M-2018-16 and M-6018-16 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, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Date: 2024/3/27

Table of Contents

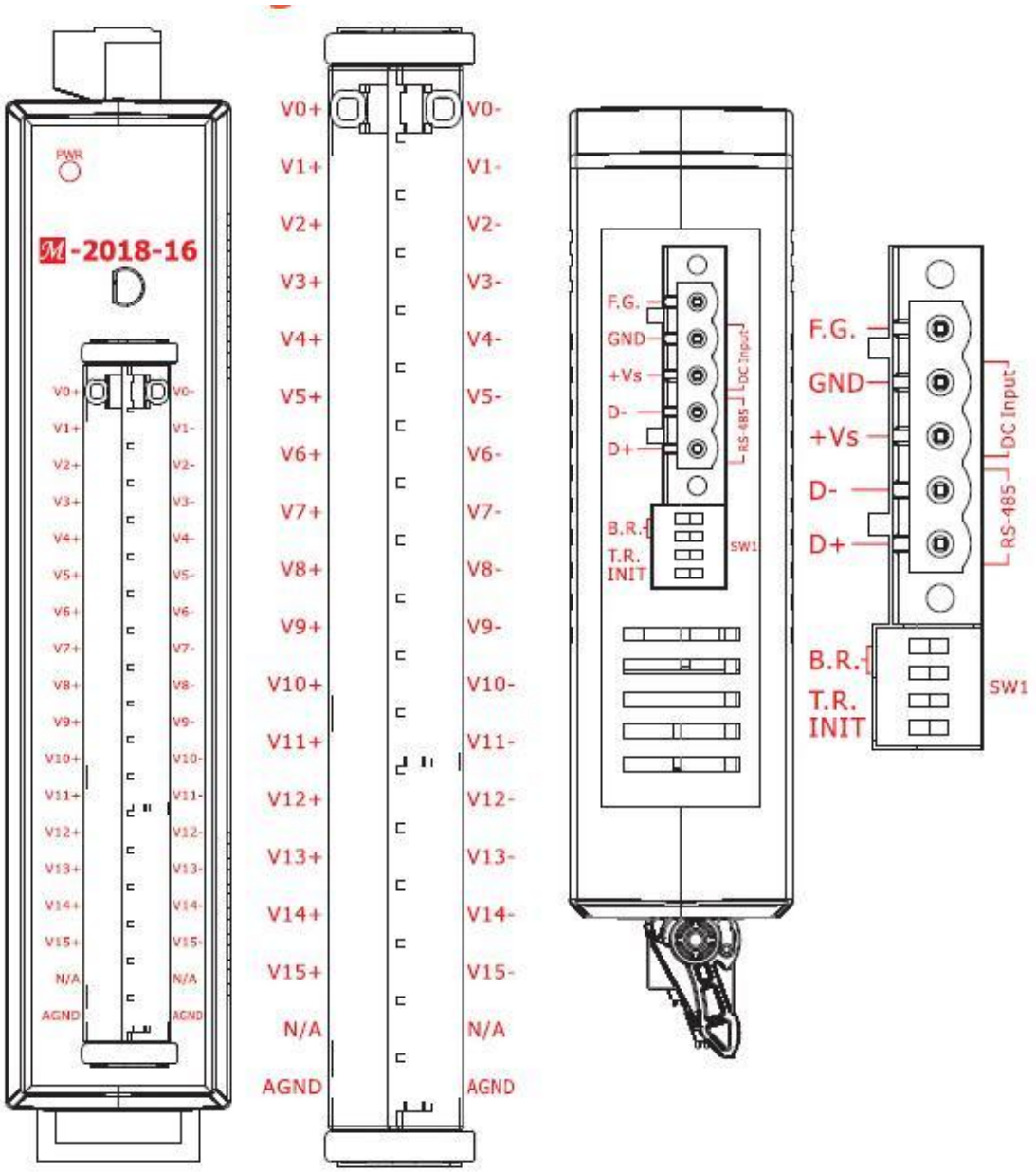
1. Introduction

The M-2000 and M-6000 series is a family of network data acquisition and control modules, providing analog-to-digital, digital-to-analog, digital input/output, timer/counter and other functions. The modules can be remotely controlled using the DCON and Modbus RTU protocols. Communication between the module and the host is via an RS-485 bi-directional serial bus standard. Baud Rates are software programmable and transmission speeds of up to 115.2K baud can be selected.

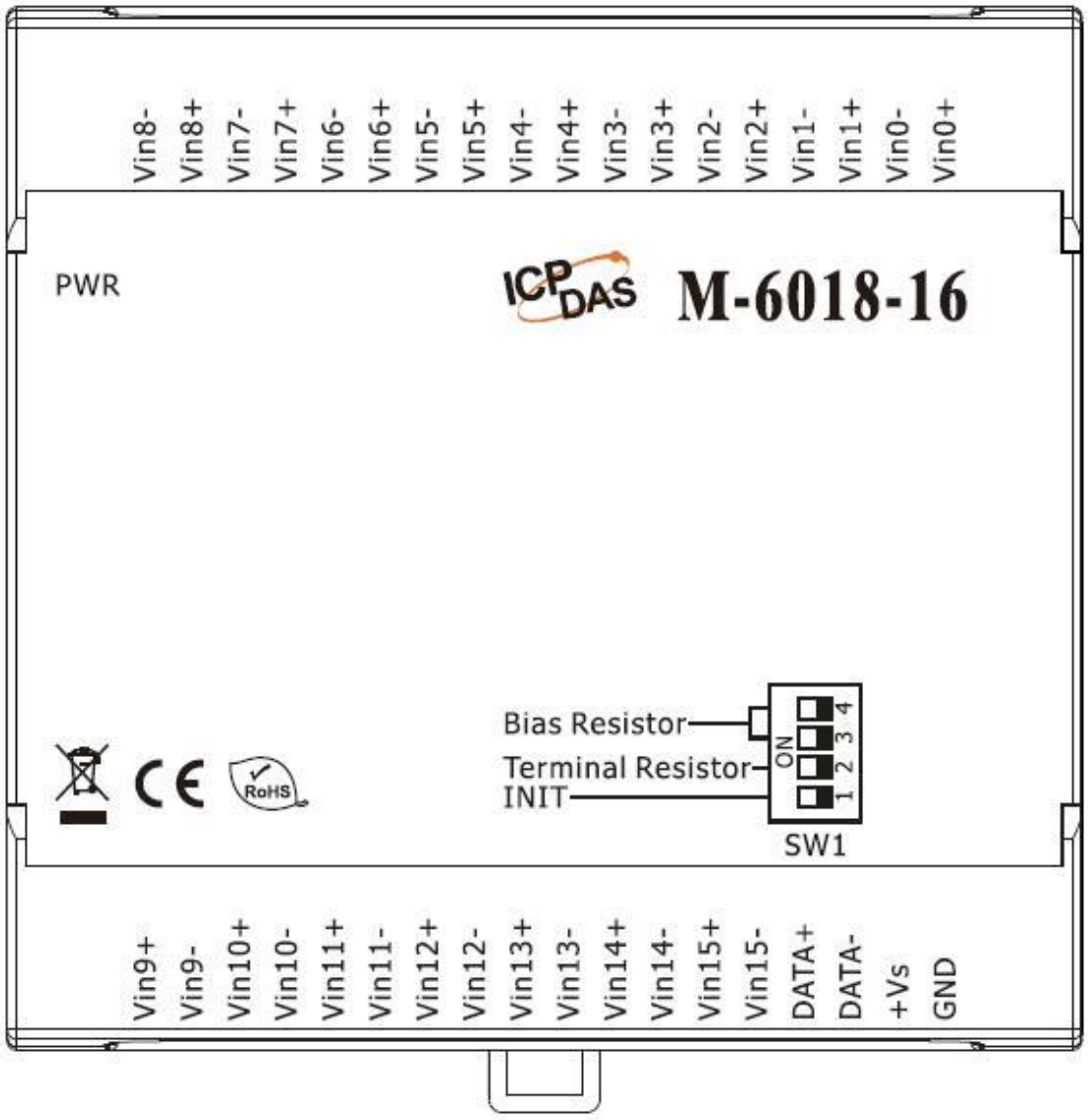
The M-2018-16 and M-6018-16 are analog input modules that include 16 differential analog input channels. The voltage input range can be from ±15 mV to ±2.5 V, the current input range can be either +4 to +20 mA, 0 to +20 mA, or ±20 mA, and types J, K, T, E, R, S, B, N, C, L, M, LDIN43710 thermocouple can be used for the thermocouple input. Overvoltage protection of up to 120 VDC is provided. The module also features per-channel open wire detection for the thermocouple input types, and provides 4 kV ESD protection as well as 3000 VDC intra-module isolation.

