ZB-2026 Series

User Manual

Warranty

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

Warning

ICP DAS assumes no liability for 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.

Table of Contents

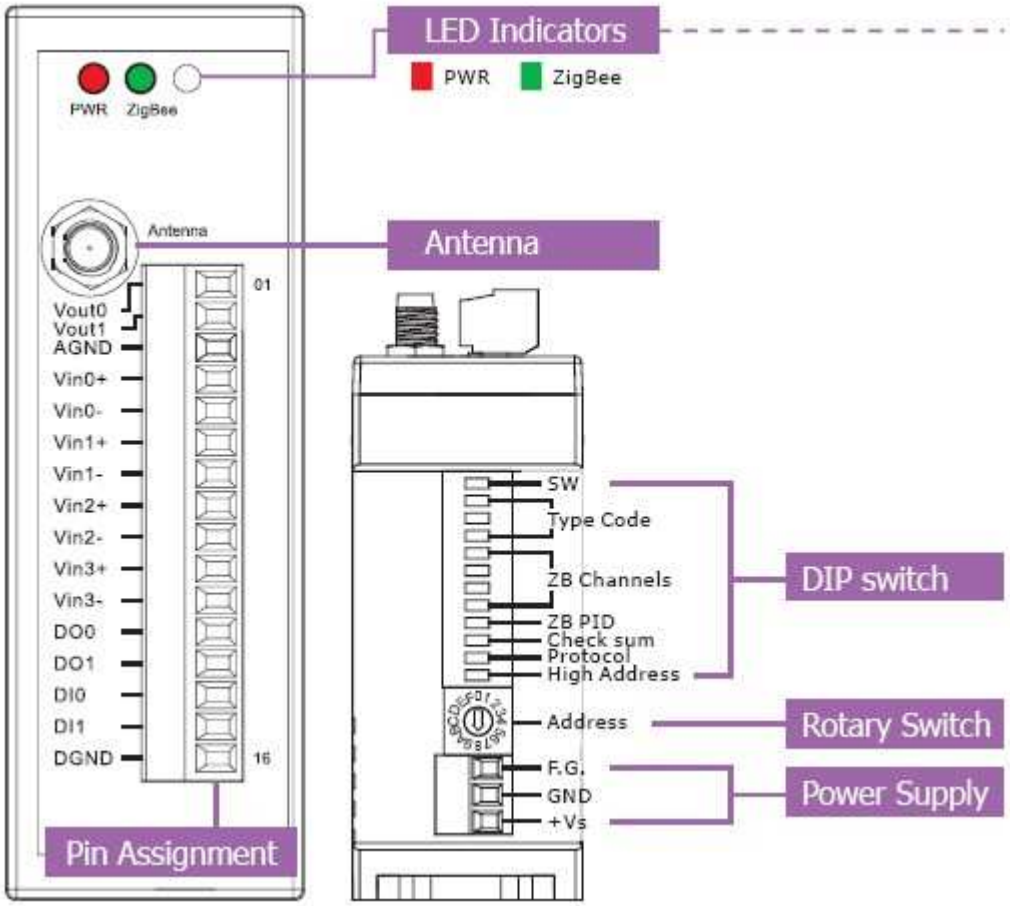
1.Introduction

The ZB-2000 series is a family of wireless ZigBee data acquisition modules that provide analog-to-digital, digital-to-analog, digital input/output, timer/counter and other functions. These modules can be remotely controlled using a set of DCON or Modbus RTU commands. The ZB-2026 series are 4-channel analog input, 2-channel analog output, 2-channel digital input and 2-channel digital output modules that can be configured via hardware. Refer to Section 1.5 “Switch Descriptions” for details.

