tM-TH8

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: 2018/8/15

Table of Contents

1. Introduction

The tM series is a family of network data acquisition and control modules, providing analog-to-digital, digital-to-analog, digital input/output and other functions. The modules can be remotely controlled using a set of commands, which we call the DCON protocol, or the standard Modbus protocol. 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 115.2 Kbps can be selected.

The tM series feature a new design for the frame ground and INIT switch as shown in the figure. The frame ground provides enhanced static protection (ESD) abilities and ensures the module is more reliable. The INIT switch allows easier access to INIT mode. Please refer to Sections A.1 and A.3 for more details.

The features of the tM-TH8 module are as follows:

1. 24-bit sigma-delta ADC to provide excellent accuracy
2. Direct thermistor connection
3. Software calibration

The tM-TH8 is an 8-channel thermistor input module

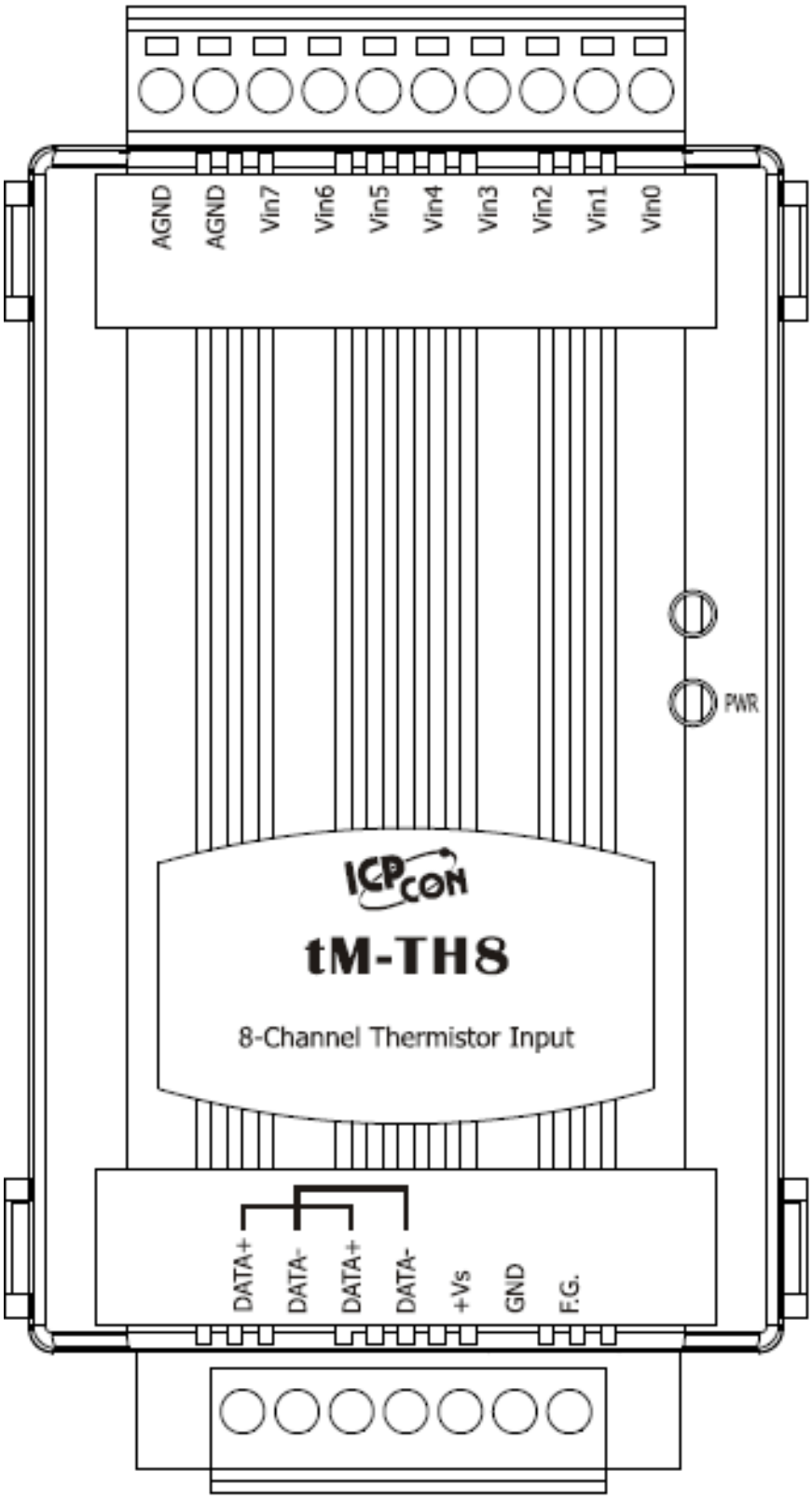
Supported thermistor types are as follows:

1. PreCon Type III, Model 3, 10,000 Ohms at 25℃ (77℉)
2. Fenwell Type U, 2000 Ohms at 25℃
3. YSI L Mix, 100 Ohms at 25℃
4. YSI L Mix, 300 Ohms at 25℃
5. YSI L Mix, 1000 Ohms at 25℃
6. YSI B Mix, 2252 Ohms at 25℃
7. YSI B Mix, 3000 Ohms at 25℃
8. YSI B Mix, 5000 Ohms at 25℃
9. YSI B Mix, 6000 Ohms at 25℃
10. YSI B Mix, 10000 Ohms at 25℃
11. YSI H Mix, 10000 Ohms at 25℃
12. YSI H Mix, 30000 Ohms at 25℃
13. User-defined

1.1 More Information

- For details of **INIT mode** operation, please refer to Section A.1 INIT Mode.
- For details of **module watchdog** and **host watchdog**, please refer to Section A.2 Dual Watchdog Operation.
- For details of **ESD protection** and **grounding**, please refer to Section A.3 Frame Ground.
- There is a way to check whether the module is reset, please refer to Section A.4 **Reset Status** for details.
- A brief introduction to thermistor is described in Section A.5 Thermistor.
- The module tM-TH8 can be used in resistance measurement. Please refer to Section A.6 Resistance Measurement for details,

1.2 Terminal Assignment



1.3 Specifications

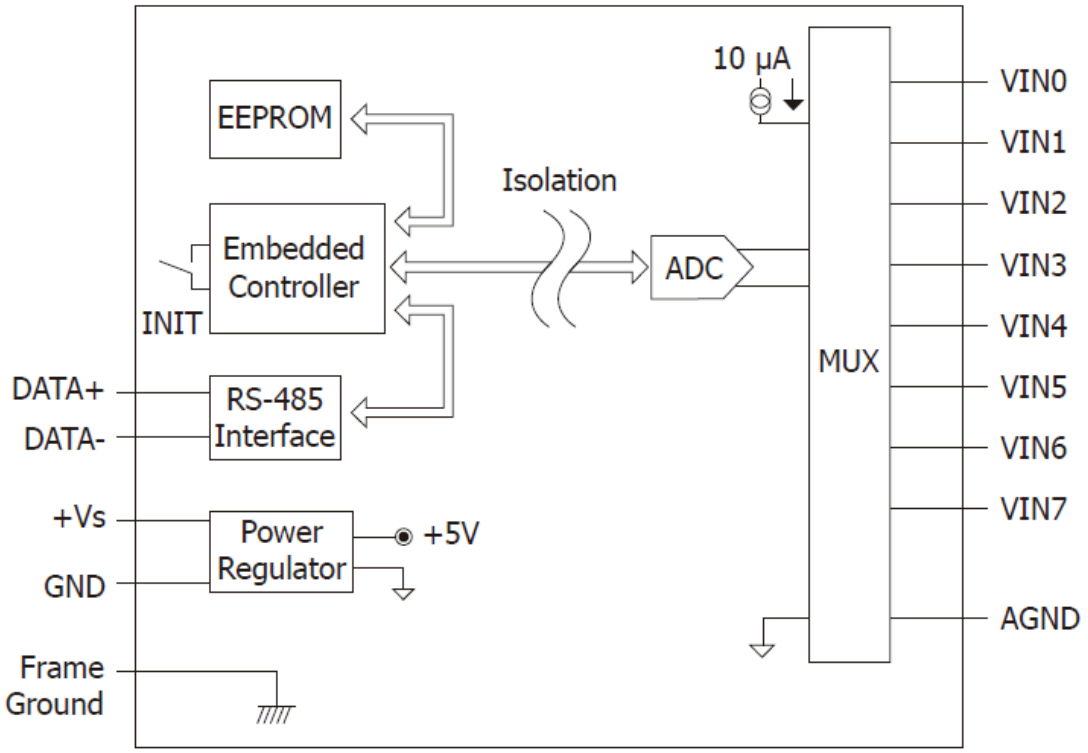
	tM-TH8
Analog Input	
Input Channels	8
Input Type	Thermistor
Thermistor Type	PreCon Type III, Fenwell Type U, YSI L 100, YSI L 300, YSI L 1000, YSI B 2252, YSI B 3000, YSI B 5000, YSI B 6000, YSI B 10000, YSI H 10000, YSI H 30000, User-defined
Temperature Scale	Celsius, Fahrenheit
Sampling Rate	8 samples/sec (total)
Bandwidth	5.24 Hz
Accuracy	±1%
Zero Drift	+/-20 μ V/℃
Span Drift	+/-25 ppm/℃
Open Wire Detection	Yes
Individual Channel Configurable	Yes
Protocol	DCON, Modbus RTU, Modbus ASCII
Overvoltage Protection	120 VDC
Intra-module Isolation	2500 VDC
Power	
Reverse Polarity Protection	Yes
Requirement	+10 to +30 VDC
Consumption	0.6 W
Environment	
Operating Temperature	-25℃ to +75℃
Storage Temperature	-30℃ to +75℃
Humidity	10 ~ 95% RH, non-condensing