1.1 Terminal Assignment

M-2018-16



M-6018-16


Vin8-
Vin8+
Vin7-
Vin7+
Vin6-
Vin6+
Vin5-
Vin5+
Vin4-
Vin4+
Vin3-
Vin3+
Vin2-
Vin2+
Vin1-
Vin1+
Vin0-
Vin0+
PWR
ICP DAS
M-6018-16
Bias Resistor
Terminal Resistor
INIT
ON
1 2 3 4
SW1
RoHS
Vin9+
Vin9-
Vin10+
Vin10-
Vin11+
Vin11-
Vin12+
Vin12-
Vin13+
Vin13-
Vin14+
Vin14-
Vin15+
Vin15-
DATA+
DATA-
+Vs
GND

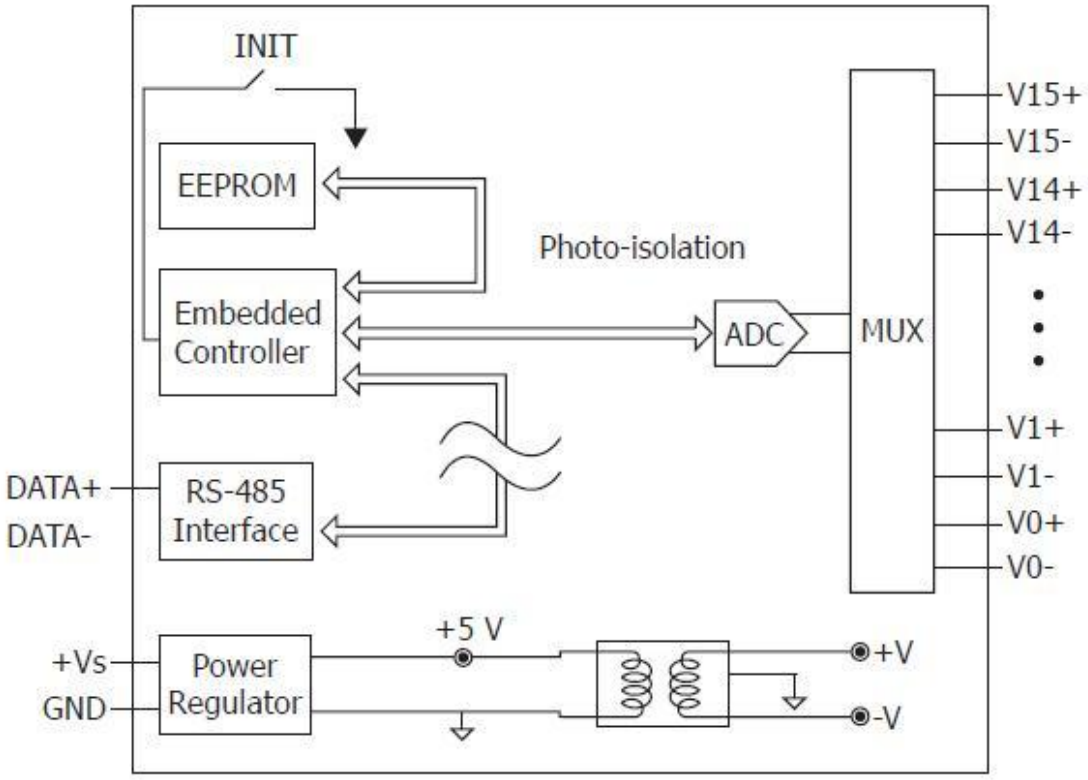
1.2 Specifications

	M-2018-16	M-6018-16	
Analog Input			
Input Channels	16 differential	16 differential	
Input Type	mV, V, mA[*1]	mV, V, mA[*1]	
Thermocouple Type	J, K, T, E, R, S, B, N, C	J, K, T, E, R, S, B, N, C	
Sampling Rate	10 samples/sec	10 samples/sec	
Accuracy	±0.1%	±0.1%	
Input Impedance	> 400KΩ	> 400KΩ	
Voltage overload Protection	120V	120V	
Isolation	3000V DC	3000V DC	
Open Wire Detection	Yes	Yes	
Modbus RTU	Yes	Yes	
Power			
Requirement	+10 to +48V DC	+10 to +48V DC	
Consumption	0.5W	0.5W	
Temperature Range			
Operating	-25℃ to +75℃	-25℃ to +75℃	
Storage	-40℃ to +85℃	-40℃ to +85℃	
*1: requires optional external 125 ohm resistor			

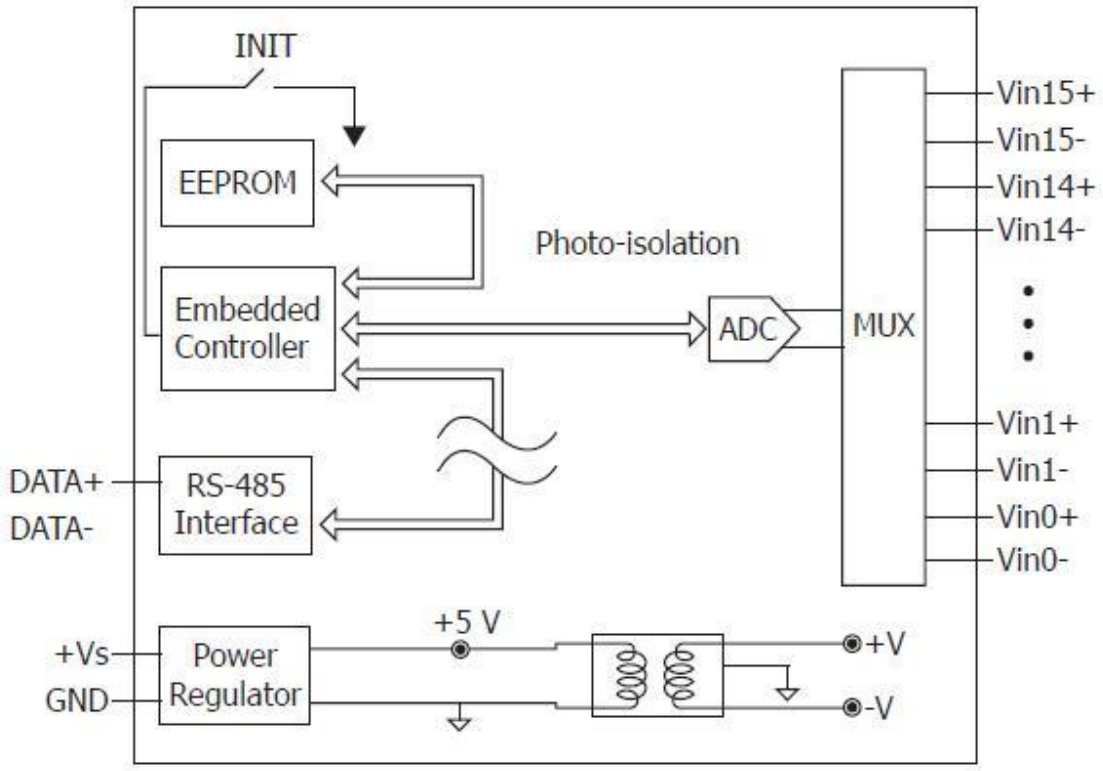
Note: A warm up period of 30 minutes is recommended in order to achieve the complete performance results described in the specifications.

