ZT-2018
User Manual

Warranty

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

Warning

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

Copyright

Trademarks

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

Technical Support

If you have any problems, please feel free to contact us via email at service@icpdas.com.

Table of Contents

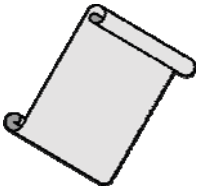
What's in the Shipping Package?

The shipping package contains the following items:


ZT-2018


ANT-124-05


DN-1824


Quick Start


CD

If any of these items are missing or damaged, please contact your local distributor for more information. Save the shipping materials and cartons in case you need to ship the module in the future.

More Information

- Documentation:

  All documentation related to the ZT Series of devices can be found on the companion CD at:

  CD: \Napdos\ZigBee\ZT_Series\Document

  Or can be downloaded from:
  http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zt_series/document

- Software:

  Utility software for the ZT Series of devices can be found on the companion CD at:

  CD: \Napdos\ZigBee\ZT_Series\Utility

  Or can be downloaded from:
  http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zt_series/utility

1 Introduction

1.1 Introduction to ZigBee

ZigBee is a specification for a suite of high-level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for personal area networks. ZigBee devices are often used in mesh network form to transmit data over longer distances, passing data through intermediate devices to reach more distant ones. This allows ZigBee networks to be formed ad-hoc, with no centralized control or high-power transmitter/receiver required in order to reach all of the devices. Any ZigBee device can be tasked with running the network.

ZigBee is targeted at applications that require a low data rate, long battery life, and secure networking. ZigBee has a defined transmission rate of 250 kbit/s, best suited for periodic or intermittent transmission of data, or for a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs.

1.2 Introduction to the ZT-2000 I/O Series

The ZT-2000 I/O series of devices are small wireless ZigBee I/O modules based on the IEEE802.15.4 standard that allow data acquisition and control via personal area ZigBee networks. See Section 3.1 for more detailed information.

The ZT-2000 I/O series is a wireless data acquisition-based client/server system. Accordingly, a Net Server for the ZigBee (ZT-2570/ZT-2550) is essential in such systems. For more information regarding any configuration issues related to the ZigBee Coordinator, please refer to the "ZT-25XX ZigBee Converter Quick Start" document, which can be found at:

http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zt_series/document/

2 Hardware Information

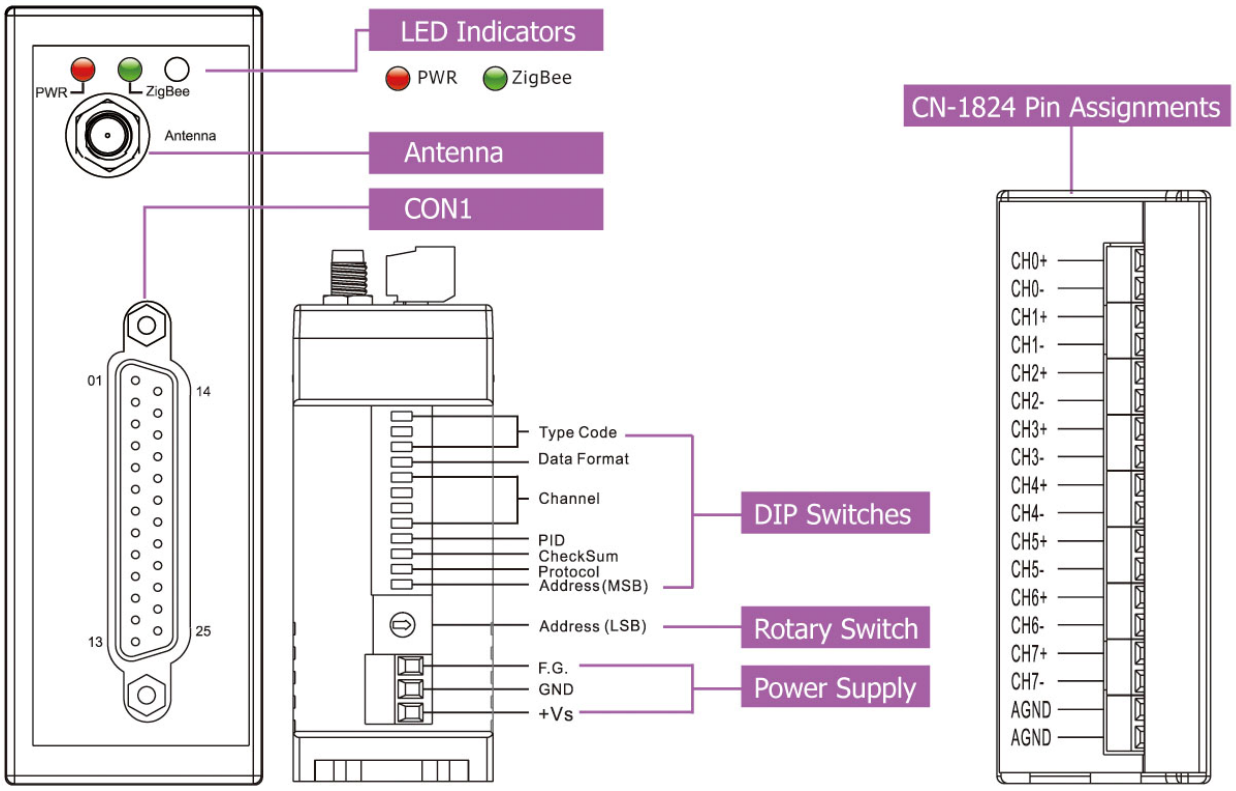
2.1 Specifications

ZT-2018

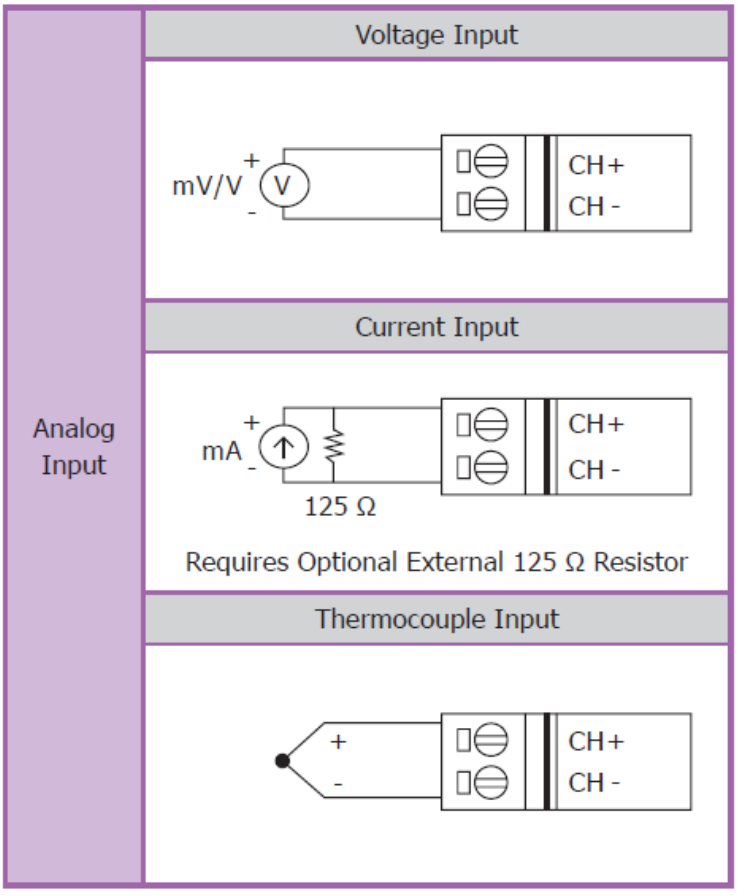
Analog Input	
Input Channels	8 Differential
Input Type	+/-15 mV, +/-50 mV, +/-100 mV, +/-500 mV, +/-1V, +/-2.5V, +/-20 mA, 0 ~ 20 mA, 4 ~ 20 mA (Current Input Requires Optional External 125 Ω Resistor). Thermocouple (J, K, T, E. R. S, B, N, C, L, M, LDIN43710)
Resolution	16-bit
Sampling Rate	16-bit, 10 Samples/Sec. (Total)
Accuracy	+/-0.1% FSR
-3dB Bandwidth	15.7 Hz
Zero Drift	+/-10 μV/° C
Span Drift	+/-25 ppm/°C
Common Mode Rejection	86 dB
Normal Mode Rejection	100 dB
Input Impedance	>400 kΩ
Open Thermocouple Detection	Yes
Overvoltage Protection	240 Vrms
Intra-module Isolation, Field-to-Logic	3000 VDC
ESD Protection	+/-4 kV Contact for each Channel
LED Indicators	
ZigBee PWR	ZigBee Device Power
ZigBee Net	ZigBee Communication Indicator
Power	
Power Consumption	0.88 W (Max.)
Environment	
Operating Temperature	-25 to 75°C
Storage Temperature	-30 to 80°C
Humidity	10 to 90%, Non-condensing

Wireless	
RF Channels	16
RF Transmit Power	11 dBm
Antenna (2.4 GHz)	5 dBi Omnidirectional
Transmission Range (LOS)	700 m (Typical)
Max. Slaves Supported	255
EMI Certification	CE/FCC, FCC ID

2.2 Pin Assignments



2.3 Wire Connections



3 Setting up the ZT-2000 I/O Device

3.1 Introduction to the Configuration Parameters

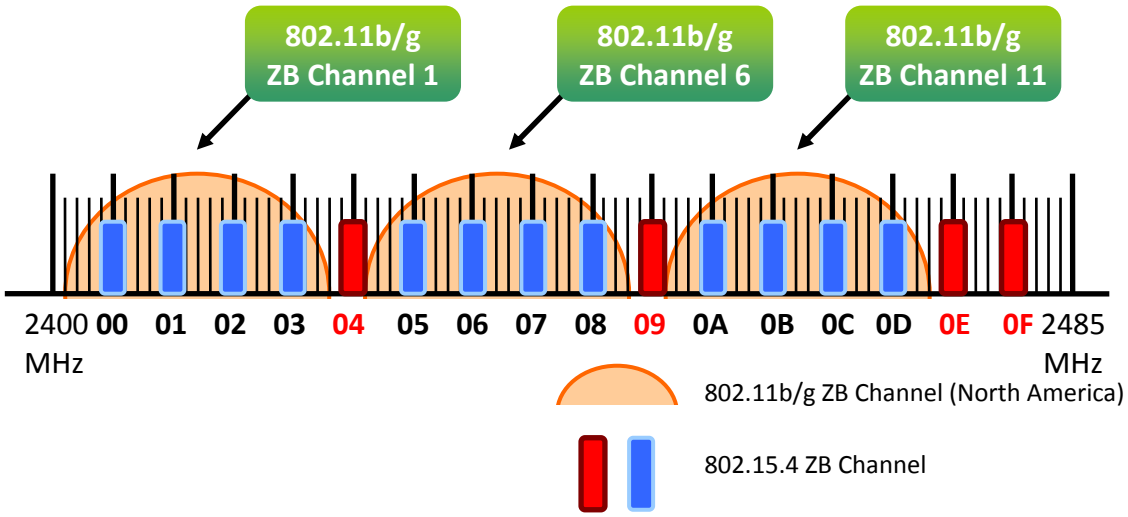
A. The "**ZB PID**" parameter is the group identity for a ZigBee network, and must be the same for all devices in the same ZigBee network.

B. The "**Node ID**" parameter is the individual identity of the specific ZigBee module, and must be unique for each device connected to the same ZigBee network.

C. The "**ZB Channel**" parameter indicates the radio frequency channel, and must be set to the same value as other modules on the same ZigBee network.

