Type Code, see Section 1.9.3 for details.

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x08
03	Type Code	1 Byte	0: OK Others: Error

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: The Type Code is out of range, the channel number is out of range, reserved bits should be filled with zero, or an incorrect number of bytes were received

3.3.7 Sub-function 32 (0x20) - Read the Firmware Version Information

This sub-function code is used to read the firmware version information for a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x20

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x20
03	Major Version	1 Byte	0x00 to 0xFF
04	Minor Version	1 Byte	0x00 to 0xFF
05	Reserved	1 Byte	0x00
06	Build Version	1 Byte	0x00 to 0xFF

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: Reserved bits should be filled with zero, or an incorrect number of bytes were received.

3.3.8 Sub-function 37 (0x25) - Read whether an Analog Input Channel is Enabled or Disabled

This sub-function code is used to read whether each channel of a module is enabled or disabled.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x25

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x25
03	Status (Enabled/disabled)	1 Byte	0x00 to 0xFF, the status of each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is disabled and 1 denotes that the channel is enabled.

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: An incorrect number of bytes were received

3.3.9 Sub-function 38 (0x26) – Enable or Disable the Analog Input Channels

This sub-function code is used to specify which channels of a module are to be enabled or disabled.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x26
03	Enabled/disabled Settings	1 Byte	0x00 to 0xFF, the settings for each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is to be disabled and 1 denotes that the channel is to be enabled.

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub function Code	1 Byte	0x26
03	Enabled/disabled Settings	1 Byte	0: OK Others: Error.

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: The enabled/disabled settings are out of range, or an incorrect number of bytes were received

3.3.10 Sub-function 41 (0x29) - Read the Miscellaneous Settings

This sub-function code is used to read the miscellaneous settings for a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x29

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x29
03	Miscellaneous Settings	1 Byte	Bit 7: Filter Settings 0: 60 Hz Rejection 1: 50 Hz Rejection Bit 6: Reserved Bit 5: Mode Settings 0: Normal Mode 1: Fast Mode Bits 4 to 0: Reserved

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: An incorrect number of bytes were received

3.3.11 Sub-function 42 (0x2A) - Write the Miscellaneous Settings

This sub-function code is used to set the miscellaneous settings for a module.

Request

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x2A
03	Miscellaneous Settings	1 Byte	Bit 7: Filter Settings 0: 60 Hz Rejection 1: 50 Hz Rejection Bit 6: Reserved Bit 5: Mode Settings 0: Normal Mode 1: Fast Mode Bits 4 to 0: Reserved

Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0x46
02	Sub-function Code	1 Byte	0x2A
03	Miscellaneous Settings	1 Byte	0: OK Others: Error

Error Response

00	Address	1 Byte	1 to 247
01	Function Code	1 Byte	0xC6
02	Exception Code	1 Byte	03: Reserved bits should be filled with zero, or an incorrect number of bytes were received

3.4 Address Mappings

The Modbus address mappings for the M-7003 is as follows:

Address	Description	Attribute
00001 ~ 00004	Reads the current status of the Digital Output or sets the Digital Output to either active or inactive	R/W
00129 ~ 00132	Reads/sets the Digital Output Safe Value	R/W
00161 ~ 00164	Reads/sets the Digital Output Power-on Value	R/W
00257	Reads/sets the Communication Protocol 0: DCON 1:Modbus RTU	R/W
00259	Reads/sets the Filter Settings 0: 60 Hz Rejection 1: 50 Hz Rejection	R/W
00260	Reads/sets the Host Watchdog Mode 0: The same as for I-7000 series modules 1: Analog Output and Digital Output commands can be used to clear the status of the Host Watchdog timeout	R/W
00261	Enables or disables the Host Watchdog, or reads the status of the Host Watchdog 0: Disable 1: Enable	R/W
00262	Enables or disables the Analog Input Alarm 0: Disable 1: Enable	R/W
00263	Reads/sets the Analog Input Alarm mode 0: Momentary 1: Latched	R/W
00264	Clears the latched Digital Input and Digital Output channels. Write 1 to clear	W
00269	Reads/sets the Modbus Analog Input Data Format 0: 2’s Complement Hexadecimal 1: Engineering Units	R/W
00270	Reads/resets the status of the Host Watchdog	R/W