1.3 Block Diagrams

1.3.1 Block diagram for the M-2018-16

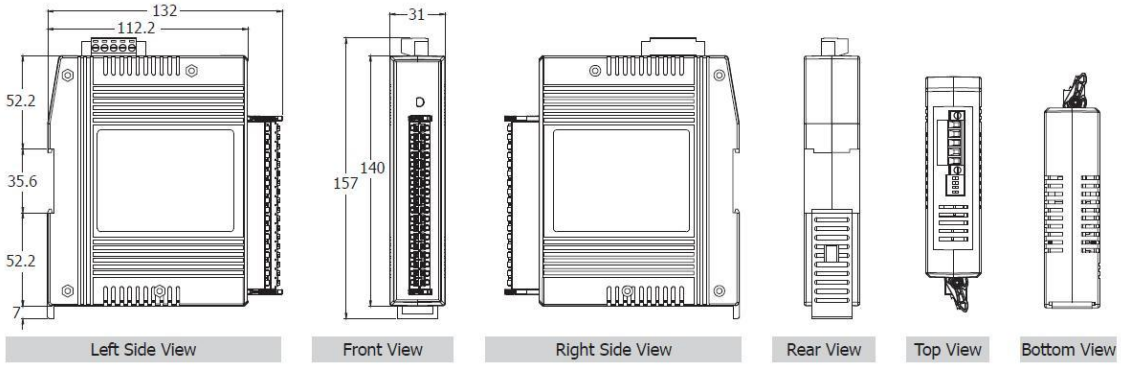


1.3.2 Block diagram for the M-6018-16

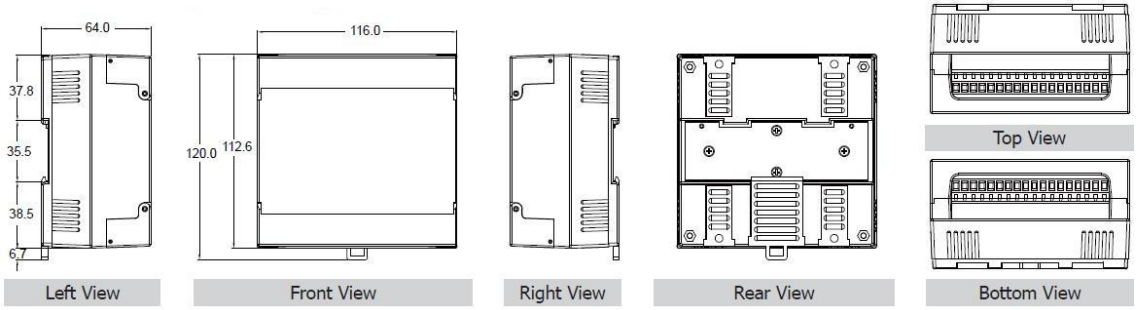


1.4 Dimensions

1.4.1 M-2018-16

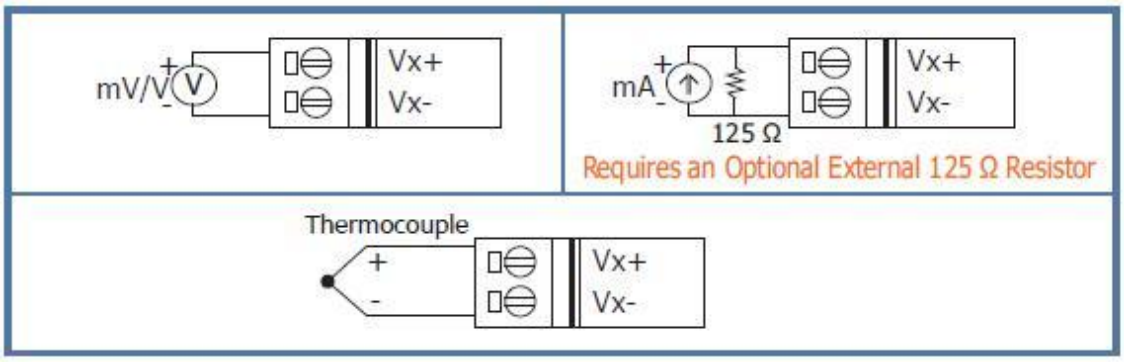


1.4.2 M-6018-16

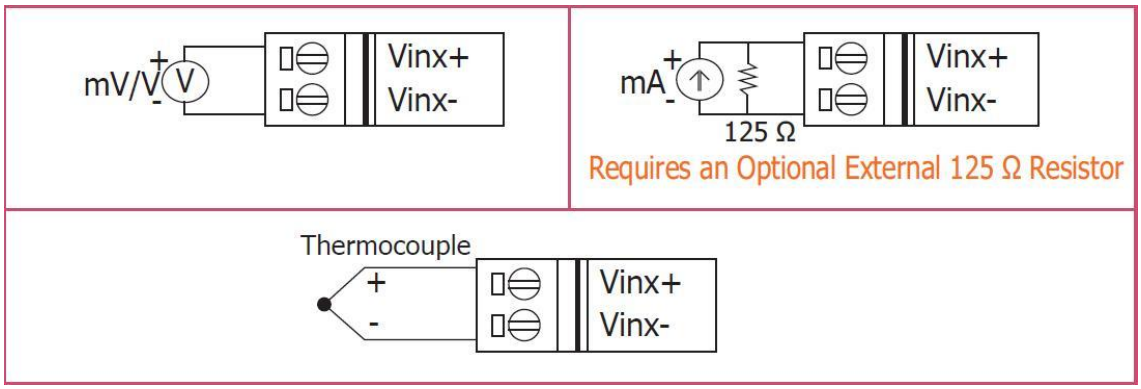


1. 5 Wiring Diagrams

1.5.1 Wiring diagram for the M-2018-16



1.5.2 Wiring diagram for the M-6018-16



1.6 Quick Start

To install the module, follow the steps below:

1. Connect the analog input. See Section 1.1 for the terminal assignment and Section 1.5 for the wiring diagram.
2. Connect the module to the RS-485 network using the DATA+ and DATA- terminals. See Section 1.1 for the terminal assignment. If the host is only equipped with an RS-232 interface, then an RS-232 to RS-485 converter will be required. Refer to the "**I-7000 Bus Converter User's Manual**" for more information.
3. Connect the module to the power supply using the +Vs and GND terminals. See Section 1.1 for the terminal assignment. Note that the voltage supplied should be in the range of +10 to +48V DC.
4. For DCON protocol, configure the module by sending the %AANNTTCCFF command. See Section 2.1 for details. For Modbus RTU protocol, configure the module using the following functions.
   - Sub-function 04h of Function 46h, see Section 3.3.2
   - Sub-function 06h of Function 46h, see Section 3.3.4
   - Sub-function 08h of Function 46h, see Section 3.3.6

   The default settings for the module can be found in Section 1.7.
5. For DCON protocol, in order to read data from the input channels, send either the #AA or #AAN command to the module. See Sections 2.2 and 2.3 for details. For Modbus RTU protocol, use Function 04h to read the data from the input channels. See Section 3.2 for details.
6. If the host is a PC with a Windows operating system installed, the DCON Utility can be used to allow easy

configuration and reading of data. The DCON Utility can be downloaded from the ICP DAS website (http://www.icpdas.com). The documentation for the DCON Utility can be found in the ”**Getting Started For I-7000 Series Modules**” manual.

Refer to the “**I-7000 Bus Converter User’s Manual**” and ”**Getting Started For I-7000 Series Modules**” manuals for more details. The “**Getting Started For I-7000 Series Modules**” manual can be downloaded from the ICP DAS website (http://www.icpdas.com).

1.7 Default Settings

Default settings for the M-2018-16 and M-6018-16 modules are:

- Protocol: Modbus RTU
- Module address: 01
- Analog input type: Type 05, -2.5V to 2.5V
- Baud Rate: 9600 bps
- Filter set at 60Hz rejection

1.8 Calibration

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

The calibration procedure is as follows:

1. Warm up the module for 30 minutes.
2. Set the type code to the type you want to calibrate. Refer to Sections 2.1 for details.
3. Enable calibration. Refer to Section 2.18 for details.
4. Apply the zero calibration voltage/current.
5. Send the zero calibration command. Refer to Section 2.5 for details.
6. Apply the span calibration voltage/current.
7. Send the span calibration command. Refer to Section 2.4 for details.
8. Repeat steps 3 to 7 three times.

Notes:

1. For the M-2018-16 and M-6018-16, connect the calibration voltage/current to channel 0.
2. When calibrating type 06 for the M-2018-16 and M-7018-16, a resistor of 125 ohms, 0.1% should be connected. Refer to Section 1.5 for details.
3. Calibration voltages and currents are shown below.
4. The M-2018-16 and M-6018-16 modules must be switched to the DCON protocol mode before calibrating. Refer to Section 3.3.4 for details of the switching protocol.

Calibration voltages/current used by the M-2018-16 and M-6018-16:

Type Code	00	01	02	03	04	05	06
Zero Input	0mV	0mV	0mV	0mV	0V	0V	0mA
Span Input	+15mV	+50mV	+100mV	+500mV	+1V	+2.5V	+20mA

1.9 Configuration Tables

Baud Rate Setting (CC)

7	6	5	4	3	2	1	0
Data		Baud					

Key	Description
Baud	Baud Rate 03: 1200 04: 2400 05: 4800 06: 9600 07: 19200 08: 38400 09: 57600 0A: 115200
Data	Data Format 0: N81 1: N82 2: E81 3: O81

Analog Input Type Setting (TT)

Type Code	Analog Input Type	Range
00	+/-15mV	-15mV ~ 15mV
01	+/-50mV	-50mV ~ 50mV
02	+/-100mV	-100mV ~ 100mV
03	+/-500mV	-500mV ~ 500mV
04	+/-1V	-1V ~ 1V
05	+/-2.5V	-2.5V ~ 2.5V
06	+/-20mA	-20mA ~ 20mA
07	+4 to +20mA	4mA ~ 20mA
0E	Type J Thermocouple	-210℃ ~ 760℃
0F	Type K Thermocouple	-270℃ ~ 1372℃
10	Type T Thermocouple	-270℃ ~ 400℃
11	Type E Thermocouple	-270℃ ~ 1000℃
12	Type R Thermocouple	0℃ ~ 1768℃
13	Type S Thermocouple	0℃ ~ 1768℃
14	Type B Thermocouple	0℃ ~ 1820℃
15	Type N Thermocouple	-270℃ ~ 1300℃
16	Type C Thermocouple	0℃ ~ 2320℃
17	Type L Thermocouple	-200℃ ~ 800℃
18	Type M Thermocouple	-200℃ ~ 100℃
19	Type L DIN43710 Thermocouple	-200℃ ~ 900℃
1A	0 to +20mA	0 ~ 20mA

Data Format Setting (FF)

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

Key	Description
DF	Data format 00: Engineering unit 01: % of FSR (full scale range) 10: 2’s complement hexadecimal
CS	Checksum settings 0: Disabled 1: Enabled
FS	Filter settings 0: 60Hz rejection 1: 50Hz rejection

Note: The reserved bits should be zero.