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

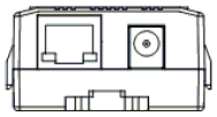
1.4 Block Diagrams

1.4.1 Block Diagram for the tM-TH8

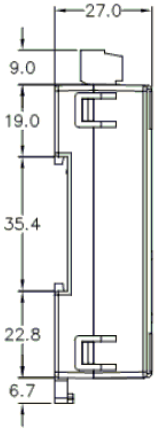
Block Diagram



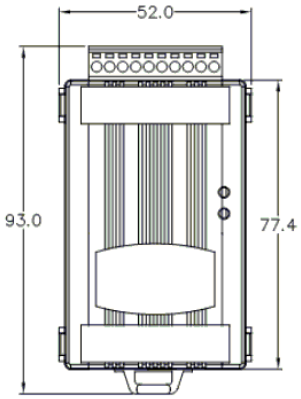
1.5 Dimensions



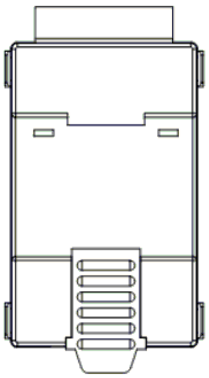
Top View



Left Side View

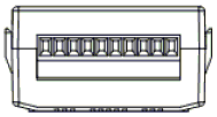


Front View



Back View

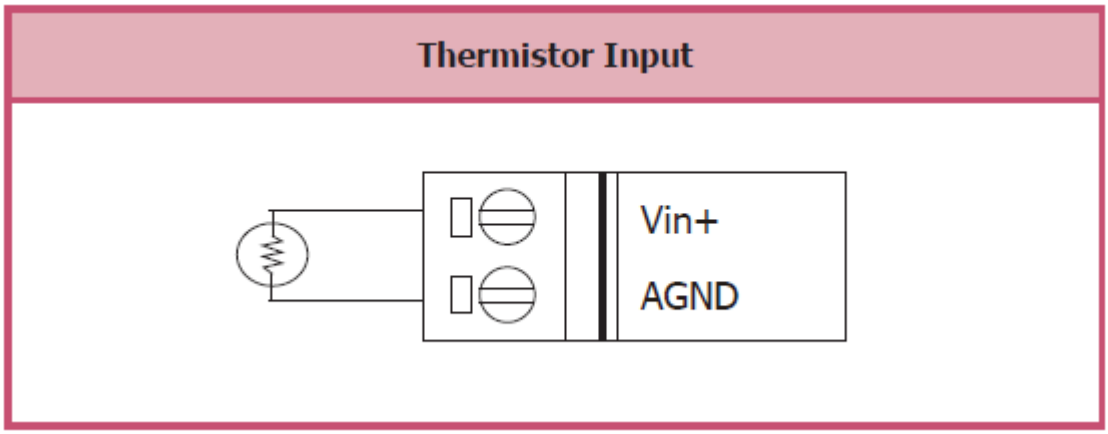
Right Side View



Bottom View

1.6 Wiring

1.6.1 tM-TH8 Wiring



1.6.2 Wiring Recommendations

- Use 26-12 AWG wire for signal connections.
- Strip the wire to a length of 7±0.5mm.
- Use a crimp terminal for wiring.
- Avoid high-voltage cables and power equipment as much as possible.
- For RS-485 communication, use insulated and twisted pair 24 AWG wire, e.g. Belden 9841.

1.7 Quick Start

Please refer to the Quick Start Guide for tM-TH8.

1.8 Default Settings

Default settings for the tM-TH8 are as follows:

- Protocol: Modbus RTU
- Module Address: 01
- Thermistor Type: Type 60, PreCon Type III, -30°F to 240°F
- Baud Rate: 9600 bps
- Temperature Scale: Celsius (the tM-TH8 also supports Fahrenheit scale)

1.9 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 at least 30 minutes.
2. Enable calibration. Refer to Section. 2.21 for details.
3. Connect the zero calibration resistor.
4. Send the zero calibration command. Refer to Section 2.5 for details.
5. Connect the span calibration resistor.
6. Send the span calibration command. Refer to Section 2.4 for details.
7. Repeat steps 2 to 6 three times.

Notes:

1. Calibration resistors are shown below.
2. The tM-TH8 modules must be switched to the DCON protocol mode before calibrating. Refer to Sections 1.12.1 and 3.6.4 for details of the switching protocol.

Calibration resistor used by the tM-TH8

Zero Calibration Resistor	Span Calibration Resistor
0 Ohms	204800 Ohms

1.10 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: eight data bits, no parity, and one stop bit 1: eight data bits, no parity, and two stop bit 2: eight data bits, even parity, and one stop bit 3: eight data bits, odd parity, and one stop bit

Temperature Sensor Type Setting (TT)

Type Code	Temperature Sensor Type	Temperature Range
60	PreCon Type III 10K @ 25℃	-30 ℉ ~ 240 ℉
61	Fenwell Type U 2K @ 25℃	-50 ℃ ~ 150 ℃
62	Fenwell Type U 2K @ 25℃	0 ℃ ~ 150 ℃
63	YSI L Mix 100 @ 25℃	-80 ℃ ~ 100 ℃
64	YSI L Mix 300 @ 25℃	-80 ℃ ~ 100 ℃
65	YSI L Mix 1000 @ 25℃	-70 ℃ ~ 100 ℃
66	YSI B Mix 2252 @ 25℃	-50 ℃ ~ 150 ℃
67	YSI B Mix 3000 @ 25℃	-40 ℃ ~ 150 ℃
68	YSI B Mix 5000 @ 25℃	-40 ℃ ~ 150 ℃
69	YSI B Mix 6000 @ 25℃	-30 ℃ ~ 150 ℃
6A	YSI B Mix 10000 @ 25℃	-30 ℃ ~ 150 ℃
6B	YSI H Mix 10000 @ 25℃	-30 ℃ ~ 150 ℃
6C	YSI H Mix 30000 @ 25℃	-10 ℃ ~ 200 ℃
70	User-defined	-50 ℃ ~ 150 ℃
71	User-defined	-50 ℃ ~ 150 ℃
72	User-defined	-50 ℃ ~ 150 ℃
73	User-defined	-50 ℃ ~ 150 ℃
74	User-defined	-50 ℃ ~ 150 ℃
75	User-defined	-50 ℃ ~ 150 ℃
76	User-defined	-50 ℃ ~ 150 ℃
77	User-defined	-50 ℃ ~ 150 ℃

Note: Please refer to Section 1.11 for details of user-defined types.