ZB Channel	0x00	0x01	·····	0x0F
Frequency (MHz)	2405	2410	·····	2480

※ **ZB channels 0x04, 0x09, 0x0E or 0x0F are recommended because they do not overlap with the Wi-Fi frequency band.**



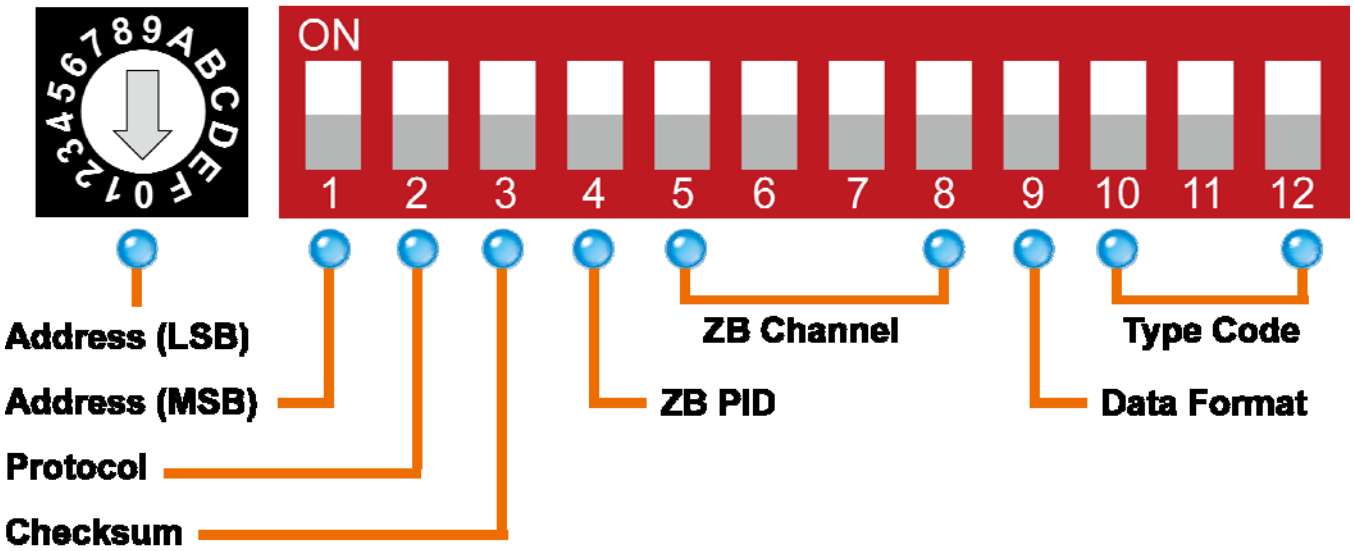
D. **Protocol/Application Mode:**

When implementing custom programs based on different protocols, the following application mode(s) are recommended in order to ensure optimal performance.

User Program Protocol	ZT-2000	ZT-2550	ZT-2570
DCON	DCON	Transparent	Transparent
Modbus RTU	Modbus RTU	Transparent Modbus Gateway	Transparent Modbus Gateway
Modbus TCP	Modbus RTU	------	Modbus Gateway

3.2 Introduction to the Rotary and DIP Switches

The configuration of the ZT-2018 can be adjusted using a combination of the external rotary switch and the DIP switches. The ZT-2000 device should only be rebooted once the configuration is complete.



➢ Rotary Switch

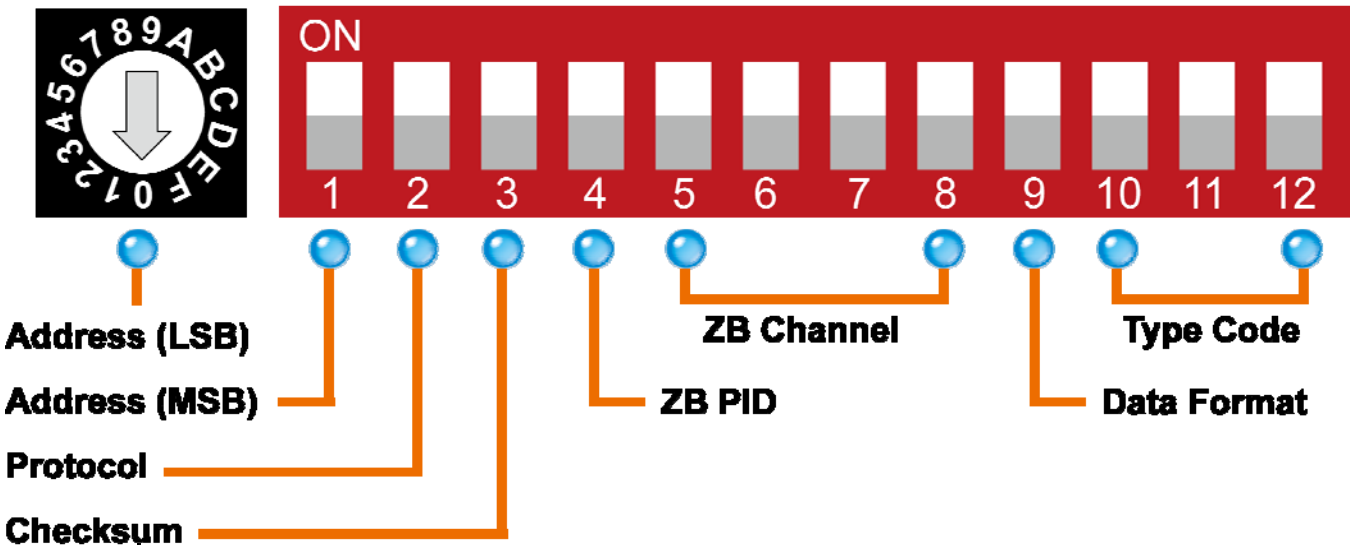
Case 1: Address MSB = 0

	0	1	2	3	4	5	6	7
Address	*Note 1	01	02	03	04	05	06	07
Node ID	*Note 1	0x0001	0x0002	0x003	0x0004	0x0005	0x0006	0x0007
	8	9	A	B	C	D	E	F
Address	08	09	0A	0B	0C	0D	0E	0F
Node ID	0x008	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E	0x000F

Case 2: Address MSB = 1

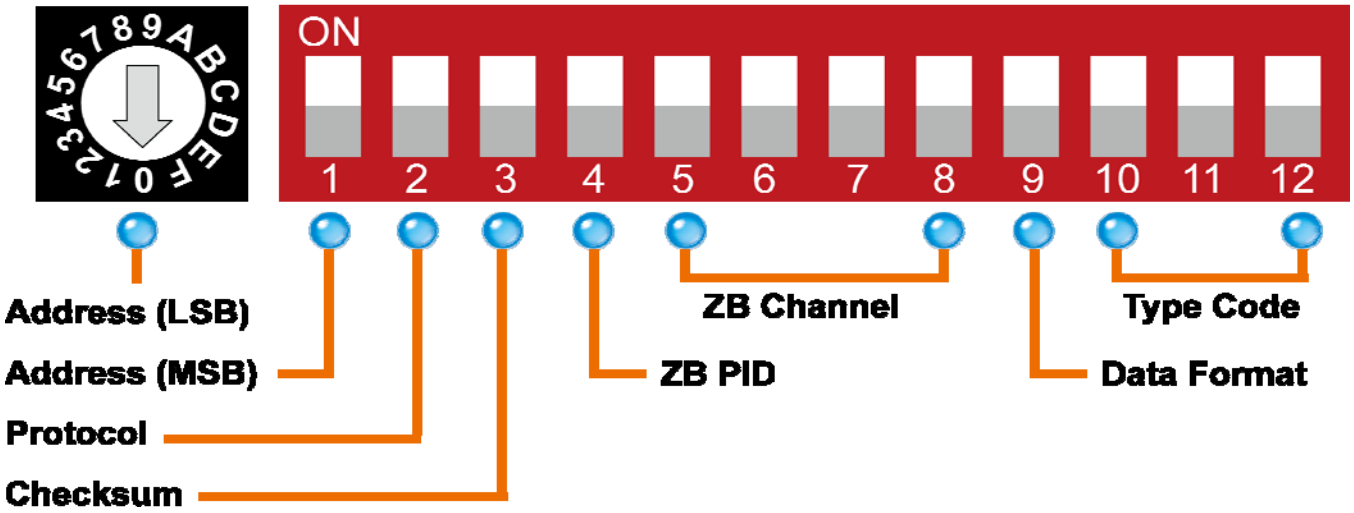
	0	1	2	3	4	5	6	7
Address	10	11	12	13	14	15	16	17
Node ID	0x0010	0x0011	0x0012	0x013	0x0014	0x0015	0x0016	0x0017
	8	9	A	B	C	D	E	F
Address	18	19	1A	0B	0C	1D	1E	1F
Node ID	0x018	0x0019	0x001A	0x001B	0x001C	0x001D	0x001E	0x001F

***Note 1: The "Address" and "Node ID" values are defined via the $AANNTTCCFF command. In software configuration mode, the DIP switches for "Address", "Data Format" and "Type Code" are ignored and can also be set via the %AANNTTCCFF and $AACiRrr commands.**



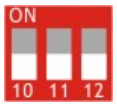
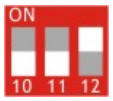
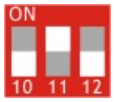
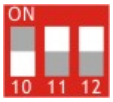
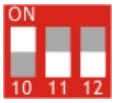
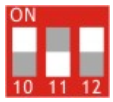
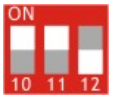
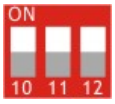
➢ DIP Switches

Number	Item	Status	Description
1	Address MSB	OFF	Valid Address (Node ID) from 0x01 to 0x0F
		ON	Valid Address (Node ID) from 0x10, 0x01 to 0x1F
2	Protocol	OFF	DCON Protocol
		ON	Modbus RTU Protocol
3	Checksum	OFF	Disabled (DCON Protocol)
		ON	Enabled (DCON Protocol)
4	ZB PID	OFF	ZigBee Pan ID = 0x0000
		ON	ZigBee Pan ID = 0x0001
5	ZB Channel	OFF	------
		ON	0x08
6		OFF	------
		ON	0x04
7		OFF	------
		ON	0x02
8		OFF	------
		ON	0x01
9	Data Format	OFF	Engineering Units Format
		ON	Hexadecimal Format



➢ Type Code

DIP switches 10–12 are used to define the input Type Code for the ZT-2018, as shown below.

Switch Position	Type Code	Switch Position	Type Code	Switch Position	Type Code
	0x00		0x01		0x02
	0x03		0x04		0x05
	0x06		0x07		

3.3 Starting the ZT-2000 I/O Device

As the ZigBee network is controlled by the ZigBee Coordinator, the ZT-2550/ZT-2570 (ZigBee Coordinator) must be configured first. Refer to the documents section below for full details of how to configure these devices.

Once configuration of the ZigBee Coordinator has been completed, set the "ZB PID" and "ZB Channel" values for the ZT-2000 I/O device to the same values as the network, and then reboot the device. The module will automatically start to function on the ZigBee network using the default protocol.

※ Documents

Helpful documentation related to the ZT-2550 and ZT-2570 can be found at:

http://ftp.icpdas.com.tw/pub/cd/usbcd/napdos/zigbee/zt_series/document/zt-255x/
http://ftp.icpdas.com.tw/pub/cd/usbcd/napdos/zigbee/zt_series/document/zt-257x/

※ Configuration Utility

A utility that can be used to configure the ZT-2000 I/O device Coordinator is available for download from:

http://ftp.icpdas.com.tw/pub/cd/usbcd/napdos/zigbee/zt_series/utility/

3.4 Communications Testing

Once the ZT-2000 I/O device has joined the ZigBee network, the signal quality can be confirmed by monitoring the status of the ZigBee Net LED indicators. If the LED indicator shows a steady light, communication with the ZT-2000 I/O device has been successfully established for data acquisition and control.

