ZT-2015 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

1 Introduction 5

1.1 Introduction to ZigBee 5

1.2 Introduction to the ZT-2000 I/O Series 6

2 Hardware Information 7

2.1 I/O Specifications 7

2.2 System Specifications 7

2.3 Pin Assignment 9

2.4 Wire Connections 10

3 Setting up the ZT-2000 I/O Device 11

3.1 Introduction to the Configuration Parameters 11

3.2 Introduction to the Rotary and DIP Switches 12

3.3 Starting the ZT-2000 I/O Device 15

3.4 Communications Testing 15

3.5 Examples 16

4 RTD Type and Data Format Table 18

5 Calibration 21

6 The DCON/Modbus RTU Command Sets 22

6.1 Communicating with the ZT-2000 I/O Device 22

6.2 The DCON Protocol Command Set 22

6.2.1 Checksum 23

6.2.2 Overview of the DCON Command Set 24

6.2.3 %AANNTTCCFF 26

6.2.4 #AA 27

6.2.5 #AAN 28

6.2.6 $AA0Ci 29

6.2.7 $AA1Ci 30

6.2.8 $AA2 31

6.2.9 $AA5 32

6.2.10 $AA5VV 33

6.2.11 $AA6 ... 34

6.2.12 $AA7CiRrr ... 35

6.2.13 $AA8Ci ... 36

6.2.14 $AAF ... 37

6.2.15 $AAM ... 38

6.2.16 $AAB ... 39

6.2.17 $AAS1 ... 40

6.2.18 ~** ... 41

6.2.19 ~AA0 ... 42

6.2.20 ~AA1 ... 43

6.2.21 ~AA2 ... 44

6.2.22 ~AA3ETT ... 45

6.2.23 ~AAEV ... 46

6.2.24 ~AAO(Name) ... 47

6.3 Modbus RTU Protocol Command set ... 48

6.3.1 Modbus Address Mapping ... 49

6.3.2 PLC Address Mapping ... 51

6.3.3 01 (0x01) Reading the Coils ... 52

6.3.4 02 (0x02) Reading the Discrete Inputs ... 53

6.3.5 03 (0x03) Reading Multiple Registers ... 54

6.3.6 04 (0x04) Reading Multiple Input Registers ... 55

6.3.7 05 (0x05) Writing a Single Coil ... 56

6.3.8 06 (0x06) Writing Multiple Registers ... 57

6.3.9 15 (0x0F) Writing Multiple Coils ... 58

6.3.10 70 (0x46) Reading/Writing the Module Settings ... 59

7 Appendix ... 69

7.1 Software Configuration Mode ... 69

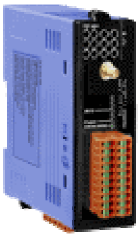
7.2 Dual Watchdog Operation ... 70

7.3 Reset Status ... 71

8 Troubleshooting ... 72

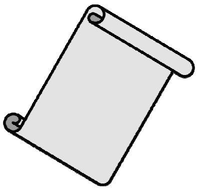
What's in the Shipping Package?

The shipping package contains the following items:



ZT-2000 DIO Module

ANT-124-05



Quick Start

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:
  CD: \Napdos\ZigBee\ZT_Series\Document
  http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zt_series/document
- Software:
  CD: \Napdos\ZigBee\ZT_Series\Utility
  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 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 Series

ZT-2000 I/O series devices are small wireless ZigBee I/O modules based on the IEEE802.15.4 standard that allow data acqusition and control via personal area ZigBee networks. See Section 2.1 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. So, if there is any configuration issue of ZigBee Coordinator, please refer to the "ZT-25XX ZigBee Converter Quick Start" ducument for more information, which can be found at the following link:

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

2 Hardware Information

2.1 I/O Specifications

Analog Input	
Input Channels	6
Input Type	2/3-wire RTD
RTD Type	Pt100, Pt1000, Ni120, Cu100, Cu1000
Resolution	16-bit
Sampling Rate	12 samples/Sec. (Total)
Accuracy	+/-0.05 %
Zero Drift	+/-0.5 μV/°C
Span Drift	+/-20 μV/°C
Common Mode Rejection	150 dB
Normal Mode Rejection	100 dB
Input Impedance	>1M Ohms
Open Wire Detection	Yes
Overvoltage Protection	120 VDC/110 VAC
Individual Channels Configurable	Yes
3-wire RTD lead resistance elimination	Yes