	timeout. Write 1 to clear.	
00271	Reads/sets the Analog Input Filter Format 0: Normal 1: Fast	R/W
00272	Loads the factory calibration parameters. Write 1 to load.	W
00273	Reads the Reset status 0: This is **NOT** the first time the module has been read since being powered on 1: This is the first time the module has been read since being powered on	R
00279	Sets the active Digital Output mode: 0: Normal 1: Inverse	W
00280	Clears the high latch values for all Analog Input channels. Write 1 to clear.	W
00281	Clears the low latch values for all Analog Input channels. Write 1 to clear.	W
00284	Enables or disables calibration, or reads the status of the calibration function 0: Disabled 1: Enabled	R/W
00513 ~ 00520	Clears the high latched Analog Input for channels 0 to 7. Write 1 to clear.	W
00545 ~ 00552	Clears the low latched Analog Input for channels 0 to 7. Write 1 to clear.	W
00705 ~ 00712	Reads/clears the status of the high alarm for channels 0 to 7. Write 1 to clear.	R/W
00737 ~ 00744	Reads/clears the status of the low alarm for channels 0 to 7. Write 1 to clear.	R/W
10073 ~ 10076	Reads the status of the high latched Digital Output channels	R
10105 ~ 10108	Reads the status of the low latched Digital Output channels	R
30001 ~ 30008	Reads the Analog Input value for channels 0 to 7	R
30513 ~ 30520	Reads the high latch value for Analog Input channels 0 to 7	R

30545 ~ 30552	Reads the low latch value for Analog Input channels 0 to 7	R
40257 ~ 40264	Reads/writes the Type Code for Analog Input channels 0 to 7	R/W
40481	Reads the Firmware Version (high word)	R
40482	Reads the Firmware Version (low word)	R
40483	Reads the Name of the Module (high word)	R
40484	Reads the Name of the Module (low word)	R
40485	Reads the Module address. The valid range is 0x1 to 0xF7	R
40486	Reads/writes the Baud Rate (Section 1.9.1 for details)	R/W
40488	Reads/writes the Response Delay Time in ms. The valid range is 0 to 30ms.	R/W
40489	Reads/writes the Host Watchdog Timeout value in 0.1s increments. The valid range is 0 to 255.	R/W
40490	Reads the status of the Analog Input channel, or sets the Analog Input channel to enabled or disabled.	R/W
40492	Reads/clears the Host Watchdog Timeout count. Write 0 to clear.	R/W
40494	Reads/sets the threshold value for Analog Input Type 4~20mA. Bit 7:0 Threshold Value Bit 8 Enables/Disables the threshold function Bit 15:9 Reserved	R/W
40502	Reads/sets the reset time after receive a command.	R/W
40577 ~ 40584	Reads/writes the Analog Input high alarm value	R/W
40609 ~ 40616	Reads/writes the Analog Input low alarm value	R/W
40865	Sets the Analog Input span/zero calibration. 0x5A45: Zero Calibration 0x5350: Span Calibration	W

Note:

The command to load the factory calibration parameters (00272) takes about 3 seconds to be processed. Subsequent commands should not be sent before this time has elapsed.

3.5 Engineering Units Data Format Table

The Modbus protocol supports the Engineering Units data format, and the Type Code information is as follows.

Type Code	Analog Input Type	-F.S.	+F.S.
07	+4 to +20 mA	4000	20000
08	-10 to +10 V	-10000	10000
09	-5 to +5 V	-5000	5000
0A	-1 to +1 V	-10000	10000
0B	-500 to +500 mV	-5000	5000
0C	-150 to +150 mV	-15000	15000
0D	-20 to +20 mA	-20000	20000
1A	0 to +20 mA	0	20000

The under-range value is –32768 and the over-range value is +32767. For details of the hexadecimal data format, refer to Section 1.9.4 and Section 1.9.5.