ICP DAS provides the "DCON Utility" which can be used to simulate DCON/Modbus communication. This software can also be used to verify the device settings and the ZigBee I/O functions.

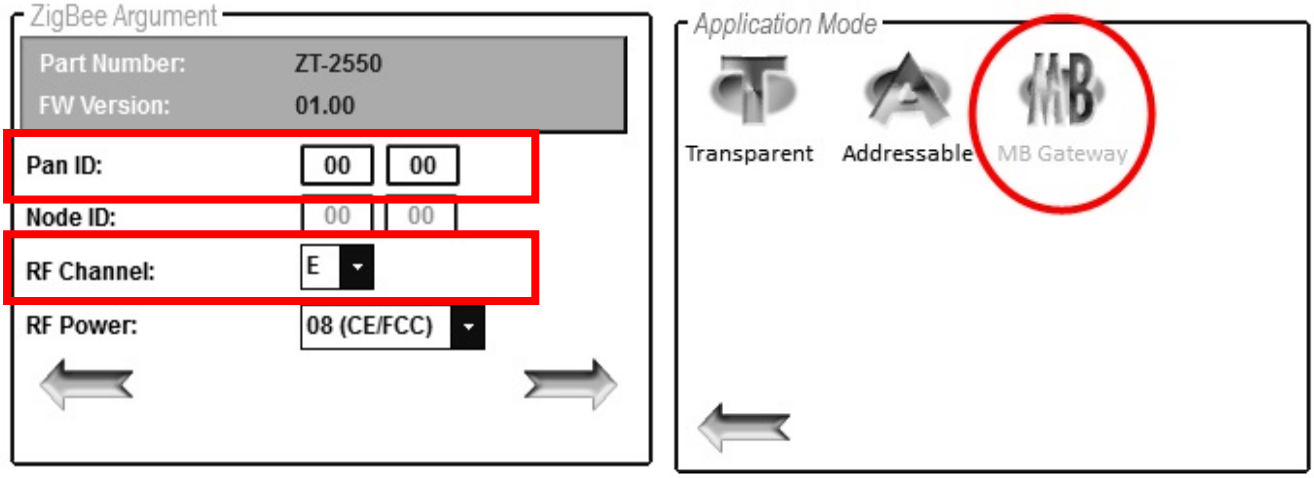
The DCON Utility can be downloaded from:

http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/

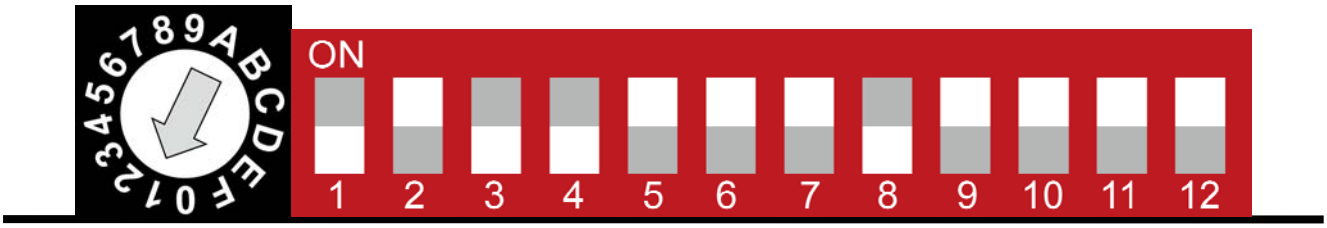
3.5 Examples

➢ Architecture Diagram

➢ Configuring the ZT-2550/ZT-2570



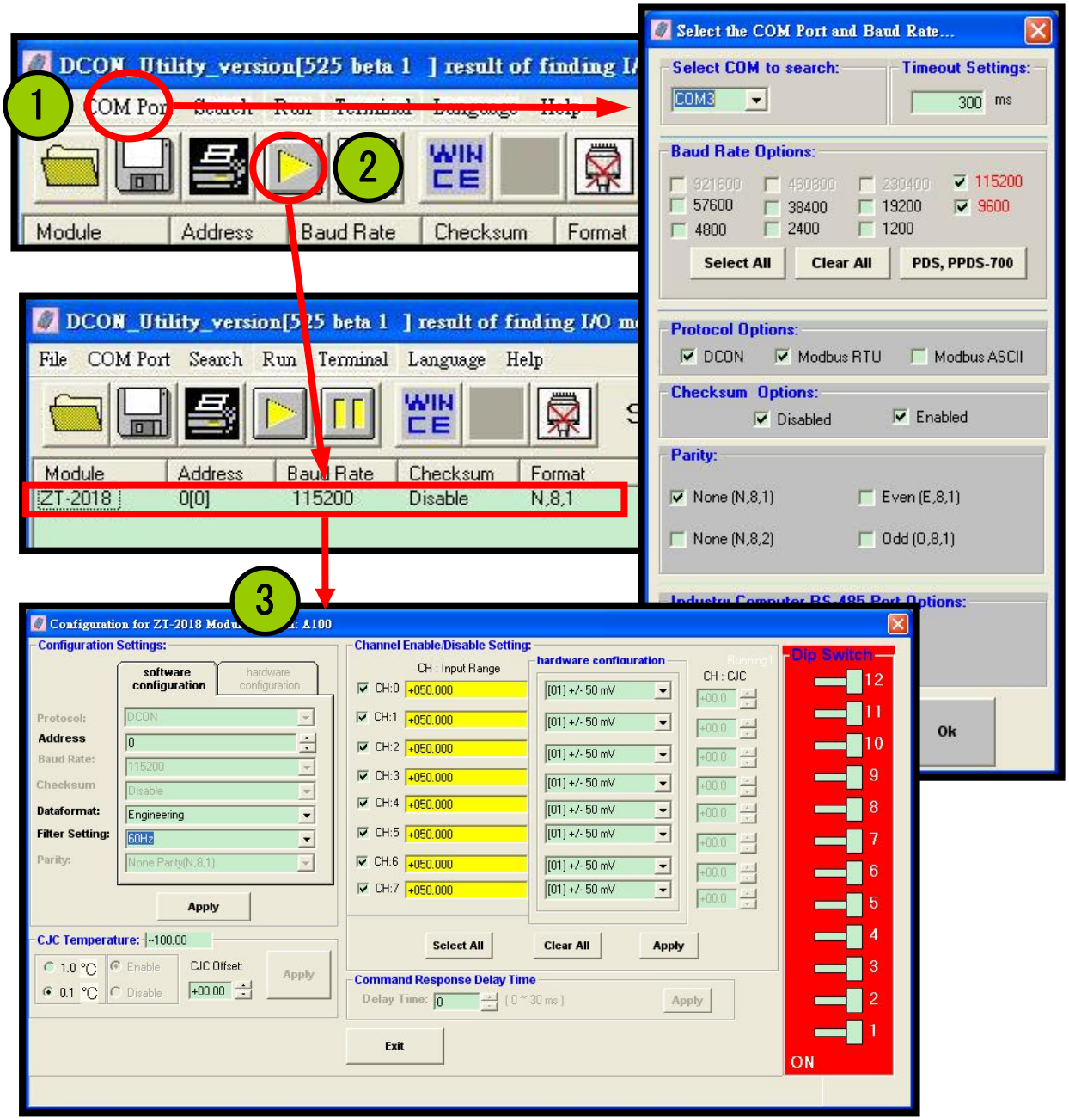
➢ Configuring the ZT-2000 I/O device



Number	Item	Status	Description	
1	Address MSB	OFF	Address/Node ID is **01** (Rotary Switch=1)	
2	Protocol	ON	Use the **Modbus RTU** Protocol	
3	Checksum	OFF	Disabled	
4	ZB PID	OFF	ZigBee Pan ID=**0x0000**	
5	ZB Channel	ON	0x08	ZigBee RF Channel = **0x0E**
6		ON	0x04	
7		ON	0x02	
8		OFF	------	

➢ Simulating I/O channel operation via the DCON Utility

1. Launch the DCON Utility and select the appropriate COM Port settings to connect to the ZigBee Coordinator (ZT-2550/ZT-2570).
2. Click the "Search" button to start searching for ZT-2000 I/O devices connected to the same ZigBee network.
3. If any ZT-2000 I/O devices are found, they will be displayed in the device list windows. Double-click the name of the name of the module to start the operatiion.



4 Analog Input Type and Data Format

Type Code	Input Type	Data Format	+F.S.	-F.S.
00	-15 to +15 mV	Engineering Units	+15.000	-15.000
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
01	-50 to +50 mV	Engineering Units	+50.000	-50.000
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
02	-100 to +100 mV	Engineering Units	+100.00	-100.00
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
03	-500 to +500 mV	Engineering Units	+500.00	-500.00
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
04	-1 to +1 V	Engineering Units	+1.0000	-1.0000
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
05	-2.5 to +2.5 V	Engineering Units	+2.5000	-2.5000
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
06	-20 to +20 mA	Engineering Units	+20.000	-20.000
		% of FSR*2	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
07	4 to +20 mA	Engineering Units	+20.000	+04.000
		% of FSR*2	+100.00	+000.00
		2's Comp. Hex	FFFF	0000
0E	Type J Thermocouple -210 ~ +760° C	Engineering Units	+760.00	-210.00
		% of FSR*2	+100.00	-027.63
		2's Comp. Hex	FFFF	DCA2
0F	Type K Thermocouple -270 ~ +1372° C	Engineering Units	+1372.0	-0270.0
		% of FSR*2	+100.00	-019.68
		2's Comp. Hex	7FFF	E6D0
10	Type T Thermocouple -270 ~ +400° C	Engineering Units	+400.00	-270.00
		% of FSR*2	+100.00	-067.50
		2's Comp. Hex	7FFF	DCA2
11	Type E Thermocouple -270 ~ +1000° C	Engineering Units	+1000.0	-0270.0
		% of FSR*2	+100.00	-027.00
		2's Comp. Hex	7FFF	DD71

12	Type R Thermocouple 0 ~ +1768° C	Engineering Units	+1768.0	-0000.0
		% of FSR*2	+100.00	-000.00
		2's Comp. Hex	7FFF	0000
13	Type S Thermocouple 0 ~ +1768° C	Engineering Units	+1768.0	-0000.0
		% of FSR*2	+100.00	-000.00
		2's Comp. Hex	7FFF	0000
14	Type B Thermocouple 0 ~ +1820° C	Engineering Units	+1820.0	-0000.0
		% of FSR*2	+100.00	-000.00
		2's Comp. Hex	7FFF	0000
15	Type N Thermocouple -270 ~ +1300° C	Engineering Units	+1300.0	-0270.0
		% of FSR*2	+100.00	-020.77
		2's Comp. Hex	7FFF	E56B
16	Type C Thermocouple 0 ~ +2320° C	Engineering Units	+2320.0	-0000.0
		% of FSR*2	+100.00	-000.00
		2's Comp. Hex	7FFF	0000
17	Type L Thermocouple -200 ~ +800° C	Engineering Units	+800.00	-200.00
		% of FSR*2	+100.00	-025.00
		2's Comp. Hex	7FFF	E000
18	Type M Thermocouple -200 ~ +100° C	Engineering Units	+100.00	-200.00
		% of FSR*2	+050.00	-100.00
		2's Comp. Hex	4000	8000
19	Type LDIN43710 Thermocouple -200 ~ +800° C	Engineering Units	+900.00	-200.00
		% of FSR*2	+100.00	-022.22
		2's Comp. Hex	7FFF	E38E
1A	0 to +20 mA	Engineering Units	+20.000	+00.000
		% of FSR*2	+100.00	+000.00
		2's Comp. Hex	FFFF	0000
*1: FSR (Full Scale Range)				

➢ Analog Input Over/Under Range Readings

	Over Range	Under Range
Engineering Units	+9999.9	-9999.9
% of FSR	+999.99	-999.99
2's Complement Hex	7FFF	8000

➢ Analog Input Over/Under Range Readings when using the Modbus RTU protocol

Over Range	Under Range
7FFFh	8000h

➢ Data Format Settings (FF)

7	6	5	4	3	2	1	0
FS	Reserved					DF	

Key	Description
DF	Data Format 00: Engineering Units 01: % of FSR 10: 2's Complement Hexadecimal
FS	Filter Settings 0: 60 Hz Rejection 1: 50 Hz Rejection

5 Calibration

➢ Warning

Performing calibration is not recommended until the process is fully understood.
The calibration procedure is as follows:

1. Warm up the module for at least 30 minutes.
2. Set the Type Code to the type you wish to calibrate. Refer to Section 4 and Section 6.2.13 for details.
3. Enable calibration. Refer to Section 6.2.31 for details.
4. Apply the zero calibration voltage/current.
5. Send the zero calibration command. Refer to Section 6.2.7 for details.
6. Apply the span calibration voltage/current.
7. Send the span calibration command. Refer to Section 6.2.6 for details.
8. Repeat steps 3 to 7 three times.