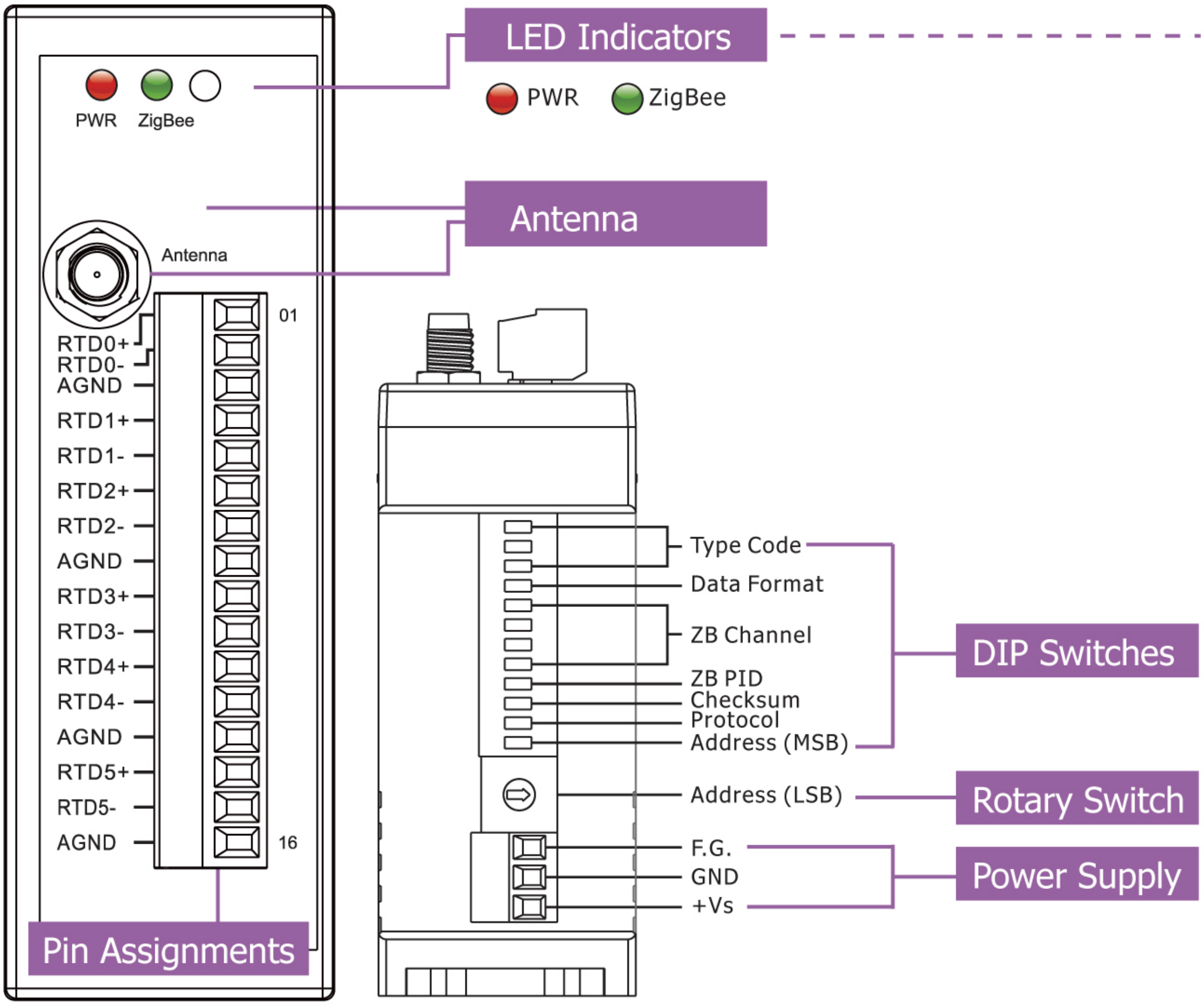
2.2 System Specifications

Communication Interface	
Wireless Standards	ZigBee 2007 Pro
Transmission Power	11 dBm (FCC Certificated) (Max 19 dBm)
2.4 GHz Antenna	5 dBi Omni directional
Transmission Range (LoS)	700 m (Typical)
Certification	CE/FCC, FCC ID
Max. Slaves in a ZigBee Network	255
Protocols	Supports DCON and Modbus RTU Protocols
Hot Swap	Rotary and DIP switch
EMS Protection	
ESD (IEC 61000-4-2)	±4 kV Contact for Power Line, Communication Line and each Channel, ±8 kV Air for Random Point
EFT (IEC 61000-4-4)	±4 kV for Power Line
Surge (IEC 61000-4-5)	±3 kV for Power Line

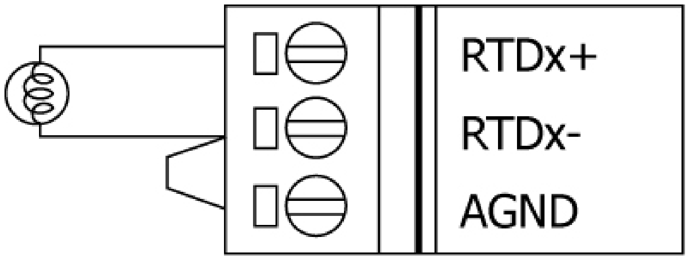
Isolation	
Intra-module Isolated, Field-to-Logic	3000 VDC
LED Indicators	
ZigBee PWR	ZigBee Device Power
ZigBee Net	Zigbee Communication Indicator
Power	
Power Consumption	1.5 W (Max.)
Mechanical	
Flammability	Fire Retardant Materials (UL94-V0 Level)
Dimensions (W x L x H)	33 mm x 87 mm x 107 mm
Installation	DIN-Rail
Environment	
Operating Temperature	-25 to 75 ℃
Storage Temperature	-30 to 80 ℃
Humidity	10 to 90%, Non-condensing

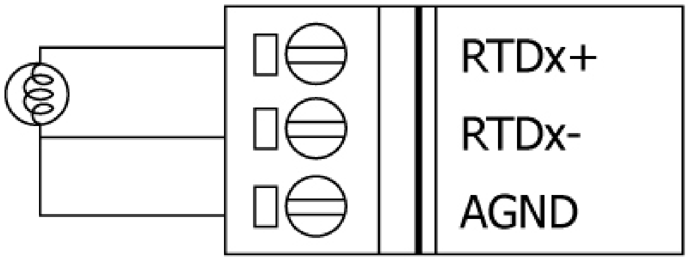
2.3 Pin Assignment

Appearance



2.4 Wire Connections





3 Setting up the ZT-2000 I/O Device

3.1 Introduction to the Configuration Parameters

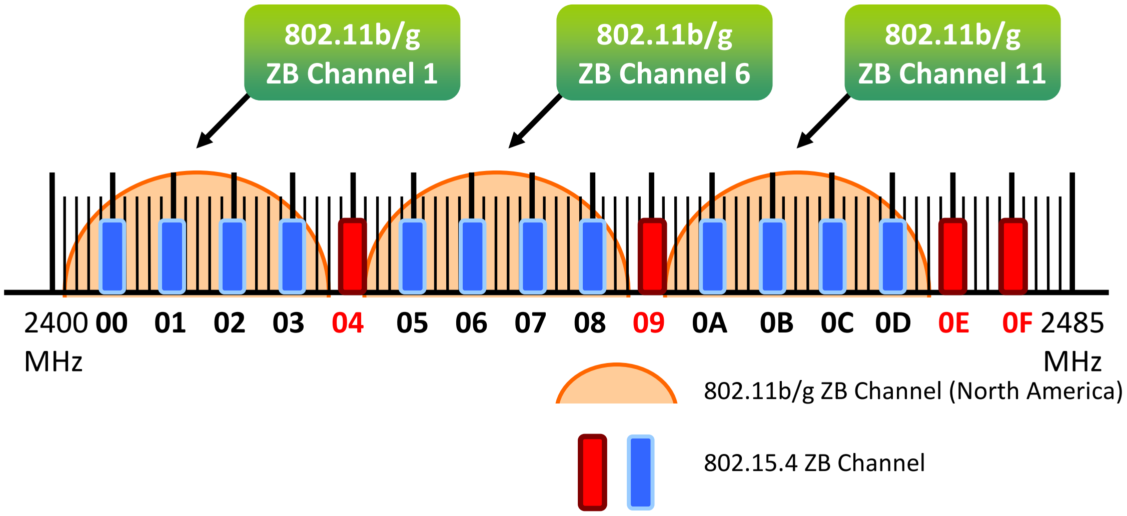
A. The **"ZB PID"** parameter is the group identity for a ZigBee network, and must be the same for all devices in the same ZigBee network.

