ZT-2017 and ZT-2017C 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 purpose only and may be registered trademarks of their respective companies.

Technology Support

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

Table of Contents

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 able 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 rate of 250 kbit/s, best suited for periodic or intermittent data or 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 Device*

The ZT-2000 I/O series devices are small-sized wireless ZigBee I/O modules based on the IEEE802.15.4 standard that allow data acqusition and control via the personal area ZigBee network. Reference Sec. 2.1 for more detail information.

The ZT-2000 I/O series devices is a wireless data acqusition based client/server system. Accordingly, A Net Server of the ZigBee(ZT-2570/ZT-2550) is essential in such system. Please refer to "ZT-25XX ZigBee converter quick start " for more information as following links:
http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zt_series/document/

2 Information to the Hardware

2.1 Specifications

ZT-2017

Analog Input	
Input Channels	8 Differential
Input Type	+/-10 V, +/-5 V, +/-1 8V, +/-500 mV, +/-150 mV, -20 mA ~ +20 mA (Requires Optional External 125 Ω Resistor)
Resolution	16bit
Sampling Rate	16bit, 10 Samples/Sec. (Total)
Accuracy	+/-0.1% FSR
-3dB Bandwidth	15.7Hz
Zero Drift	+/-20 μV/℃
Span Drift	+/-25 ppm/℃
Common Mode Rejection	86 dB
Normal Mode Rejection	100 dB
Input Impedance	>2 MΩ
Overvoltage Protection	240 Vrms
Individual Channels Configurable	Yes
Intra-module Isolated, Field-to-Logic	3000 VDC
ESD Protection	+/-4 kV Contact for each channel
LED Indicator	
ZigBee PWR	ZigBee Device Power
ZigBee Net	Zigbee Communication Indicator
Power	
Power Consumption	1.7W (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 (2.4GHz)	5 dBi Omni-directional antenna
Transmit Range (LOS)	700 m (Typical)
Max. Slaves Supported	255
EMI Certification	CE/FCC, FCC ID

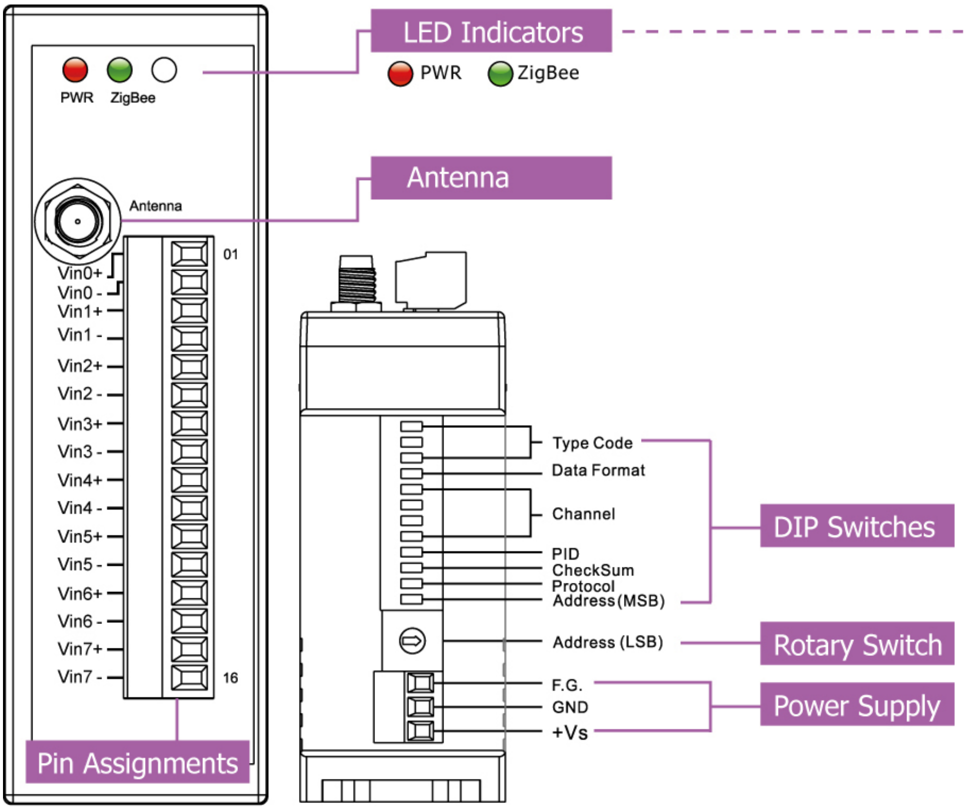
ZT-2017C

Analog Input	
Input Channels	8 Differential
Input Type	-20 mA ~ +20 mA, 0 mA ~ +20 mA, +4 mA ~ +20 mA
Resolution	16bit
Sampling Rate	16bit, 10 Samples/Sec. (Total)
Accuracy	+/-0.1% FSR
-3dB Bandwidth	15.7Hz
Zero Drift	+/-20 μV/℃
Span Drift	+/-25 ppm/℃
Common Mode Rejection	86 dB
Normal Mode Rejection	100 dB
Common Voltage	+/-200 VDC
Individual Channels Configurable	Yes
Open Wire Detection for 4 ~ 20 mA	Yes
Intra-module Isolated, Field-to-Logic	3000 VDC
ESD Protection	+/-4 kV Contact for each channel
LED Indicator	
ZigBee PWR	ZigBee Device Power
ZigBee Net	Zigbee Communication Indicator
Power	
Power Consumption	1.7W (Max.)
Environment	
Operating Temperature	-25 to 75 °C
Storage Temperature	-30 to 80 °C
Humidity	10 to 90%, Non-condensing

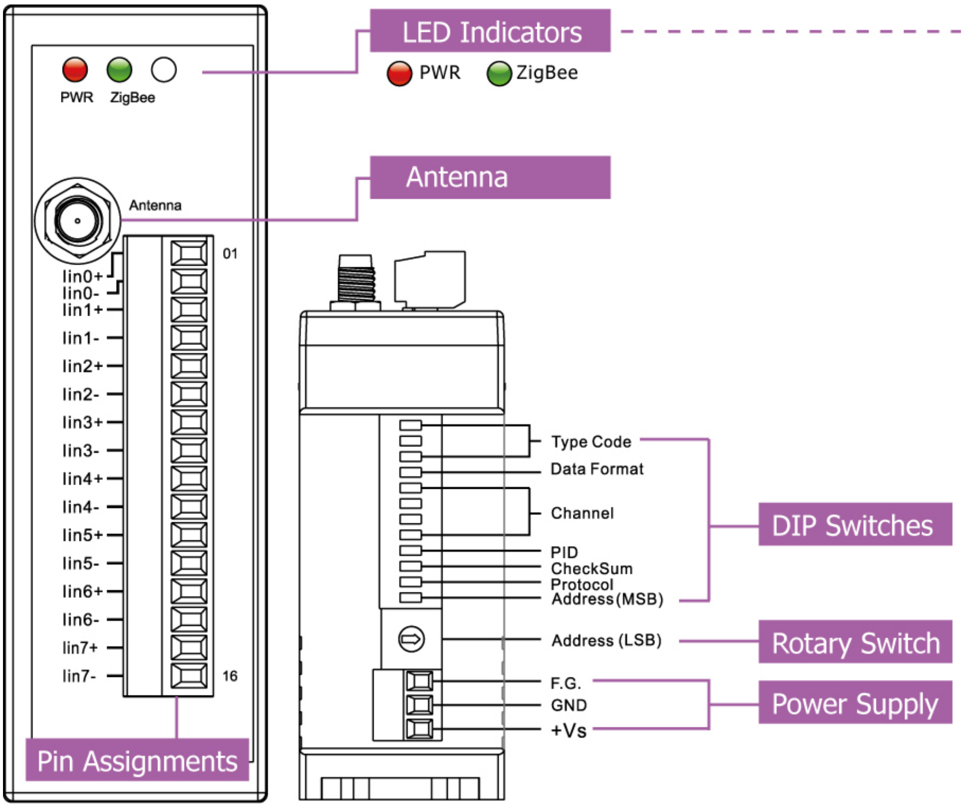
Wireless	
RF Channels	16
RF Transmit Power	11 dBm
Antenna (2.4GHz)	5 dBi Omni-directional antenna
Transmit Range (LOS)	700 m (Typical)
Max. Slaves Supported	255
EMI Certification	CE/FCC, FCC ID