Analog Input Type and Data Format Table

Type code	Input Type	Data Format	+F.S	-F.S.
00	-15 to +15 mV	Engineering unit	+15.000	-15.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
01	-50 to +50 mV	Engineering unit	+50.000	-50.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
02	-100 to +100 mV	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
03	-500 to +500 mV	Engineering unit	+500.00	-500.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
04	-1 to +1 V	Engineering unit	+1.0000	-1.0000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
05	-2.5 to +2.5 V	Engineering unit	+2.5000	-2.5000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
06	-20 to +20 mA	Engineering unit	+20.000	-20.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
07	+4 to +20 mA	Engineering unit	+20.000	+04.000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000
0E	Type J Thermocouple -210 ~ 760℃	Engineering unit	+760.00	-210.00
		% of FSR	+100.00	-027.63
		2's comp HEX	7FFF	DCA2
0F	Type K Thermocouple -270 ~ 1372℃	Engineering unit	+1372.0	-0270.0
		% of FSR	+100.00	-019.68
		2's comp HEX	7FFF	E6D0

Type code	Input Type	Data Format	+F.S	-F.S.
10	Type T Thermocouple -270 ~ 400℃	Engineering unit	+400.00	-270.00
		% of FSR	+100.00	-067.50
		2's comp HEX	7FFF	A99A
11	Type E Thermocouple -270 ~ 1000℃	Engineering unit	+1000.0	-0270.0
		% of FSR	+100.00	-027.00
		2's comp HEX	7FFF	DD71
12	Type R Thermocouple 0 ~ 1768℃	Engineering unit	+1768.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
13	Type S Thermocouple 0 ~ 1768℃	Engineering unit	+1768.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
14	Type B Thermocouple 0 ~ 1820℃	Engineering unit	+1820.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
15	Type N Thermocouple -270 ~ 1300℃	Engineering unit	+1300.0	-0270.0
		% of FSR	+100.00	-020.77
		2's comp HEX	7FFF	E56B
16	Type C Thermocouple 0 ~ 2320℃	Engineering unit	+2320.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
17	Type L Thermocouple -200 ~ 800℃	Engineering unit	+800.00	-200.00
		% of FSR	+100.00	-025.00
		2's comp HEX	7FFF	E000
18	Type M Thermocouple -200 ~ 100℃	Engineering unit	+100.00	-200.00
		% of FSR	+050.00	-100.00
		2's comp HEX	4000	8000
19	Type L DIN43710 Thermocouple -200 ~ 900℃	Engineering unit	+900.00	-200.00
		% of FSR	+100.00	-022.22
		2's comp HEX	7FFF	E38E
1A	0 to +20 mA	Engineering unit	+20.000	+00.000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000

Thermocouple Over Range/Under Range Reading for the M-2018-16 and M-6018-16 with DCON protocol

	Over Range	Under Range
Engineering Unit	+9999.9	-9999.9
% of FSR	+999.99	-999.99
2’s Complement HEX	7FFF	8000

Thermocouple Over Range/Under Range Reading for the M-2018-16 and M-6018-16 with Modbus RTU protocol

Over Range	Under Range
7FFFh	8000h

4 ~ 20mA Under Range Reading

	Modbus RTU	DCON
Engineering Unit	-32768	-9999.9
% of FSR		-999.99
2’s Complement HEX	0000h	0000

1.10 M-2000 and M-6000 Notes

The M-2000 and M-6000 series support both the DCON and Modbus RTU protocols and the Modbus RTU communication protocol is the default protocol. The communication Baud Rates for the Modbus RTU protocol can be in the range of 1200 bps to 115200 bps.

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

1.10.1 Protocol Switching

To switch to the DCON protocol:

1. Sets the Modbus register 00257, base 1, to a value of 0.
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 INIT switch should be set to the on position. See Section 2.15 for details.
2. After a power-on reset, the communication protocol will be changed to the Modbus RTU protocol.

1.10.2 INIT Mode

When the module is powered on, with the INIT switch set to the on position, the module is in INIT mode, (see Section A.1 for details), 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 the INIT mode and use the above settings to communicate with the module. To read the current settings, send the commands $AA2, (see Section 2.6), and $AAP, (see Section 2.14). To set new settings, send the commands %AANNTTCCFF, (see Section 2.1) and $AAPN, (see Section 2.15). The new communication settings will be effective after the next power-on reset.

1.11 Technical Support

Should you encounter problems while using the M-2000 and M-6000 module, and are unable to find the help you need in this manual or on our website, please contact ICP DAS Product Support.

Email: service@icpdas.com
Website: http://www.icpdas.com.tw/contact_us/contact_us.html

When requesting technical support, be prepared to provide the following information about your system:

1. Module name and serial number: The serial number can be found printed on the barcode label attached to the cover of the module.
2. Firmware version: See Section 2.12 and 3.3.7 for information regarding the command used to identify the firmware version.
3. Host configuration (type and operating system)
4. If the problem is reproducible, please give full details describing the procedure used to reproduce the problem.
5. Specific error messages displayed. If a dialog box with an error message is displayed, please include the full text of the dialog box, including the text in the title bar.
6. If the problem involves other programs or hardware devices, please describe the details of the problem in full.
7. Any comments and suggestions related to the problem are welcome.

ICP DAS will reply to your request by email within three business days.

2. DCON Protocol

With DCON protocol, all communication with M-2000 and M-6000 modules consists of commands generated by the host and responses transmitted by the M-2000 and M-6000 modules. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The ID is 01 by default and can be changed using a user command. All commands to the modules contain the ID address, meaning that only the addressed module will respond. The only exception to this is command ~** (Section 2.20), which are sent to all modules, but in these cases, the modules do not reply to the command.

Command Format:

Leading Character	Module Address	Command	[CHKSUM]	CR

Response Format:

Leading Character	Module Address	Data	[CHKSUM]	CR

CHKSUM A 2-character checksum that is present when the checksum setting is enabled. See Section 1.9 and 2.1 for details.

CR End of command character, carriage return (0x0D)

Checksum Calculation:

1. Calculate the ASCII code sum of all the characters in the command/response string except for the carriage return character (CR).
2. The checksum is equal to the sum masked by 0ffh.

Example:

Command string: $012(CR)

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

Response string: !01200600(CR)

1. 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 response string with the checksum = !01200600AA(CR)

Note:

All characters should be in upper case.

General Command Sets			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Set Module Configuration	2.1
#AA	>(Data)	Reads the Analog Inputs of All Channels	2.2
#AAN	>(Data)	Reads the Analog Input of the Specified Channel	2.3
$AA0	!AA	Performs a Span Calibration	2.4
$AA1	!AA	Performs a Zero Calibration	2.5
$AA2	!AANNTTCCFF	Reads the Module Configuration	2.6
$AA3	>(Data)	Reads the CJC Temperature	2.7
$AA5VVVV	!AA	Enables/Disables the Channel	2.8
$AA6	!AAVVVV	Reads the Channel Enable/Disable Status	2.9
$AA9	!AA(Data)	Reads the CJC Offset	2.10
$AA9SNNNN	!AA	Sets the CJC Offset	2.11
$AAF	!AA(Data)	Reads the Firmware Version	2.12
$AAM	!AA(Data)	Reads the Module Name	2.13
$AAP	!AASC	Reads the Protocol	2.14
$AAPN	!AA	Sets the Protocol	2.15
~AAC	!AAN	Reads the CJC Enable/Disable	2.16
~AACN	!AA	Enables/Disables the CJC	2.17
~AAEV	!AA	Enables/Disables the Calibration	2.18
~AAO(Name)	!AA	Sets the Module Name	2.19
~AAEO	!AAN	Reads the Open Wire Detection Enable/Disable	2.25
~AAEON	!AA	Enable/Disable Open Wire Detection	2.26

Host Watchdog Command Sets			
Command	**Response**	**Description**	**Section**
~**	No Response	Host OK	2.20
~AA0	!AASS	Reads the Host Watchdog Status	2.21
~AA1	!AA	Resets the Host Watchdog Status	2.22
~AA2	!AAETT	Reads the Host Watchdog Timeout Settings	2.23
~AA3ETT	!AA	Sets the Host Watchdog Timeout Settings	2.24

2.1 %AANNTTCCFF

Description:

Sets the configuration of an analog input module.