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.*



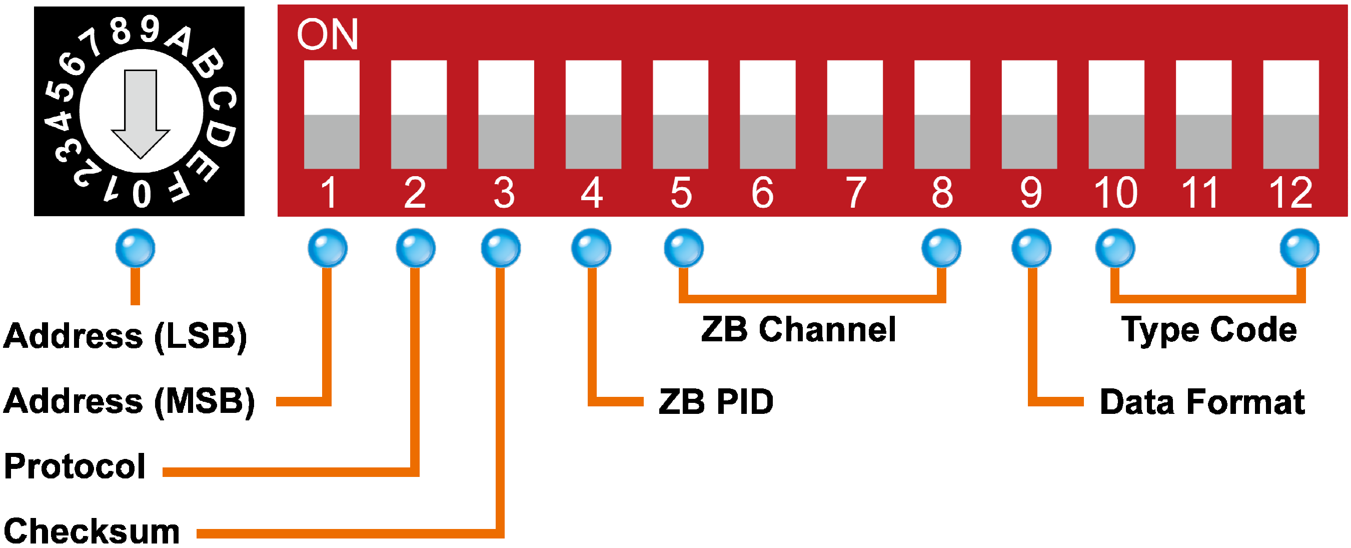
D. Protocol/Application Mode:

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-2017/2017C can be adjusted using a combination of the external rotary switch and the DIP switches. The ZT-2000 device shoule only be rebooted once the configuration is complete.



➢ Rotary Switch

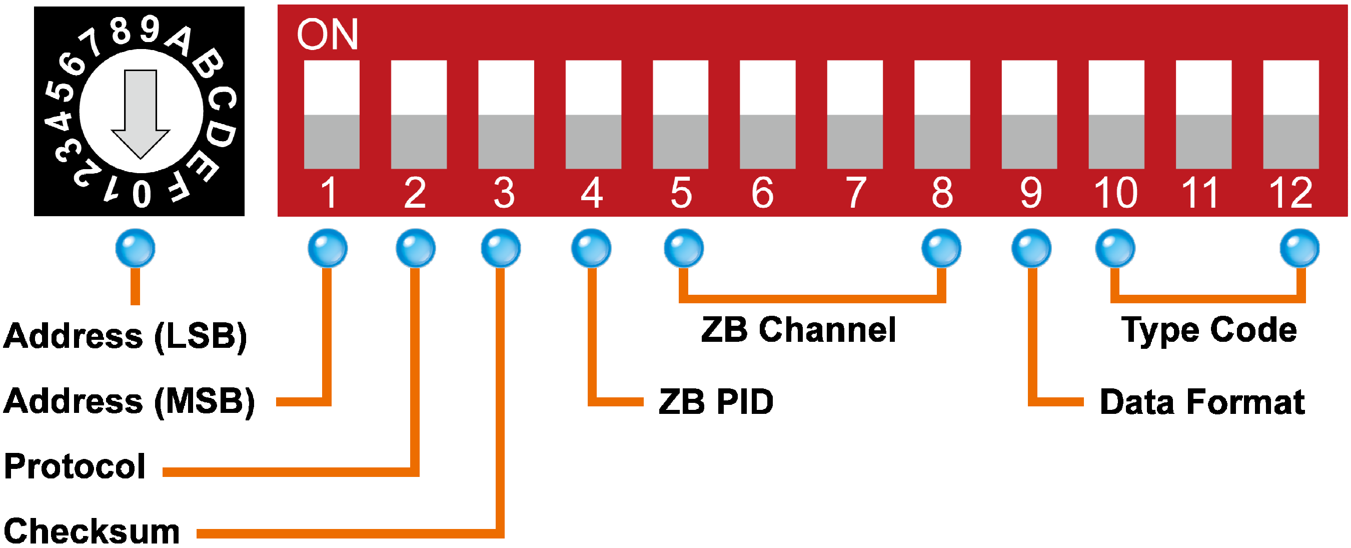
Case1: Address MSB = 0

	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

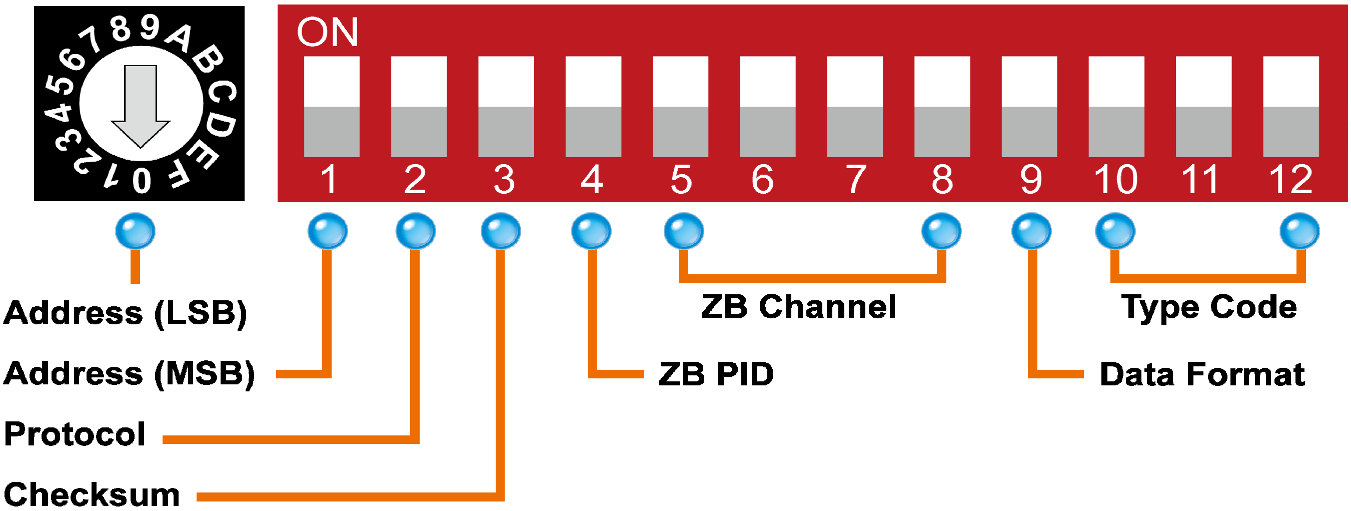
	0	1	2	3	4	5	6	7
Address	10	11	12	13	14	15	16	17
Node ID	0x0010	0x0011	0x0012	0x013	0x0014	0x0015	0x0016	0x0017
	8	9	A	B	C	D	E	F
Address	18	19	1A	0B	0C	1D	1E	1F
Node ID	0x018	0x0019	0x001A	0x001B	0x001C	0x001D	0x001E	0x001F