Data Format Setting (FF)

7	6	5	4	3	2	1	0
RS	CS	RS				DF	

Key	Description
DF	Data format 00: Engineering unit 01: % of FSR (full scale range) 10: 2’s complement hexadecimal 11: Ohms
CS	Checksum setting 0: Disabled 1: Enabled
RS	Reserved

Note: The reserved bits should be zero.

Thermistor Type and Data Format Table

Type Code	Thermistor Type	Data Format	+F.S.	-F.S.
60	PreCon Type III 10K @ 25℃ -30 ~ 240℉	Engineering unit	+240.00	-030.00
		% of FSR	+100.00	-012.50
		2's comp HEX	7FFF	F000
		Ohms	+000539.4	+173600.0
61	Fenwell U 2K @ 25℃ -50 ~ 150℃	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+000037.2	+134020.0
62	Fenwell U 2K @ 25℃ 0 ~ 150℃	Engineering unit	+150.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+000037.2	+006530.0
63	YSI L Mix 100 @ 25℃ -80 ~ 100℃	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+000014.3	+014470.0
64	YSI L Mix 300 @ 25℃ -80 ~ 100℃	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+000035.8	+067660.0
65	YSI L Mix 1000 @ 25℃ -70 ~ 100℃	Engineering unit	+100.00	-070.00
		% of FSR	+100.00	-070.00
		2's comp HEX	7FFF	A667
		Ohms	+000106.4	+132600.0
66	YSI B Mix 2252 @ 25℃ -50 ~ 150℃	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+000041.8	+151000.0
67	YSI B Mix 3000 @ 25℃ -40 ~ 150℃	Engineering unit	+150.00	-040.00
		% of FSR	+100.00	-026.67
		2's comp HEX	7FFF	DDDE
		Ohms	+000055.6	+101000.0

Type Code	Thermistor Type	Data Format	+F.S.	-F.S.
68	YSI B Mix 5000 @ 25℃ -40 ~ 150℃	Engineering unit	+150.00	-040.00
		% of FSR	+100.00	-026.67
		2's comp HEX	7FFF	DDDE
		Ohms	+000092.7	+168300.0
69	YSI B Mix 6000 @ 25℃ -30 ~ 150℃	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000111.5	+106200.0
6A	YSI B Mix 10K @ 25℃ -30 ~ 150℃	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000185.9	+177000.0
6B	YSI H Mix 10K @ 25℃ -30 ~ 150℃	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000237.0	+135200.0
6C	YSI H Mix 30K @ 25℃ -10 ~ 200℃	Engineering unit	+200.00	-010.00
		% of FSR	+100.00	-005.00
		2's comp HEX	7FFF	F99A
		Ohms	+000186.7	+158000.0
70 ~ 77	User-defined -50 ~ 150℃	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+000000.0	+000000.0

Note:

1. For user-defined types, if the resistance is larger than 204800 ohms, then it is treated as under range
2. Refer to Section 1.11 for more information regarding user-defined types.

Thermistor Over Range/Under Range Reading for the tM-TH8 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

Thermistor Over Range/Under Range Reading for the tM-TH8 with Modbus protocol

	Over Range	Under Range
Engineering Unit	32767	-32768
2’s Complement HEX	7FFF	8000

1.11 User-defined Types

For a typical thermistor, the relationship between resistance and temperature can be expressed by the Steinhart–Hart equation

$$1/T = A + B \ln R_T + C (\ln R_T)^3$$

where R_T is the resistance in ohms at temperature T in degrees Kelvin (K = ℃ + 273.15). The values of A, B, and C are called Steinhart Coefficients. The error of the equation is less than +/- 0.01℃ in a 100℃ span.

The tM-TH8 supports user-defined types by specifying the Steinhart coefficients using the @AASxTttC(data) command, see Section 2.33. The data sent is a 32-bit hexadecimal value in IEEE-754 standard format:

Bits	Description
31 (sign)	0 = positive, 1 = negative
30-23 (exponent)	The exponent base is 2. The actual exponent is calculated by subtracting 127 from the stored value.
22-00 (mantissa)	The mantissa is expressed as 1.f where f is the fractional part and is stored in this field.

Example:
For the hexadecimal value C3694000h

1. Bit 31is 1, indicating a negative number.
2. The value of bits 30-23 is 10000110 binary or 134 decimal. The exponent is 7, 134-127.
3. The mantissa is 1.11010010100000000000000 binary.
4. Adjust the mantissa for the exponent. The adjusted mantissa is 11101001.0100000000000000 binary. That is 233.25 decimal.
5. Considering the sign bit. The floating-point number of C3694000h is –233.25.

The following methods can be used to obtain the Steinhart Coefficients of a thermistor.

1. Ask the manufacturer. Usually, thermistor manufacturers publish data for the Steinhart Coefficients for their thermisotr products.
2. Solve the coefficients by using three known data sets: (R1, T1), (R2, T2) and (R3, T3), which can be obtained by referring to the published Resistance versus Temperature data table of the thermistor. They can be obtained by measurement if no published data table is available. In order to achieve the most accurate results, the following values are recommended:
   -40℃ $\leqq$ T1, T2, T3 $\leqq$ 150℃ and
   |T2 – T1| $\leqq$ 50℃, |T3 – T2| $\leqq$ 50℃.

The DCON Utility, available for downloaded from the ICP DAS website (http://www.icpdas.com), can assist with the calculation of the Steinhart Coefficients and the conversion to IEEE-754 format. Documentation for the DCON Utility can be found in the ”**Getting Started For I-7000 Series Modules**” manual.

1.12 tM-TH8 Notes

The Modbus RTU communication protocol is the default protocol of the tM series. 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.

1.12.1 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 3.6.4 for details. Or, set both of the Modbus registers 00257 and 00258 to zero.
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 for the tM-TH8, the slide switch of the module should be set to the INIT position, see the figure on the next page. See Section 2.17 for details.
2. After a power-on reset, the communication protocol will be changed to the Modbus RTU protocol.

1.12.2 INIT Mode

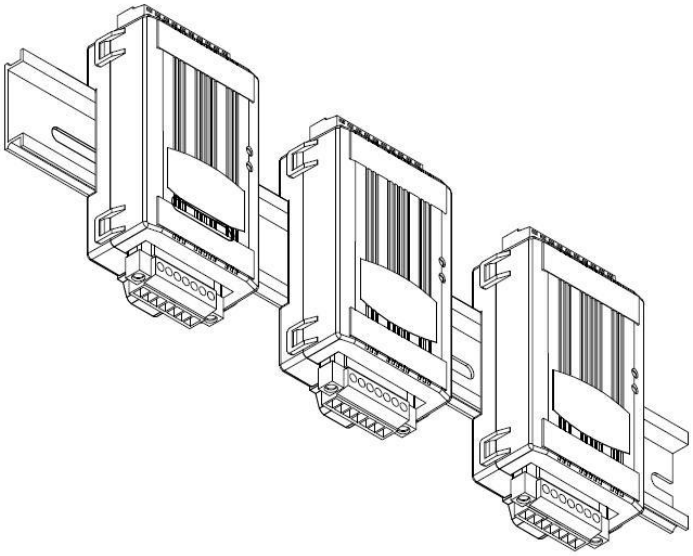
When the module is powered on, with the slide switch set to the INIT position as shown in the figure below for the tM-TH8, the module is in INIT mode, (see Section A.1 for details), and the communication settings are as follows:

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 2.16). To set new settings, send the commands %AANNTTCCFF, (see Section 2.1) and $AAPN, (see Section 2.17). The new communication settings will be effective after the next power-on reset.