Syntax:

%AANNTTCCFF[CHKSUM](CR)

%	Delimiter character
AA	Address of the module to be configured in hexadecimal format (00 to FF)
NN	New address of the module in hexadecimal format (00 to FF)
TT	New type code, see Section 1.9 for details.
CC	New Baud Rate code, see Section 1.9 for details. To change the Baud Rate, the INIT switch must be set to the on position. See Section A.1 for details.
FF	Used to set the data format, checksum, and filter settings (Section 1.9). To change the checksum setting, the INIT switch must be set to the on position. See Section A.1 for details.

Response:

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

! Delimiter character for a valid response

? Delimiter character for an invalid response. If changing the **Baud Rate** or **checksum** settings without switching the INIT switch to the on position, the module will return an invalid command.

AA Address of the 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: %0102000600 Response: !02

Change the address of module 01 to 02. The module returns a valid response.

Command: %0202000602 Response: !02

Set the data format of module 02 to be 2 (2's complement hexadecimal). The module returns a valid response.

Command: %0101000A00 Response: ?01

Change the Baud Rate of module 01 to 115200bps. The module returns an invalid command, because it is not in INIT* mode.

Command: %0101000A00 Response: !01

Change the Baud Rate of module 01 to 115200bps and the module is in INIT* mode. The module returns a valid response.

Related Commands:

Section 2.6 $AA2

Related Topics:

Section 1.9 Configuration Tables, Section A.1 INIT Mode

Notes:

1. 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 on the next power on reset.

2.2 #AA

Description:
Reads the data from every analog input channel.

Syntax:
#AA[CHKSUM](CR)

#	Delimiter character
AA	Address of the module to be read (00 to FF)

Response:
Valid Response: **>(Data)[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**

>	Delimiter character for a valid response
?	Delimiter character for an invalid response
(Data)	Data from every analog input channels, see Section 1.9 for the details of 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+003.24+015.35+008.07+014.79

Reads module 01 and receives the data in engineering format.

Command: #02 Response:
>4C532628E2D683A20F2ADBA16284BA71

Reads module 02 and receives the data in hexadecimal format.

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

Reads module 03 and the data is under range.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:

Section 1.9 Configuration Tables

2.3 #AAN

Description:

Reads the analog input of channel N.

Syntax:

#AAN[CHKSUM](CR)

#	Delimiter character
AA	Address of the module to be read (00 to FF)
N	The channel to be read, zero based.

Response:

Valid Response: **>(Data)[CHKSUM](CR)**

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

>	Delimiter character for a valid response
?	Delimiter character for an invalid response. An invalid command is returned if the specified channel is incorrect.
(Data)	Analog input data of the specified channel, see Section 1.9 for details of the data format.
AA	Address of the responding module (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: #032 Response: >+025.13

Reads data from channel 2 of module 03.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:

Section 1.9 Configuration Tables

2.4 $AA0

Description:
Performs a span calibration.

Syntax:
$AA0[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be calibrated (00 to FF)
0	Command for the span calibration

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: !01

Performs a span calibration on module 01 and returns a valid response.

Command: $020 Response: ?02

Performs a span calibration on module 02. An invalid command is returned because the "enable calibration" command was not sent in advance.

Related Commands:

Section 2.5 $AA1, Section 2.18 ~AAEV

Related Topics:

Section 1.8 Calibration

Notes:

1. The "enable calibration" command, ~AAEV, must be sent before this command is used, see Section 1.8 for details.

2.5 $AA1

Description:
Performs a zero calibration.

Syntax:
$AA1[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
1	Command for the zero calibration

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: $011 Response: !01

Performs a zero calibration on module 01 and returns a valid response.

Command: $021 Response: ?02

Performs a zero calibration on module 02. An invalid command is returned because the "enable calibration" command was not sent in advance.

Related Commands:

Section 2.4 $AA0, Section 2.18 ~AAEV

Related Topics:

Section 1.8 Calibration

Notes:

1. The "enable calibration" command, ~AAEV, must be sent before this command is used, see Section 1.8 for details.

2.6 $AA2

Description:
Reads the module configuration.

Syntax:
$AA2[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
2	Command to read the module configuration

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
TT	Type code of the module, see Section 1.9 for details.
CC	Baud Rate code of the module, see Section 1.9 for details.
FF	Data format, checksum settings and filter settings of the module, see Section 1.9 for details.

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: !01050600

Reads the configuration of module 01.

Command: $022 Response: !02030602

Reads the configuration of module 02.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.9 Configuration Tables

,

2.7 $AA3

Description:

Reads the CJC (cold junction compensation) temperature.

Syntax:

$AA3[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
3	Command to read the CJC temperature

Response:

Valid Response: **>(Data)[CHKSUM](CR)**

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

>	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
(Data)	CJC temperature in degrees Celsius, consisting of a sign byte, '+' or '-', and followed by 5 decimal digits with a fixed decimal point in tenths of a degree.

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: $013 Response: >+0031.2

Reads the CJC temperature of module 01 and the module responds with 31.2℃.

Related Commands:

Section 2.10 $AA9, Section 2.11 $AA9SNNNN, Section 2.16 ~AAC, Section 2.17 ~AACN

Note:

1. For M-2018-16 and M-6018-16 modules, the CJC offset is included in the reported CJC temperature.

2.8 $AA5VVVV

Description:

Specifies the channel(s) to be enabled.

Syntax:

$AA5VV(VV)[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
5	Command to set the channel(s) to enabled
VVVV	A four-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the channel is enabled and 0 means that the channel is disabled.

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response. An invalid command is returned if an attempt is made to enable a channel that is not present.
AA	Address of the responding module (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: $015003A Response: !01

Enables channels 1, 3, 4, and 5 and disables all other channels of module 01. The module returns a valid response.

Command: $016 Response: !01003A

Reads the channel status of module 01 and returns a response of 3A, meaning that channels 1, 3, 4, and 5 are enabled and all other channels are disabled.

Related Commands:

Section 2.9 $AA6

Note:

1. It is recommended that only the channels that will be used are enabled.

2.9 $AA6

Description:

Reads the enabled/disabled status of each channel.

Syntax:

$AA6[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
6	Command to read the channel status

Response:

Valid Response: **!AAVV(VV)[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
VVVV	A four-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the channel is enabled and 0 means that the channel is disabled.

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: $015003A Response: !01

Enables channels 1, 3, 4, and 5 and disables all other channels of module 01. The module returns a valid response.

Command: $016 Response: !01003A

Reads the channel status of module 01 and returns a response of 3A, meaning that channels 1, 3, 4, and 5 are enabled and all other channels are disabled.

Related Commands:

Section 2.8 $AA5VVVV

2.10 $AA9

Description:

Reads the CJC (cold junction compensation) offset value that is set by the $AA9SNNNN command (Section 2.11).

Syntax:

$AA9[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
9	Command to read the CJC offset value

Response:

Valid Response: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
(Data)	CJC offset value consisting of a sign byte, '+' or '-', followed by 4 hexadecimal digits. Each count is equal to 0.01℃.

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: $019 Response: !01+0010

Reads the CJC offset of module 01 and the module responds with +10 which means +0.16℃.

Related Commands:

Section 2.7 $AA3, Section 2.11 $AA9SNNNN, Section 2.17 ~AACN

2.11 $AA9SNNNN

Description:

Sets the CJC (cold junction compensation) offset value to adjust the error produced by the CJC sensor.

Syntax:

$AA9SNNNN[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
9	Command to set the CJC offset value
S	Sign byte, '+' or '-', of the offset value
NNNN	The absolute value of the offset in four hexadecimal digits, which must be less than or equal to 1000h. Each count is equal to 0.01°C.

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: $019+0010 Response: !01

Sets the CJC offset of module 01 to +0.16℃ and returns a valid response.

Related Commands:

Section 2.7 $AA3, Section 2.10 $AA9, Section 2.17 ~AACN

2.12 $AAF

Description:
Reads the firmware version of a module.

Syntax:
$AAF[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
F	Command to read the firmware version

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
(Data)	A string indicating the firmware version of the module

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: !01A2.0

Reads the firmware version of module 01, and shows that it is version A2.0.