***Note 1: The “Address” and “Node ID” are defined 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.**



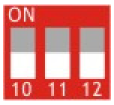
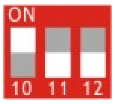
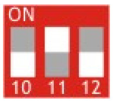
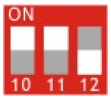
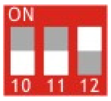
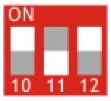
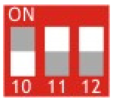
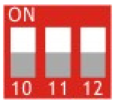
➢ DIP Switches

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-2015, as shown below.

Switch Position	Type Code	Switch Position	Type Code	Switch Position	Type Code
	0x20		0x23		0x24
	0x27		0x28		0x2A
	0x80		0x81		

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 shown below for full details of how to configure these devices.

Once configuration of the ZigBee Coordinator has been completed, set the "Pan ID" and "RF 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**

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 the ZT-2000 I/O device Coordinator)

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.

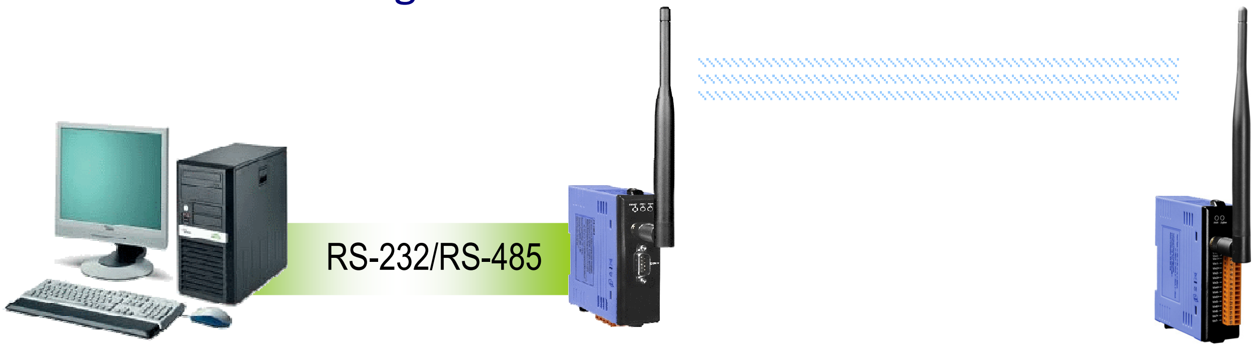
ICP DAS also provides the "DCON Utility", which can alse be used to simulate DCON/Modbus communication. This software can be used to verify the device settings and ZigBee I/O functions.

※ **The Download DCON Utility can be downloaded from:**

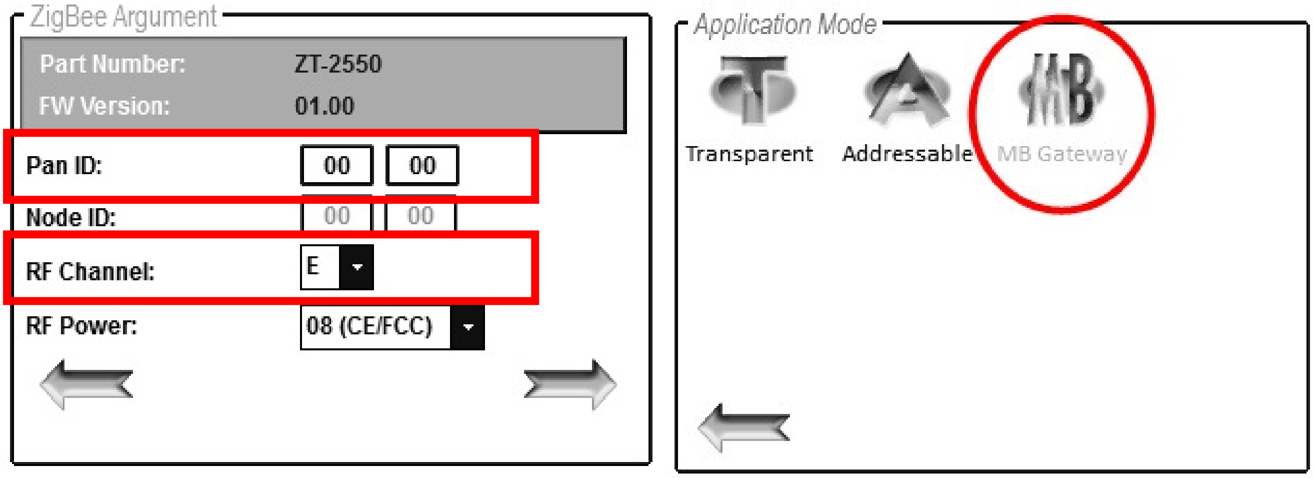
http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/