4. Troubleshooting

If you are having difficulties using your M-7003 module, here are some suggestions that may help. If you cannot find the answers you need in this guide, contact ICP DAS Product Support.

4.1 Communicating with the Module

If you attempt to communicate with the module and receive no response, first check the following:

- ☐ Ensure that the supplied power is within the range of +10 to +30 V_{DC}. If the supplied power is sufficient, then the power LED should be on.
- ☐ When the module receives a command, the power LED will be set to "OFF". The power LED will be shown as "ON" after the module responds. This method can be used to check whether or not the module has received a command sent from the host.
- ☐ If possible, use another device that is known to be functional to check whether the host can communicate with the device through the same network.
- ☐ If the host is a PC installed with a Windows operating system, execute the DCON Utility to determine whether the module can be found. The DCON Utility can be downloaded from the ICP DAS website at http://www.icpdas.com. Documentation for the DCON Utility can be found in the "Getting Started for I-7000 Series Modules" manual.
- ☐ Set the module to "INIT mode" and communicate with the module using address 00 and the DCON protocol. See Section 1.9 for more details related to configuration settings.

4.2 Reading Data

If the data read from the input channel is not correct, first check the following:

- ☐ Ensure that the Type Code and data format settings are correct. The Type Code is set by using the $AA7CiRrr command (see Section 2.12). The data format is set by using the %AANNTTCCFF command (see Section 2.1). If you are using the Modbus RTU protocol, the Type Code is set by using sub-function 08h of the 46h function.
- ☐ If the voltage read by the module is incorrect, it may be because the calibration parameters stored in the non-volatile memory are corrupted. You can calibrate the module by yourself, but be sure to read the information contained in Section 1.8 before performing any calibration. Use the $AAS1 command (Section 2.21) to reload the factory calibration parameters.

5. Appendix

5.1 INIT Mode

Each I-7000 and M-7000 module contains a built-in EEPROM memory that can be used to store configuration information, such as the module address, the Type Code, and the Baud Rate, etc. Occasionally, the configuration of a module may be forgotten and there may be no visual indications of the configuration of the module. It is difficult to communicate with the module when the configuration of the module is unknown. To help avoid this problem, the I-7000 and M-7000 series has a special mode called the “INIT mode”. When the module is powered on in “INIT mode” the configuration of the module is reset to the default settings shown below, allowing it to be operated as normal.

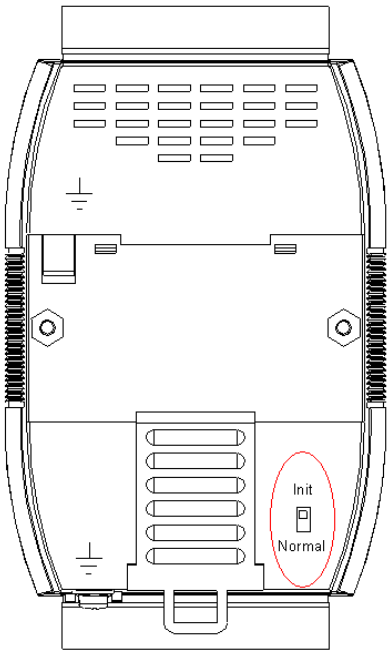
1. Address: 00
2. Baud Rate: 9600 bps
3. No checksum
4. Protocol: DCON

The configuration information stored in the EEPROM is not changed and can be read by sending the $AA2(CR) command (see Section 2.7) at 9600 bps.

There are also other commands that require the module to set to INIT mode before being used. They are:

1. %AANNTTCCFF, which is used when changing the Baud Rate and the checksum settings. See Section 2.1 for details.
2. $AAPN, see Section 2.20 for details.