2.13 $AAM

Description:
Reads the name of a module.

Syntax:
$AAM[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
M	Command to read the module name

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
(Name)	A string showing the name of the module

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: $01M Response: !012018

Reads the module name of module 01 and returns the name “2018”.

Related Commands:

Section 2.19 ~AAO(Name)

2.14 $AAP

Description:

Reads the communication protocol information.

Syntax:

$AAP[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
P	Command to read the communication protocol

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
S	The protocols supported by the module 0: only DCON protocol is supported 1: both the DCON and Modbus RTU protocols are supported
C	Current protocol saved in EEPROM that will be used at the next power on reset 0: the protocol set in EEPROM is DCON 1: the protocol set in 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: $01P Response: !0110

Reads the communication protocol of module 01 and returns a response of 10 meaning that it supports both the DCON and Modbus RTU protocols and the protocol that will be used at the next power on reset is DCON.

Related Commands:

Section 2.15 $AAPN

2.15 $AAPN

Description:

Sets the communication protocol.

Syntax:

$AAPN[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
P	Command to set the communication protocol
N	0: DCON protocol 1: Modbus RTU protocol Before using this command, the INIT switch must be in the on position, see Section A.1 for details. The new protocol is saved in the EEPROM and will be effective after the next power on reset.

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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

Sets the communication protocol of module 01 to Modbus RTU and returns an invalid response because the module is not in INIT mode.

Command: $01P1 Response: !01

Sets the communication protocol of module 01 to Modbus RTU and returns a valid response.

Related Commands:

Section 2.14 $AAP

Related Topics:

Section A.1 INIT Mode

2.16 ~AAC

Description:

Reads the CJC (cold junction compensation) enabled/disabled status.

Syntax:

~AAC[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
C	Command to read the CJC enabled/disabled status

Response:

Valid Response: **!AAN[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
N	0: CJC disabled 1: CJC 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: ~01C Response: !011

Sends a command to read the CJC enabled/disabled status of module 01 and shows that the CJC is enabled.

Related Commands:

Section 2.7 $AA3, Section 2.10 $AA9, Section 2.11 $AA9SNNNN, Section 2.17 ~AACN

2.17 ~AACN

Description:

Enable/disable CJC (cold junction compensation).

Syntax:

~AACN[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
C	Command to enable/disable CJC
N	0: disable CJC 1: enable CJC

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: ~01C0 Response: !01

Sends a command to disable CJC of module 01 and returns a valid response.

Related Commands:

Section 2.7 $AA3, Section 2.10 $AA9, Section 2.11 $AA9SNNNN, Section 2.16 ~AAC

2.18 ~AAEV

Description:

Enable/Disable module calibration.

Syntax:

~AAEV[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
E	Command to enable/disable calibration
V	1: enable calibration 0: disable calibration

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: ?01

Sends the command to perform a span calibration on module 01. It returns an invalid response because the "enable calibration" command was not sent in advance.

Command: ~01E1 Response: !01

Enables calibration on module 01 and returns a valid response.

Command: $010 Response: !01

Sends the command to perform a span calibration on module 01 and returns a valid response.

Related Commands:

Section 2.4 $AA0, Section 2.5 $AA1

Related Topics:

Section 1.8 Calibration

2.19 ~AAO(Name)

Description:

Sets the name of a module.

Syntax:

~AAO(Name)[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
O	Command to set the module name
(Name)	New name of the module (max. 6 characters).

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: ~01O2018A Response: !01

Sets the name of module 01 to be “2018A” and returns a valid response.

Command: $01M Response: !012018A

Reads the name of module 01 and returns “2018A”.

Related Commands:

Section 2.13 $AAM

2.20 ~**

Description:
Informs all modules that the host is OK.

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

Response:
No response.

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

Related Commands:
Section 2.21 ~AA0, Section 2.22 ~AA1, Section 2.23 ~AA2, Section 2.24 ~AA3EVV

Related Topics:
Section A.2 Dual Watchdog Operation

Note:
After sending this command, there must be a 2ms delay before the next command can be sent.

2.21 ~AA0

Description:

Reads the host watchdog status of a module.

Syntax:

~AA0[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
0	Command to read the module status

Response:

Valid Response: **!AASS[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
SS	Two hexadecimal digits that represent the host watchdog status, where: Bit 7: 0 indicates that the host watchdog is disabled and 1 indicates the host watchdog is enabled, Bit 2: 1 indicates that a host watchdog time out has occurred and 0 indicates that no host watchdog time out has occurred. The host watchdog status is stored in EEPROM and can only be reset using the ~AA1 command.

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: !0100

Reads the host watchdog status of module 01 and returns 00, meaning that the host watchdog is disabled and no host watchdog time out has occurred.

Command: ~020 Response: !0204

Reads the host watchdog status of module 02 and returns 04, meaning that a host watchdog timeout has occurred.

Related Commands:

Section 2.20 ~**, Section 2.22 ~AA1, Section 2.23 ~AA2, Section 2.24 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.22 ~AA1

Description:

Resets the host watchdog time out status of a module.

Syntax:

~AA1[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
1	Command to reset the host watchdog time out status

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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 host watchdog status of module 01 and shows that a host watchdog time out has occurred.

Command: ~011 Response: !01

Resets the host watchdog time out status of module 01 and returns a valid response.

Command: ~010 Response: !0100

Reads the host watchdog status of module 01 and shows that no host watchdog time out has occurred.

Related Commands:

Section 2.20 ~**, Section 2.21 ~AA0, Section 2.23 ~AA2, Section 2.24 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.23 ~AA2

Description:

Reads the host watchdog time out value of a module.

Syntax:

~AA2[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
2	Command to read the host watchdog time out value

Response:

Valid Response: **!AAEVV[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
E	1: the host watchdog is enabled 0: the host watchdog is disabled
VV	Two hexadecimal digits to represent the time out value in tenths of a second, for example, 01 means 0.1 seconds and FF means 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: ~012 Response: !011FF

Reads the host watchdog time out value of module 01 and returns FF, meaning that the host watchdog is enabled and the host watchdog time out value is 25.5 seconds.

Related Commands:

Section 2.20 ~**, Section 2.21 ~AA0, Section 2.22 ~AA1, Section 2.24 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.24 ~AA3EVV

Description:

Enables/disables the host watchdog and set the host watchdog time out value of a module.

Syntax:

~AA3EVV[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
3	Command to set the host watchdog
E	1: enable the host watchdog 0: disable the host watchdog
VV	Two hexadecimal digits to represent the time out value in tenths of a second, for example, 01 means 0.1 seconds and FF means 25.5 seconds.

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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 of module 01 and sets the host watchdog time out value to 10.0 seconds. The module returns a valid response.

Command: ~012 Response: !01164

Reads the host watchdog time out value of module 01. The module returns 164, meaning that the host watchdog is enabled and the host watchdog time out value is 10.0 seconds.

Related Commands:

Section 2.20 ~**, Section 2.21 ~AA0, Section 2.22 ~AA1, Section 2.23 ~AA2

Related Topics:

Section A.2 Dual Watchdog Operation

2.25 ~AAEO

Description:

Reads the open wire detection enabled/disabled status.

Syntax:

~AAEE[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
EO	Command to read the open wire detection enabled/disabled status

Response:

Valid Response: **!AAN[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
N	0: open wire detection disabled 1: open wire detection 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: ~01EO Response: !011

Sends a command to read the open wire detection enabled/disabled status of module 01 and shows that the open wire detection is enabled.

Related Commands:

Section 2.26 ~AAEON

2.26 ~AAEON

Description:
Enable/disable open wire detection.

Syntax:
~AAEEN[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
EO	Command to enable/disable open wire detection
N	0: disable open wire detection 1: enable open wire detection

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (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: ~01EO0 Response: !01

Sends a command to disable open wire detection of module 01 and returns a valid response.

Related Commands:

Section 2.25 ~AAEO

3. Modbus RTU Protocol

The Modbus protocol is developed by Modicon Inc., originally developed for Modicon controllers. Detailed information can be found at http://www.modicon.com/techpubs/toc7.html. You can also visit http://www.modbus.org to find more valuable information.

M-2000 and M-6000 series modules support the Modbus RTU protocol. The communication Baud Rates range from 1200bps to 115200bps. The following Modbus functions are supported.