Pin Assignment

ZT-2017

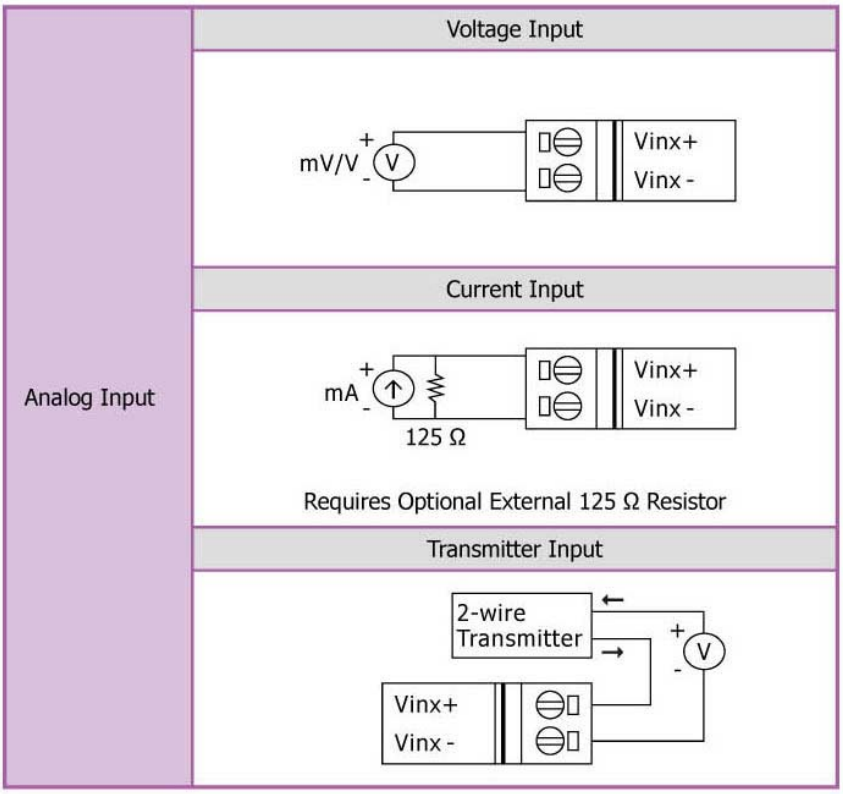


ZT-2017C

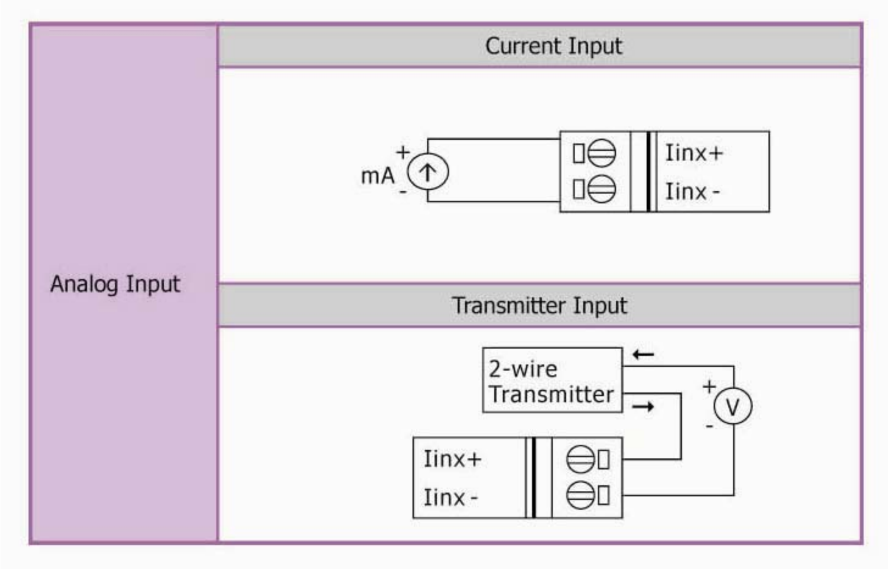


2.2 *Wire Connection*

ZT-2017



ZT-2017C



3 Setting up the ZT-2000 I/O Device

3.1 Introduction of configurations

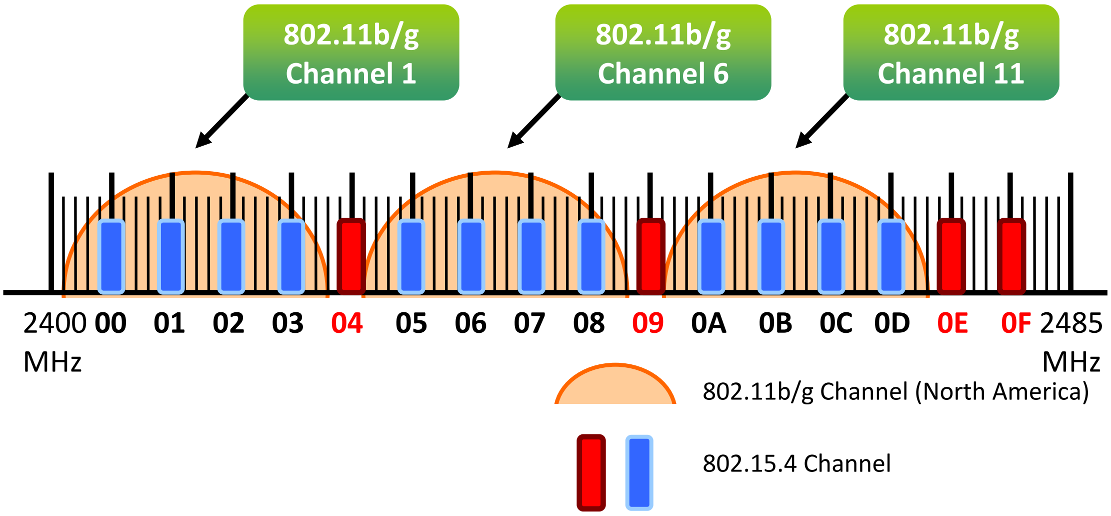
A. “Pan ID” is the group identity of a ZigBee network, and must be set to the same if they are in the same ZigBee network.

B. “Node ID” is the identity of the ZigBee module.
The identity number must be unique if it is in the same ZigBee network as other ZigBee module.

C. “RF Channel” indicates the radio frequency channel, and must be set to the same channel if the module is in the same ZigBee network as other ZigBee modules.

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

※ *In addition, the RF channels 0x04, 0x09, 0x0E or 0x0F are recommended because they do not overlap with frequencies Wi-Fi.*



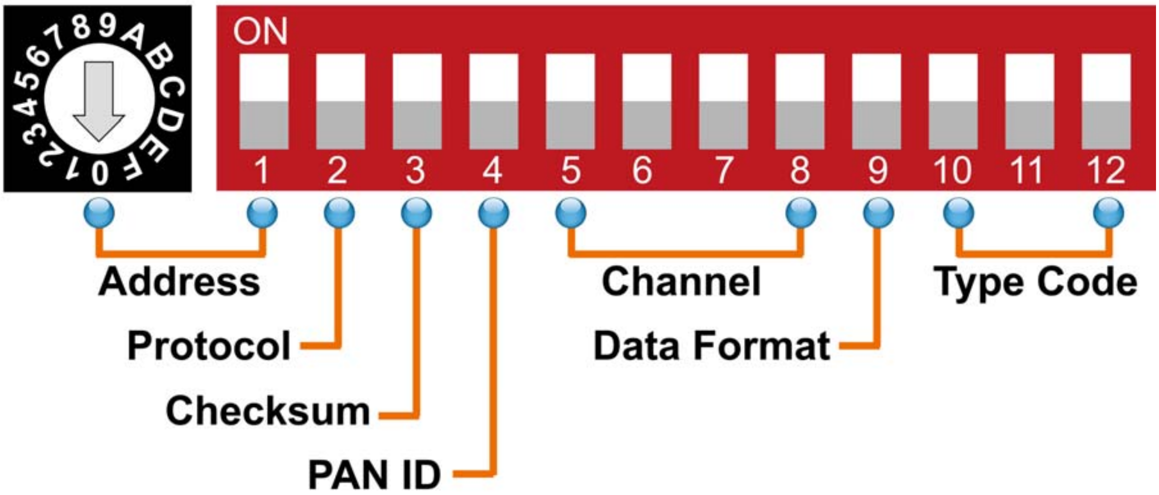
D. Protocol/Application Mode:

For using different protocol on the user program, the following recommended application mode works together.

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 Rotation and DIP Switch

The configurations is adjusted by the external rotation switch and DIP switch. User only reboot the power when ZT-2000 device configuration completed.