Originally, INIT mode was accessed by connecting the INIT* terminal to the GND terminal. Newer I-7000 and M-7000 modules include an INIT switch located on the rear of the module to allow easier access to INIT mode. For these modules, INIT mode is accessed by sliding the INIT switch to the Init position, as shown below.



INIT Switch

5.2 Dual Watchdog Operation

Dual Watchdog = Module Watchdog + Host Watchdog

The Module Watchdog is a hardware reset circuit that monitors the operating status of the module. While working in harsh or noisy environments, the module may be shut down by external signals. The Watchdog circuit allows the module to operate continuously without disruption.

The Host Watchdog is a software function that monitors the operating status of the host. Its purpose is to prevent problems due to network/communication errors or host malfunctions. When a Host Watchdog timeout occurs, the module will reset all outputs to a safe state in order to prevent the controlled target from performing any erroneous operations.

M-7000 series modules include an internal Dual Watchdog, making the control system more reliable and stable.

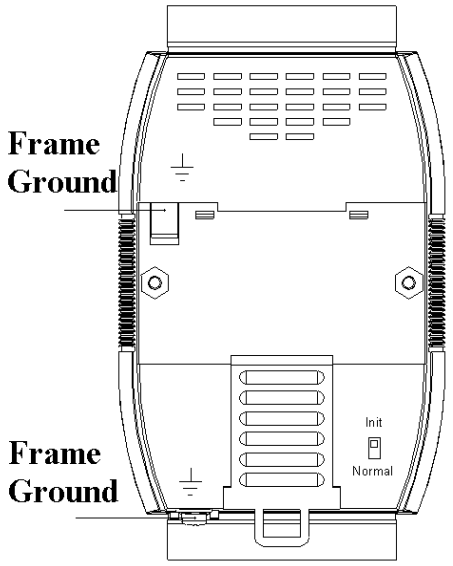
For more information regarding the Dual Watchdog, refer to Chapter 5 of the “Getting Started for M-7003 Modules” manual that can be downloaded from the ICP DAS website

5.3 Frame Ground

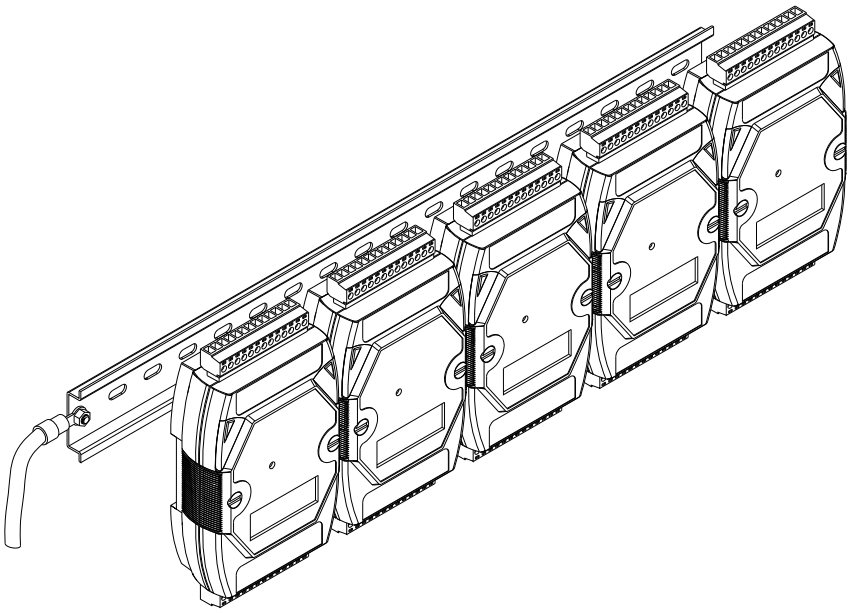
Electronic circuits are constantly vulnerable to Electrostatic Discharge (ESD), which becomes worse in a continental climate area. Some I-7000 and M-7000 modules feature a new design for the frame ground, which provides a path for bypassing ESD, allowing enhanced static protection (ESD) capabilities and ensures that the module is more reliable.