3.5 Examples

- Architecture Diagram



- Configuring the ZT-2550/ZT-2570


ZigBee Argument

Part Number: ZT-2550

FW Version: 01.00

Pan ID: 00 00

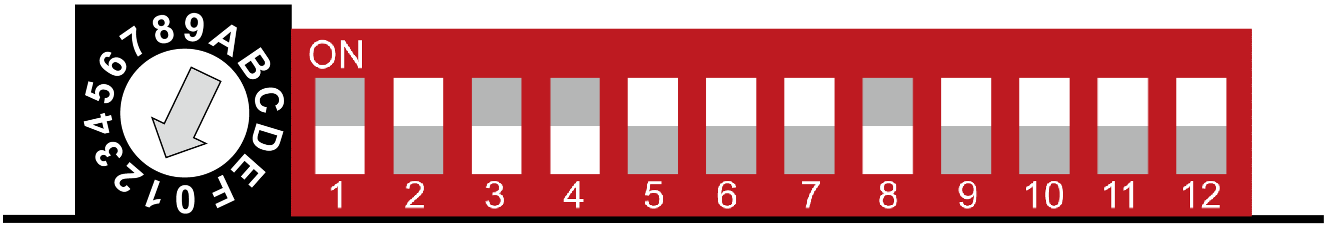
Node ID: 00 00

RF Channel: E

RF Power: 08 (CE/FCC)

Application Mode

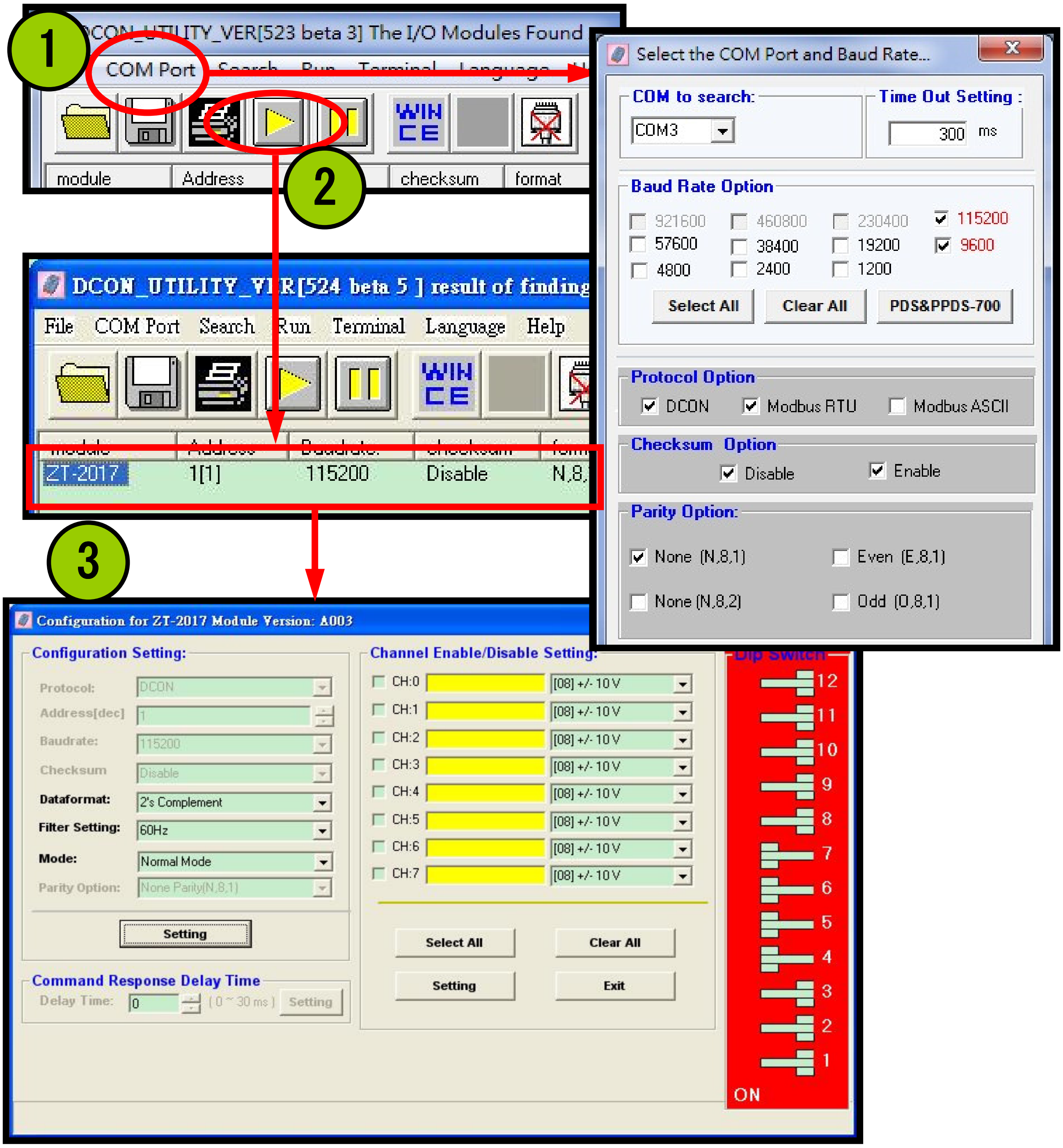
- Configuring the ZT-2000 I/O device



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 operating platform.


1
2
3
DCON_UTILITY_VER[523 beta 3] The I/O Modules Found
COM Port Search Run Terminal Language
module Address checksum format

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

Select All | Clear All | PDS&PPDS-700

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

File COM Port Search Run Terminal Language Help

module	Address	Baudrate	checksum	form
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)

Setting

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

Select All | Clear All
Setting | Exit

Dip Switch: 12 11 10 9 8 7 6 5 4 3 2 1 ON

4 RTD Type and Data Format Table

Type Code	RTD Type	Data Format	+F.S.	-F.S.
20	Platinum 100 α= 0.00385 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2’s Comp. Hex	7FFF	8000
		Ohms	+138.50	+060.60
21	Platinum 100 α= 0.00385 0 ~ 100° C	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+100.00
		2’s Comp. Hex	7FFF	0000
		Ohms	+138.50	+100.00
22	Platinum 100 α= 0.00385 0 ~ 200° C	Engineering Units	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2’s Comp. Hex	7FFF	0000
		Ohms	+175.84	+100.00
23	Platinum 100 α= 0.00385 0 ~ 600° C	Engineering Units	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2’s Comp. Hex	7FFF	0000
		Ohms	+313.59	+100.00
24	Platinum 100 α= 0.003916 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2’s Comp. Hex	7FFF	8000
		Ohms	+139.16	+060.60
25	Platinum 100 α= 0.003916 0 ~ 100° C	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2’s Comp. Hex	7FFF	0000
		Ohms	+139.16	+100.00
26	Platinum 100 α= 0.003916 0 ~ 200° C	Engineering Units	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2’s Comp. Hex	7FFF	0000
		Ohms	+177.14	+100.00

Type Code	RTD Type	Data Format	+F.S.	-F.S.
27	Platinum 100 α= 0.00385 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2’s Comp. Hex	7FFF	8000
		Ohms	+138.50	+060.60
28	Platinum 100 α= 0.00385	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+100.00