➢ Rotation Switch

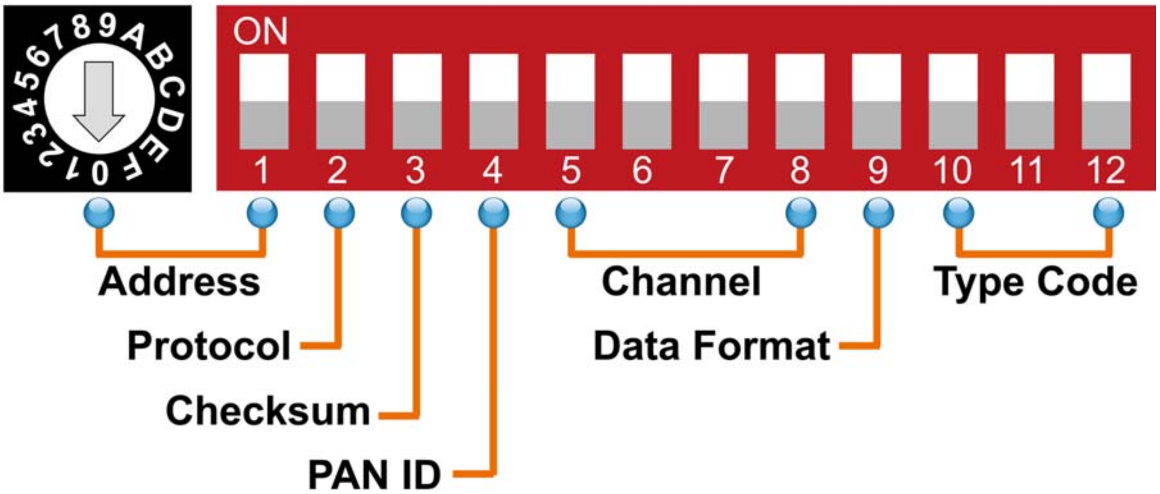
Case1 : Address MSB = 0

	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

Case1 : Address MSB = 1

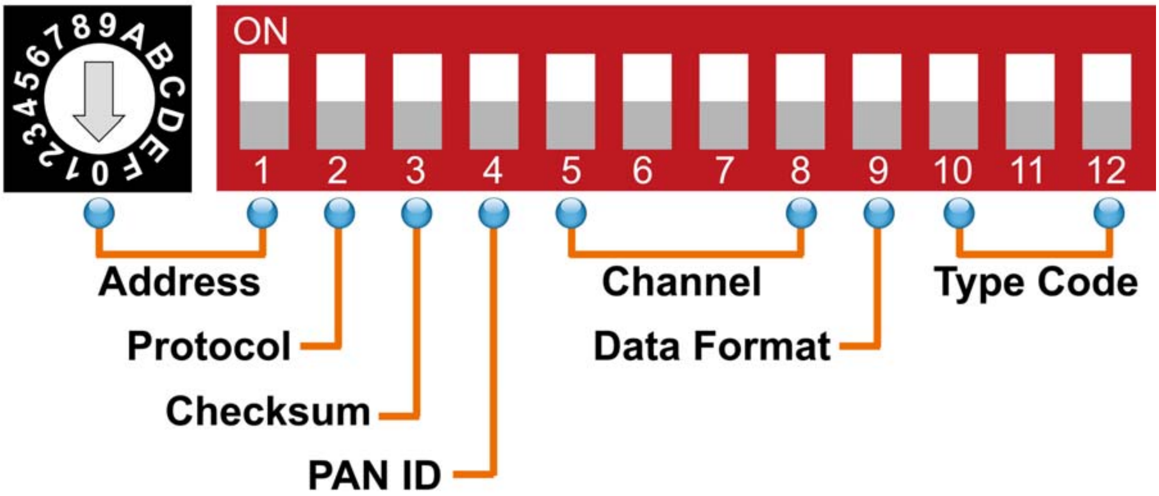
	0	1	2	3	4	5	6	7
Address	10	11	12	13	14	15	16	17
Node ID	0x0010	0x0011	0x0012	0x013	0x0014	0x0015	0x0016	0x0017
	8	9	A	B	C	D	E	F
Address	18	19	1A	0B	0C	1D	1E	1F
Node ID	0x018	0x0019	0x001A	0x001B	0x001C	0x001D	0x001E	0x001F

***Note 1 : The "Address" and "Node ID" are defined through the command. In the software configuration mode, the dip switch of "Address", "Data Format" and AI Type Code" are ignored and "Address", "Data Format" and "AI Type Code" can be set through the command.**



➢ DIP Switch

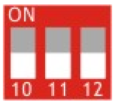
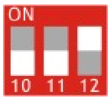
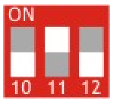
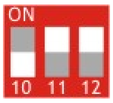
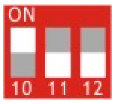
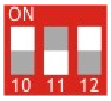
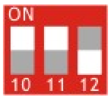
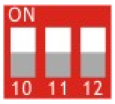
Number	Item	Status	Explain
1	Address MSB	OFF	Valid address(Node ID) from 0x01 to 0x0F
		ON	Valid address(Node ID) from 0x10, 0x01 to 0x1F
2	Protocol	OFF	DCON Protocol
		ON	Modbus RTUProtocol
3	Checksum	OFF	Disabled
		ON	Enabled
4	ZigBee Pan ID	OFF	Pan ID = 0x0000
		ON	Pan ID = 0x0001
5	ZigBee RF Channel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



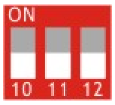
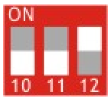
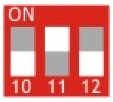
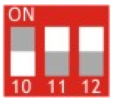
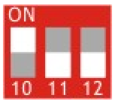
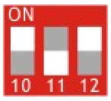
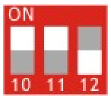
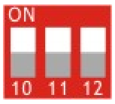
➢ Type Code

Dip switches 10-12 define the input type code of the ZT-2017 or ZT-2017C, as shown below.

ZT-2017

Switch Value	Type Code	Switch Value	Type Code	Switch Value	Type Code
	0x08		0x09		0x0A
	0x0B		0x0C		0x0D
	0x07		0x1A		

ZT-2017C

Switch Value	Type Code	Switch Value	Type Code	Switch Value	Type Code
	0x0D		0x0D		0x0D
	0x0D		0x0D		0x0D
	0x07		0x1A		

3.3 Start-up ZT-2000 I/O Device

Because the ZigBee network is in charged by the ZigBee coordinator, so user must first configure ZT-2550/ZT-2570 (ZigBee coordinator). Please see the configuration details in the documents as below links.

Once the ZigBee coordinator has completed the configuration, you only configure the ZT-2000 I/O device into the same "Pan ID" and "RF channel" and then reboot power. It will start working in the ZigBee network via the default protocol.

※ Documents

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 (Used to configure ZT-2000 I/O device Coordinator)

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

3.4 Communication Test

Once the ZT-2000 I/O device has joined ZigBee network, user may confirm the signal quality via the LED status of ZigBee Net LED indicators. If the LED indicator is steady light, it is allowed communicating with ZT-2000 I/O device for data acquisition and controlling.

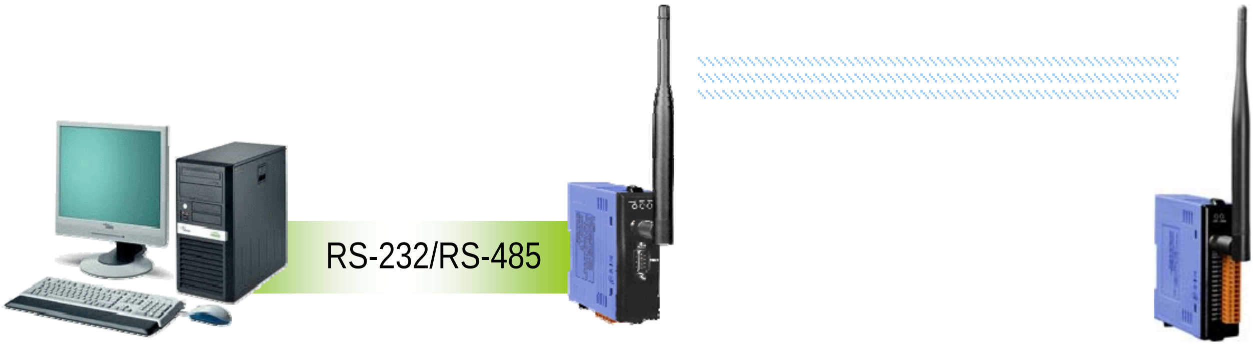
ICP DAS also provides a software "DCON Utility" to simulate the DCON/Modus communication, user may use this software to verify the setting and ZigBee I/O functions.

※ Download DCON Utility

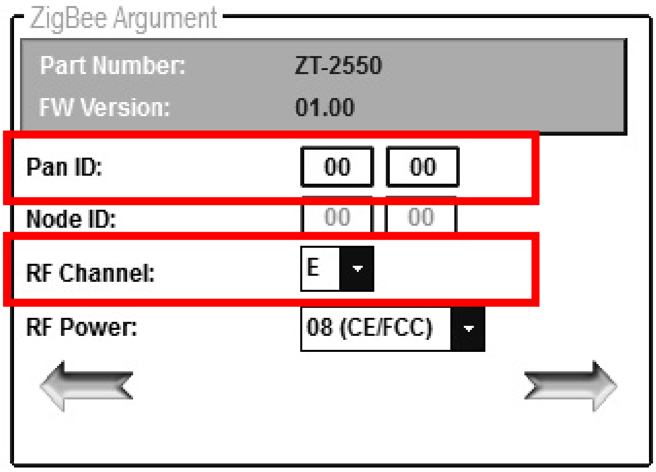
http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/