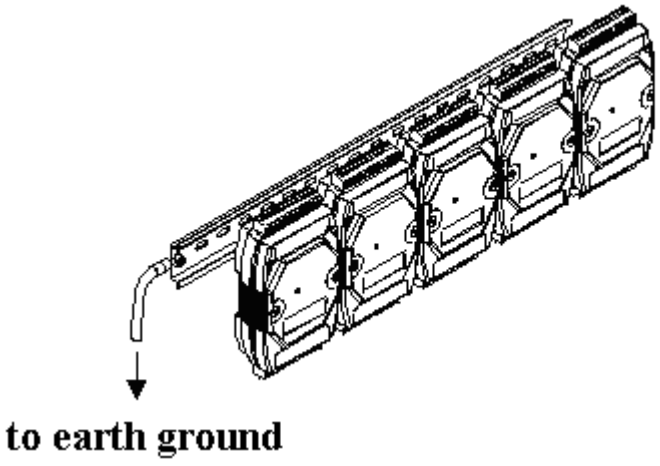
1.13 Mounting

1.13.1 Din-Rail Mounting

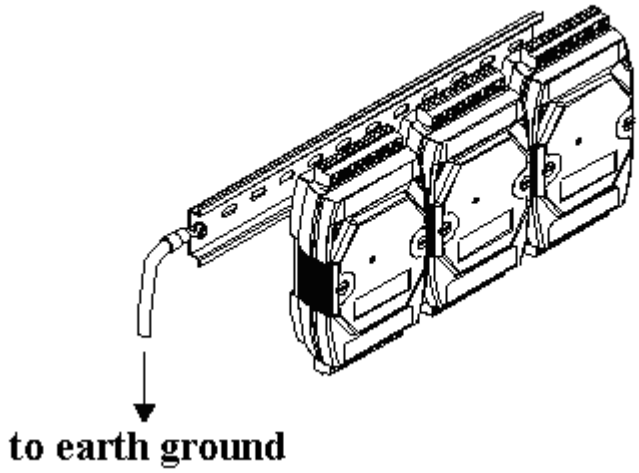


There are three new DIN rail models available. Each is made of stainless steel, which is stronger than those made of aluminum. There is a screw at one end and a ring terminal is included so that it can be easily connected to the earth ground. The three new DIN rail models are as follows.

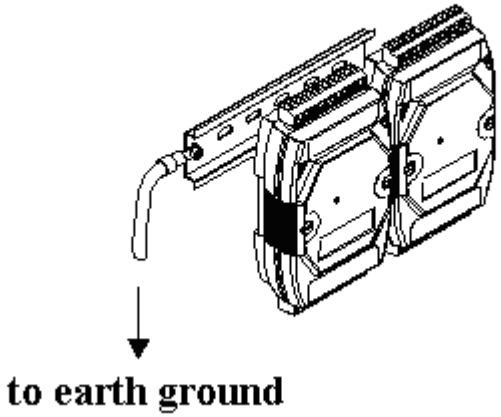
Part number	Max. number of modules	Dimensions
DRS-360	6	360mm x 35mm



Part number	Max. number of modules	Dimensions
DRS-240	4	240mm x 35mm



Part number	Max. number of modules	Dimensions
DRS-125	2	125mm x 35mm



Note: It is recommended that a 16 – 14 AWG wire is used to connect the DIN rail to the earth ground.

1.14 Technical Support

Should you encounter any problems while using the tM-TH8 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 2.13 and 3.6.7 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. Any 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

All communication with I-7000 modules consists of commands generated by the host and responses transmitted by the I-7000 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 2.34), which are sent to all modules, but in both of 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 which is present when the checksum setting is enabled. See Sections 1.10 (Data Format Setting) 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	Sets the module configuration	2.1
#AA	>(Data)	Reads the analog inputs of all channels	2.2
#AAN	>(Data)	Reads the analog input of a specified channel	2.3
$AA0	!AA	Performs span calibration	2.4
$AA1	!AA	Performs zero calibration	2.5
$AA2	!AANNTTCCFF	Reads the module configuration	2.6
$AA5	!AAS	Reads the module reset status	2.7
$AA5VV	!AA	Enables/Disables a channel	2.8
$AA6	!AAVV	Reads the channel enabled/disabled status	2.9
$AA7CiRrr	!AA	Sets a single channel range configuration	2.10
$AA8Ci	!AACiRrr	Reads a single channel range configuration	2.11
$AAB	!AANN	Reads the channel diagnostic status	2.12
$AAF	!AA(Data)	Reads the firmware version	2.13
$AAI	!AAS	Reads the INIT status	2.14
$AAM	!AA(Data)	Reads the module name	2.15
$AAP	!AASC	Reads the communication protocol	2.16
$AAPN	!AA	Sets the communication protocol	2.17
$AAS1	!AA	Reloads the default calibration parameters	2.18
~AAD	!AAT	Reads the temperature scaling	2.19
~AADT	!AA	Sets the temperature scaling to either ℃ or ℉	2.20
~AAEV	!AA	Enables/Disables calibration	2.21
~AAI	!AA	Soft INIT	2.22
~AAO(Name)	!AA	Sets the module name	2.23
~AARD	!AAVV	Reads the response delay time	2.24
~AARDVV	!AA	Sets the response delay time	2.25
~AATnn	!AA	Sets the soft INIT timeout	2.26

User-defined Thermistor Type Setting Command Sets			
Command	**Response**	**Description**	**Section**
@AAA2CiToo	!AA	Sets the temperature offset of a channel	2.27
@AAA3Ci	!AAoo	Reads the temperature offset of a channel	2.28
@AAA6CiRrr	!AA	Sets the resistance offset of a channel	2.29
@AAA7Ci	!AArr	Reads the resistance offset of a channel	2.30
@AAGxTtt	!AA(data)	Reads the Steinhart coefficient of a user-defined type	2.31
@AARTTttR(data)	!AA(data)	Reads the temperature corresponding to the resistance of a user-defined type	2.32
@AASxTttC(data)	!AA	Sets the Steinhart coefficient of a user-defined type	2.33

Host Watchdog Command Sets			
Command	**Response**	**Description**	**Section**
~**	No Response	Host is OK	2.34
~AA0	!AASS	Reads the host watchdog status	2.35
~AA1	!AA	Resets the host watchdog status	2.36
~AA2	!AAETT	Reads the host watchdog timeout settings	2.37
~AA3ETT	!AA	Sets the host watchdog timeout settings	2.38

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 1.10 for details. Not used by the tM-TH8. For the tM-TH8 use the $AA7CiRrr command to set the type of each channel, see Section 2.10 for details.
CC	New Baud Rate code, see Section 1.10 for details. For the tM-TH8, the slide switch must be moved to the INIT position in order to change Baud Rates. See Section A.1 for details.
FF	Used to set the data format, checksum, and filter settings (Section 1.10). For the tM-TH8, the slide switch must be moved to the INIT position in order to change the checksum setting. See Section A.1 for details.

Response:

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

! Delimiter for a valid command

? Delimiter for an invalid command. If the **Baud Rate** or **checksum** settings are changed without switching the slide switch to the INIT 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: %0102200600 Response: !02

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

Command: %0202200603 Response: !02

Sets the data format of module 02 to be 3 (Ohms). The module returns a valid response.

Command: %0101200A00 Response: ?01

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

Command: %0101200A00 Response: !01

Changes 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, Section 2.22 ~AAI, Section 2.26 ~AATnn

Related Topics:

Section 1.10 Configuration Tables, Section A.1 INIT* pin Operation

Notes:

1. Changes to the address, type code and data format 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. For the tM-TH8, changing the Baud Rate and checksum settings can be achieved by software only and is performed by using the following commands:
   I. Send a ~AATnn command. See Section 2.26 for details.
   II. Send a ~AAI command. See Section 2.22 for details.
   III. Send a %AANNTTCCFF command.

   If the command is valid, the Baud Rate and checksum settings will be changed after the module responds with !AA.

2.2 #AA

Description:

Reads the data from all analog input channels.