Function Code	Description	Section
02 (0x02)	Read input status	3.1
04 (0x04)	Read input channels	3.2
70 (0x46)	Read/write module settings	3.3

If the function specified in the message is not supported, then the module responds as follows.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	Function code | 0x80
02	Exception code	1 Byte	01

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

3.1 02 (0x02) Read Input Status

This function code is used to read the thermocouple wire opening status of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02 ~ 03	Starting channel	2 Bytes	0x80 to 0x8F, where 0x80 corresponds to channel 0, 0x81 corresponds to channel 1, etc
04 ~ 05	Number of input channels	2 Bytes	N, 1 to 16; (Starting channel + N) should be less than or equal to 0x90

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02	Byte count	1 Byte	1
03	Data of input channels	1 Byte	A bit corresponds to a channel. When the bit is 1 it denotes that the channel is enabled and is either over-range, under-range or wire opening. If the bit is 0 it denotes that the channel is disabled or normal.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x82
02	Exception code	1 Byte	02: starting channel out of range 03: (starting channel + number of input channels) out of range, incorrect number of bytes received

3.2 04 (0x04) Read Input Channels

This function code is used to read from contiguous analog input channels or the CJC temperature.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x04
02 ~ 03	Starting channel	2 Bytes	0 to F for reading analog inputs 0x80 for reading CJC temperature
04 ~ 05	Number of input channels (N)	2 Bytes	1 to 16; (Starting channel + N) <= 16 for reading analog inputs. 1 for reading CJC temperature.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x04
02	Byte count	1 Byte	2 x N
03 ~	Data of input channels	2 x N Bytes	When used for the CJC temperature, this is a 2's complement hex value in 0.01℃ increments.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x84
02	Exception code	1 Byte	02: starting channel out of range 03: (starting channel + number of input channels) out of range, incorrect number of bytes received

3.3 70 (0x46) Read/Write Module Settings

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

Sub-function Code	Description	Section
00 (0x00)	Read the module name	3.3.1
04 (0x04)	Set the module address	3.3.2
05 (0x05)	Read the communication settings	3.3.3
06 (0x06)	Set the communication settings	3.3.4
07 (0x07)	Read the type code	3.3.5
08 (0x08)	Set the type code	3.3.6
32 (0x20)	Read the firmware version	3.3.7
37 (0x25)	Read the channel enable/disable status	3.3.8
38 (0x26)	Set the channel enable/disable	3.3.9
41 (0x29)	Read the miscellaneous settings	3.3.10
42 (0x2A)	Write the miscellaneous settings	3.3.11
43 (0x2B)	Read the CJC offset	3.3.12
44 (0x2C)	Write the CJC offset	3.3.13
45 (0x2D)	Read the CJC enable/disable status	3.3.14
46 (0x2E)	Set the CJC enable/disable	3.3.15

If the module does not support the sub-function code specified in the message, then it responds as follows.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	02: invalid sub-function code

3.3.1 Sub-function 00 (0x00) Read module name

This sub-function code is used to read the name of a 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 0x20 0x18 0x00 for M-2018-16 modules 0x00 0x60 0x18 0x00 for M-6018-16 modules

Error Response

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

3.3.2 Sub-function 04 (0x04) Set module address

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
03	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	Set address result	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: new address out of range, reserved bytes should be filled with zero, incorrect number of bytes received

3.3.3 Sub-function 05 (0x05) Read communication settings

This sub-function code is used to read the communication protocol settings of a 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	Baud Rate code, see Section 1.9 for details.
05 ~ 07	Reserved	3 Bytes	0x00 0x00 0x00
08	Mode	1 Byte	0: DCON protocol 1: Modubs RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Note: This information is the data saved in the EEPROM and will be used for the next power-on reset. It is not the currently used settings.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: reserved byte should be filled with zero, incorrect number of bytes received

3.3.4 Sub-function 06 (0x06) Set communication settings

This sub-function code is used to set the communication protocol of a 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	Baud Rate code, see Section 1.9 for details.
05 ~ 07	Reserved	3 Bytes	0x00 0x00 0x00
08	Mode	1 Byte	0: DCON protocol 1: Modubs RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 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 ~ 07	Reserved	3 Bytes	0x00 0x00 0x00
08	Mode	1 Byte	0: OK, others: error
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Note: The new Baud Rate and protocol will be effective after the next power-on reset.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: Baud Rate or mode out of range, reserved bytes should be filled with zero, incorrect number of bytes received

3.3.5 Sub-function 07 (0x07) Read type code

This sub-function code is used to read the type code information 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 Bytes	0x00
04	Channel	1 Byte	0x00 for M-2018-16 and M-6018-16 modules

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	Type code, see Section 1.9 for details.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: reserved bytes should be filled with zero, incorrect number of bytes received

3.3.6 Sub-function 08 (0x08) Set type code

This sub-function code is used to set the type code 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	1 Byte	0x00 for M-2018-16 and M-6018-16 series modules
05	Type code	1 Byte	Type code, see Section 1.9 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: type code out of range, reserved bytes should be filled with zero, incorrect number of bytes received

3.3.7 Sub-function 32 (0x20) Read firmware version

This sub-function code is used to read the firmware version information of 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 ~ 0xFF
04	Minor version	1 Byte	0x00 ~ 0xFF
05	Build version	1 Byte	0x00 ~ 0xFF

Error Response

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

3.3.8 Sub-function 37 (0x25) Read channel enabled/disabled status

This sub-function code is used to read the enabled/disabled status of each channel in a module.

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	Enabled/disabled status	2 Byte	0x0000 ~ 0xFFFF, enabled/disabled status of each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it denotes that the channel is enabled and 0 denotes that the channel is disabled.

Error Response

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

3.3.9 Sub-function 38 (0x26) Set channel enable/disable

This sub-function code is used to specify the channels to be enabled in a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x26
03	Enable/disable setting	2 Byte	0x0000 ~ 0xFFFF, enable/disable setting of each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it denotes that the channel is enabled and 0 denotes that the channel is disabled.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x26
03	Enable/disable setting	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: enable/disable setting out of range, incorrect number of bytes received

3.3.10 Sub-function 41 (0x29) Read miscellaneous settings

This sub-function code is used to read the miscellaneous settings of 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 setting 0: 60Hz rejection 1: 50Hz rejection Bit 6~0: reserved

Note: The reserved fields are filled with zeros.

Error Response

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

3.3.11 Sub-function 42 (0x2A) Write miscellaneous settings

This sub-function code is used to set the miscellaneous settings of 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 setting 0: 60Hz rejection 1: 50Hz rejection Bit 6~0: reserved

Note: The reserved fields are filled with zeros.

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, incorrect number of bytes received

3.3.12 Sub-function 43 (0x2B) Read CJC offset

This sub-function code is used to read the CJC offset setting of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2B
03	Channel	1 Byte	0x00 for module CJC offset

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2B
03 ~ 04	CJC offset	2 Bytes	This is a 2's complement hex CJC offset value. For module CJC offset, it is in 0.01℃ increments. For channel CJC offset, it is in 0.1℃ increments where 00 denotes 0℃, 7F denotes 12.7℃, FF denotes -0.1℃ and 80 denotes -12.8℃.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes received, reserved byte is not zero

3.3.13 Sub-function 44 (0x2C) Write CJC offset

This sub-function code is used to set the CJC offset setting of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2C
03	Channel	1 Byte	0x00 for module CJC offset
04 ~ 05	CJC offset	2 Bytes	This is a 2's complement hex CJC offset value. For module CJC offset, it is in 0.01℃ increments and the absolute value should be less than or equal to 0x1000. For channel CJC offset, it is in 0.1℃ increments and in the range 00 ~ FF, where 00 denotes 0℃, 7F denotes 12.7℃, FF denotes -0.1℃ and 80 denotes -12.8℃.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2C
03	Set CJC offset	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 byte should be zero, CJC offset value out of range, incorrect number of bytes received

3.3.14 Sub-function 45 (0x2D) Read CJC enabled/disabled status

This sub-function code is used to read the CJC enabled/disabled status of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2D
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	0x2D
03	CJC enabled/disabled status	1 Byte	0: CJC disabled 1: CJC enabled

Error Response

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

3.3.15 Sub-function 46 (0x2E) Set CJC enable/disable

This sub-function code is used to enable/disable the CJC of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2E
03	Reserved	1 Byte	0x00
04	Enable/disable CJC	1 Byte	0: disable CJC 1: enable CJC

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2E
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 byte is not zero, setting byte incorrect, incorrect number of bytes received