➢ Notes

1. The calibration voltage/current source should be connected to channel 0.
2. Calibration voltages and currents are shown below.
3. Switch to DCON protocol mode before calibrating the module. Refer to Section 3.2 for details of how to switch protocols.

➢ Calibration Voltage Types used by the ZT-2018

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

6 The DCON/Modbus RTU Command Sets

6.1 Communicating with the ZT-2000 I/O Device

ICP DAS ZT-2000 I/O devices can be operated using either the DCON or the Modbus RTU protocol, which can be selected by adjusting the position of DIP switch 2 to OFF (DCON) or ON (Modbus RTU) and then rebooting the ZT-2000 I/O device to user the new protocol.

6.2 The DCON Protocol Command Set

All ZT-2000 I/O series devices are controlled via wireless broadcast commands, so each device must have a unique address that is saved in the EEPROM of the device.

Consequently, all command and response formats contain the address of the destination module. When an I/O device receives a command, it will determine whether or not to respond based on the address contained in the command. However, there are two exceptions to this, the #** and ~** commands.

➢ DCON Command Format

Delimiter Character	Module Address	Command	[CHECKSUM]	CR

➢ DCON Response Format

Delimiter Character	Module Address	Data	[CHECKSUM]	CR

※ **Note: 'CR' is the end of command (carriage return) character used to end a frame.**

※ **All characters should be expressed in capital letters.**

6.2.1 Checksum

➢ Calculating the Checksum:

Sum the ASCII codes of all the characters contained in the command in addition to the 'CR' terminator. The Checksum is the sum value expressed in Hexadecimal format.

➢ Example: Command "$012(CR)"

Sum = '$' + '0' + '1' + '2' = 24h + 30h + 31h + 32h = B7h

Checksum = "B7"

DCON Command with Checksum ="$012B7(CR)"

➢ Example: Response "!01200600(CR)"

Sum = '!' + '0' + '1' + '2' + '0' + '0' + '6' + '0' + '0'

= 21h+30h+31h+32h+30h+30h+36h+30h+30h

= 1AAh

Checksum = "AA"

DCON Response with Checksum = "!01200600AA(CR)"

※ Note: The Checksum is the sum value expressed in capital letters.

6.3 Overview of the DCON Command Set

General Command Set			
Command	Response	Description	Section
%AANNTTCCFF	!AA	Sets the Configuration of the Module	6.2.3
#AA	>(Data)	Reads the Analog Input Data from all Channels	6.2.4
#AAN	>(Data)	Reads the Analog Input Data from a Specific Channel	6.2.5
$AA0	!AA	Performs an Analog Input Span Calibration	6.2.6
$AA1	!AA	Performs an Analog Input Zero Calibration	6.2.7
$AA2	!AANNTTCCFF	Reads the Configuration of the Module	6.2.8
$AA5	!AAS	Reads the Reset Status of the Module	6.2.10
$AA5VV	!AA	Enables or Disables Specific Analog Input Channels	6.2.11
$AA6	!AAVV	Reads whether each Analog Input Channel is Enabled or Disabled	6.2.12
$AA7CiRrr	!AA	Sets the Type Code for a Specific Analog Input Channel	6.2.13
$AA8Ci	!AACiRrr	Reads the Type Code for a Specific Analog Input Channel	6.2.14
$AAF	!AA(Data)	Reads the Firmware Version of the Module	6.2.21
$AAM	!AA(Data)	Reads the Name of the Module	6.2.22
$AAS1	!AA	Reloads the Default Calibration Parameters	6.2.23
~AAEV	!AA	Enables or Disables Calibration for the Module	6.2.31
~AAO(Name)	!AA	Sets the Name of the Module	6.2.32
@AACH	!AA	Clears the High Latch Values for all Analog Input Channels	6.2.33
@AACHi	!AA	Clears the High Latch Value for a Specific Analog Input Channel	6.2.34
@AACHCi	!AA	Clears the Status of the High Alarm for a Specific Analog Input Channel	6.2.35

@AACL	!AA	Clears the Low Latch Values for all Analog Input Channels	6.2.36
@AACLi	!AA	Clears the Low Latch Value for a Specific Analog Input Channel	6.2.37
@AACLCi	!AA	Clears the Status of the Low Alarm for a Specific Analog Input Channel	6.2.38
@AADHCi	!AA	Disables the High Alarm for a Specific Analog Input Channel	6.2.39
@AADI	!AAHHLL	Reads the Status of the Alarms for all Analog Input Channels	6.2.40
@AADLCi	!AA	Disables the Low Alarm for a Specific Analog Input Channel	6.2.41
@AAHI(Data)CiT	!AA	Sets the High Alarm Value and Type for a Specific Analog Input Channel	6.2.42
@AALO(Data)CiT	!AA	Sets the Low Alarm Value and Type for a Specific Analog Input Channel	6.2.43
@AARH	!AA(Data)	Reads the High Latch Values for all Analog Input Channels	6.2.45
@AARHi	!AA(Data)	Reads the High Latch Value for a Specific Analog Input Channel	6.2.46
@AARHCi	!AA(Data)S	Reads the High Alarm Value for a Specific Analog Input Channel	6.2.47
@AARL	!AA(Data)	Reads the Low Latch Values for all Analog Input Channels	6.2.48
@AARLi	!AA(Data)	Reads the Low Latch Value for a Specific Analog Input Channel	6.2.49
@AARLCi	!AA(Data)S	Reads the Low Alarm Value for a Specific Analog Input Channel	6.2.50

CJC Command Sets			
Command	Response	Description	Section
$AA3	>(Data)	Reads the CJC Temperature	6.2.9
$AA9	!AA(Data)	Reads the CJC Offset Value	6.2.15
$AA9SNNNN	!AA	Sets the CJC Offset Value	6.2.16
$AA9Ci	!AA(Data)	Reads the CJC Offset Value for a Specific Analog Input Channel	6.2.17
$AA9SNNNNCi	!AA	Sets the CJC Offset Value for a Specific Analog Input Channel	6.2.18
$AAA	!AAi	Reads the CJC Temperature Update Settings	6.2.19
$AAAi	!AA	Sets the CJC Temperature Update Settings	6.2.20
~AAC	!AAN	Read whether the CJC Function is Enabled or Disabled	6.2.29
~AACN	!AA	Enables or Disables the CJC Function	6.2.30
@AAOD	!AAS	Reads the Status of the CJC Connection	6.2.44

Host Watchdog Command Sets			
Command	Response	Description	Section
~**	No Response	The command to inform all module that the Host is OK	6.2.24
~AA0	!AASS	Reads the Status of the Host Watchdog	6.2.25
~AA1	!AA	Resets the Status of the Host Watchdog Timeout	6.2.26
~AA2	!AAETT	Reads the Timeout Settings for the Host Watchdog	6.2.27
~AA3ETT	!AA	Enables or Disables the Host Watchdog and sets the Host Watchdog Timeout Value	6.2.28

%AANNTTCCFF

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

Syntax	
%AANNTTCCFF[CHECKSUM](CR)	
%	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
NN	The new address of the module in hexadecimal format (00 to FF)
TT	00 (Reserved)
CC	0A (Reserved)
FF	The command used to set the data format, checksum, and filter settings. See Section 4 for details of the data format.

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	%0320000A80
Response	!03
In Normal mode, the address 0x20 is saved to the EEPROM and the data format for module 03 is set to 80 (50 Hz rejection). The module returns a response indicating that the command was successful.	
Command	%0320000A80
Response	!20
In Software Configuration mode, the address 0x20 is saved to the EEPROM and the data format for module 03 is set to 80 (50 Hz rejection). The module returns a response indicating that the command was successful.	
Command	%0303000000
Response	?03
Attempts to set the configuration for module 03, but returns a response indicating that an error occurred because the “CC”parameter must be 0A.	

※Related Commands: Section 6.2.8 $AA2
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.1 #AA

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

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

Response	
Valid Command	>(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
>	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The data from all the Analog Input channels. See Section 4 for details of the data format. Data from disabled channels is filled with space characters.
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	#03
Response	>+15.000+15.000+15.000+15.000+15.000+15.000+15.000+15.000
Reads data from the Analog Input channels of module 03 and returns a response indicating that the command was successful, with the data from all Analog Input channels in engineering units format.	

Command	#03
Response	>-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9
Attempts to read data from the Analog Input channels of module 03, but returns a response indicating that the command was unsuccessful because the data is not within the valid range.	

※Related Commands: Section 6.2.3 %AANNTTCCFF, Section 6.2.8 $AA2, Section 6.2.13 $AA7CiRrr

※Related Topics: Section 4 Analog Input Type and Data Format.
Section 7.1 Software Configuration Mode

6.3.2 #AAN

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

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

Response	
Valid Command	>(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
>	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command.
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The data from the specified Analog Input channel. See Section 4 for details of the data format. If the specified channel is disabled, then the data field will be filled with space characters.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

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

Command	#039
Response	?03
Attempts to read data from Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.3 %AANNTTCCFF, Section 6.2.8 $AA2, Section 6.2.13 $AA7CiRrr

※Related Topics: Section 4 Analog Input Type and Data Format.
Section 7.1 Software Configuration Mode

6.3.3 $AA0

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command.
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$030
Response	?03
Attempts to perform an Analog Input span calibration on module 03, but a response indicating that the command was unsuccessful is returned because the “Enable Calibration” command (~AAEV, see Section 6.2.31) was not sent in advance.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a response indicating that the command was successful.	

Command	$030
Response	!03
Performs an Analog Input span calibration on module 03 and returns a response indicating that the command was successful.	

※Related Commands: Section 6.2.7 $AA1, Section 6.2.31 ~AAEV

※Related Topics: Section 5 Calibration

※Notes: The “Enable Calibration” command, ~AAEV, and the “Zero Calibration” command, $AA1, must be sent before this command is used. See Sections 6.2.31 and 6.2.7 for details.

6.3.4 $AA1

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$031
Response	?03
Attempts to perform an Analog Input zero calibration on module 03, but a response indicating that the command was unsuccessful is returned because the “Enable Calibration” command (~AAEV, see Section 6.2.31) was not sent in advance.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a response indicating that the command was successful.	

Command	$031
Response	!03
Performs an Analog Input zero calibration on module 03 and returns a response indicating that the command was successful.	

※Related Commands: Section 6.2.6 $AA0, Section 6.2.31 ~AAEV
※Related Topics: Section 5 Calibration
※Notes:

1. The “Enable Calibration” command, ~AAEV, must be sent before this command is used. See Section 6.2.31 for details.
2. This command must be sent before the “Span Calibration” command, $AA0, is used. See Section 6.2.6 for details.