Syntax:

#AA[CHKSUM](CR)

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

Response:

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

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

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	Data from all analog input channels, see Section 1.10 for the data format. For the tM-TH8, data from disabled channels is filled with space characters.
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: #01 Response: >+026.35

Reads module 01 and receives the data in engineering format.

Command: #02 Response: >4C53

Reads module 02 and receives the data in hexadecimal format.

Command: #03 Response: >-9999.9

Reads module 03 and the data is under range.

Command: #04 Response: >+025.12+054.12+150.12

Reads module 04, an I-7033, and receives the data from 3 channels.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:

Section 1.10 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 Command: **>(Data)[CHKSUM](CR)**

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

>	Delimiter character for a valid command
?	Delimiter character for an invalid command. An invalid command is returned if the specified channel is incorrect.
(Data)	Analog input data of the specified channel, see Section 1.10 for the data format. For the tM-TH8, if the specified channel is disabled, then the data field will be filled with space characters.
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.

Command: #029 Response: ?02

Reads data from channel 9 of module 02. An error is returned because channel 9 is invalid.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:

Section 1.10 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 Command: **!AA[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command. An invalid command is returned if the specified channel is incorrect.
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 and returns a valid response.

Command: $030 Response: ?03

Performs a span calibration on module 03. An invalid command is returned because the "enable calibration" command was not sent in advance.

Related Commands:

Section 2.5 $AA1, Section 2.21 ~AAEV

Related Topics:

Section 1.9 Calibration

Notes:

The "enable calibration" command, ~AAEV, must be sent before this command is used, see Section 1.9 for details.

2.5 $AA1

Description:

Performs a zero calibration.

Syntax:

$AA1[CHKSUM](CR)

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

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command. An invalid command is returned if the specified channel is incorrect.
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 Receive: !01

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

Command: $021 Receive: !02

Performs a zero calibration on module 02 and returns a valid response.

Command: $031 Receive: ?03

Performs a zero calibration on module 03. An invalid command is returned because the “enable calibration” command was not sent in advance.

Related Commands:

Section 2.4 $AA0, Section 2.21 ~AAEV

Related Topics:

Section 1.9 Calibration

Notes:

The “enable calibration” command, ~AAEV, must be sent before this command is used, see Section 1.9 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 Command: **!AATTCCFF[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
TT	Type code of the module, see Section 1.10 for details.
CC	Baud Rate code of the module, see Section 1.10 for details.
FF	Data format, checksum settings and filter settings of the module, see Section 1.10 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: !01200600

Reads the configuration of module 01.

Command: $022 Response: !02230602

Reads the configuration of module 02.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.10 Configuration Tables

2.7 $AA5

Description:

Reads the reset status of a module.

Syntax:

$AA5[CHKSUM](CR)

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

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
S	Reset status of the module 1: This is the first time the command has been sent since the module was powered on. 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.

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 response shows 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 response shows that there has been no module reset since the last $AA5 command was sent.

2.8 $AA5VV

Description:

Specifies the channels to be enabled.

Syntax:

$AA5VV[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
5	Command to set the channels to enabled
VV	A two-digit hexadecimal value, 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:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command. 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: $0153A 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: !013A

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

Related Commands:

Section 2.9 $AA6

Notes:

It is recommended that only the channels to 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 Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, 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.

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: $0153A 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: !013A

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

Related Commands:

Section 2.8 $AA5VV

2.10 $AA7CiRrr

Description:

Sets the type code of a channel.

Syntax:

$AA7CiRrr[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
7	Command to set the channel range code
Ci	i specifies the input channel to be set (0-7)
Rrr	rr represents the type code of the channel to be set. Refer to the Temperature Sensor Type Setting table in Section 1.10.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
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: $017C0R60 Response: !01

Sets the type code for channel 0 of module 01 to be 60 and the module returns a valid response.

Command: $027C5R72 Response: !02

Sets the type code for channel 5 of module 02 to be 72 and the module returns a valid response.

Command: $037C1R30 Response: ?03

Sets the type code for channel 1 of module 03 to be 30. The module returns an invalid response because the type code is invalid.

Related Commands:

Section 2.11 $AA8Ci

Related Topics:

Section 1.10 Configuration Tables

2.11 $AA8Ci

Description:

Reads the type code information of a channel.

Syntax:

$AA8Ci[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
8	Command to read the type code of a channel
Ci	Specifies which channel to access for the type code information (i = 0 ~ 7)

Response:

Valid command: **!AACiRrr[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid channel
AA	Address of the responding module (00 to FF)
Ci	Specifies which input channel the type code information relates to.
Rrr	Represents the type code of the specified input channel. Refer to the Temperature Sensor Type Setting table in Section 1.10.

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: $018C0 Response: !01C0R70

Reads the input range of channel 0 of module 01 and returns 70.

Related Commands:

Section 2.10 $AA7CiRrr

Related Topics:

Section 1.10 Configuration Tables

2.12 $AAB

Description:

Diagnoses the analog inputs for over-range, under-range, and wire opening conditions.

Syntax:

$AAB[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be diagnosed (00 to FF)
B	Command to diagnose the analog inputs

Response:

Valid command: **!AANN[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
NN	Represents the diagnostic results of all the analog input channels (00 to FF), 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 it is in either over-range, under-range or wire opening condition. If the bit is 0 it denotes that the channel is disabled or normal.

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: $01B Response: !0101

Diagnoses the analog inputs of module 01. The module returns a valid response denoting that channel 1 is in either over-range, under-range or wire opening condition.

2.13 $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 command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Data)	Firmware version string 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.

Command: $02F Response: !02B1.1

Reads the firmware version of module 02, and shows that it is version B1.1.

2.14 $AAI

Description:

Reads the INIT status of a module.

Syntax:

$AAI[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be diagnosed (00 to FF)
I	Command to read the module INIT status

Response:

Valid command: **!AAS[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
S	INIT switch status of the module 0: The INIT switch is moved to the INIT position 1: The INIT switch is moved to 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 INIT status of module 01. The response shows that the INIT switch is moved to the INIT position.

2.15 $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 command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Name)	Name string 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: !01tTH8

Reads the module name of module 01 and returns the name “tTH8”.

Related Commands:

Section 2.23 ~AAO(Name)

2.16 $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 3: The DCON, Modbus RTU and Modbus ASCII 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 3: the protocol set in EEPROM is Modbus ASCII

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: !0130

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

Related Commands:

Section 2.17 $AAPN

2.17 $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 3: Modbus ASCII protocol Before using this command, the slide switch must be in the INIT position for tM-TH8, 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 2.16 $AAP

Related Topics:

Section A.1 INIT Mode

2.18 $AAS1

Description:

Reloads the factory default calibration parameters, including the internal calibration parameters.

Syntax:

$AAS1[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to have the default parameters reloaded (00 to FF)
S1	Command to reload the factory default calibration parameters

Response:

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: $01S1 Response: !01

Sends the command to reload the factory default calibration parameters for module 01 and returns a valid response.

Related Topics:

Section 1.9 Calibration

2.19 ~AAD

Description:

Reads the temperature scale information of a module.

Syntax:

~AAD[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
D	Command to read the temperature scale

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
T	Temperature scale 0: the temperature scale is Celsius. 1: the temperature scale is Fahrenheit.

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: ~01D Response: !010

Reads the temperature scale of module 01 and returns Celsius.

Command: ~02D Response: !021

Reads the temperature scale of module 02 and returns Fahrenheit.

Related Commands:

Section 2.20 ~AADT

Notes:

The default temperature scale is Celsius.

2.20 ~AADT

Description:

Sets the temperature scale of a module to Celsius or Fahrenheit.

Syntax:

~AADT[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
D	Command to set the temperature scale
T	C: set the temperature scale to Celsius. F: set the temperature scale to Fahrenheit.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: ~01DC Response: !01

Sets the temperature scale of module 01 to Celsius, and returns a valid response.

Command: ~02DF Response: !02

Sets the temperature scale of module 02 to Fahrenheit, and returns a valid response.