The ZB-2000 series modules are not able to operate as standalone modules and need to be connected to a ZigBee host device, such as the ZB-2550(P)(-T) or the ZB-2570(P)(-T), in order to communicate with the ZB-2000 series. For more information, refer to the “ZigBee Converter Quick Start“ at available the following address:
http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zigbee_converter/

1.1. Pin Assignments

ZB-2026 Series



1.2. Specifications

I/O Specifications

Models	ZB-2026-T	ZB-2026-PA
Analog Input		
Input Channels	4 (Differential)	
Input Type	+/-10 V_{DC}, +/-5 V_{DC}, +/-1 V_{DC}, +/-500 mV, +/-150 mV, -20 mA ~ +20 mA (Requires Optional External 125 Ω Resistor)	
Resolution	16-bit	
Sampling Rate	10 Samples/Sec. (Total)	
Accuracy	+/-0.1% of FSR	
-3dB Bandwidth	15.7 Hz	
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	
Analog Output		
Output Channels	2	
Output Type	+/-10 V_{DC}, +/-5 V_{DC}, 0 ~ 10 V_{DC}, 0 ~ 5 V_{DC}	
Resolution	12-bit	
Accuracy	+/-0.1% of FSR	
Zero Drift	+/-30 μV/℃	
Span Drift	+/-25 ppm/℃	
Programmable Output Slope	0.0625 ~ 1024 V/Sec.	
Voltage Capability	20mA@10V	
Power-ON and Safe Value	Yes	
Digital Input		

Input Channels	2 (Sink)
On Voltage Level	3.5 V_{DC} ~ 50 V_{DC}
Off Voltage Level	1 V_{DC} max.
Input Impedance	10K ohm
Event Counters	Channels : 2
	Max. Counts: 16-bit (65535)
	Max. Input Frequency : 50 Hz
	Min. Pulse Width : 10 ms
Digital Output	
Output Channels	2 (Sink)
Output Type	Isolated Open Collector
Max.Load Current	700 mA/channel
Load Voltage	+5 V_{DC} ~+50 V_{DC}
Short Circuit Protection	Yes