Either of the following options will provide better protection for the module:

1. If the module is DIN-Rail mounted, connect the DIN-Rail to the earth ground. This is because the DIN-Rail is in contact with the upper frame ground, as shown in the figure below.
2. Alternatively, connect the lower frame ground terminal to a wire and connect the wire to the earth ground, as shown in the figure below.

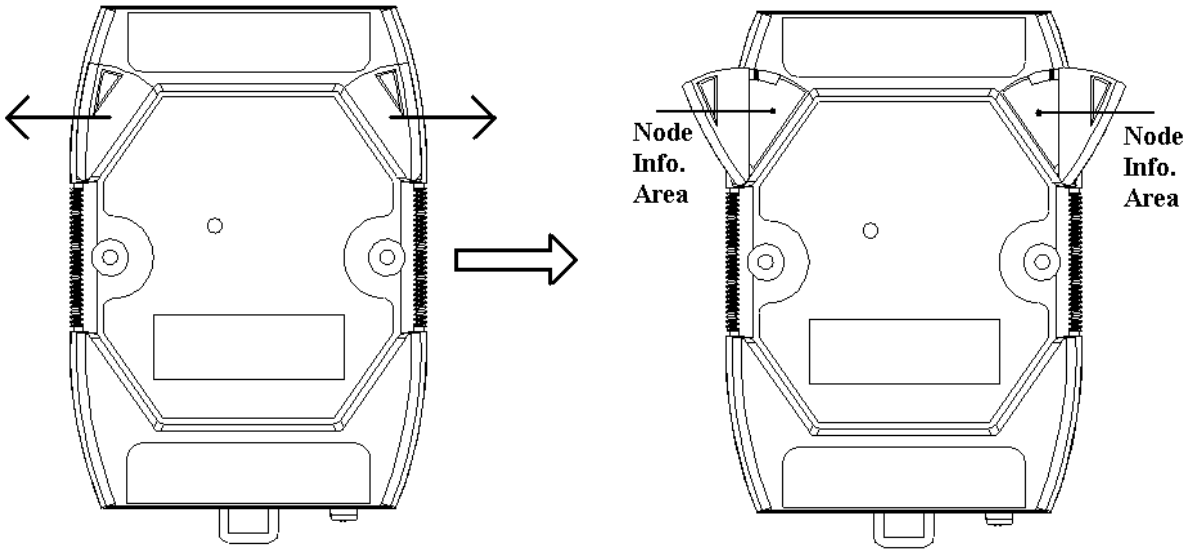


New DIN-Rail models are available that can easily be connected to the earth ground. Each rail is made of stainless steel, which is stronger than those made of aluminum. There is a screw at one end and a ring terminal is included, as shown in the figure below.



5.4 Node Information Area

Each I-7000 and M-7000 module has a built-in EEPROM that can be used to store configuration information, such as the module address, the Type Code, and the Baud Rate, etc. One minor drawback is that there may be no visual indications of the configuration of the module. Newer I-7000 and M-7000 modules include “Node Information” areas that are protected by a cover, as shown below, and can be used to make a written record of the node information, such as the module address and the Baud Rate, etc. To access the node information areas, first slide the covers outward, as shown in the figure below.



5.5 Reset Status

The reset status of a module is set when the module is first powered on, or when the module is reset by the Module Watchdog, and is cleared after responding to the first $AA5 command (see Section 2.9). This command can be used to check whether the module has recently been reset. If the response from the $AA5 command indicates that the reset status has been cleared, it means that the module has not been reset since the last $AA5 command was sent. If the response from the $AA5 command indicates that the reset status is set and it is not the first time an $AA5 command has been sent, it means that the module has been reset and the Digital Output value has been changed to either the default power-on value or the safe value.