6.3.5 $AA2

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

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

Response	
Valid Command	!NNTTCCFF[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
NN	The address of the module that is saved in the EEPROM in hexadecimal format (00 to FF)
TT	00 (Reserved)
CC	0A (Reserved)
FF	The data format, checksum settings and filter settings for the module. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$032
Response	!FF000A00
In Normal mode, reads the configuration of module 03. The response indicates that the command was successful, and shows that the address stored in the EEPROM is 0xFF, that the filter is set to 60 Hz rejection, and that the data format is Engineering Units.	

Command	$FF2
Response	!FF000A00
In Software Configuration mode, reads the configuration of module FF. The response indicates that the command was successful, and shows that the address stored in the EEPROM is 0xFF, that the filter is set to 60 Hz rejection, and that the data format is Engineering Units.	

※Related Commands: Section 6.2.3 %AANNTTCCFF
※Related Topics: Section 4 Analog Input Type and Data Format
Section 7.1 Software Configuration Mode

6.3.6 $AA3

Description
This command is used to read the CJC(cold junction compensation) temperature for a specified module.

Syntax	
$AA3[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
3	The command to read the CJC temperature

Response	
Valid Command	>(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
>	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The CJC temperature in degrees Celsius, consisting of a sign byte, '+' or '–', followed by 5 decimal digits with a fixed decimal point indicating the temperature in tenths of a degree Celsius.
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	$033
Response	>+0031.2
Reads the CJC temperature for module 03, and returns a response indicating that the command was successful and that the temperature is 31.2° C.	

※Related Commands: Section 6.2.15 $AA9, Section 6.2.16 $AA9SNNNN,Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.7 $AA5

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

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

Response	
Valid Command	!AAS[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
S	The reset status of the module: 0: This is not the first time the command has been sent since the module was powered on, which denotes that there has been no module reset since the last $AA5 command was sent. 1: This is the first time the command has been sent since the module was powered on.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$035
Response	!031
Reads the reset status of module 03. The module returns a response indicating that the command was successful and that it is the first time the $AA5 command has been sent since the module was powered on.	

Command	$035
Response	!030
Reads the reset status of module 03. The module returns a response indicating that the command was successful and that there has been no module reset since the last $AA5 command was sent.	

6.3.8 $AA5VV

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$0353A
Response	!03
Enables Analog Input channels 1, 3, 4, and 5 on module 03 and disables all other Analog Input channels. The module returns a response indicating that the command was successful.	

Command	$036
Response	!033A
Reads the status of the Analog Input channels on module 03, and returns a response indicating that the command was successful, with a value of 3A, which denotes that Analog Input channels 1, 3, 4, and 5 are enabled and all other Analog Input channels are disabled.	

※Related Commands: Section 6.2.12 $AA6

6.3.9 $AA6

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

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

Response	
Valid Command	!AAVV[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
VV	A two-digit hexadecimal value representing the Analog Input channel, where bit 0 corresponds to Analog Input channel 0, and bit 1 corresponds to Analog Input channel 1, etc. When the bit is 0, it denotes that the Analog Input channel is disabled, and 1 denotes that the Analog Input channel is enabled.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$0353A
Response	!03
Enables Analog Input channels 1, 3, 4, and 5 and disables all other Analog Input channels on module 03. The module returns a response indicating that the command was successful.	

Command	$036
Response	!033A
Reads the status of the Analog Input channels on module 03, and returns a response indicating that the command was successful, with a value of 3A, which denotes that Analog Input channels 1, 3, 4, and 5 are enabled and all other Analog Input channels are disabled.	

※Related Commands: Section 6.2.11 $AA5VV

6.3.10 $AA7CiRrr

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

Syntax	
$AA7CiRrr[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
7	The command to set the channel Type Code
Ci	i specifies the Analog Input channel to be set (0-7)
Rrr	rr represents the Type Code to be set for the Analog Input channel. See Section 4 for details of the data format.

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$037C0R08
Response	!03
Sets the Type Code for Analog Input channel 0 of module 03 to 8 (-10 ~ +10V), and the module returns a response indicating that the command was successful.	

Command	$037C5R09
Response	!03
Sets the Type Code for Analog Input channel 5 of module 03 to 9 (-5 ~ +5V), and the module returns a response indicating that the command was successful.	

Command	$037C1R80
Response	?03
Attempts to set the Type Code for Analog Input channel 1 of module 03 to 80. The module returns a response indicating that the command was unsuccessful because the Type Code is incorrect.	

※Related Commands: Section 6.2.14 $AA8Ci
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.11 $AA8Ci

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

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

Response	
Valid Command	!AACiRrr[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
Ci	i specifies which Analog Input channel the Type Code information relates to
Rrr	rr represents the Type Code used for the specified Analog Input channel. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$038C0
Response	!03C0R08
Reads the Analog Input Type Code information for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with a value of 08 denoting that the input range is -10 ~ +10V.	

Command	$038C9
Response	?03
Attempts to read the Type Code information for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because the Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.4 #AA, Section 6.2.5 #AAN, Section 6.2.13 $AA7CiRrr
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.12 $AA9

Description
This command is used to read the CJC (cold junction compensation) offset value for a specified module. The CJC offset value is set using the $AA9SNNNN command (see Section 6.2.16 for details).

Syntax	
$AA9[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
9	The command to read the CJC offset value

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The CJC offset value, consisting of a sign byte, '+' or '−', followed by 4 hexadecimal digits. The interval between each value is equal to 0.01° C.
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.	

Example	
Command	$039+0010
Response	!03
Sets the CJC offset value for module 03 to 0.16° C and returns a response indicating that the command was successful.	

Command	$039
Response	!03+0010
Reads the CJC offsetvalue for module 03 and returns a response indicating that the command was successful, with a value of +0010 (0.16° C).	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.16 $AA9SNNNN, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.13 $AA9SNNNN

Description
This command is used to set the CJC (cold junction compensation) offset value on a specified module to compensate for any error produced by the CJC sensor.

Syntax	
$AA9SNNNN[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
9	The command to set the CJC offset value
S	The sign, '+' or '-', of the offset value
NNNN	The absolute value of the CJC offset represented by four hexadecimal digits, which must be less than or equal to 10000. The interval between each value is equal to 0.01° C.

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$039+0010
Response	!03
Sets the CJC offset value for module 03 to +0010 (0.16° C) and returns a response indicating that the command was successful.	

Command	$039
Response	!03+0010
Reads the CJC offset value for module 03 and returns a response indicating that the command was successful, with a value of +0010 (0.16° C).	

Command	$039+3000
Response	?03
Attempts to set the CJC offset value for module 03 to +3000 (122.88° C), but returns a response indicating that the command was unsuccessful because the value given for the CJC offset was not within the valid range.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.15 $AA9, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.14 $AA9Ci

Description
This command is used to read the CJC (cold junction compensation) offset value for a specific Analog Input channel of a specified module.

Syntax	
$AA9Ci[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
9	The command to read the CJC offset value for a specific Analog Input channel
Ci	Specifies which Analog Input channel to access for the CJC offset value (0-7)

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The CJC offset value consisting of a sign byte, '+' or '-', followed by 4 hexadecimal digits. The interval between each value is equal to 0.01° C.
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.	

Example	
Command	$039+0010C1
Response	!03
Sets the CJC offset value for Analog Input channel 1 of module 03 to +0010 (0.16°C) and returns a response indicating that the command was successful.	

Command	$039C1
Response	!03+0010
Reads the CJC offset for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, with a value of +0010 (0.16°C).	

Command	$039C9
Response	?03
Attempts to read the CJC offset for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.18 $AA9SNNNNCi, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.15 $AA9SNNNNCi

Description
This command is used to set the CJC(cold junction compensation) offset value for a specific Analog Input channel of a specified module.

Syntax	
$AA9SNNNN[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
9	The command to set the CJC offset value
S	The Sign, '+' or '−', of the offset value
NNNN	The absolute value of the CJC offset represent by four hexadecimal digits, which must be less than or equal to 10000h. The interval between each value is equal to 0.01° C.
Ci	Specifies which Analog Input channel to set (0-7)

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$039+0010C1
Response	!03
Sets the CJC offset value for Analog Input channel 1 of module 03 to +0010 (0.16°C) and returns a response indicating that the command was successful.	

Command	$039C1
Response	!03+0010
Reads the CJC offset calue for the ※channel 1 of module 03 and returns a response indicating that the command was successful, with a value of +0010 (0.16°C).	

Command	$039+0010C9
Response	!03
Attempts to set the CJC offset value for Analog Input channel 9 of module 03 to +0010 (0.16° C), but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.17 $AA9Ci, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.16 $AAA

Description
This command is used to read the CJC (cold junction compensation) temperature update settings for a specified module.

Syntax	
$AAA[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
A	The command to read the CJC temperature update settings

Response	
Valid Command	!AAi[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
i	The CJC temperature update settings: 0: The CJC temperature update function has been stopped 1: The CJC temperature update function has been started 2: The CJC temperature will only be update once
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.	

Example	
Command	$03A1
Response	!03
Starts the CJC temperature update function for module 03 and returns a response indicating that the command was successful.	

Command	$03A
Response	!031
Reads the CJC temperature update settings for module 03 and returns a response indicating that the command was successful, with a value of 1, meaning that the CJC temperature update function has been started.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.15 $AA9, Section 6.2.16 $AA9SNNNN, Section 6.2.17 $AA9Ci, Section 6.2.18 $AA9SNNNNCi, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN

6.3.17 $AAAi

Description
This command is used to configure the CJC (cold hunction compensation) temperature update settings for a specified module.

Syntax	
$AAAi[CHECKSUM](CR)	
$	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
A	The command to configure the CJC temperature update settings
i	CJC temperature update settings: 0: Stops the CJC temperature update function 1: Starts the CJC temperature update function 2: The CJC temperature will only be updated once

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$03A1
Response	!03
Starts the CJC temperature update function for module 03 and returns a response indicating that the command was successful.	

Command	$03A
Response	!031
Reads the CJC temperature update settings for module 03 and returns a response indicating that the command was successful, with a value of 1, meaning that the CJC temperature update function has been started.	

Command	$03A3
Response	!03
Attempts to start the CJC temperature update function for module 03, but returns a response indicating that the command was unsuccessful because the settings parameter was invalid.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.15 $AA9, Section 6.2.16 $AA9SNNNN, Section 6.2.17 $AA9Ci, Section 6.2.18$AA9SNNNNCi, Section 6.2.19 $AAA, Section 6.2.30 ~AACN

6.3.18 $AAF

Description
This command is used to read the firmware version of a specified module.

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The firmware version of the module as a string value
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$03F
Response	!03A1.0
Reads the firmware version of module 03, and returns a response indicating that the command was successful and showing that firmware is version A1.0.	

6.3.19 $AAM

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

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The name of the module as a string value
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	~03OZT-2018
Response	!03
Sets the name of module 03 to “ZT-2018” and returns a response indicating that the command was successful.	

Command	$03M
Response	!03ZT-2018
Reads the name of module 03 and returns a response indicating that the command was successful, and that the name of the module is “ZT-2018”.	

※Related Commands: Section 6.2.32 ~AAO(Name)

6.3.20 $AAS1

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

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

Response	
Valid Command	!AA [CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$03S1
Response	!03
Sends a command to reload the factory default calibration parameters for module 03 and returns a response indicating that the command was successful.	

※Related Commands: Section 6.2.6 ~AA0, Section 6.2.7 $AA1, Section 6.2.31 ~AAEV
※Related Topics: Section 5 Calibration

6.3.21 ~**

Description
This command is used to inform all modules that the Host is OK.

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

Response
There is no response to this command.