	0 ~ 100° C	2's Comp. Hex	7FFF	0000
		Ohms	+138.50	+100.00
29	Platinum 100 α= 0.00385 0 ~ 200° C	Engineering Units	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+175.84	+100.00
2A	Platinum 100 α= 0.00385 0 ~ 600° C	Engineering Units	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+313.59	+100.00
2B	Platinum 100 α= 0.003916 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
		Ohms	+139.16	+060.60
2C	Platinum 100 α= 0.003916 0 ~ 100° C	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+139.16	+100.00
2D	Platinum 100 α= 0.003916 0 ~ 200° C	Engineering Units	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+177.14	+100.00
2C	Platinum 100 α= 0.003916 0 ~ 100° C	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+139.16	+100.00

2F	Platinum 100 α= 0.00385 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
		Ohms	+138.50	+060.60
80	Platinum 100 α= 0.00385 0 ~ 100° C	Engineering Units	+100.00	+000.00
		% of FSR	+100.00	+100.00
		2's Comp. Hex	7FFF	0000
		Ohms	+138.50	+100.00
81	Platinum 100 α= 0.00385 0 ~ 200° C	Engineering Units	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+175.84	+100.00
82	Platinum 100 α= 0.00385 0 ~ 600° C	Engineering Units	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's Comp. Hex	7FFF	0000
		Ohms	+313.59	+100.00

83	Platinum 100 α= 0.003916 -100 ~ 100° C	Engineering Units	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's Comp. Hex	7FFF	8000
		Ohms	+139.16	+060.60

➢ RTD Over/Under Range Reading

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

➢ RTD Over/Under Range Reading 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 (Full Scale Range) 10: 2' s complement hexadecimal 11: Ohms
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.
3. Enable calibration. Refer to Section 2.29 for details.
4. Connect a zero calibration resistor.
5. Send the zero calibration command. Refer to Section 2.6 and 2.7 for details.
6. Connect the span calibration resistor.
7. Send the span calibration command. Refer to Section 2.5 and 2.8 for details.
8. Repeat steps 3 to 7 three times.

➢ Notes

1. Use the 2-wire RTD connection to connect the calibration resistor.
2. Each channel should be calibrated separately and only the channel being calibrated should be enabled during calibration.
3. Calibration resistor types are shown below.

➢ Calibration resistor types used by the ZT-2015

Type	Zero Calibration Resistor	Span Calibration Resistor
2B	0 Ohms	200 Ohms
20	0 Ohms	375 Ohms
2A	0 Ohms	3200 Ohms

➢ Notes

1. Types 21 to 29, 2E, 2F, 80 81 and 83 use the same calibration parameters as type 20.
2. Types 2C and 82 use the same calibration parameters as type 2B.
3. Type 2D uses the same calibration parameters as type 2A

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 operate using both the DCON and the Modbus RTU protocol. Adjust the DIP switch number 2 to select the DCON or Modbus RTU protocol and reboot the ZT-2000 I/O device to correct 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 to denote the difference.

Consequently, all command and response formats contain the destination address of the 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, #** and ~** commands.

- DCON Command Format

Delimit Character	Module Address	Command	[CHECKSUM]	CR

- DCON Response Format

Delimit Character	Module Address	Data	[CHECKSUM]	CR

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

6.2.1 Checksum

Calulating 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.

6.2.2 Overview of the DCON Command Set

General Command Set			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the Module Configuration	6.2.3
#AA	>(Data)	Reads Data from the Analog Inputs	6.2.4
#AAN	>(Data)	Reads Data from the Analog Input of a Channel	6.2.5
$AA0	!AA	Performs a Span Calibration	6.2.6
$AA1	!AA	Performs a Zero Calibration	6.2.7
$AA2	!AANNTTCCFF	Reads the Module Configuration	6.2.8
$AA5	!AAS	Reads the Reset Status of the Module	6.2.9
$AA5VV	!AA	Enables/Disables each Channel	6.2.10
$AA6	!AAVV	Reads the Enabled/Disabled Status of each Channel	6.2.11
$AA7CiRrr	!AA	Sets the Type Code of a Channel	6.2.12
$AA8Ci	!AACiRrr	Reads the Type Code of a Channel	6.2.13
$AAF	!AA(Data)	Reads the Firmware Version of the Modeule	6.2.14
$AAM	!AA(Data)	Reads the Name of the Module	6.2.15
$AAS1	!AA	Reloads the Default Calibration Parameters	6.2.16
~AAEV	!AA	Enables/Disables Calibration	6.2.22
~AAO(Name)	!AA	Sets the Name of the Module	6.2.23

Host Watchdog Command Sets			
Command	Response	Description	Section
~**	No Response	Host OK Command	6.2.17
~AA0	!AASS	Reads the Status of the Host Watchdog	6.2.18
~AA1	!AA	Resets the Host Watchdog Timeout Status	6.2.19
~AA2	!AAETT	Reads the Timeout Settings for the Host Watchdog	6.2.20
~AA3ETT	!AA	Sets the Host Watchdog Timeout Settings	6.2.21

6.2.3 %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)

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 and returns a response indicating that an error occurred is returned because the "CC" parameter have to be 0A.	

※Related Commands: $AA2

6.2.4 #AA

Description
This command is used to read the 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
(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.
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	#03
Response	>+025.12+054.12+150.12
Reads the analog input channels of module 03 and returns a response indicating that the command was successful, with the data for 3 channels in engineering format.	

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

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

※Section 7.1 Software Configuration Mode

6.2.5 #AAN

Description
This command is used to read the analog input data from a specific 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 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. Note that a response indicating that the command was successful will be returned if the specified channel is incorrect.
(Data)	The analog input data from the specified 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.
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	#032
Response	>+025.13
Reads data from channel 2 of module 03 and returns a response indicating that the command was successful, and the analog input value is +025.13mV.	
Command	#039
Response	?03
Attempts to read data from channel 9 of module 03. A response indicating that an error occurred is returned because channel 9 does not exist.	

※Related Commands: %AANNTTCCFF, $AA2

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

6.2.6 $AA0Ci

Description
This command is used to perform a zero calibration on the specified channel.

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 span calibration
C	The command to specify the channel to be calibrated
i	Specifies the channel to be calibrated

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	$010C0
Response	!01
Attempts to performs a zero calibration on channel 0 of module 01 and returns a valid response.	
Command	$030C1
Response	?03
Attempts to perform a zero calibration on channel 1 of module 03. An invalid command is returned because the “enable calibration” command was not sent in advance.	