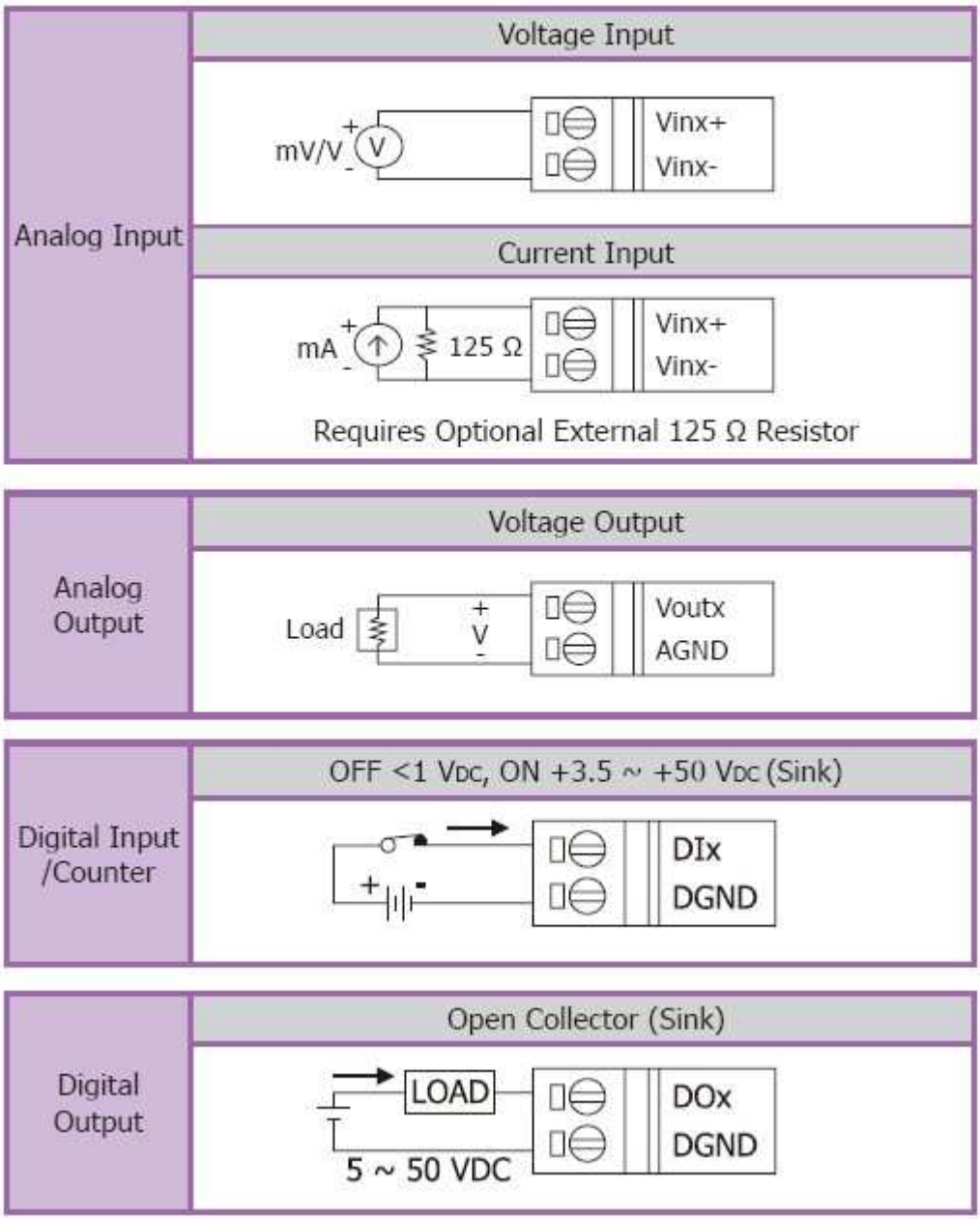
System Specifications

Models	**ZB-2026-T**	**ZB-2026-PA**
Communication Interface		
Wireless	ZigBee, IEEE 802.15.4 Standard	
Transmission power	4 dBm	22 dBm
Antenna 2.4 GHz -	3 dBi Omni-Directional antenna	5 dBi Omni-Directional antenna
Transmission range (LOS)	100 m	700 m(Typical) 1 km(Max.)
Certification	CE/FCC,FCC ID	No
Max. Slaves in a zigbee network	254	
ZB-100R/ZB-100T Supported	Yes	

Protocols	Supports DCON and Modbus RTU Protocols
Hot Swap	By Rotary and DIP switch
LED Indicators	
Power	1 LED, Red
ZigBee Communication	1 LED, Green
Isolation	
Intra-module Isolated, Field-to-Logic	2500 V_{DC} (for AI, AO, DI and DO)
EMS Protection	
ESD (IEC 61000-4-2)	4 kV Contact for Power Line, Communication Line and Each Channels, 8 kV Air for Random Point
EFT (IEC 61000-4-4)	4 kV for Power Line
Power	
Input Volatge Range	+10 V_{DC} ~ +30 V_{DC} (Reverse Polarity Protectionn)
Power Consumption	2.1 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 Mounting
Environment	
Operating Temperature	-25 ℃ ~ +75 ℃
Storage Temperature	-30 ℃ ~ +80 ℃
Humidity	10 ~ 90% RH, Non-condensing