Related Commands:

Section 2.19 ~AAD

Notes:

1. The default temperature scale is Celsius.
2. The commands affected are #AA, #AAN, and @AARTTttR(Data).

2.21 ~AAEV

Description:

Enables/Disables the module calibration.

Syntax:

~AAEV[CHKSUM](CR)

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

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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. An invalid response is returned because the “enable calibration” command has not yet been sent.

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 1.9 Calibration

2.22 ~AAI (Soft INIT command)

Description:

The Soft INIT command is used to enable modification of the Baud Rate and checksum settings using software only.

Syntax:

~AAI[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
I	Command to set the Soft INIT

Response:

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: ~01I Response: !01

Sets the soft INIT of module 01 and returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.26 ~AATnn

Related Topics:

Section A.1 INIT* Pin Operation

Notes:

The ~AATnn command should be sent prior to sending this command, see Section 2.26 for details.

2.23 ~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 command: **!AA[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: ~01O7005N Response: !01

Sets the name of module 01 to be “7005N” and returns a valid response.

Command: $01M Response: !017005N

Reads the name of module 01 and returns the name “7005N”.

Related Commands:

Section 2.15 $AAM

2.24 ~AARD

Description:

Reads the response delay time value of a module.

Syntax:

~AARD[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
RD	Command to read the response delay time value

Response:

Valid command : **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	Two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30ms. The max allowable value is 30 (1Eh).

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: ~01RD Response: !0102

Reads the response delay time value of module 01 and returns 02, which denotes that the response delay time value is 2ms.

Related Commands:

Section 2.25 ~AARDVV

2.25 ~AARDVV

Description:

Sets the response delay time value of a module.

Syntax:

~AARDVV[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
RD	Command to set the response delay time
VV	Two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30s. The max allowable value is 30 (1Eh).

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: ~01RD06 Response: !01

Sets the response delay time value to 6ms. The module returns a valid response.

Command: ~01RD Response: !0106

Reads the response delay time value of module 01. The module returns 06, which denotes that the response delay time value is 6ms.

Related Commands:

Section 2.24 ~AARD

2.26 ~AATnn

Description:

Sets the soft INIT timeout value.

Syntax:

~AATnn[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
T	Command to set the soft INIT timeout
nn	Two hexadecimal digits representing the timeout value in seconds. The maximum timeout value is 60 seconds. When changing the Baud Rate or checksum settings without altering the INIT* status, the ~AAI and %AANNTTCCFF commands should be sent consecutively and the time interval between the two commands should be less than the soft INIT timeout. If the soft INIT timeout is 0, then the Baud Rate and checksum settings cannot be changed using software only. The power-on reset value of the soft INIT timeout is 0.

Response:

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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: ~01I Response: !01

Sets the soft INIT timeout value of module 01 and returns a valid response.

Command: %0101000700 Response: ?01

Attempts to change the Baud Rate of module 01 to 19200 without first altering the INIT * status. The module returns an invalid response because the soft INIT timeout value is 0.

Command: ~01T10 Response: !01

Sets the soft INIT timeout value of module 01 to 16 seconds and returns a valid response.

Command: ~01I Response: !01

Sets the soft INIT of module 01 and returns a valid response.

Command: %0101000700 Response: !01

Changes the Baud Rate of module 01 to 19200 without first altering INIT * status. The module returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.22 ~AAI

Related Topics:

Section A.1 INIT* Pin Operation

Notes:

It is recommended that the soft INIT timeout value is reset to 0 once any changes to the Baud Rate and checksum settings have been completed.

2.27 @AAA2CiToo

Description:

Sets the temperature offset of a channel.

Syntax:

@AAA2CiToo[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
A2	Command to set the temperature offset
Ci	i specify the input channel to be set (0 ~ 7)
Too	oo represents the temperature offset in 0.1℃/℉, -12.8 ~ 12.7. It is in hexadecimal format and 01 for 0.1, 02 for 0.2, FF for -0.1, FE for -0.2, etc.

.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
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: @01A2C2T0A Response: !01

Sets the temperature offset of channel 2 to 1.0, and the module returns a valid response.

Command: @01A2C5TF0 Response: !01

Sets the temperature offset of channel 5 to -1.6, and the module returns a valid response.

Related Commands:

Section 2.28 @AAA3Ci

2.28 @AAA3Ci

Description:

Reads the temperature offset of a channel.

Syntax:

@AAA3Ci [CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
A3	Command to read the temperature offset
Ci	i specify the input channel to be set (0 ~ 7)

.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
AA	Address of the responding module (00 to FF)
(Data)	Two hexadecimal digits to represent the temperature offset. It is 01 for 0.1, 02 for 0.2, FF for -0.1, FE for -0.2, etc.

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: @01A3C2 Response: !010A

Reads the temperature offset of channel 2, and the module returns a value of 1.0.

Command: @01A3C5 Response: !01F0

Reads the temperature offset of channel 5, and the module returns a value of -1.6.

Related Commands:

Section 2.27 @AAA2CiToo

2.29 @AAA6CiRrr

Description:

Sets the resistance offset of a channel.

Syntax:

@AAA6CiRrr[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
A6	Command to set the resistance offset
Ci	i specify the input channel to be set (0 ~ 7)
Rrr	rr represents the resistance offset in 0.1 ohms, 0.0 ohms ~ 25.5 ohms. It is in hexadecimal format and 01 for 0.1, 02 for 0.2, FF for 25.5, etc.

.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
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: @01A6C2T0A Response: !01

Sets the resistance offset of channel 2 to 1.0, and the module returns a valid response.

Command: @01A6C5TF0 Response: !01

Sets the resistance offset of channel 5 to 24.0, and the module returns a valid response.

Related Commands:

Section 2.30 @AAA7Ci

2.30 @AAA7Ci

Description:

Reads the resistance offset of a channel.

Syntax:

@AAA7Ci [CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
A7	Command to read the resistance offset
Ci	i specify the input channel to be set (0 ~ 7)

.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
AA	Address of the responding module (00 to FF)
(Data)	Two hexadecimal digits to represent the resistance offset. It is 01 for 0.1 ohms, 02 for 0.2 ohms, FF for 25.5 ohms, FE for 25.4 ohms, etc.

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: @01A7C2 Response: !010A

Reads the resistance offset of channel 2, and the module returns a value of 1.0.

Command: @01A7C5 Response: !01F0

Reads the resistance offset of channel 5, and the module returns a value of 24.0.

Related Commands:

Section 2.29 @AAA6CiRrr

2.31 @AAGxTtt

Description:

Reads the Steinhart coefficient of a user-defined type.

Syntax:

@AAGxTtt[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
G	Command to read the Steinhart coefficient
x	A: reads Steinhart coefficient A B: reads Steinhart coefficient B C: reads Steinhart coefficient C
Ttt	tt represents the type code of the user-defined type to read the Steinhart coefficient.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
AA	Address of the responding module (00 to FF)
(Data)	Eight hexadecimal digits to represent the Steinhart coefficient in IEEE-754 format. See Section 1.11 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: @01GAT70 Response: !013A94030A

Reads the A coefficient of type 70 of module 01 and returns a value of 3A94030A, which is equivalent to $1.129241x10^{-3}$.

Command: @01GBT70 Response: !0139757ACF

Reads the B coefficient of type 70 of module 01 and returns a value of 39757ACF, which is equivalent to $2.341077x10^{-4}$.

Command: @01GCT70 Response: !0133BC73A5

Reads the C coefficient of type 70 of module 01 and returns a value of 33BC73A5, which is equivalent to $8.775468x10^{-8}$.

Related Commands:

Section 2.32 @AARTTttR(data), Section 2.33 @AASxTttC(data)

Related Topics:

Section 1.11 User-defined Types, Section A.5 Thermistor

2.32 @AARTTttR(Data)

Description:

Reads the temperature associated with the input resistance of a user-defined type.

Syntax:

@AARTTttR(Data)[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
RT	Command to read the temperature associated with the input resistance of a user-defined type
Ttt	tt represents the type code of the user-defined type.
R(Data)	The (Data) consists of seven decimal digits, or five digits, decimal point and one additional digit, to represent the input resistance.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
AA	Address of the responding module (00 to FF)
(Data)	The (Data) consists of a +/- sign, three digits, decimal point and two additional digits to represent the temperature associated with the input resistance.