3.5 Examples

- Architecture Chart



- Configurations of ZT-2550/ZT-2570


ZigBee Argument

Part Number: ZT-2550

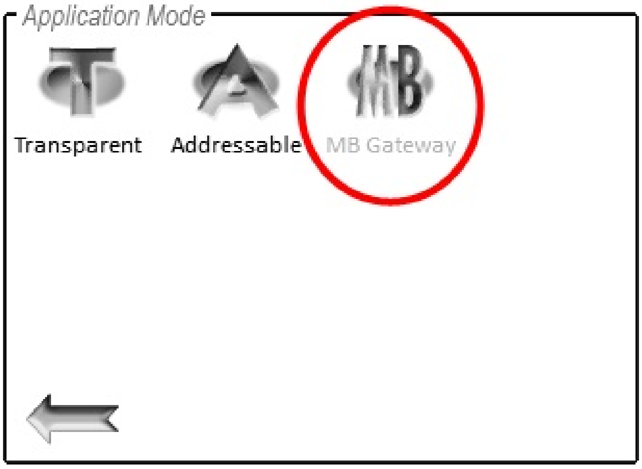
FW Version: 01.00

Pan ID: 00 00

Node ID: 00 00

RF Channel: E

RF Power: 08 (CE/FCC)

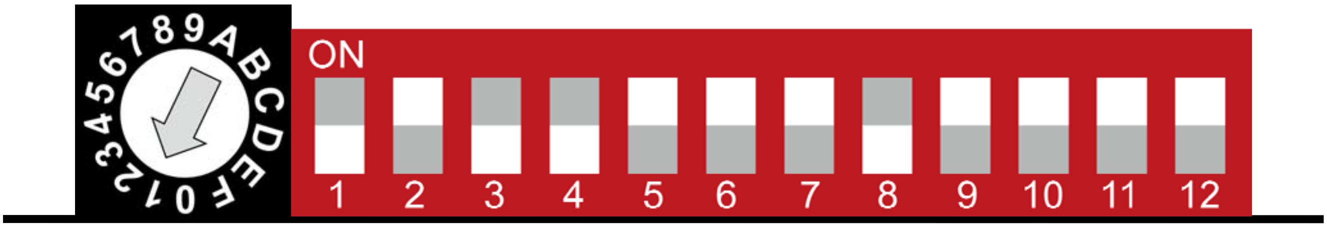

Application Mode

Transparent

Addressable

MB Gateway

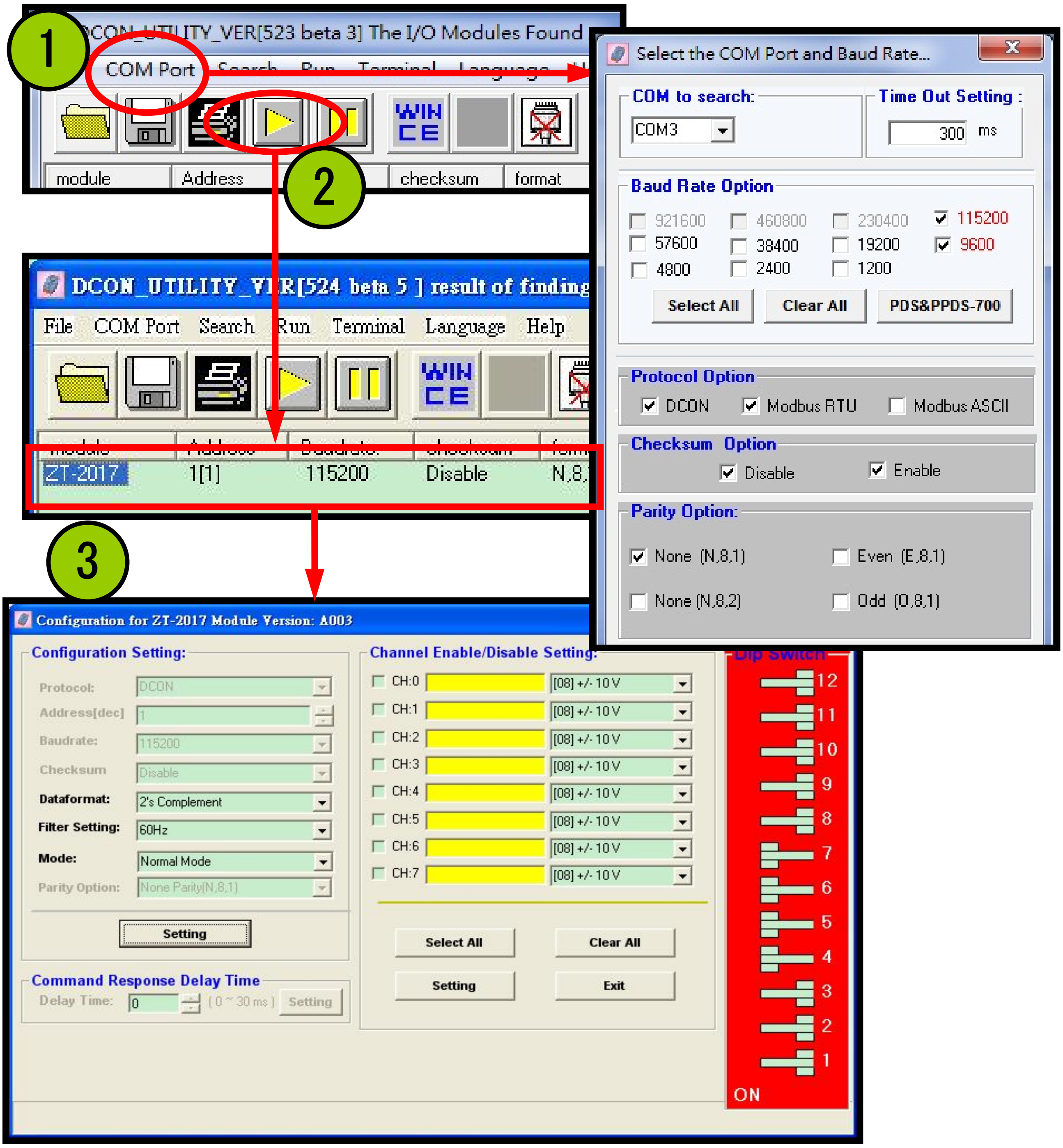
- Configurations of ZT-2000 I/O device



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

➢ Simulate I/O channel operating via using DCON Utility

1. Launch DCON Utility and select the correct COM Port settings to connect the ZigBee Coordinator (ZT-2550/ZT-2570).
2. Clicking "Search" button will start searching which ZT-2000 I/O device is in the same ZigBee network.
3. If there is any ZT-2000 I/O devices displayed, double clicking the "module name" will start the I/O channels operated platform.



CON_UTILITY_VER[523 beta 3] The I/O Modules Found

Select the COM Port and Baud Rate...

COM to search: COM3

Time Out Setting : 300 ms

Baud Rate Option

□ 921600 □ 460800 □ 230400 ☑ 115200
□ 57600 □ 38400 □ 19200 ☑ 9600
□ 4800 □ 2400 □ 1200

Protocol Option: ☑ DCON ☑ Modbus RTU □ Modbus ASCII

Checksum Option: ☑ Disable ☑ Enable

Parity Option:
☑ None (N,8,1) □ Even (E,8,1)
□ None (N,8,2) □ Odd (O,8,1)

DCON_UTILITY_VER[524 beta 5] result of finding

module	Address	Baudrate	checksum	format
ZT-2017	1[1]	115200	Disable	N,8,

Configuration for ZT-2017 Module Version: A003

Configuration Setting:

Protocol: DCON
Address[dec]: 1
Baudrate: 115200
Checksum: Disable
Dataformat: 2's Complement
Filter Setting: 60Hz
Mode: Normal Mode
Parity Option: None Parity(N,8,1)

Command Response Delay Time
Delay Time: 0 (0 ~ 30 ms) Setting

Channel Enable/Disable Setting:

□ CH:0 [08] +/- 10 V
□ CH:1 [08] +/- 10 V
□ CH:2 [08] +/- 10 V
□ CH:3 [08] +/- 10 V
□ CH:4 [08] +/- 10 V
□ CH:5 [08] +/- 10 V
□ CH:6 [08] +/- 10 V
□ CH:7 [08] +/- 10 V

4 Analog Input Type and Data Format Table

Type Code	Input Type	Data Format	+F.S.	-F.S.
07	+4 to +20 mA	Engineering units	+20.000	+04.000
		% of FSR	+100.00	+000.00
		2’s comp HEX	FFFF	0000
08*1	-10 to +10 V	Engineering units	+10.000	-10.000
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
09*1	-5 to +5 V	Engineering units	+5.0000	-5.0000
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
0A*1	-1 to +1 V	Engineering units	+1.0000	-1.0000
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
0B*1	-500 to +500 mV	Engineering units	+500.00	-500.00
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
0C*1	-150 to +150 mV	Engineering units	+150.000	-150.00
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
0D	-20 to +20 mA	Engineering units	+20.000	-20.000
		% of FSR	+100.00	-100.00
		2’s comp HEX	7FFF	8000
1A	0 to +20 mA	Engineering units	+20.000	+00.000
		% of FSR	+100.00	+000.00
		2’s comp HEX	FFFF	0000

*1: only available with the ZT-2017
*2: FSR (FULL Scale Range)

- Analog Inputy Over/Under Range Readings

	Over Range	Under Range
Engineering Unit	+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 Sections 1.8 and 2.10 for details.
3. Enable calibration. Refer to Section 2.20 for details.
4. Apply the zero calibration voltage/current.
5. Send the zero calibration command. Refer to Section 2.5 for details.
6. Apply the span calibration voltage/current.
7. Send the span calibration command. Refer to Section 2.4 for details.
8. Repeat steps 3 to 7 three times.