3.4 Address Mappings

3.4.1 M-2018-16 and M-6018-16 Address Mappings (Base 1)

<table>
<tr><th>Address</th><th>Description</th><th>Attribute</th></tr>
<tr><td>30001 ~ 30016
40001 ~ 40016</td><td>Analog input value of channel 0 to 15</td><td>R</td></tr>
<tr><td>30129
40129</td><td>CJC temperature in 0.01℃</td><td>R</td></tr>
<tr><td>40353 ~ 40368</td><td>CJC offset of channel 0 to 15 in 0.1℃.
1 for 0.1, 127 for 12.7, 255 for –0.1, 128 for –12.8</td><td>R/W</td></tr>
<tr><td>40481</td><td>Firmware version (low word)</td><td>R</td></tr>
<tr><td>40482</td><td>Firmware version (high word)</td><td>R</td></tr>
<tr><td>40483</td><td>Module name (low word)</td><td>R</td></tr>
<tr><td>40484</td><td>Module name (high word)</td><td>R</td></tr>
<tr><td>40485</td><td>Module address, valid range: 1 ~ 247</td><td>R/W</td></tr>
<tr><td>40486</td><td>Bits 5:0
Baud rate, 0x03 ~ 0x0A
<table>
<tr><td>Code</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td></tr>
<tr><td>Baud</td><td>1200</td><td>2400</td><td>4800</td><td>9600</td></tr>
<tr><td>Code</td><td>0x07</td><td>0x08</td><td>0x09</td><td>0x0A</td></tr>
<tr><td>Baud</td><td>19200</td><td>38400</td><td>57600</td><td>115200</td></tr>
</table>
Bits 7:6
00: no parity, 1 stop bit
01: no parity, 2 stop bits
10: even parity, 1 stop bit
11: odd parity, 1 stop bit</td><td>R/W</td></tr>
</table>

Address	Description	Attribute
40487	Type code	R/W
40488	Modbus response delay time in ms, valid range: 0 ~ 30	R/W
40489	Host watchdog timeout value, 0 ~ 255, in 0.1s	R/W
40490	Channel enable/disable, 0000h ~ FFFFh	R/W
40491	Module CJC offset in 0.01℃	R/W
40492	Host watchdog timeout count, write 0 to clear	R/W
10129 ~ 10144 00129 ~ 00144	Over/under range status of channel 0 to 15	R
00257	Protocol, 0: DCON, 1: Modbus RTU	R/W
00259	Filter setting, 0: 60Hz rejection, 1: 50Hz rejection	R/W
00260	Modbus host watchdog mode 0: same as I-7000 1: can use AO and DO command to clear host watchdog timeout status	R/W
00261	1: enable, 0: disable host watchdog	R/W
00268	1: enable, 0: disable CJC	R/W
00269	Modbus data format, 0: hex, 1: engineering	R/W
00270	Host watch dog timeout status, write 1 to clear host watch dog timeout status	R/W
00273	Reset status, 1: first read after powered on, 0: not the first read after powered on	R

3.5 Engineering Data Format Table

Type Code	Input Type	Min.	Max.
00	-15 mV ~ +15 mV	-15000	15000
01	-50 mV ~ + 50 mV	-5000	5000
02	-100 mV ~ +100 mV	-10000	10000
03	-500 mV ~ +500 mV	-5000	5000
04	-1 V ~ +1 V	-10000	10000
05	-2.5 V ~ +2.5 V	-25000	25000
06	-20 mA ~ +20 mA	-20000	20000
07	+4 mA ~ +20 mA	4000	20000
0E	Type J Thermocouple	-2100	7600
0F	Type K Thermocouple	-2700	13720
10	Type T Thermocouple	-2700	4000
11	Type E Thermocouple	-2700	10000
12	Type R Thermocouple	0	17680
13	Type S Thermocouple	0	17680
14	Type B Thermocouple	0	18200
15	Type N Thermocouple	-2700	13000
16	Type C Thermocouple	0	23200
17	Type L Thermocouple	-2000	8000
18	Type M Thermocouple	-20000	10000
19	Type $L_{DIN43710}$ Thermocouple	-2000	9000
1A	0 ~ +20 mA	0	20000

The under range value is –32768 and the over range value is +32767. For the hex data format, please refer to Section 1.9.

4. Troubleshooting

If you are having difficulty using the M-2000 or M-6000 module, here are some suggestions that may help. If you cannot find the answers you need in these guides, contact ICP DAS Product Support. Contact information is located in Section 1.11.

4.1 Communicating with the module

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

- Make sure the supplied power is within the range of +10 to +48 V DC. If the supplied power is OK, then the power LED should be on.
- When the module receives a command, the power LED is set to "off". The power LED is shown as "on" after the module responds. This method can be used to check whether the module has received a command sent from the host.
- If possible, use another device to check whether the host can communicate with the device through the same RS-485 network.
- If the host is a PC installed with a Windows operating system, then execute the DCON Utility to determine whether the module can be found. The DCON Utility can be downloaded from the ICP DAS website http://www.icpdas.com. The DCON Utility documentation 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 the following settings: address 00, Baud Rate 9600bps, no checksum and DCON protocol. See Section A.1 for details.

4.2 Reading Data

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

- Make sure the type code and data format settings are correct. The type code is set by using the %AANNTTCCFF command, see Section 2.1 for details. The data format is set by using the %AANNTTCCFF command. For the Modbus RTU protocol, the type code is set by using sub-function 08h of the function 46h.
- If the voltage read by the module is incorrect, then it may be because the calibration parameters stored in the non-volatile memory are corrupted. You can calibrate the module by yourself. Be sure to read Section 1.8 in detail before doing any calibration.

A. Appendix

A.1 INIT Mode

Each M-2000 and M-6000 module has a built-in EEPROM to store configuration information such as module address, type code, Baud Rate, etc. Occasionally, the configuration of a module may be forgotten and there are 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 M-2000 and M-6000 series has a special mode called **"INIT mode"**. When the module is powered on in **"INIT mode"** the configuration of the module is reset as follows, 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 they can be read by sending the $002(CR) command at 9600bps.

There are commands that require the module to be in INIT mode. They are:

1. %AANNTTCCFF when changing Baud Rate and checksum settings. See Section 2.1 for details.
2. $AAPN, see Section 2.15 for details.

The INIT mode is accessed by switching the INIT switch to the on position.

A.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 circuit allows the module to work 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 time out occurs, the module will reset all outputs to a safe state in order to prevent any erroneous operations of the controlled target.

M-2000 and M-6000 series modules include an internal Dual Watchdog, making the control system more reliable and stable.

For more information regarding the Dual Watchdog, please refer to Chapter 5 of the "**Getting Started For I-7000 Series Modules**" manual that can be downloaded from the ICP DAS website http://www.icpdas.com.

A.3 Thermocouple

When two wires composed of dissimilar homogeneous metals are joined at one end, a thermoelectric electromotive force (emf) appears that depends only on the metals and the junction temperature. This is called the Seebeck effect. A pair of different metals with a fixed junction at one end constitutes a **thermocouple**. For small changes in temperature, the emf is linearly proportional to the temperature. This implies that the temperature reading can be obtained by measuring the emf.

We cannot measure the emf, V_1, directly because when a voltmeter is connected to the thermocouple, another emf, V_2, is created at the (cold) junction of the thermocouple and the voltmeter. The cold junction compensation method is used to resolve the problem. Using another sensor, e.g. a thermistor, to measure the cold junction temperature, T_2, we can calculate the emf, V_2, which corresponds to T_2. The thermocouple emf, V_1, can be obtained by adding V_2 to that measured by the voltmeter and then the temperature.

A.4 Frame Ground

Electronic circuits are constantly vulnerable to ESD which become worse in a continental climate area. The M-2000 and M-6000 modules feature a new design for the frame ground. The frame ground provides a path for bypassing ESD, which provides enhanced static protection (ESD) abilities and ensures the module is more reliable.

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

1. If the module is DIN rail mounted, connect the DIN rail to the earth ground.
2. Alternatively, connect the frame ground terminal to a wire and connected the wire to the earth ground.

A.5 Hexadecimal Data Conversion

There are two types for hexadecimal data conversion, one for the 4 to 20mA and the 0 to 20mA ranges, and the other for other ranges.

I. for the 4 to 20mA and the 0 to 20mA ranges
The mappings are 0000h maps to the minimum value and FFFFh maps to the maximum value. The formula for the data conversion is
real_data = hex_data * (max_value - min_value) / 65535 + min_value
For example, for the 4 to 20mA range, the formula is
real_data = hex_data * (20.0 - 4.0) / 65535 + 4.0

II. for other ranges
The mappings are 8000h, -32768, maps to -MAX, 0000h, 0, maps to 0, and 7FFFh, 32767, maps to +MAX, where MAX is the larger absolute value of the minimum value and the maximum value. The formula for the data conversion is
If hex_data >= 0 then
    real_data = hex_data * MAX / 32767
else
    real_data = hex_data * MAX / 32768
For example, for type K thermocouple, -270 ~ 1372℃, MAX = 1372, the formula is
If hex_data >= 0 then
    real_data = hex_data * 1372.0 / 32767
else
    real_data = hex_data * 1372.0 / 32768

Revision History

Revision	Date	Changes Made
B2.8	2024/3/27	Modify section 3.4.1 to add over/under range status