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

※Related Commands: Section 6.2.6 ~AA0, Section 6.2.7 ~AA1, Section 6.2.8 ~AA2, Section 6.2.28 ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation.

6.3.22 ~AA0

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

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

Response	
Valid Command	!AASS[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
SS	Two hexadecimal digits that represent the status of the Host Watchdog, where: Bit 2: 0 indicates that no Host Watchdog timeout has occurred, and 1 indicates that a Host Watchdog timeout has occurred. Bit 7: 0 indicates that the Host Watchdog is disabled, and 1 indicates that the Host Watchdog is enabled. The status of the Host Watchdog is stored in the EEPROM, and can only be reset by using the ~AA1 command. See Section 6.2.26 for more 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	~030
Response	!0380
Reads the status of the Host Watchdog for module 03 and returns a response indicating that the command was successful, with a value of 80, meaning that the Host Watchdog is enabled and no Host Watchdog timeout has occurred.	

Command	~030
Response	!0304
Reads the status of the Host Watchdog for module 03 and returns a response indicating that the command was successful, with a value of 04, meaning that a Host Watchdog timeout has occurred.	

※Related Commands: Section 6.2.24 ~**, Section 6.2.26 ~AA1, Section 6.2.27 ~AA2, Section 6.2.28 ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation

6.3.23 ~AA1

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	~030
Response	!0304
Reads the status of the Host Watchdog for module 03 and returns a response indicating that the command was successful, and that a Host Watchdog timeout has occurred.	

Command	~031
Response	!03
Resets the status of the Host Watchdog timeout for module 03 and returns a response indicating that the command was successful.	

Command	~030
Response	!0300
Reads the status of the Host Watchdog for module 03 and returns a response indicating that the command was successful, and showing that no Host Watchdog timeout has occurred.	

※Related Commands: Section 6.2.24 ~**, Section 6.2.25 ~AA0, Section 6.2.27 ~AA2, Section 6.2.28 ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation

6.3.24 ~AA2

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

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

Response	
Valid Command	!AAETT[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
E	The status of the Host Watchdog 0: The Host Watchdog is disabled 1: The Host Watchdog is enabled
TT	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.	

Example	
Command	~033164
Response	!03
Enables the Host Watchdog for module 03 and sets the Host Watchdog timeout value to 64 (10.0 seconds). The module returns a response indicating that the command was successful.	

Command	~032
Response	!03164
Reads the Host Watchdog timeout value for module 03 and returns a response indicating that the command was successful, with a value of 164, which denotes that the Host Watchdog is enabled and the Host Watchdog timeout value is 64 (10.0 seconds).	

※Related Commands: Section 6.2.24 ~**, Section 6.2.25 ~AA0, Section 6.2.26 ~AA1, Section 6.2.28 ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation

6.3.25 ~AA3ETT

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

Syntax	
~AA3ETT[CHECKSUM](CR)	
~	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
3	The command to enable or disable the Host Watchdog
E	The command to set the Host Watchdog: 0: Disables the Host Watchdog 1: Enables the Host Watchdog
TT	Two hexadecimal digits to represent the Host Watchdog timeout value in tenths of a second. For example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	~033164
Response	!03
Enables the Host Watchdog for module 03 and sets the Host Watchdog timeout value to 10.0 seconds. The module returns a response indicating that the command was successful.	

Command	~032
Response	!03164
Reads the Host Watchdog timeout value for module 03. The module returns a response indicating that the command was successful, with a value of 164, which denotes that the Host Watchdog is enabled and that the Host Watchdog timeout value is 10.0 seconds.	

※Related Commands: Section 6.2.24 ~**, Section 6.2.25 ~AA0, Section 6.2.26 ~AA1, Section 6.2.27 ~AA2

※Related Topics: Section 7.2 Dual Watchdog Operation

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

6.3.26 ~AAC

Description
This command is used to read whether the CJC (cold junction compensation) function for a specified module is enabled or disabled.

Syntax	
~AAC[CHECKSUM](CR)	
~	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
C	The command to read whether the CJC function is enabled or disabled

Response	
Valid Command	!AAN[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
N	The Status of the CJC function: 0: CJC function is disabled 1: CJC function is enabled
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$03C1
Response	!03
Enables CJC function for module 03, and returns a response indicating that the command was successful.	

Command	~03C
Response	!031
Reads the status of the CJC function for module 03 and returns a response indicating that the command was successful, and that the CJC function is enabled.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.15 $AA9, Section 6.2.16 $AA9SNNNN, Section 6.2.17 $AA9Ci, Section 6.2.18 $AA9SNNNNCi, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.30 ~AACN, Section 6.2.44 @AAOD

6.3.27 ~AACN

Description
This command is used to enable or disable CJC(cold junction compensation) of a function specified module.

Syntax	
~AACN[CHECKSUM](CR)	
~	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
C	The command to enable or disable the CJC function
N	The Status of the CJC function: 0: Disables CJC function 1: Enables CJC function

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	$03C1
Response	!03
Enables CJC function for module 03, and, and returns a response indicating that the command was successful.	

Command	~03C
Response	!031
Reads the status of the CJC function for module 03 and returns a response indicating that the command was successful, andf that CJC function is enabled.	

※Related Commands: Section 6.2.9 $AA3, Section 6.2.15 $AA9, Section 6..2.16 $AA9SNNNN, Section 6.2.17 $AA9Ci, Section 6.2.18 $AA9SNNNNCi, Section 6.2.19 $AAA, Section 6.2.20 $AAAi, Section 6.2.29 ~AAC, Section 6.2.44 @AAOD

6.3.28 ~AAEV

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$030
Response	?03
Attempts to send a command to perform a span calibration on module 03, but returns a response indicating that the command was unsuccessful because the “Enable Calibration” command (~AAEV) has not yet been sent.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a response indicating that the command was successful.	

Command	$030
Response	!03
Sends a command to perform a span calibration on module 03 and returns a response indicating that the command was successful.	

※Related Commands: Section 6.2.6 $AA0, Section 6.2.7 $AA1, Section 6.2.23 $AAS1
※Related Topics: Section 5 Calibration
※Note: This command must be sent before any other calibration commands can be used.

6.3.29 ~AAO(Name)

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

Syntax	
~AAO(Name)[CHECKSUM](CR)	
~	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
O	The command to set the name of the module
(Name)	The new name of the module (Max. 8 characters)

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	~03OZT-2018
Response	!03
Sets the name of module 03 to “ZT-2018” and returns a response indicating that the command was successful.	

Command	$03M
Response	!03ZT-2018
Reads the name of module 03 and returns a response indicating that the command was successful, with the name “ZT-2018”.	

Command	~03O123456789ABCDEF
Response	?03
Attempts to set the name of module 03 to “123456789ABCDEF”, but returns a response indicating that the command was unsuccessful, because the name is longer than 8 characters..	

※Related Commands: Section 6.2.22 $AAM

6.3.30 @AACH

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

Syntax	
@AACH[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
CH	The command to clear the high latch values for all Analog Input channels

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RH0
Response	!03+05.000
Reads the high latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with a value of +05.000 (+5.0 V).	

Command	@03CH
Response	!03
Clears the high latch values for all Analog Input channels of module 03 and returns a response indicating that the command was successful.	

Command	@03RH0
Response	!03+00.000
Reads the high latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the high latch has been cleared.	

※Related Commands: Section 6.2.34 @AACHi, Section 6.2.45 @AARH, Section 6.2.46 @AARHi

6.3.31 @AACHi

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RH1
Response	!03+06.000
Reads the high latch value for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, with a value of +06.000 (+6.0 V).	

Command	@03CH1
Response	!03
Clears the high latch value for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful.	

Command	@03RH1
Response	!03+00.000
Reads the high latch value for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying that the high latch value has been cleared.	

Command	@03CH9
Response	?03
Attempts to clear the high latch value for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.33 @AACH, Section 6.2.45 @AARH, Section 6.2.46 @AARHi

6.3.32 @AACHCi

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

Syntax	
@AACHCi[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
CHC	The command to clear the status of the high alarm for the Analog Input channel
i	The Analog Input channel to be cleared, zero based

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03DI
Response	!038000
Reads the current status of the alarms for module 03 and returns a response indicating that the command was successful, and that a high alarm has occurred on Analog Input channel 7.	

Command	@03CHC7
Response	!03
Clears the status of the high alarm for Analog Input channel 7 of module 03 and returns a response indicating that the command was successful.	

Command	@03DI
Response	!030000
Reads the current status of the alarms for module 03 and returns a response indicating that the command was successful, and that no alarms have occurred.	

Command	@03CHC9
Response	?03
Attempts to clear the status of the high alarm for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.39 @AADHCi, Section 6.2.40 @AADI, Section 6.2.42 @AAHI(Data)CiT, Section 6.2.47 @AARHCi

6.3.33 @AACL

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RL0
Response	!03-05.000
Reads the low latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with a value of -05.000 (-5.0 V).	

Command	@03CL
Response	!03
Clears the low latch values for all Analog Input channels of module 03 and returns a response indicating that the command was successful.	

Command	@03RL0
Response	!03+00.000
Reads the low latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with a value of +00.000 (0.0 V) signifying denoting that the low latch value has been cleared.	

※Related Commands: Section 6.2.37 @AACLi, Section 6.2.48 @AARL, Section 6.2.49 @AARLi

6.3.34 @AACLi

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

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

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RL1
Response	!03-06.000
Reads the low latch value for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, with a value of -06.000 (-6.0 V).	

Command	@03CL1
Response	!03
Clears the low latch value for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful.	

Command	@03RL1
Response	!03+00.000
Reads the low latch value for Analog Input channel 1 of module 03 and returns a response indication that the command was successful, with a value of +00.000 (0.0 V) signifying denoting that the low latch value has been cleared.	

Command	@03CL9
Response	?03
Attempts to clear the low latch value for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.36 @AACL, Section 6.2.48 @AARL, Section 6.2.49 @AARLi

6.3.35 @AACLCi

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

Syntax	
@AACLCi[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
CLC	The command to clear the status of the low alarm for the Analog Input channel
i	The Analog Input channel to be cleared, zero based

Response	
Valid Command	!AA [CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03DI
Response	!030080
Reads the current status of the alarms for module 03 and returns a response indicating that the command was successful, and that a low alarm has occurred on Analog Input channel 7.	

Command	@03CLC7
Response	!03
Clears the status of the low alarm for Analog Input channel 7 of module 03 and returns a response indicating that the command was successful.	

Command	@03DI
Response	!030000
Reads the current status of the alarms for module 03 and returns a response indicating that the command was successful, and that no alarms have occurred.	

Command	@03CLC9
Response	?03
Attempts to clear the status of the low alarm for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.40 @AADI, Section 6.2.41 @AADLCi, Section 6.2.43 @AALO(Data)CiT, Section 6.2.50 @AARLCi

6.3.36 @AADHCi

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

Syntax	
@AADHCi[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
DH	The command to disable the high alarm for the Analog Input channel
Ci	The Analog Input channel where the alarm is to be disabled, zero based

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03DI
Response	!03FFFF
Reads the status of the alarms for module 03 and returns a response indicating that the command was successful, and that the high alarm for the all Analog Input channels are enabled.	

Command	@03DHC0
Response	!03
Disables the high alarm for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful.	

Command	@03DI
Response	!03FEFF
Reads the status of the alarms for module 03 and returns a response indicating that the command was successful, and that the high alarm for Analog Input channel 0 is disabled and all others are enabled.	

Command	@03DHC9
Response	?03
Attempts to disable the high alarm for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.35 @AACHCi, Section 6.2.40 @AADI, Section 6.2.42 @AAHI(Data)CiT, Section 6.2.47 @AARHCi

6.3.37 @AADI

Description
This command is used to read the status of the alarms for all Analog Input channels of a specified module.