1.3. Wire Connections

ZB-2026 Series

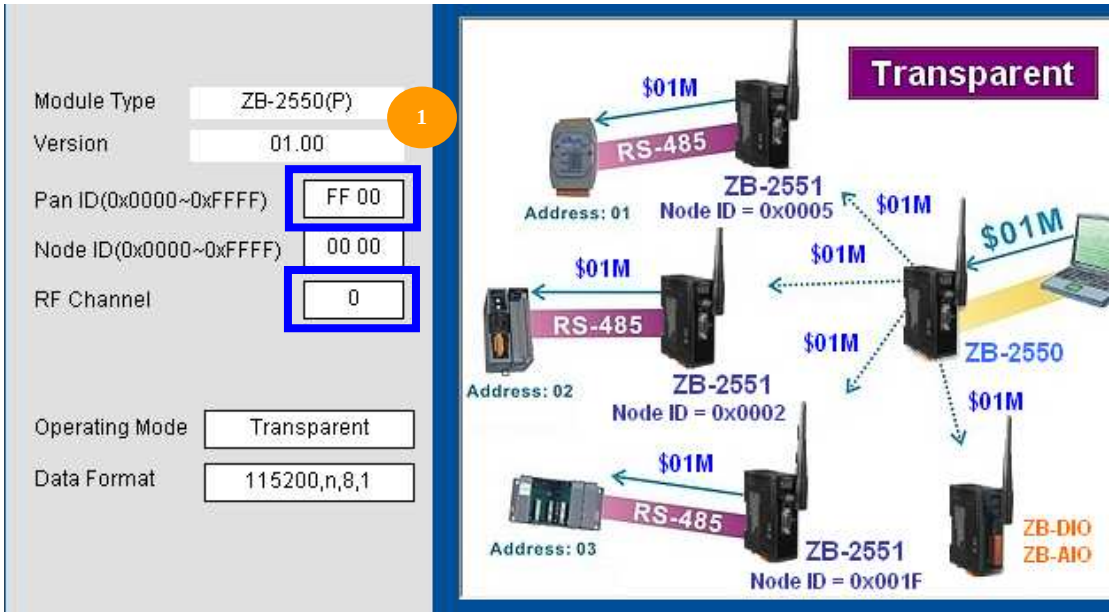


1.4. Quick Start

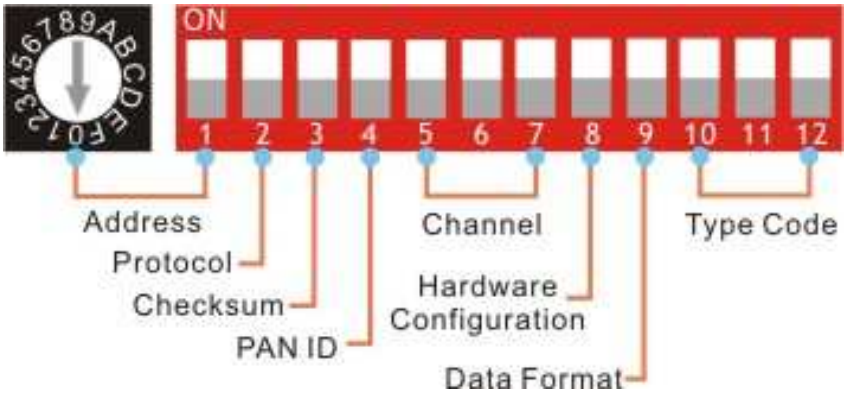
This Quick Start document describes the methods used to quickly set up and test ZB-2000 series modules using the ICP DAS DCON Utility.

First, you must set the ZB-2570(P)(-T)/ZB-2550(P)(-T) before using any ZB-2000 modules because the ZB-2570(P)(-T)/ZB-2550(P)(-T) is a ZigBee Net Server. For more information about the ZB-2570(P)(-T)/ZB-2550(P)(-T), please refer to the “ZigBee Converter Quick Start“ at the following address:
http://ftp.icpdas.com/pub/cd/usbcd/napdos/zigbee/zigbee_converter

If you have already installed the ZB-257x/ZB-255x series Utility, you only need to set the “PAN ID” and the “ZB RF Channel” for the ZB-2570(P)(-T)/ZB-2550(P)(-T) so that it is the same as the ZB-AIO setting. The ZB-AIO will then operate correctly.



1.5. Switch Descriptions



The following are description of each dip switch:

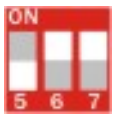
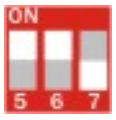
1. **Address:** The module address is defined using two components. The first is dip switch 1 and the second is a 16-position rotary switch. The address is a hexadecimal value that allows you set addresses maping from 0x01 to 0x1F (0x00 is used for initialization mode). Dip switch 1 defines the high 4 bits of the address value and the 16-position rotary switch defines the low 4 bits of the address.
   The address value is equal to the ZigBee PAN ID value. A unique node ID should be set for all ZigBee slave devices, such as ZigBee IO modules, ZigBee converters and ZigBee repeaters.
   The ZB-2026 series will change to INIT mode when the address value is 0. Refer to Section 1.7 “INIT mode” for more information.