➢ Notes

1. Connect the calibration voltage/current to channel 0.
2. Calibration voltages and currents are shown as below.
3. Switch to DCON protocol mode before calibrating. Refer to Section 1.5 for details of how to switch the protocol.

➢ Calibration voltage type used by the ZT-2017 and ZT-2017C:

Type Code	08*1	09*1	0A*1	0B*1	0C*1	0D
Zero Input	0V	0V	0V	0mV	0mV	0mA
Span Input	+10V	+5V	+1V	+500mV	+150mV	+20mA
*1: only available with the ZT-2017						

6 DCON/Modbus RTU Command set

6.1 How to communicate with ZT-2000 I/O Device

ICP DAS ZT-2000 I/O devices provides DCON and Modbus RTU protocols. Through by the wireless transmission, user may easily control and monitor I/O channels. The following documents shows the details of DCON and Modbus RTU protocols command set as below link.

http://ftp.icpdas.com/pub/cd/8000cd/napdos/7000/manual/modbusdio.pdf

6.2 DCON Protocol Command set

All the ZT-2000 I/O series devices are controlled via wireless broadcasting commands, so there must be a unique adjustable address saved in the EEPROM to show the difference.

In other words, all the command formats contain the destination address. When I/O devices receive commands, it will decide whether to respond or not in according own address. But, there are still two exception commands #** and ~**.

➢ DCON Command Format

Leading Character	Module Address	Command	[CheckSum]	CR

➢ DCON Response Command Format

Leading Character	Module Address	Data	[CheckSum]	CR

※ Note: ‘CR’ is a characters used to end a frame.

6.2.1 Checksum

- Calulate Checksum：

  Sum all the ASCII code of characters to the command in addition to the ‘CR’ terminator. The Checksum value is the sum expressed to the Hexadecimal. The Checksum is the sum value expressed to 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 Command ” !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: Checksum is the sum value in capital letters expressed.

6.2.2 Quick Start

General Command Sets			
Command	Response	Description	Section
%AANNTTCCFF	!AA	Sets the module configuration	2.1
#AA	>(Data)	Reads data from the analog inputs	2.2
#AAN	>(Data)	Reads data from the analog input of a channel	2.3
$AA0	!AA	Performs a zero calibration	2.4
$AA1	!AA	Performs a span calibration	2.5
$AA2	!AANNTTCCFF	Reads the module configuration	2.6
$AA5	!AAS	Reads the module reset status	2.7
$AA5VV	!AA	Enables/Disables the channel	2.8
$AA6	!AAVV	Reads the enabled/disabled status of the channel	2.9
$AA7CiRrr	!AA	Sets the range configuration of a channel	2.10
$AA8Ci	!AACiRrr	Reads the range configuration of a channel	2.11
$AAF	!AA(Data)	Reads the firmware version	2.12
$AAM	!AA(Data)	Reads the module name	2.13
$AAS1	!AA	Reloads the default calibration parameters	2.14
~AAEV	!AA	Enables/Disables calibration	2.20
~AAO(Name)	!AA	Sets the module name	2.21
@AACH	!AA	Clears the high latches	2.22
@AACHi	!AA	Clears the high latch of a specific channel	2.23
@AACHCi	!AA	Clears the high latched alarm of a specific channel	2.24
@AACL	!AA	Clears the low latches	2.25
@AACLi	!AA	Clears the low latch of a specific channel	2.26
@AACLCi	!AA	Clears the low latched alarm of a specific channel	2.27
@AADHCi	!AA	Disables the high alarm of a specific channel	2.28
@AADI	!AAHHLL	Reads the alarm status	2.29

@AADLCi	!AA	Disables the low alarm of a specific channel	2.30
@AAHI(data)CiT	!AA	Sets the the high alarm of a specific channel	2.31
@AALO(data)CiT	!AA	Sets the low alarm of a specific channel	2.32
@AARH	!AA(data)	Reads the high latches	2.33
@AARHCi	!AA(data)S	Reads the high alarm of a specific channel	2.34
@AARHi	!AA(data)	Reads the high latch of a specific channel	2.35
@AARL	!AA(data)	Reads the low latches	2.36
@AARLi	!AA(data)	Reads the low latch of a specific channel	2.37
@AARLCi	!AA(data)S	Reads the low alarm of a specific channel	2.38

Host Watchdog Command Sets

Command	Response	Description	Section
~**	No Response	Host is OK	2.15
~AA0	!AASS	Reads the Host Watchdog status	2.16
~AA1	!AA	Resets the Host Watchdog status	2.17
~AA2	!AAETT	Reads the Host Watchdog timeout settings	2.18
~AA3ETT	!AA	Sets the Host Watchdog timeout settings	2.19

6.2.3 %AANNTTCCFF

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

Syntax	
%AANNTTCCFF[CHKSUM](CR)	
%	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
NN	The address of the module to be configured in hexadecimal format (00 to FF)
TT	00 (Reserved)
CC	0A (Reserved)
FF	Used to set the data format, checksum, and filter settings (See Section 1.8 for details)

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

Examples	
Command	%0320000A80
Response	!03
In the normal mode, saves the address 0x20 into the EEPROM is 0x20 and sets the data format of module 03 to 80 (50Hz rejection). The module returns a valid response.	

Command	%0320000A80
Response	!20
In the software configuration mode, saves the address 0x20 into the EEPROM is 0x20 and sets the data format of module 03 to 80 (50Hz rejection). The module returns a valid response.	

※Related Commands: $AA2

6.2.4 #AA

Description
This command is used to read the data from all analog input channels

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

Response	
Valid Command	>(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	The data from all analog input channels, see Section 1.8 for details of the data format. Data from disabled channels is filled with space characters.
AA	The address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	#01
Response	>+10.000+10.000+10.000+10.000+10.000+10.000+10.000+10.000
Reads module 01 and receives the data in engineering format.	

※Related Commands: %AANNTTCCFF、$AA2、$AA7CiRrr

※Related Topics: Section 1.8 Configuration Tables.

6.2.5 #AAN

Description
This command is used to read the analog input of a specific channel

Syntax	
#AAN[CHKSUM](CR)	
#	Delimiter character
AA	The address of the module to be read (00 to FF)
N	The channel to be read, zero based

Response	
Valid Command	>(Data)[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
>	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
(Data)	The analog input data from the specified channel, see Section 1.8 for details of the data format. If the specified channel is disabled, then the data field will be filled with space characters.
AA	The address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	#032
Response	>+025.13
Reads data from channel 2 of module 03.	

Command	#039
Response	?03
Reads data from channel 9 of module 03. An error is returned because channel 9 is invalid.	

※Related Commands: %AANNTTCCFF、$AA2、$AA7CiRrr

※Related Topics: Section 1.8 Configuration Tables.

6.2.6 $AA0

Description
This command is used to perform a span calibration

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

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

Example	
Command	$030
Response	?03
Performs a span calibration of module 03. An invalid command is returned because the "enable calibration" command (~AAEV, see Section 2.20) was not sent in advance.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a valid response.	

Command	$030
Response	!03
Performs a span calibration of module 03 and returns a valid response.	

※Related Commands：$AA1、~AAEV

※Related Topics：Section 1.9 Calibration

※Notes：The "enable calibration" command, ~AAEV, and the "zero calibration" command, $AA1, must be sent before this command is used, see Sections 2.20 and 2.5 for details.

6.2.7 $AA1

Description
This command is used to perform a zero calibration

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

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

Example	
Command	$031
Response	?03
Performs a zero calibration of module 03. An invalid command is returned because the "enable calibration" command (~AAEV, see Section 2.20) was not sent in advance.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a valid response.	

Command	$031
Response	!03
Performs a zero calibration of module 03 and returns a valid response.	

※Related Commands : $AA0、~AAEV

※Related Topics : Section 1.9 Calibration

※Notes :

1. The "enable calibration" command, ~AAEV, must be sent before this command is used, see Section 2.20 for details.
2. This command must be sent before the "span calibration" command, $AA0, is used.

6.2.8 $AA2

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

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

Response	
Valid Command	!NNTTCCFF[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
NN	The new address is saved in the EEPROM (00 to FF)
TT	00 (Reserved)
CC	0A (Reserved)
FF	The data format, checksum settings and filter settings of the module, see Section 1.8 for details.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	$032
Response	!FF000A00
In the normal mode, reads the configuration of module 03. The response shows that the address data in the EEPROM is 0xFF.	

Command	$FF2
Response	!FF000A00
In software configuration mode, reads the configuration of module FF.	

※Related Commands：%AANNTTCCFF

※Related Topics：Section 1.8 Configuration Tables

6.2.9 $AA5

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

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

Response	
Valid Command	!AAS[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (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.	

Example	
Command	$035
Response	!031
Reads the reset status of module 03. The response shows that it is a 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 response shows that there has been no module reset since the last $AA5 command was sent.	