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: @01RTT70R0104500
Response: !01-032.64

Reads the temperature associated with 104500 ohms of type 70 of module 01, and the module returns a value of –32.64 degree.

Command: @01RTT70R00801.2
Response: !01+072.62

Reads the temperature associated with 801.2 ohms of type 70 of module 01, and the module returns a value of 72.62 degree.

Related Commands:

Section 2.19 ~AAD, Section 2.20 ~AADT, Section 2.31 @AAGxTtt, Section 2.33 @AASxTttC(data)

Related Topics:

Section A.6 Thermistor

Notes:

The temperature scale is set using the ~AADT command and can be read using the ~AAD command. See Sections 2.19 and 2.20 for details.

2.33 @AASxTttC(data)

Description:

Sets the Steinhart coefficient of a user-defined type.

Syntax:

@AASxTttC(data)[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be set (00 to FF)
S	Command to set the Steinhart coefficient
x	A: sets Steinhart coefficient A B: sets Steinhart coefficient B C: sets Steinhart coefficient C
Ttt	tt represents the type code of the user-defined type to set the Steinhart coefficient.
C(Data)	The (Data) is eight hexadecimal digits in IEEE-754 format to represent the Steinhart coefficient. See Section 1.11 for details.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid type code
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: @01SAT70C3A94030A Response: !01

Sets the A coefficient of type 70 of module 01 to be 3A94030A, which is equivalent to 1.129241×10^{-3}, and the module returns a valid response.

Command: @01SBT70C39757ACF Response: !01

Sets the B coefficient of type 70 of module 01 to be 39757ACF, which is equivalent to 2.341077×10^{-4}, and the module returns a valid response.

Command: @01SCT70C33BC73A5 Response: !01

Sets the C coefficient of type 70 of module 01 to be 33BC73A5, which is equivalent to 8.775468×10^{-8}, and the module returns a valid response.

Related Commands:

Section 2.31 @AAGxTtt, Section 2.32 @AARTTttR(data)

Related Topics:

Section 1.11 User-defined Types, Section A.5 Thermistor

Notes:

If the resistance versus temperature data table of the thermistor is available, it is suggested that the @AARTTttR(data) command be used to verify the accuracy of the user-defined type.

2.34 ~**

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 2.35 ~AA0, Section 2.36 ~AA1, Section 2.37 ~AA2, Section 2.38 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.35 ~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 command: **!AASS[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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 that the host watchdog is enabled, Bit 2: 1 indicates that a host watchdog timeout has occurred, and 0 indicates that no host watchdog timeout has occurred. The host watchdog status is stored in EEPROM and can only be reset by 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 timeout 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 2.34 ~**, Section 2.36 ~AA1, Section 2.37 ~AA2, Sec 2.38 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.36 ~AA1

Description:

Resets the host watchdog timeout status of a module.

Syntax:

~AA1[CHKSUM](CR)

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

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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 timeout has occurred.

Command: ~011 Response: !01

Resets the host watchdog timeout 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 timeout has occurred.

Related Commands:

Section 2.34 ~**, Section 2.35 ~AA0, Section 2.37~AA2, Section 2.38~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.37 ~AA2

Description:

Reads the host watchdog timeout 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 timeout value

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
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 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: ~012 Response: !011FF

Reads the host watchdog timeout value of module 01 and returns FF, which denotes that the host watchdog is enabled and the host watchdog timeout value is 25.5 seconds.

Related Commands:

Section 2.34 ~**, Section 2.35 ~AA0, Section 2.36 ~AA1, Section 2.38 ~AA3EVV

Related Topics:

Section A.2 Dual Watchdog Operation

2.38 ~AA3EVV

Description:

Enables/disables the host watchdog and sets the host watchdog timeout 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 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 for a valid command
?	Delimiter character for an invalid command
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 timeout value to 10.0 seconds. The module returns a valid response.

Command: ~012 Response: !01164

Reads the host watchdog timeout value of module 01. The module returns 164, which denotes that the host watchdog is enabled and the host watchdog timeout value is 10.0 seconds.

Related Commands:

Section 2.34 ~**, Section 2.35 ~AA0, Section 2.36 ~AA1, Section 2.37 ~AA2, Section

Related Topics:

Section A.2 Dual Watchdog Operation

Notes:

When a host watchdog timeout occurs, the host watchdog is disabled. The ~AA3EVV command should be sent again to reenable the host watchdog.

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.

The tM-TH8 supports the Modbus RTU and Modbus ASCII protocols. The communication Baud Rates range from 1200bps to 115200bps. The following Modbus functions are supported by the tM-TH8.

Function Code	Description	Section
01 (0x01)	Read digital output status	3.1
02 (0x02)	Read input status	3.2
04 (0x04)	Read input channels	3.3
05 (0x05)	Write single digital output	3.4
15 (0x0F)	Write multiple digital outputs	3.5
70 (0x46)	Read/write module settings	3.6

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 01 (0x01) Read Coil Registers

This function code is used to read the digital output channel status of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x01
02 ~ 03	Starting channel	2 Bytes	256 to 272
04 ~ 05	Number of digital output channels	2 Bytes	N, 1 to 16

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x01
02	Byte count	1 Byte	1
03	Data of digital output channels	1 Byte	A bit corresponds to a channel. When the bit is 1 it denotes that the channel is ON. If the bit is 0 it denotes that the channel is OFF.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x81
02	Exception code	1 Byte	02: starting channel out of range 03: out of range, incorrect number of bytes received

3.2 02 (0x02) Read Input Status

This function code is used to read the thermistor 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 0x87, where 0x80 corresponds to channel 0, 0x81 corresponds to channel 1, etc
04 ~ 05	Number of input channels	2 Bytes	N, 1 to 8; (Starting channel + N) should be less than or equal to 0x88

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

3.3 04 (0x04) Read 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 7
04 ~ 05	Number of input channels (N)	2 Bytes	1 to 8; (Starting channel + N) <= 8.

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	Data in 2's complement hex format.

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

3.4 05 (0x05) Write Single Coil Register

This function code is used to write to one digital output channel.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x05
02 ~ 03	Channel	2 Bytes	256 to 271
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	Channel	2 Bytes	256 to 271
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	02: channel out of range

3.5 15 (0x0F) Write Multiple Coil Registers

This function code is used to write to multiple digital output channels.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x0F
02 ~ 03	Starting channel	2 Bytes	256 to 271
04 ~ 05	Number of channels (N)	2 Bytes	1 to 16
06	Byte count	1 Byte	1
07 ~	Data of digital output channels	1 Byte	A bit corresponds to a coil. When the bit is 1 it denotes that the coil is ON. If the bit is 0 it denotes that the coil is OFF.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x0F
02 ~ 03	Starting channel	2 Bytes	256 to 271
04 ~ 05	Number of channels (N)	2 Bytes	1 to 16

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x8F
02	Exception code	1 Byte	02: starting channel out of range 03: out of range, incorrect number of bytes received

3.6 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	3.6.1
04 (0x04)	Set the module address	3.6.2
05 (0x05)	Read the communication settings	3.6.3
06 (0x06)	Set the communication settings	3.6.4
07 (0x07)	Read the type code	3.6.5
08 (0x08)	Set the type code	3.6.6
32 (0x20)	Read the firmware version	3.6.7
37 (0x25)	Read the channel enable/disable status	3.6.8
38 (0x26)	Set the channel enable/disable	3.6.9
41 (0x29)	Read the miscellaneous settings	3.6.10
42 (0x2A)	Write the miscellaneous settings	3.6.11

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

3.6.1 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	0x07 0x00 0x80 0x03 for tM-TH8