2. **Protocol:** Dip switch 2 defines the protocol. The ON position means that the “Modbus RTU” protocol will be enabled and the OFF position denotes that the “DCON” protocol will be need.

3. **Checksum:** Dip switch 3 defines the checksum status. The ON positon enables the checksum and the OFF position disables the checksum. This option is only effective when the DCON protocol is enabled.

4. **PAN ID:** Dip switch 4 defines the ZigBee network PAN ID. Only 0xFF00 or 0xFF01 is allowed for ZigBee IO series modules. 0xFF01 is selected by moving the dip switch to the ON position and 0xFF00 is selected by moving the dip switch to the OFF position.

5. **Channel:** Dip switches 5-7 define the ZigBee operating channel. The configuration is as follows:

Switch Value	Channel	Switch Value	Channel	Switch Value	Channel
	0		1		2
	3		4		9
	14		15		

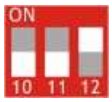
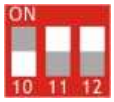
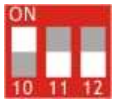
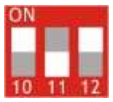
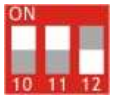
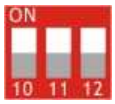
6. **Hardware configuration:** Dip switch 8 defines whether the ZB-2026 series type code is to be configured using firmware or hardware. The ON position indicates that configuration is via hardware and the OFF position defines that configuration is performed via firmware. The differences between firmware and hardware configuration are as follows:

Type Code is set to firmware configuration	Type Code is set to hardware configuration
Allows different type codes to be set for each channel.	All channels will use the same type code that is defined using the values set via dip switches 10-12.
The type code value is configured based on the EEPROM of the ZB-2026 series.	The type code value is configured based on the dip switch of the ZB-2026 series.
The data format is configured using software commands.	The data format is configured using the dip switch. Data format configuration commands are ignored in this mode.
When you use a command to read the current type code, you will get the value that is stored in the EEPROM of the ZB-2026 series.	When you use a command to read the current type code, you will get the value that has been configured via the dip switch of the ZB-2026 series.

7. **Data Format:** Dip switch 9 defines the data format of the ZB-2026 series. The ON position indicates hex format and the OFF position indicates engineering format. This dip switch is only valid when the “Hardware configuration” dip switch is in the ON position.

8. **Type Code:** Dip switches 10-12 define the input type code of the ZB-2026 series, as shown below.

ZB-2026 Series

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

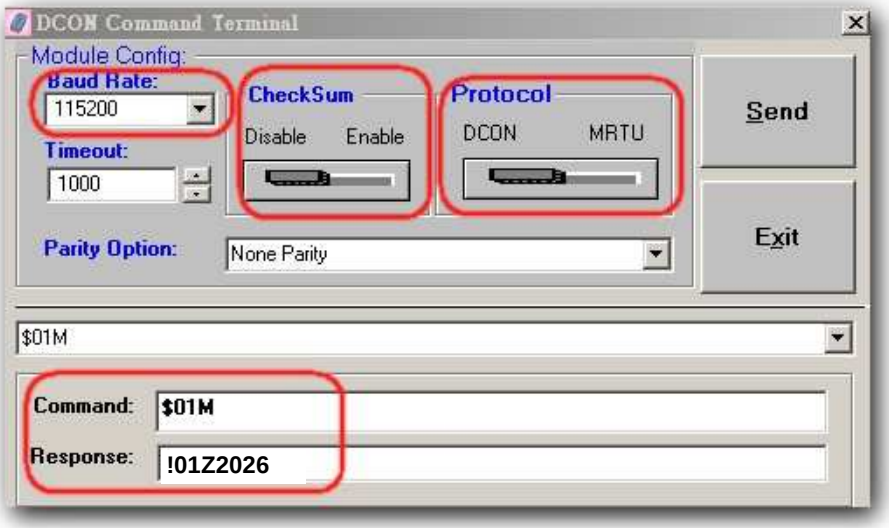
1.6. Connection

A ZigBee host must already exist in a ZigBee network. If you want to communicate with ZB-2000 IO modules, you need to use a ZB-2550(P)(-T) or a ZP-2570(P) to connect to your controller.