6.2.10 $AA5VV

Description
This command is used to specify the channels to be enabled

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

Response	
Valid Command	!AA[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command. An invalid command is returned if an attempt is made to enable a channel that is not present.
AA	The address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	$0353A
Response	!03
Enables channels 1, 3, 4, and 5 and disables all other channels on module 03. The module returns a valid response.	

Command	$036
Response	!033A
Reads the channel status of module 03. The module returns a response of 3A, which denotes that channels 1, 3, 4, and 5 are enabled and all other channels are disabled.	

※Related Commands : $AA6

6.2.11 $AA6

Description
This command is used to read the enabled/disabled status of each channel

Syntax	
$AA6[CHKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read (00 to FF)
6	The command to read the status of the channel

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

Example	
Command	$0353A
Response	!03
Enables channels 1, 3, 4, and 5 and disables all other channels on module 03. The module returns a valid response.	

Command	$036
Response	!033A
Reads the channel status of module 03. The module returns a response of 3A, which denotes that channels 1, 3, 4, and 5 are enabled and all other channels are disabled.	

※Related Commands：$AA5VV

6.2.12 $AA7CiRrr

Description
This command is used to set the type code of a specific channel

Syntax	
$AA7CiRrr[CHKSUM](CR)	
$	Delimiter character
AA	The address of the module to be set (00 to FF)
7	The command to set the channel range code
Ci	i specifies the input channel to be set (0-7)
Rrr	rr represents the type code of the channel to be set. Refer to the Temperature Sensor Type Settings table in Section 1.8.

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

Example	
Command	$037C0R08
Response	!03
Sets the type code for channel 0 of module 03 to 8 (-10 ~ +10V) and the module returns a valid response.	

Command	$037C5R09
Response	!03
Sets the type code for channel 5 of module 03 to 9 (-5 ~ +5V) and the module returns a valid response.	

Command	$037C1R80
Response	?03
Sets the type code for channel 1 of module 03 to 80. The module returns an invalid response because the type code is invalid.	

※Related Commands：$AA8Ci

※Related Topics：Section 1.8 Configuration Tables

6.2.13 $AA8Ci

Description
This command is used to read the type code information for a specific channel.

Syntax	
$AA8Ci[CHKSUM](CR)	
$	Delimiter character
AA	The address of the module to be read (00 to FF)
8	The command to read the type code of a channel
Ci	Specifies which channel to access for the type code information (0-7)

Response	
Valid Command	!AACiRrr[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command or invalid channel
AA	The address of the responding module (00 to FF)
Ci	Specifies which input channel the type code information relates to.
Rrr	Represents the type code of the specified input channel. Refer to the Temperature Sensor Type Settings table in Section 1.8.
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	$038C0
Response	!03C0R08
Reads the input range of channel 0 of module 03 and returns 8 (-10 ~ +10V).	

※Related Commands : $AA7CiRrr

※Related Topics : Section 1.8 Configuration Tables

6.2.14 $AAF

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

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

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (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.	

Example	
Command	$03F
Response	!01A1.0
Reads the firmware version of module 01, and shows that it is version A1.0.	

※Related Commands：$AALS

6.2.15 $AAM

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

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

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (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.	

Example	
Command	$03M
Response	!03ZT-2017
Reads the name of module 03 and returns the name "ZT-2017".	

※Related Commands：~AAO (Name)

6.2.16 $AAS1

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

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

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

Example	
Command	$03S1
Response	!03
Sends a command to reload the factory default calibration parameters for module 03 and returns a valid response.	

※Related Topics：Section 1.9 Calibration

6.2.17 ~**

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

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

Response	
None response	

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

※Related Topics：Section 5.1 Dual Watchdog Operation.

6.2.18 ~AA0

Description
This command is used to read the status of a module’ s Host Watchdog.

Syntax	
~AA0CHKSUM](CR)	
~	Delimiter character
AA	The address of the module to be read (00 to FF)
0	The command to read the status of the Host Watchdog

Response	
Valid Command	!AASS[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (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 EEPROM and can only be reset by using the ~AA1 command.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	~030
Response	!0300
Reads the status of the Host Watchdog of module 03 and returns 00, meaning that the Host Watchdog is disabled and no Host Watchdog timeout has occurred.	

Command	~030
Response	!0304
Reads the status of the Host Watchdog of module 03 and returns 04, meaning that a Host Watchdog timeout has occurred.	

※Related Commands：~**, ~AA1, ~AA2, ~AA3ETT

※Related Topics：Section 5.1 Dual Watchdog Operation

6.2.19 ~AA1

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

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

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

Example	
Command	~030
Response	!0304
Reads the status of the Host Watchdog of module 03 and shows that a Host Watchdog timeout has occurred.	

Command	~031
Response	!03
Resets the timeout status of the Host Watchdog of module 03 and returns a valid response.	

Command	~03.
Response	!0300
Reads the status of the Host Watchdog of module 03 and shows that no Host Watchdog timeout has occurred.	

※Related Commands：~**, ~AA0, ~AA2, ~AA3ETT

※Related Topics：Section 5.1 Dual Watchdog Operation

6.2.20 ~AA2

Description
This command is used to read the timeout value of a module’s Host Watchdog.

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

Response	
Valid Command	!AAEVV[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
E	0: The Host Watchdog is disabled 1: The Host Watchdog is enabled
VV	Two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Example	
Command	~032
Response	!031FF
Reads the Host Watchdog timeout value of module 03 and returns FF, which denotes that the Host Watchdog is enabled and the Host Watchdog timeout value is 25.5 seconds.	

※Related Commands : ~**, ~AA0, ~AA1, ~AA3ETT

※Related Topics : Section 5.1 Dual Watchdog Operation

6.2.21 ~AA3ETT

Description
This command is used to enable/disable a module' s Host Watchdog and sets the Host Watchdog timeout value of a module.

Syntax	
~AA3ETT[CHKSUM](CR)	
~	Delimiter character
AA	The address of the module to be set (00 to FF)
3	The command to set the Host Watchdog
E	0: Disables the Host Watchdog 1: Enables the Host Watchdog
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.

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

Example	
Command	~033164
Response	!01
Enables the Host Watchdog of module 03 and sets the Host Watchdog timeout value to 10.0 seconds. The module returns a valid response.	

Command	~032
Response	!01164
Reads the Host Watchdog timeout value of module 03. The module returns 164, which denotes that the Host Watchdog is enabled and the Host Watchdog timeout value is 10.0 seconds.	

※Related Commands : ~**, ~AA0, ~AA1, ~AA2

※Related Topics : Section 5.1 Dual Watchdog Operation

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

6.2.22 ~AAEV

Description
This command is used to enable/disable calibration of the module.

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

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

Example	
Command	$030
Response	?03
Sends a command to perform a span calibration on module 03. An invalid response is returned because the “enable calibration” command has not yet been sent.	

Command	~03E1
Response	!03
Enables calibration on module 03 and returns a valid response.	

Command	$030
Response	!03
Sends a command to perform a span calibration on module 03 and returns a valid response.	

※Related Commands：$AA0, $AA1

※Related Topics：1.9 Calibration

6.2.23 ~AAO(Name)

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

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

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

Example	
Command	~03OZT-2017
Response	!03
Sets the name of module 03 to “ZT-2017” and returns a valid response.	

Command	$03M
Response	!03ZT-2017
Reads the name of module 03 and returns the name “ZT-2017”.	

※Related Commands : $AAM

6.2.24 @AACH

Description
This command is used to clear the high latch value of all channels.

Syntax	
@AACH[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
CH	The command to clear the high latches

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

Example	
Command	@03RH0
Response	!03+05.000
Reads the high latch value of channel 0 and returns +05.000.	

Command	@03CH
Response	!03
Clears the high latch value of module 03 and returns a valid response.	

Command	@03RH0
Response	!03+00.000
Reads the high latch value of channel 0 and returns +00.000.	

※Related Commands：@AACHi, @AARH, @AARHi

6.2.25 @AACHi

Description
This command is used to clear the high latch value of a specific channel.

Syntax	
@AACHi[CHKSUM](CR)	
~	Delimiter character
AA	The address of the module to be set (00 to FF)
CH	The command to clear the high latch value
i	The channel to be cleared, zero based

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

Example	
Command	@03RH1
Response	!03+06.000
Reads the high latch value of channel 1 and returns +06.000.	

Command	@03CH
Response	!03
Clears the high latch value of module 03 and returns a valid response.	

Command	@03RH1
Response	!03+00.000
Reads the high latch value of channel 1 and returns +00.000.	

※Related Commands：@AACH, @AARH, @AARHi

6.2.26 @AACHCi

Description
This command is used to clear the high alarm status of a specific channel.

Syntax	
@AACHCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
CHC	The command to clear the high alarm status
i	The channel to be cleared, zero based

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

Example	
Command	@03DI
Response	!038000
Reads the alarm status and returns a response indicating that a high alarm of channel 7 has occurred.	

Command	@03CHC7
Response	!03
Clears the high alarm status of channel 7 and returns a valid response.	

Command	@03DI
Response	!030000
Reads the alarm status and returns a response indicating that neither a high alarm nor a low alarm has occurred.	