※Related Commands: $AA1Ci, ~AAEV

※Notes:

1. The “enable calibration” command, ~AAEV, must be sent before this command is used, see Section 1.9 for details.
2. This command must be sent before the “span calibration” command, $AA1Ci, is used.

6.2.7 $AA1Ci

Description
This command is used to perform a span calibration on the specified channel.

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 zero calibration
C	The command to specify the channel to be calibrated
i	Specifies the channel to be calibrated

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	$011C0
Response	!01
Attempts to perform a span calibration on channel 0 of module 01 and returns a valid response.	
Command	$031C1
Response	?03
Attempts to perform a span calibration on channel 1 of module 03. An invalid command is returned because the “enable calibration” command was not sent in advance.	

※Related Commands: $AA0, ~AAEV

※Related Topics: Section 5 Calibration

※Notes:

3. The “enable calibration” command, ~AAEV, and the “zero calibration” command, $AA0Ci, must be sent before this command is used, see Sections 1.9, 2.4 and 2.21 for details.

6.2.8 $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.
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 indicating that the command was successful and shows that the address stored in the EEPROM is 0xFF, 60 Hz rejection and engineering units format.	
Command	$FF2
Response	!FF000A00
In Software Configuration mode, reads the configuration of module FF. The response indicating that the command was successful and shows that the address stored in the EEPROM is 0xFF, 60 Hz rejection and engineering units format.	

※Related Commands: %AANNTTCCFF

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

6.2.9 $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 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 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.	

6.2.10 $AA5VV

Description
This command is used to specify the 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 set the channels to enabled
VV	A two-digit hexadecimal value, 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	
Valid Command	!AA[CHECKSUM](CR)
Invalid Command	?AA[CHECKSUM](CR)
!	Delimiter character to indicate a valid command
?	Delimiter character to indicate an invalid command. Note that a response indicating that the command was invalid will be returned if an attempt is made to enable a channel that is not present.
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 channels 1, 3, 4, and 5 on module 03 and disables all other channels. The module returns a response indicating that the command was successful.	
Command	$036
Response	!033A
Reads the status of the channels of module 03, and returns a response indicating that the command was successful , with a value 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 of a specified module.

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

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, where bit 0 corresponds to channel 0, and bit 1 corresponds to channel 1, etc. When the bit is 0, it denotes that the channel is disabled, and 1 denotes that the channel is enabled.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	$0353A
Response	!03
Enables channels 1, 3, 4, and 5 and disables all other channels on module 03. The module returns a response indicating that the command was successful.	
Command	$036
Response	!033A
Reads the status all of the channels of module 03, and returns a response indicating that the command was successful with a value 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 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 range code
C	The command to specify the channel to specify the input channel
i	Specify the input channel to be set (0-5)
R	The command to specify the channel to specify the type code
rr	Represent the type code of the channel to be set. See Section 4 for details.

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	$037C0R20
Response	!03
Sets the type code for channel 0 of module 03 to 0x20 (PT100, -100 ~ +100 ℃), and the module returns a response indicating that the command was successful.	
Command	$037C1R90
Response	?03
Attempts to set the type code for channel 1 of module 03 to 0x90. The module returns a response indicating that the command was unsuccessful because the type code is incorrect.	

※Related Commands: $AA8Ci

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

6.2.13 $AA8Ci

Description
This command is used to read the type code information for a specific 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 of the channel
C	The command to specify the channel to specify the input channel
i	Specify the input channel to be set (0-5)

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 input channel the type code information relates to.
Rrr	rr represents the type code of the specified input channel. See Section 4 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	$038C0
Response	!03C0R20
Reads the Type Code for channel 0 of module 03 and returns a response indicating that the command was successful, with a value of 0x20 (PT100, -100 ~ +100 ℃).	
Command	$038C9
Response	?03
Attempts to read the type code for channel 9 of module 03 and returns a response indicating that the command was unsuccessful because the channel 9 does not exist.	

※Related Commands: $AA7CiRrr

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

6.2.14 $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.	

Example	
Command	$03F
Response	!031.0
Reads the firmware version of module 03, and returns a response indicating that the command was successful and showing that firmware is version 1.0.	

6.2.15 $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.	

Example	
Command	$03M
Response	!03ZT-2015
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-2015".	

※Related Commands: ~AAO(Name)

6.2.16 $AAB

Description
This command is used to diagnose the analog inputs for over-range, under-range, and wire opening conditions.

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)
B	The command to diagnose the analog inputs

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	$03B
Response	!03
Sends a command to Diagnoses the analog inputs of module 03. The module returns a valid response denoting that channel 0 is in either an over-range, an under-range or a wire opening condition.	

6.2.17 $AAS1

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

Example	
Command	$03S1
Response	!03
Sends a command to reload the factory default calibration parameters for module 01 and returns a valid response.	

6.2.18 ~**

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.

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

※Related Commands: ~AA0, ~AA1, ~AA2, ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation.

6.2.19 ~AA0

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

Syntax	
~AA0CHKSUM](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 EEPROM, and can only be reset by using the ~AA1 command.
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.	

Examples	
Command	~030
Response	!0300
Reads the status of the Host Watchdog for module 03 and returns a response indicating that the command was successful, with a value of 00, meaning that the Host Watchdog is disabled and no Host Watchdog timeout has occurred.	
Command	~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: ~**, ~AA1, ~AA2, ~AA3ETT

※Related Topics: Section 7.2 Dual Watchdog Operation

6.2.20 ~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, showing that no Host Watchdog timeout has occurred.	

※Related Commands: ~**, ~AA0, ~AA2, ~AA3ETT
※Related Topics: Section 7.2 Dual Watchdog Operation

6.2.21 ~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	!AAEVV[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
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 for module 03 and returns a response indicating that the command was successful, with a value of 1FF, 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 7.2 Dual Watchdog Operation

6.2.22 ~AA3ETT

Description
This command is used to enable/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	!01
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	!01164
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: ~**, ~AA0, ~AA1, ~AA2

※Related Topics: Section 7.2 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.

6.2.23 ~AAEV

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

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

Example	
Command	$030
Response	?03
Attempts to send a command to perform a span calibration on module 03, and 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: $AA0, $AA1, $AAS1
※Related Topics: 5 Calibration

6.2.24 ~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)
0	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	~030ZT-2015
Response	!03
Sets the name of module 03 to "ZT-2015" and returns a response indicating that the command was successful.	
Command	$03M
Response	!03ZT-2015
Reads the name of module 03 and returns a response indicating that the command was successful, with the name "ZT-2015".	