To create a ZigBee network, the “Channel” and “PAN ID” parameters of all ZigBee modules should be individually configured to the same value.

For instance, use the software utility to configure the PAN ID of the ZB-2550(P)(-T) as 0xFF00 and the Channel as 0. Then set dip switches 4-7 of the ZB-2026 series to the OFF position to set the PAN ID to 0xFF00 and the Channel to 0. Based on the above configuration, the ZB-2550(P)(-T), the ZB-2026 series can now communicate with each other. Set Dip switches 2 and 3 to the OFF position to set the protocol for the module to DCON and disable the checksum.

Rotating the rotary switch of the ZB-2026 series to position 1 will set the module address to 0x01. You can then use the “DCON Utility” on the host PC that is connected to the ZB-2550(P)(-T) to send a command to the ZB-2026 series. The response you receive should be similar to that shown below:



ZB-2026 series are command-based data acquisition modules. A number of commands are provided that can be used to configure and read AI data. Refer to Section 2 for details.

The ZB-2026 series also support the Modbus RTU protocol. The Configuration command format for the Modbus RTU is as follows:

Field ❶	Field ❷	Field ❸	Field ❹ ~ Field *n	Field (❹+*n)
Module Address	Function code	Sub function	Configuration field	CRC16

*n: This value depends on the Sub-function code.

Eg: To set channels 0, 1, 2 and 3 to enabled and channels 4, 5, 6 and 7 to disabled, the following command should be sent:

01 46 26 0F BA 69

The supported AI/O commands are as follows:

Function Code	Description
0x01	Read coils
0x02	Read discrete inputs
0x03	Read multiple registers
0x04	Read multiple input registers
0x05	Write single coils
0x0F	Write multiple coils

Eg: To read the current AI value of channels 0 to 7, the following command should be sent:

01 03 00 00 00 07 04 08

Eg: To set the filter to 50Hz, the following command should be sent:

01 05 01 02 FF 00 2C 06

To install the ZigBee AIO module, follow the steps below:

1. Connect the analog input.
2. Connect the ZigBee AIO module to the power supply using the +Vs and GND terminals.

3. In order to read data from the input channels when using the DCON protocol, send either a #AA or #AAN command to the module. See Sections 2.3 and 2.4 for details. When using the Modbus RTU protocol, use the Function 04h to read the data from the input channels. See Section 3.2 for details.

1.7. INIT Mode

Each ZigBee module has an internal EEPROM that is used to store its configuration, such as it's module address, ZigBee PAN ID, ZigBee channel, etc. If you forget the module's configuration information, you can use INIT mode to reset the ZB-2026 series to the default settings, then you can re-configure the module. To change to INIT mode, you only need to adjust the address value to 0.

The default settings for ZB AIO modules are:

- Protocol: DCON
- Module Address: 0
- Checksum: Disable

1.8. 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: FSR (FULL Scale Range)				

Data Format Settings (FF)

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

Key	Description
DF	Data Format 00: Engineering units 01: % of FSR 10: 2's Complement Hexadecimal
MS	Mode Settings 0: Normal mode 1: Fast mode
FS	Filter Settings 0: 60 Hz rejection 1: 50 Hz rejection.

Note: The reserved bits should be zero.