※Related Commands：@AACH, @AARH, @AARHi

6.2.27 @AACL

Description
This command is used to clear the low latch values of all channels.

Syntax	
@AACL[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
CL	The command to clear the low latch values

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

Example	
Command	@03RL0
Response	!03-05.000
Reads the low latch value of channel 0 and returns -05.000.	

Command	@03CL
Response	!03
Clears the low latch value of module 03 and returns a valid response.	

Command	@03RL0
Response	!03+00.000
Reads the low latch value of channel 0 and returns +00.000.	

※Related Commands：@AACLi, @AARL, @AARLi

6.2.28 @AACLi

Description
This command is used to clear the low latch value of a specific channel.

Syntax	
@AACLi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FFF)
CL	The command to clear the low latch value
i	The channel to be cleared, zero based

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

Example	
Command	@03RL1
Response	!03-06.000
Reads the low latch value of channel 1 and returns -06.000.	

Command	@03CL1
Response	!03
Clears the low latch value of channel 1 and returns a valid response.	

Command	@03RL1
Response	!03+00.000
Reads the low latch value of channel 1 and returns +00.000.	

※Related Commands：@AACL, @AARL, @AARLi

6.2.29 @AACLCi

Description
This command is used to clear the low alarm status of a specific channel.

Syntax	
@AACLCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
CLC	The command to clear the low alarm status
i	The channel to be cleared, zero based

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

Example	
Command	@03DI
Response	!030080
Reads the alarm status and returns a response indicating that a low alarm of channel 7 has occurred.	

Command	@03CHC7
Response	!03
Clears the low alarm status of channel 7 and returns a valid response.	

Command	@03DI
Response	!030000
Reads the alarm status and returnsa response indicating that neither a high alarms nor a low alarms has occurred.	

※Related Commands : @AACL, @AARL, @AARLi

6.2.30 @AADHCi

Description
This command is used to disable the high alarm of a specific channel.

Syntax	
@AADHCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
DH	The command to disable the high alarm
Ci	The channel where the alarm is to be disabled, zero based

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

Example	
Command	@03DHC0
Response	!01
Disables the high alarm of channel 0.	

Command	@03DI
Response	Response: !01FEFF
Reads the alarm status and returns a response indicating that the high alarm of channel 0 is disabled and others are enabled	

※Related Commands : @AADI

6.2.31 @AADI

Description
This command is used to read the alarm status.

Syntax	
@AADI[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
DI	The command to read the alarm status

Response	
Valid Command	!AAHHLL[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
HH	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to 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 channel 0, bit 1 corresponds to 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.	

Example	
Command	@03DI
Response	!034008
Reads the alarm status of module 03 and returns a response indicating that a high alarm of channel 6 and a low alarm of channel 3 have ocurred.	

※Related Commands：@AADHCi, @AADLCi, @AAHI(Data)CiT, @AALO(Data)CiT

6.2.32 @AADLCi

Description
This command is used to disable the low alarm of a specific channel.

Syntax	
@AADLCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
DL	The command to disable the low alarm
Ci	The channel where the alarm is to be disabled, zero based

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

Example	
Command	@03DLC5
Response	!03
Disables the low alarm of channel 5 and returns a valid response.	

Command	@03DI
Response	!03FFDF
Reads the alarm status and returns a response indicating that the low alarm of channel 5 is disabled and others are enabled.	

※Related Commands：@AADI

6.2.33 @AAHI(Data)CiT

Description
This command is used to set the high alarm of a specific channel.

Syntax	
@AAHI(Data)CiT[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
HI	The command to set the high alarm
(Data)	The high alarm limit， which should be consistent with the data format. Refer to Section 1.8 for details.
Ci	The channel to be set， zero based
T	The alarm type： M： Momentary alarm L： Latched alarm

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

Example	
Command	@03HI+09.000C0M
Response	!03
Sets the high alarm of channel 0. The high alarm limit is +09.000 and the type is momentary and returns a valid response.	

Command	@03DI
Response	!030100
Reads the alarm status and returns a response indicating that the high alarm of channel 0 is enabled and others are disabled.	

※Related Commands：@AADI

6.2.34 @AALO(Data)CiT

Description
This command is used to set the low alarm of a specific channel.

Syntax	
@AALO(Data)CiT[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be set (00 to FF)
LO	The command to set the low alarm
(Data)	The high alarm limit, which should be consistent with the data format. Refer to Section 1.8 for details.
Ci	The channel to be set, zero based
T	The alarm type: M: Momentary alarm L: Latched alarm

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

Example	
Command	@03LO-03.000C1L
Response	!03
Sets the high alarm of channel 1. The low alarm limit is -03.000 and the type is latched and returns a valid response.	

Command	@03DI
Response	!030002
Reads the alarm status and returns a response indicating that the high alarm of channel 1 is enabled and others are disabled.	

※Related Commands : @AADI

6.2.35 @AARH

Description
This command is used to read the high latch values of all channels.

Syntax	
@AARH[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
RH	The command to read the high latch values

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The high latch values of all channels, see Section 1.8 for defaults 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	@03RH
Response	!03+08.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the high latch values of module 03 and returns the data in engineering format.	

※Related Commands : @AACH, @AACHi, @AARHi

6.2.36 @AARHCi

Description
This command is used to read the high alarm status of a specific channel.

Syntax	
@AARHCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
RH	The command to read the high alarm status
Ci	The channel to be read, zero based

Response	
Valid Command	!AA(Data)S[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The high latch values of all channels, see Section 1.8 for defaults 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	@03RHC0
Response	!03+08.0002
Reads the high alarm status of channel 0 and returns a response indicating that the high alarm limit is +08.000 and the type is latched.	

※Related Commands: @AAHI(Data)CiT, @AADHCi, @AADI

6.2.37 @AARHi

Description
This command is used to read the high latch value of a specific channel.

Syntax	
@AARHCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
RH	The command to read the high latch value
i	The channel to be read, zero based

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The high latch value of a specific channel, see Section 1.8 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	@03RH0
Response	!03+08.000
Reads the high latch value of channel 0 and returns the data in engineering format.	

※Related Commands : @AACH, @AACHi, @AARH

6.2.38 @AARL

Description
This command is used to read the low latch values for all channels.

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

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The low latch values of all channels, see Section 1.8 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	@03RL
Response	!03-02.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000
Reads the low latch values of module 01 and returns the data in engineering format.	

※Related Commands : @AACL, @AACLi, @AARLi

6.2.39 @AARLCi

Description
This command is used to read the low alarm status of a specific channel.

Syntax	
@AARLCi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
RL	The command to read the low alarm status
Ci	The channel to be read, zero based

Response	
Valid Command	!AA(Data)S[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The low alarm status of a specific channel, see Section 1.8 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	@03RLC0
Response	!03-03.0001
Reads the low alarm status of channel 0 and returns a response indicating that the high alarm limit is -03.000 and the type is momentary.	

※Related Commands：@AALO(Data)CiT, @AADI, @AADLCi

6.2.40 @AARLi

Description
This command is used to read the low latch value of a specific channel.

Syntax	
@AARLi[CHKSUM](CR)	
@	Delimiter character
AA	The address of the module to be read (00 to FF)
RL	The command to read the low latch value
i	The channel to be read, zero based

Response	
Valid Command	!AA(Data)[CHKSUM](CR)
Invalid Command	?AA[CHKSUM](CR)
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	The high latch value of a specific channel, see Section 1.8 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 of channel 0 and returns the data in engineering format.	

※Related Commands：@AACL, @AACLi, @AARL

6.3 Modbus RTU Protocol Command set

The Modbus protocol is developed by Modicon Inc., originally developed for Modicon controllers. Detailed information can be found more valuable information at

http://www.modicon.com

http://www.modbus.org

➢ Modbus RTU Command Format

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

Function Code	Description
0x04	Read input channels
0x46	Read/write module settings

Example：

A. Read the analog input value of the module 01, the following command should be sent

   01 04 00 00 00 08 F1 CC

B. Read the module name, the following command should be sent:

   01 46 00 12 60

6.3.1 Modbus Address Mapping

Address Mapping		
Address	Description	Attribute
00259	Filter settings, 0: 60Hz rejection, 1: 50Hz rejection	R/W
00260	Modbus Host Watchdog mode 0: The same as I-7000 series modules 1: The AO and DO commands can be used to clear the Host Watchdog timeout status	R/W
00261	Enable/disable the Host Watchdog 0: disable 1: enable	R/W
00269	Modbus data format 0: hex 1: engineering	R/W
00270	The Host Watchdog timeout status, write 1 to clear the Host Watchdog timeout status	W
00272	The factory calibration parameters, write to load	W
00273	The reset status 0: not the first time the status has been read after being powered on 1: the first time the status has been read after being powered on	R
00280	The high latch of channels 0 to 7, write 1 to clear	W
00281	The low latch of channels 0 to 7, write 1 to clear	W
00513 ~ 00520	The high latch of channels 0 to 7, write 1 to clear	W
00545 ~ 00552	The low latch of channels 0 to 7, write 1 to clear	W
00577 ~ 00584	Enable/disable the high alarm of channels 0 to 7 0: disable 1: enable	R/W
00609 ~ 00616	Enable/disable the low alarm of channels 0 to 7 0: disable 1: enable	R/W
00641 ~ 00648	The high alarm mode of channels 0 to 7 0: momentary 1: latch	R/W
00673 ~ 00680	The low alarm mode of channels 0 to 7 0: momentary 1: latch	R/W
00705 ~	The high alarm status of channels 0 to 7	R/W