※Related Commands: $AAM

6.3 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

6.3.1 Modbus Address Mapping

Address Mapping		
Address	Description	Attribute
00259	The filter settings. 0: 60Hz rejection 1: 50Hz rejection	R/W
00260	The Modbus Host Watchdog mode: 0: The same as I-7000 series modules 1: The AO and DO commands can be used to clear the status of the Host Watchdog timeout	R/W
00261	Enables or disabsle the Host Watchdog: 0: Disable 1: Enable	R/W
00269	The Modbus Data Format: 0: Hexadecimal 1: Engineering Units	R/W
00270	The Host Watchdog timeout status. 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
00275	1: force to return 32767 for wire opening	R/W
10129 ~ 10134	The under range status of channels 0 to 5 (supports types 0x7 and 0x1A only)	R
30001 ~ 30006	The analog input value for channels 0 to 5	R
40257 ~ 40262	The type code for channels 0 to 5	R/W
40289 ~ 40294	Temperature offset of channel 0 to 5 in 0.1° C divisions/increment, valid range: -128 ~ 127	R/W
40385 ~ 40390	Resistance offset of channel 0 to 5 in 0.1 ohm increment, valid range: 0 ~ 255	R/W
40481	The Firmware Version (Low Word)	R
40482	The Firmware Version (Ligh 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:	R

	Bits 5:0 Baud Rate. Always set to 0x0A Bits 7:6 Reserved	
40489	The Host Watchdog timeout value. Valid range is 0 ~ 255, in 0.1 s intervals	R/W
40490	Enables or disables a specific channel	R/W
40492	The Host Watchdog timeout count. Write 0 to clear	R/W

6.3.2 PLC Address Mapping

Function Code	Description	Section
0x01	Reads the Coils	6.3.3
0x02	Reads 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 responds as 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.

6.3.3 01 (0x01) Reading the Coils

Description
This function code is used to read the current digital output readback 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

6.3.4 02 (0x02) Reading the Discrete Inputs

Description
This function code is used to read the current digital input 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

6.3.5 03 (0x03) Reading Multiple Registers

Description
This function code is used to read the current digital 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	0x03
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	0x03
02	Byte Count	1	Byte Count of 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

6.3.6 04 (0x04) Reading Multiple Input Registers

Description
This function code is used to read the current digital 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	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	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

6.3.7 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	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

6.3.8 06 (0x06) Writing Multiple Registers

Description
This function code is used to configure 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

6.3.9 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 Cchannel 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 ‘1’, it denotes that the configuration of the channel that was set is ON or Enable. If the bit is ‘0’, it denotes that the configuration of the channel that was set is OFF or Disable.

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	Input 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

6.3.10 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 the Enabled/Disabled Status of a Specific Channel	A.6
38 (0x26)	Sets a Specific Channel to Enabled/Disabled	A.7
41 (0x29)	Reads the Miscellaneous Settings	A.8
42 (0x2A)	Writes the Miscellaneous Settings	A.9

If the module does not support the sub-function code specified in the message, then it will response as foolows:

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 0x15 0x00(ZT-2015)

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

Description
This sub-function code is used to set the address fo 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	1 to 247
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

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

Description
This sub-function code is used to read the type code 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	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.

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.4 08 (0x08) Setting the Type Code

Description
This sub-function code is used to set the type code for 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.

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

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

A.6 37 (0x25) Reading the Channel Enabled/Disabled Status

Description
This sub-function code is used to read the enabled/disabled status for each 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	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

A.7 38 (0x26) Enabling/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

A.8 41 (0x29) Reading the Miscellaneous Settings

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

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

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x29
03	Miscellaneous Settings	1	The data format. See Section 4 for details.

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 42(0x2A) Writing the Miscellaneous Settings

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

Request			
Byte	Description	Length	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x2A
03	Miscellaneous Settings	1	The data format. See Section 1.8 for details.

Response			
Byte	Description	Length	Value
00	Address	1	1 to 247
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	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 AI type code and other information. When the module is powered on with 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(Address(Node ID), data format and AI 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 llows 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

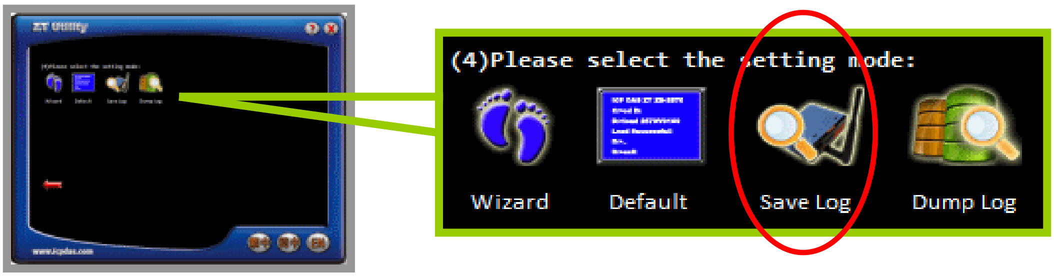
(1) Technical Support.

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:

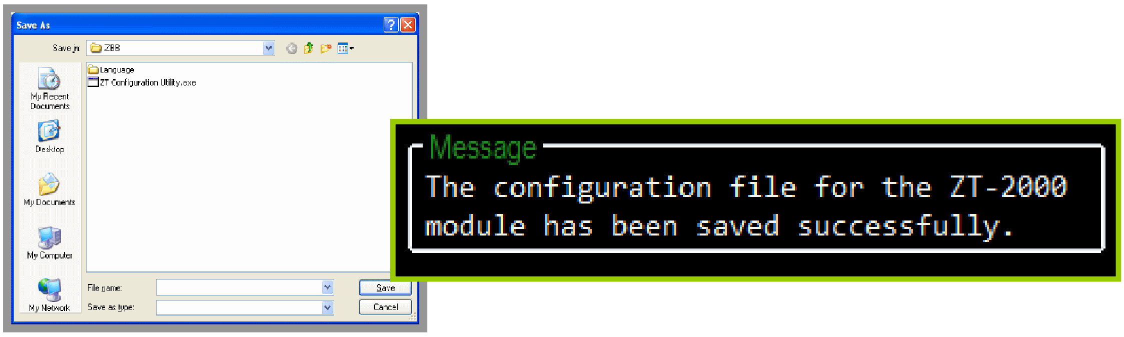
- *A description or diagram of the current DIP switch positions.*
- *A copy of the configuration file for the ZT-2000 coordinator. This file can be obtained using the procedure outlined below and should be attached to your email.*

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



(4)Please select the setting mode:

b. 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.



Message

The configuration file for the ZT-2000 module has been saved successfully.