Syntax	
@AADI[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
DI	The command to read the status of the alarms for all Analog Input channels

Response	
Valid Command	!AAHHLL[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
HH	A two-digit hexadecimal value, where bit 0 corresponds to Analog Input channel 0, and bit 1 corresponds to Analog Input channel 1, etc. When the bit is 0, it denotes that a high alarm has not occurred, and 1 denotes that a high alarm has occurred.
LL	A two-digit hexadecimal value, where bit 0 corresponds to Analog Input channel 0, and bit 1 corresponds to Analog Input channel 1, etc. When the bit is 0, it denotes that a low alarm has not occurred, and 1 denotes that a low alarm has occurred.
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	@03DI
Response	!034008
Reads the status of the alarms for module 03 and returns a response indicating that the command was successful, and that a high alarm has occurred on Analog Input channel 6 and a low alarm has occurred on Analog Input channel 3.	

※Related Commands: Section 6.2.35 @AACHCi, Section 6.2.38 @AACLCi, Section 6.2.39 @AADHCi, Section 6.2.41 @AADLCi, Section 6.2.42 @AAHI(Data)CiT, Section 6.2.43 @AALO(Data)CiT, Section 6.2.47 @AARHCi, Section 6.2.50 @AARLCi

6.3.38 @AADLCi

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

Syntax	
@AADLCi[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
DL	The command to disable the low alarm for the Analog Input channel
Ci	The Analog Input channel where the alarm is to be disabled, zero based

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03DLC5
Response	!03
Disables the low alarm for Analog Input channel 5 of module 03 and returns a response indicating that the command was successful.	

Command	@03DI
Response	!03FFDF
Reads the status of the alarms for module 03 and returns a response indicating that the command was successful, and that the low alarm for Analog Input channel 5 is disabled and all others are enabled.	

Command	@03DLC9
Response	?03
Attempts to disable the low alarm for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.38 @AACLCi, Section 6.2.40 @AADI, Section 6.2.41 @AADLCi, Section 6.2.43 @AALO(Data)CiT, Section 6.2.50 @AARLCi

6.3.39 @AAHI(Data)CiT

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

Syntax	
@AAHI(Data)CiT[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
HI	The command to enable the high alarm for the Analog Input channel
(Data)	The high alarm limit, which should be consistent with the engineering units format. Refer to Section 4 for details of the data format.
Ci	The Analog Input channel where the high alarm is to be enabled, zero based
T	The Alarm Type: M: Momentary Alarm L: Latched Alarm

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03HI+09.000C0M
Response	!03
Sets the high alarm limit for Analog Input channel 0 of module 03 to +09.000 (+9.0 V) and sets the Alarm Type to momentary, and returns a response indicating that the command was successful.	

Command	@03RHC0
Response	!03+09.0001
Reads the status of the high alarms for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, and that the high alarm limit is +09.000 (+9.0 V), and the Alarm Type is momentary.	

Command	@03HI+09.000C9M
Response	!03
Attempts to set the high alarm limit for Analog Input Analog Input channel 9 of module 03 to +09.000 (+9.0 V) and sets the Alarm Type to momentary, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.35 @AACHCi, Section 6.2.39 @AADHCi, Section 6.2.40 @AADI, Section 6.2.47 @AARHCi

※Related Topics: Section 4 Analog Input Type and Data Format

6.3.40 @AALO(Data)CiT

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

Syntax	
@AALO(Data)CiT[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LO	The command to enable the low alarm for the Analog Input channel
(Data)	The low alarm limit, which should be consistent with the engineering units format. Refer to Section 4 for details of the data format.
Ci	The Analog Input channel where the low alarm is to be enabled, zero based
T	The Alarm Type: M: Momentary Alarm L: Latched Alarm

Response	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03LO-03.000C1L
Response	!03
Sets the low alarm limit for Analog Input channel 1 of module 03 to -03.000 (-3.0 V) and sets the Alarm Type to latched, and returns a response indicating that the command was successful.	

Command	@03RLC1
Response	!03-03.0002
Reads the status of the low alarms for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, and that the low alarm limit is -03.000 (-3.0 V) and the Alarm Type is latched.	

Command	@03LO-03.000C9L
Response	!03
Attempts to set the low alarm limit for Analog Input channel 9 of module 03 to -03.000 (-3.0 V) and the Alarm Type to latched, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.38 @AACLCi, Section 6.2.40 @AADI, Section 6.2.41 @AADLCi, Section 6.2.50 @AARLCi

※Related Topics: Section 4 Analog Input Type and Data Format

6.3.41 @AAOD

Description
This command is used to read the status of the CJC (cold hunction compensation) connection for a specified module.

Syntax	
@AAOD[CHECKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
OD	The command to read the status of the CJC connection

Response	
Valid Command	!AAS[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
S	The status of the CJC connection: 0: The CJC is Disconnected 1: The CJC is Connected
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	@03OD
Response	!031
Reads the status of the CJC connection for module 03 and returns a response indicating that the command was successful, and that the CJC is connected.	

6.3.42 @AARH

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

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The high latch values for all Analog Input channels. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RH
Response	!03+08.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the high latch values for all Analog Input channels on module 03 and returns a response indicating that the command was successful, with the data in engineering units format.	

※Related Commands: Section 6.2.33 @AACH, Section 6.2.34 @AACHi, Section 6.2.46 @AARHi
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.43 @AARHi

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

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The high latch value for the specified Analog Input channel. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RH0
Response	!03+08.000
Reads the high latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with the data in engineering units format.	

Command	@03RH9
Response	?03
Attempts to read the high latch value for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.33 @AACH, Section 6.2.34 @AACHi, Section 6.2.45 @AARH
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.44 @AARHCi

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

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

Response	
Valid Command	!AA(Data)S[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The high alarm value for the specified Analog Input channel in engineering units format. See Section 4 for details of the data format.
S	The Alarm Type: 0: Alarm Disabled 1: Momentary Alarm 2: Latched Alarm
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	@03HI+09.000C0M
Response	!03
Sets the high alarm limit for Analog Input channel 0 of module 03 to +09.000 (+9.0 V) and sets the Alarm Type to momentary, and returns a response indicating that the command was successful.	

Command	@03RHC0
Response	!03+09.0001
Reads the status of the high alarm for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, and that the high alarm limit is +09.000 (+9.0 V) and the Alarm Type is momentary.	

Command	@03REC9
Response	?03
Attempts to read the status of the high alarm for Analog Input channel 9 of module 01, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.35 @AACHCi, Section 6.2.39 @AADHCi, Section 6.2.40 @AADI, Section 6.2.42 @AAHI(Data)CiT

※Related Topics: Section 4 Analog Input Type and Data Format

6.3.45 @AARL

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

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The low latch values for all Analog Input channels. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	@03RL
Response	!03-02.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the low latch values for all Analog Input channels on module 03 and returns a response indicating that the command was successful, with the data in engineering units format.	

※Related Commands: Section 6.2.36 @AACL, Section 6.2.37 @AACLi, Section 6.2.49 @AARLi
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.46 @AARLi

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

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

Response	
Valid Command	!AA(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The low latch value for the specified Analog Input channel. See Section 4 for details of the data format.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	@03RL0
Response	!03-02.000
Reads the low latch value for Analog Input channel 0 of module 03 and returns a response indicating that the command was successful, with the data in engineering units format.	

Command	@03RL9
Response	?03
Attempts to read the low latch value for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.36 @AACL, Section 6.2.37 @AACLi, Section 6.2.48 @AARL
※Related Topics: Section 4 Analog Input Type and Data Format

6.3.47 @AARLCi

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

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

Response	
Valid Command	!AA(Data)S[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The status of the low alarm for the specified Analog Input channel in engineering units format. See Section 4 for details of the data format.
S	The Alarm Type: 0: Alarm Disabled 1: Momentary Alarm 2: Latched Alarm
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.	

Example	
Command	@03LO-03.000C1L
Response	!03
Sets the low alarm limit for Analog Input channel 1 of module 03 to -03.000 (-3.0 V) and sets the Alarm Type to latched, and returns a response indicating that the command was successful.	

Command	@03RLC1
Response	!03-03.0002
Reads the status of the alarms for Analog Input channel 1 of module 03 and returns a response indicating that the command was successful, and that the low alarm limit is -03.000 (-3.0 V) and the Alarm Type is latched.	

Command	@03RLC9
Response	?03
Attempts to reads the status of the low alarm for Analog Input channel 9 of module 03, but returns a response indicating that the command was unsuccessful because Analog Input channel 9 does not exist.	

※Related Commands: Section 6.2.38 @AACLCi, Section 6.2.40 @AADI, Section 6.2.41 @AADLCi, Section 6.2.43 @AALO(Data)CiT

※Related Topics: Section 4 Analog Input Type and Data Format

6.4 Modbus RTU Protocol Command set

The Modbus Protocol was developed by Modicon Inc., and was originally designed for Modicon controllers. Detailed information regarding the Modbus RTU Protocol can be found at:
http://www.modicon.com
and http://www.modbus.org

➢ Modbus RTU Command Format

Field 1	Field 2	Field 3	Field 4~n	Field n+1~n+2
Module Address	Function Code	Sub Function	Configuration Field	CRC16

Function Code	Description
0x04	Reads the input channels
0x46	Reads/writes the module settings

Examples:

A. To read the Analog Input value for module 01, the following command should be sent:
   01 04 00 00 00 08 F1 CC
B. To read the name of the module, the following command should be sent:
   01 46 00 12 60

6.4.1 Modbus Address Mapping

Address Mapping		
Address	Description	Attribute
00259	The Filter settings. 0: 60 Hz rejection 1: 50 Hz rejection	R/W
00260	The Modbus Host Watchdog mode: 0: The same as the I-7000 series modules 1: The Analog Output and Digital Output commands can be used to clear the status of the Host Watchdog timeout	R/W
00261	Enables or disabsles the Host Watchdog: 0: Disable 1: Enable	R/W
00268	Enables or disabsles the CJC: 0: Disable 1: Enable	R/W
00269	The Modbus Data Format: 0: Hexadecimal 1: Engineering Units	R/W
00270	The status of the Host Watchdog timeout. Write 1 to clear.	W
00272	The factory calibration parameters. Write 1 to load.	W
00273	The Reset status: 0: This is NOT the first time the module has been read after being powered on 1: This is the first time the module has been read after being powered on	R
00279	The CJC connection status: 0: Disconnected 1: Connected	R
00280	The high latch values for all Analog Input channels. Write 1 to clear.	W
00281	The low latch values for all Analog Input channels. Write 1 to clear.	W
00513 ~ 00520	The high latch values for Analog Input channels 0 to 7. Write 1 to clear.	W
00545 ~ 00552	The low latch values for Analog Input channels 0 to 7. Write 1 to clear.	W

00577 ~ 00584	Enables or disables the high alarm for Analog Input channels 0 to 7: 0: Disable 1: Enable	R/W
00609 ~ 00616	Enables or disables the low alarm for Analog Input channels 0 to 7: 0: Disable 1: Enable	R/W
00641 ~ 00648	The high Alarm Type for Analog Input channels 0 to 7: 0: Momentary 1: Latch	R/W
00673 ~ 00680	The low Alarm Type for Analog Input channels 0 to 7: 0: Momentary 1: Latch	R/W
00705 ~ 00712	The status of the high alarm for Analog Input channels 0 to 7	R/W
00737 ~ 00744	The status of the low alarm for Analog Input channels 0 to 7	R/W
10129 ~ 10136	The under range status of Analog Input channels 0 to 7 (supports types 0x7 and 0x1A only)	R
30001 ~ 30008	The Analog Input value for Analog Input channels 0 to 7	R
30129	The CJC temperature	R
30513 ~ 30520	The high latch value for Analog Input channels 0 to 7	R
30545 ~ 30552	The low latch value for Analog Input channels 0 to 7	R
40257 ~ 40264	The Type Code for Analog Input channels 0 to 7	R/W
40353 ~ 40360	The CJC offset value for Analog Input channels 0 to 7	R/W
40481	The Firmware Version (Low Word)	R
40482	The Firmware Version (High Word)	R
40483	The Module Name (Low Word)	R
40484	The Module Name (High Word)	R
40485	The Module Address. Valid Range: 1 ~ 247	R
40486	The Baud Rate: Bits 5:0 Baud Rate. Always set to 0x0A Bits 7:6 Reserved	R
40489	The Host Watchdog timeout value. The valid range is 0 ~ 255, in 0.1 second intervals	R/W

40490	Enables or disables a specific Analog Input channel	R/W
40491	The CJC offset value	R/W
40492	The Host Watchdog timeout counter value. Write 0 to clear	R/W
40577 ~ 40584	The high alarm value for Analog Input channels 0 to 7	R/W
40609 ~ 40616	The low alarm value for Analog Input channels 0 to 7	R/W

6.4.2 PLC Address Mapping

Function Code	Description	Section
0x01	Reads the Coils	6.3.3
0x02	Reads the Discrete Inputs	6.3.4
0x03	Reads Multiple Registers	6.3.5
0x04	Reads Multiple Input Registers	6.3.6
0x05	Writes a Single Coil	6.3.7
0x06	Writes Multiple Registers	6.3.8
0x0F	Writes Multiple Coils	6.3.9
0x46	Reads/Writes the Module Settings	6.3.10

If the function specified in the message is not supported, then the module will respond with an error code as per the table below. Note that the address mapping for the Modbus protocol is Base 0.