00712		
00737 ~ 00744	The low alarm status of channels 0 to 7	R/W
10129 ~ 10136	The under range status of channels 0 to 7 (supports types 0x7 and 0x1A only)	R
30001 ~ 30008	The analog input value of channels 0 to 7	R
30513 ~ 30520	The high latch value	R
30545 ~ 30552	The low latch value	R
40257 ~ 40264	The type code of 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: 0x00 ~ 0xF7	R
40486	Bits 5:0 Baud Rate, 0x0A Bits 7:6 Reserved	R
40489	The Host Watchdog timeout value, 0 ~ 255, in 0.1s	R/W
40490	Enable/disable the channel	R/W
40492	The Host Watchdog timeout count, write 0 to clear	R/W
40577 ~ 40584	The high alarm value	R/W
40609 ~ 40616	The low alarm value	R/W

6.3.2 PLC Address Mapping

Function code	Description	Section
0x01	Read coils	3.1
0x02	Read discrete inputs	3.2
0x03	Read multiple registers	3.3
0x04	Read multiple input registers	3.4
0x05	Write single coils	3.5
0x06	Write multiple registers	3.6
0x0F	Write multiple coils	3.7
0x46	Read/Write module settings	3.8

If the function specified in the message is not supported, then the module responds as follows. Address mapping of Protocol(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.

6.3.3 01(0x01)Read Coils

Description
This function code is used to read the current digital output readback value of the ZT-2000 I/O device

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x01
02-03	Starting channel numbers or address mapping	2	
04-05	Output channel number or bit count	2	

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x01
02	Byte Count	1	Byte count of response (B=(bit count + 7)/8)
03	Bit values	B	(Bit values)
Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x81
02	Exception code	1	Refer to Modbus standard for more details

6.3.4 02(0x02)Read Discrete Inputs

Description
This function code is used to read the current digital input value of the ZT-2000 I/O module.

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x02
02-03	Starting channel numbers or address mapping	2	
04-05	Output channel number or bit count	2	

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x02
02	Byte Count	1	Byte count of response (B=(bit count + 7)/8)
03	Bit values	B	(Bit values)

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x82
02	Exception code	1	Refer to Modbus standard for more details

6.3.5 03(0x03)Read Multiple Registers

Description
This function code is used to read the current digital input counter value of the ZT-2000 I/O device

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x03
02-03	Starting channel numbers or address mapping	2	
04-05	Output channel number or bit count	2	

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x03
02	Byte Count	1	Byte count of response (B=2 * word count)
03~	Bit values	B*2	Register values

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x83
02	Exception code	1	Refer to Modbus standard for more details

6.3.6 04(0x04)Read Multiple Input Registers

Description
This function code is used to read the current digital input counter value of the ZT-2000 I/O device

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x04
02-03	Starting channel numbers or address mapping	2	
04-05	Output channel number or bit count	2	

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x04
02	Byte Count	1	Byte count of response (B=2 * word count)
03~	Bit values	B*2	Register values

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x84
02	Exception code	1	Refer to Modbus standard for more details

6.3.7 05(0x05)Write Single Coils

Description
This function code is used to write the digital output value of the ZT-2000 I/O device.

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x05
02-03	Starting channel numbers	2	
04-05	Output value	2	A value 0xFF00 sets the output to ON; A value 0x0000 sets the output to OFF

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x05
02~03	Address	2	This value is the same as byte 02 and 03 of the Request
04~05	Output channel numbers	2	This value is the same as byte 04 and 05 of the Request

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x85
02	Exception code	1	Refer to Modbus standard for more details

6.3.8 06(0x06)Write Multiple Registers

Description
This function code is used to set the settings of the module.

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x06
02-03	Address mapping	2	
04-05	Register value	2	Sets watchdog timeout value

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x06
02~03	Address mapping	2	The value is the same as byte 02 and 03 of the Request
04~05	Register value	2	Register value

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x86
02	Exception code	1	Refer to Modbus standard for more details

6.3.9 15(0x0F)Write multiple coils

Description
This function code is used to write the digital output value of the ZT-2000 I/O device.

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x0F
02-03	Starting channel numbers	2	
04-05	Output channel number	2	
06	Byte Count	1	B=(bit count + 7)/8
07	Output value	2	A bit corresponds to a channel. When the bit is '1' it denotes that the value of the channel that was set is ON. If the bit is '0' it denotes that the value of the channel that was set is OFF.

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x0F
02~03	Starting channel numbers	2	The value is the same as byte 02 and 03 of the Request
04~05	Input channel number	2	0x0001 ~ 0x0020

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x8F
02	Exception code	1	Refer to Modbus standard for more details

6.3.10 70(0x46)Read/Write module settings

<table>
<tr><th colspan="3">Description</th></tr>
<tr><td colspan="3">This function code is used to read the settings of the module or change the settings of the module. The following sub-function codes are supported</td></tr>
<tr><th>Sub-Function code</th><th>Description</th><th>Section</th></tr>
<tr><td>00 (0x00)</td><td>Reads the module name</td><td>A.1</td></tr>
<tr><td>04 (0x04)</td><td>Set the module address</td><td>A.2</td></tr>
<tr><td>07 (0x07)</td><td>Reads the type code</td><td>A.3</td></tr>
<tr><td>08 (0x08)</td><td>Sets the type code</td><td>A.4</td></tr>
<tr><td>32 (0x20)</td><td>Reads the firmware version</td><td>A.5</td></tr>
<tr><td>37 (0x25)</td><td>Reads the channel enabled/disabled status of a channel</td><td>A.6</td></tr>
<tr><td>38 (0x26)</td><td>Sets the channel to enabled/disabled</td><td>A.7</td></tr>
<tr><td>41 (0x29)</td><td>Reads the miscellaneous settings</td><td>A.8</td></tr>
<tr><td>42 (0x2A)</td><td>Writes the miscellaneous settings</td><td>A.9</td></tr>
</table>

A.1 00(0x00) Read Module Name

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x00

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x00
03~06	Module name	4	0x54 0x20 0x17 0x00(ZT-2017) 0x54 0x20 0x17 0x13(ZT-2017C)

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.2 04(0x04) Set the module address

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x04
03	New address	1	0x01 ~ 0xF7
04~06	Reserved	3	0x00 0x00 0x00

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x04
03	New address	1	0x00
04~06	Reserved	3	0x00 0x00 0x00

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.3 07(0x07) Read type code

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x07
03	Reserved	1	0x00
04	Channel	1	0x00 ~ 0x07

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x07
03	Type code	1	Type code, see Section 1.8 for details.

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.4 08(0x08) Set type code

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x08
03	Reserved	1	0x00
04	Channel	1	0x00 ~ 0x07
05	Type code	1	Type code, see Section 1.8 for details.

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
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	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

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

A.5 32(0x20) Read firmware version

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x20

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x20
03	Major version	1	0x00 ~ 0xFF
04	Minor version	1	0x00 ~ 0xFF
05	Reserved	1	0x00
06	Build version	1	0x00 ~ 0xFF

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.6 37(0x25) Read channel enabled/disabled status

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x25

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x25
03	Enabled/disabled status	1	0x00 ~ 0xFF, the enabled/disabled status of each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is disabled and 1 denotes that the channel is enabled.

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.7 38(0x26) Set channel enable/disable

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x26
03	Enabled/disabled settings	1	0x00 ~ 0xFF, the enabled/disabled settings for each channel, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is disabled and 1 denotes that the channel is enabled.

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
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	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.8 41(0x29) Read miscellaneous settings

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x29

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x29
03	Miscellaneous settings	1	Data format, see Section 1.8 for details.

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

A.9 42(0x2A) Write miscellaneous settings

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

Request			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x2A
03	Miscellaneous settings	1	Data format, see Section 1.8 for details.

Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0x46
02	Sub-Function code	1	0x2A
03	Miscellaneous settings	1	0: OK Others: error

Error Response			
Byte	Description	Length	Value
00	Address	1	0x01 ~ 0xF7
01	Function code	1	0xC6
02	Exception code	1	Refer to Modbus standard for more details

7 Appendix

7.1 Software Configuration Mode

Each ZT-2000 I/O device has a build-in EEPROM to store configuration information such as address, data format, AI type code and other information. When the module is powered on with Address(Node ID) is 0x00, the ZT-2000 I/O device is into the software configuration mode. In software configuration mode, the configurations(address(Node ID), data format and AI type code) are loaded from the EEPROM and change the setting with command %AANNTTCCFF, $AA7CiRrr. When the ZT-2000 I/O device in software configuration mode, the switch setting 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 circuit allows the module to work continuously without disruption.

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

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