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

3.6.2 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

3.6.3 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	Protocol supported	1 Byte	0x00: Modbus RTU 0x03: Modbus RTU & ASCII
04	Baud Rate	1 Byte	Baud Rate code, see Section 1.10 for details.
05	Reserved	1 Byte	0x00
06	Data format	1 Byte	0x00: no parity, 1 stop bit 0x01: no parity, 2 stop bits 0x02: even parity, 1 stop bit 0x03: odd parity, 1 stop bit
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0x00: DCON protocol 0x01: Modbus RTU protocol 0x03: Modbus ASCII 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

3.6.4 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 1.10 for details.
05	Reserved	1 Byte	0x00
06	Data format	1 Byte	0x00: no parity, 1 stop bit 0x01: no parity, 2 stop bits 0x02: even parity, 1 stop bit 0x03: odd parity, 1 stop bit
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0x00: DCON protocol 0x01: Modbus RTU protocol 0x03: Modbus ASCII 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	0x00: OK, others: error
05	Reserved	1 Byte	0x00 0x00 0x00
06	Data format	1 Byte	0x00: OK, others: error
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0x00: 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

3.6.5 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 ~ 0x07 for tM-TH8

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 1.10 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, channel out of range, incorrect number of bytes received

3.6.6 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 ~ 0x07 for tM-TH8
05	Type code	1 Byte	Type code, see Section 1.10 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, channel out of range, reserved bytes should be filled with zero, incorrect number of bytes received

3.6.7 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

3.6.8 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	1 Byte	0x00 ~ 0xFF, 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

3.6.9 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	1 Byte	0x00 ~ 0xFF, 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

3.6.10 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	0, not used for tM-TH8

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

3.6.11 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	0, not used for tM-TH8

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

3.7 Modbus Address Mappings

Address	Description	Attribute
00257	Protocol, 0: DCON, 1: Modbus	R/W
00258	Modbus protocol, 0: RTU, 1: ASCII	
00261	1: enable, 0: disable host watchdog	R/W
00267	Temperature scale, 0: Fahrenheit, 1: Celsius	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
00272	Write 1 to load factory calibration parameters	W
00273	Reset status, 1: first read after powered on, 0: not the first read after powered on	R
10129 ~ 10136 00129 ~ 00136	Over/under range status of channel 0 to 7	R
30001 ~ 30008 40001 ~ 40008	Analog input value of channel 0 to 7	R
40257 ~ 40264	Type code of channel 0 to 7	R/W
40385 ~ 40392	Resistance offset of channel 0 to 7 in 0.1 ohms, valid range: 0 ~ 255	R/W
40449 ~ 40456	Temperature offset of channel 0 to 7 in 0.1℃/℉, valid range: -128 ~ 127	R/W

<table>
<tr><th>Address</th><th>Description</th><th>Attribute</th></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>
<tr><td>40488</td><td>Modbus response delay time in ms, valid range: 0 ~ 30</td><td>R/W</td></tr>
<tr><td>40489</td><td>Host watchdog timeout value, 0 ~ 255, in 0.1s</td><td>R/W</td></tr>
<tr><td>40490</td><td>Channel enable/disable</td><td>R/W</td></tr>
<tr><td>40492</td><td>Host watchdog timeout count, write 0 to clear</td><td>R/W</td></tr>
<tr><td>40769 ~ 40784</td><td>Steinhart Coefficient A of type code 70 to 77</td><td>R/W</td></tr>
<tr><td>40801 ~ 40816</td><td>Steinhart Coefficient B of type code 70 to 77</td><td>R/W</td></tr>
<tr><td>40833 ~ 40848</td><td>Steinhart Coefficient C of type code 70 to 77</td><td>R/W</td></tr>
</table>

3.8 Modbus Engineering Data Format Table

Type Code	Thermistor Type	-F.S.	+F.S.
60	PreCon Type III 10K @ 25℃	-3000	24000
61	Fenwell Type U 2K @ 25℃	-5000	15000
62	Fenwell Type U 2K @ 25℃	0	15000
63	YSI L Mix 100 @ 25℃	-8000	10000
64	YSI L Mix 300 @ 25℃	-8000	10000
65	YSI L Mix 1000 @ 25℃	-7000	10000
66	YSI B Mix 2252 @ 25℃	-5000	15000
67	YSI B Mix 3000 @ 25℃	-4000	15000
68	YSI B Mix 5000 @ 25℃	-4000	15000
69	YSI B Mix 6000 @ 25℃	-3000	15000
6A	YSI B Mix 10000 @ 25℃	-3000	15000
6B	YSI H Mix 10000 @ 25℃	-3000	15000
6C	YSI H Mix 30000 @ 25℃	-1000	20000
70	User-defined	-5000	15000
71	User-defined	-5000	15000
72	User-defined	-5000	15000
73	User-defined	-5000	15000
74	User-defined	-5000	15000
75	User-defined	-5000	15000
76	User-defined	-5000	15000
77	User-defined	-5000	15000

The under range value is –32768 and the over range value is +32767. For the hex data format, please refer to Section 1.10.

4. Troubleshooting

If you are having difficulty using the tM series 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 1.14.

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 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 and no checksum. 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:

- Ensure that the type code and data format settings are correct. For the tM-TH8, the type code is set by using the $AA7CiRrr command, see Section 2.10 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.
- Connect a resistor, instead of thermistor sensor, using the thermistor connection as shown in Section 1.6. The resistance of the resistor should be less than 180000 ohms. Change the data format to ohms, see Sections 1.10 and 2.1, and check whether the data reading is correct.
- If the resistance read by the module is still incorrect, then it may be because the calibration parameters stored in the non-volatile memory are corrupted, which means that the module should be recalibrated. Be sure to read Section 1.9 in detail before performing the calibration. For the tM-TH8, you can send the $AAS1 command, see Section 2.18 for details, to reload the factory calibration parameters.

A. Appendix

A.1 INIT Mode

Each tM 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 tM 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 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 2.17 for details.

The tM modules have the INIT switch located on the right side of the module 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.

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 timeout occurs, the module will reset all outputs to a safe state in order to prevent any erroneous operations of the controlled target.

The tM 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 Frame Ground

Electronic circuits are constantly vulnerable to ESD which become worse in a continental climate area. The tM 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.

Connect the frame ground terminal to a wire/DIN rail and connect the wire/DIN rail to the earth ground will provide a better protection for the module.
New DIN rail models are available that can easily be connected to the earth ground. Each 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. Refer to Section 1.13.1 for more information about the new DIN rail models.

A.4 Reset Status

The reset status of a module is set when the module is powered-on or when the module is reset by the module watchdog. It is cleared after the responding of the first $AA5 command. This can be used to check whether the module had been reset. When the $AA5 command responds that the reset status is cleared, that means the module has not been reset since the last $AA5 command was sent. When the $AA5 command responds that the reset status is set and it is not the first time $AA5 command is sent, it means the module has been reset and the digital output value had been changed to the power-on value.

A.5 Thermistor

A thermistor is a temperature sensitive resistor that can be used in temperature measurement, control and compensation circuitry. For most thermistors, the resistance decreases as the temperature rises and the relationship can be expressed as:

$$1/T = A + B \ln R_T + C (\ln R_T)^3$$

where R_T is the resistance in ohms at the temperature T in Kelvins (K = ℃ + 273.15). This is called the Steinhart-Hart equation.

Thermistors typically work over a relatively small temperature range of -60℃ to 300℃. They are less stable than RTDs. However, they exhibit a large change in resistance proportional to a small change in temperature than RTDs and a much higher nominal resistance. They are less sensitive to lead resistance noise effects and can be transmitted over a large distance. No compensation for ambient temperature is needed. They are one of the most accurate types of temperature sensors.

A.6 Resistance Measurement

The tM-TH8 modules can be used for resistance measurement. To measure the resistance, connect the resistor using thermistor connection, as shown in Section 1.6, and set the data format to ohms. To set the data format, send the %AANNTTCCFF command to the module, see Section 2.1 for details, or use the DCON utility, refer to Chapter 2 of the "**Getting Started For I-7000 Series Modules**" manual that can be downloaded from the ICP DAS website http://www.icpdas.com.

The maximum resistance that can be measured by tM-TH8 is 204800 ohms.