Error Response

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

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

6.4.3 01 (0x01) Reading the Coils

Description
This function code is used to read the current Digital Output values from the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x01
02~03	Starting Channel Number or Address Mapping	2	See Section 6.3.1 for details
04~05	Output Channel Number or Bit Count	2	0x0001 to 0x0020

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x01
02	Byte Count	1	Byte Count of the Response (B=(Bit Count + 7)/8)
03	Bit Values	B	(Bit Values)

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x81
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.4 02 (0x02) Reading the Discrete Inputs

Description
This function code is used to read the current Digital Output values from the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x02
02~03	Starting Channel Number or Address Mapping	2	See Section 6.3.1 for details
04~05	Output Channel Number or Bit Count	2	0x0001 to 0x0020

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x02
02	Byte Count	1	Byte Count of the Response (B=(Bit Count + 7)/8)
03	Bit Values	B	(Bit Values)

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x82
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.5 03 (0x03) Reading Multiple Registers

Description
This function code is used to read the current Analog Input values from the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x03
02~03	Starting Channel Number or Address Mapping	2	See Section 6.3.1 for details
04~05	Input Channel Number or Bit Count	2	0x0001 to 0x0020

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x03
02	Byte Count	1	Byte Count of the Response (B=2 * Word Count)
03~	Register Values	B*2	Register Values

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x83
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.6 04 (0x04) Reading Multiple Input Registers

Description
This function code is used to read the current Analog Input counter values from the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x04
02~03	Starting Channel Number or Address Mapping	2	See Section 6.3.1 for details
04~05	Input Channel Number or Bit Count	2	0x0001 to 0x0020

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x04
02	Byte Count	1	Byte Count of the Response (B=2 * Word Count)
03~	Register Values	B*2	Register Values

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x84
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.7 05 (0x05) Writing a Single Coil

Description
This function code is used to write the Digital Output value for the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x05
02~03	Starting Channel Number or Address Mapping	2	See Section 6.3.1 for details
04~05	Output Value	2	A value of 0xFF00 sets the output to ON. A value of 0x0000 sets the output to OFF.

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x05
02~03	Output Channel Number	2	This value is the same as bytes 02 and 03 of the Request
04~05	Output Value	2	This value is the same as bytes 04 and 05 of the Request

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x85
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.8 06 (0x06) Writing Multiple Registers

Description
This function code is used to configure the settings for the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x06
02~03	Address Mapping	2	See Section 6.3.1 for details
04~05	Register Value	2	Register Value

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x06
02~03	Address Mapping	2	The value is the same as bytes 02 and 03 of the Request
04~05	Register Value	2	Register value

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x86
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.9 15 (0x0F) Writing Multiple Coils

Description
This function code is used to write the Digital Output value for the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x0F
02~03	Starting Channel Number	2	See Section 6.3.1 for details
04~05	Output Channel Number	2	0x0001 to 0x0020
06	Byte Count	1	B=(Bit Count + 7)/8
07	Output Value	2	A bit corresponds to a channel. When the bit is '0', it denotes that the channel that was set is OFF or Disable. If the bit is '1', it denotes that the channel that was set is ON or Enable.

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x0F
02~03	Starting Channel Number	2	The value is the same as bytes 02 and 03 of the Request
04~05	Output Channel Number	2	0x0001 ~ 0x0020

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x8F
02	Exception Code	1	Refer to the Modbus standard for more details

6.4.10 70 (0x46) Reading/Writing the Module Settings

Description		
This function code is used to read the configuration settings from the module or to change the settings for the module. The following sub-function codes are supported.		
Sub-function Code	Description	Section
00 (0x00)	Reads the Name of the Module	A.1
04 (0x04)	Sets the Address of the Module	A.2
07 (0x07)	Reads the Type Code	A.3
08 (0x08)	Sets the Type Code	A.4
32 (0x20)	Reads the Firmware Version	A.5
37 (0x25)	Reads whether a Specific Channel is Enabled or Disabled	A.6
38 (0x26)	Sets a Specific Channel to Enabled or Disabled	A.7
41 (0x29)	Reads the Miscellaneous Settings	A.8
42 (0x2A)	Writes the Miscellaneous Settings	A.9
43 (0x2B)	Reads the CJC offset for a specific channel	A.10
44 (0x2C)	Sets the CJC offset for a specific channel	A.11
45 (0x2D)	Reads the CJC function is enabled or disable	A.12
46 (0x2E)	Enables or disables the CJC function	A.13
47 (0x2F)	Reads the CJC update settings	A.14
48 (0x30)	Sets the CJC update settings	A.15

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

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.1 00 (0x00) Reading the Name of a Module

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

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

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x00
03~06	Module Name	4	0x54 0x20 0x18 0x00(ZT-2018)

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 00 [12 60]
Response	01 46 00 54 20 18 00 [1E 9C]

A.2 04(0x04) Setting the Address of the Module

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

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x04
03	New Address	1	1 to 247
04~06	Reserved	3	0x00 0x00 0x00

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x04
03	New Address	1	0: OK Others: Error
04~06	Reserved	3	0x00 0x00 0x00

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 04 02 00 00 00 [F5 1E]
Response	01 46 04 00 00 00 00 [F4 A6]

A.3 07 (0x07) Reading the Type Code

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

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

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x07
03	Type Code	1	The Type Code. See Section 4 for details of the data format.

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 07 00 01 [7C 89]
Response	01 46 07 00 [E2 3D]

A.4 08 (0x08) Setting the Type Code

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

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x08
03	Reserved	1	0x00
04	Channel Number	1	0x00 ~ 0x07
05	Type Code	1	The Type Code. See Section 4 for details of the data format.

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

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 20 [13 B8]
Response	01 46 20 01 00 00 [D2 05]

A.5 32 (0x20) Reading the Firmware Version Information

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

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

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

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 20 [13 B8]
Response	01 46 20 0A 01 00 00 [D6 B9]

A.6 37 (0x25) Reading whether a Channel is Enabled or Disabled

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

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

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

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 25 [D3 BB]
Response	01 46 25 07 [BB 5F]

A.7 38 (0x26) Enabling or Disabling a Channel

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

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

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x26
03	Enabled/Disabled Settings	1	0: OK Others: Error.

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

Example	
Command	01 46 26 01 [3B AD]
Response	01 46 26 00 [FA 6D]

A.8 43(0x2B) Reading the CJC (cold junction compensation) Offset

Description
This sub-function code is used to read the CJC offset for a module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2B
03	Channel Number	1	0x00: The total CJC offset 0x80~0x87: The CJC offset for channels 0~7

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2B
03~04	CJC Offset	2	The CJC offset value (units: 0.01°C)

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.9 44(0x2C) Writing the CJC (cold junction compensation) Offset

Description
This sub-function code is used to set the CJC offset for a module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2C
03	Channel Number	1	0x00: The total CJC offset 0x80~0x87: The CJC offset for channels 0~7
04~05	CJC Offset	2	The CJC offset value (units: 0.01°C)

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2C
03	Miscellaneous Settings	1	0: OK Others: Error

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.10 45(0x2D) Reading whether the CJC Function is Enabled or Disabled

Description
This sub-function code is used to read whether the CJC function for a module is enabled or disabled.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2D
03	Reserved	1	0x00

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2D
03	CJC Status	1	0x00: Disabled 0x01: Enabled

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.11 46(0x2E) Enabling or Disabling the CJC Function

Description
This sub-function code is used to enable or disable the CJC function for a module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2E
03	Reserved	1	0x00
04	CJC Status	1	0x00: Disabled 0x01: Enabled

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2E
03	Miscellaneous Settings	1	0: OK Others: Error

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.12 47(0x2F) Reading the CJC Update Status

Description
This sub-function code is used to read the status of the CJC update for a module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2F

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2F
03	CJC Update Status	1	0x00: Stopped 0x01: Started 0x02: The CJC status will be updated only once

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

A.13 48(0x30) Writing the CJC Update Status

Description
This sub-function code is used to write the status of the CJC update for a module.

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x30
03	CJC Update Status	1	0x00: Stopped 0x01: Started 0x02: Update the CJC status only once

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x30
03	Miscellaneous Settings	1	0: OK Others: Error

Error Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details

7 Appendix

7.1 Software Configuration Mode

Each ZT-2000 I/O device contains a built-in EEPROM memory that is used to store configuration information, such as the address, the data format, the Analog Input Type Code and other information. When the module is powered on with the Address (Node ID) set to 0x00, the ZT-2000 I/O device will be set to the software configuration mode. In this mode, the configuration details (Address(Node ID), data format and Analog Input Type Code) are loaded from the EEPROM. The settings can then be changed using the %AANNTTCCFF and $AA7CiRrr commands. When the ZT-2000 I/O device is set to software configuration mode, the switch settings are ignored.

7.2 Dual Watchdog Operation

Dual Watchdog = Module Watchdog + Host Watchdog

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

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

ZT-2000 series devices include an internal Dual Watchdog, making the control system more reliable and stable.

7.3 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, and is cleared after responding to the first $AA5 command. This can be used to check whether the module has been previously reset. When the response to the $AA5 command indicates that the reset status has been cleared, it means that the module has not been reset since the last $AA5 command was sent. When the response to the $AA5 command indicates that the reset status has been set, and it is not the first time the $AA5 command has been sent, it means that the module has been reset and the Digital Output value has been changed to the power-on value.

8 Troubleshooting

A. Technical Support.

If you have any difficulties using your ZT-2000 series I/O device, please send a description of the problem to service@icpdas.com

Include the following items in your email:

- *A description or diagram of the current DIP switch positions.*
- *A copy of the configuration file for the ZT-2000 coordinator. This file can be obtained using the procedure outlined below and should be attached to your email.*

B. Set the DIP switch for the ZT-255x device to the [ZBSET] position then reboot the device. Launch the ZT Configuration Utility and select the [Save Log] icon to save the configuration of the ZT-255x as a file.

C. After clicking the [Save Log] icon, enter the "File Name" and the "File Path" in the Windows "Save" dialog box. Once the configuration has been successfully saved, the following message will be displayed.