1.9. Analog output Type and Slew Rate Table

Type Code	Input Type	Data Format	+F.S.	-F.S.
32	+0 to +10V	Engineering units	+10.000	+0.000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000
33	-10 to +10V	Engineering units	+10.000	-10.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
34	+0 to +5V	Engineering units	+5.0000	+0.0000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000
35	-5 to +5V	Engineering units	+5.0000	-5.0000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
*1: FSR (FULL Scale Range)				

Slew Rate Control

0 Immediate chang
1 0.0625V/Second
2 0.125V/Second
3 0.25V/Second
4 0.5V/Second
5 1.0V/Second
6 2.0V/Second
7 4.0V/Second
8 8.0V/Second
9 16V/Second
A 32V/Second
B 64V/Second
C 128V/Second
D 256V/Second
E 512V/Second
F 1024V/Second

1.10. Calibration

Warning: *Performing calibration is not recommended until the process is fully understood.*

1.10.1 Analog Input

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 2.19 for details.
3. Enable calibration. Refer to Section 2.46 for details.
4. Apply the zero calibration voltage/current.
5. Send the zero calibration command. Refer to Section 2.7 for details.
6. Apply the span calibration voltage/current.
7. Send the span calibration command. Refer to Section 2.6 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 ZB-2026 series:

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

1.10.2 Analog Output

The calibration procedure is as follows:

1. Warm up the module for 30 minutes.

2. Set the type code to the type you want to calibrate. Refer to Section 2.23 for details.
3. Enable calibration. Refer to Section 2.46 for details.
4. Set the zero analog output. Refer to Section 2.5 for details.
5. Check the meter and trim the output until zero output. Refer to Section 2.11 for details.
6. Perform the analog output zero calibration command. Refer to Section 2.8 for details.
7. Set the span analog output. Refer to Section 2.5 for details.
8. Check the meter and trim the output until span output. Refer to Section 2.11 for details.
9. Perform the analog output span calibration command. Refer to Section 2.9 for details.

Notes:

1. Connect the calibration voltage to special channel you want.
2. Calibration voltages and currents are shown below.

Calibration voltages:

Type Code	32	33	34	35
Zero Input	0V	0V	0V	0V
Span Input	+10V	+10V	+5V	+5V

1.11. Technical Support

Should you encounter any problems while using the ZB-2026 series module, and are unable to find the help you need in this manual or on our website, please contact ICP DAS Product Support.

Email: support@icpdas.com
Website: http://www.icpdas.com/service/support.htm

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

1. Module name and serial number: The serial number can be found printed on the barcode label attached to the cover of the module.
2. Firmware version: See Sections 2.29 and 3.3.4 for information regarding the command used to identify the firmware version.
3. Host configuration (type and operating system)
4. If the problem is reproducible, please give full details describing the procedure used to reproduce the problem.
5. Any specific error messages displayed. If a dialog box with an error message is displayed, please include the full text of the dialog box, including the text in the title bar.
6. If the problem involves other programs or hardware devices, please describe the details of the problem in full.
7. Any comments and suggestions related to the problem are welcome.

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

2. DCON Protocol

All communication with ZB AIO modules consists of commands generated by the host and responses transmitted by the ZB AIO modules. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The ID is 01 by default and can be changed by transmitting the prescribed user command. All commands to the modules contain the ID address, meaning that only the addressed module will respond. The only exception to this is command ~**(Section 2.33) which is sent to all modules, but in this case, the modules do not reply to the command.

Command Format:

Leading Character	Module Address	Command	[CHKSUM]	CR

Response Format:

Leading Character	Module Address	Data	[CHKSUM]	CR

CHKSUM A 2-character checksum which is present when the checksum setting is enabled. See Section1.8 (Data Format Settings) for details.

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

Checksum Calculation:

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

Example:

Command string: $012(CR)

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

Response string: !01200600(CR)

1. Sum of the string = “!”+”0”+”1”+”2”+”0”+”0”+”6”+”0”+”0” = 21h+30h+31h+32h+30h+30h+36h+30h+30h = 1AAh
2. Therefore the checksum is AAh, and so CHKSUM = “AA”
3. The response string with the checksum = !01200600AA(CR)

Note:
All characters should be in upper case.

General Command Sets			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the module configuration	2.1
#**	No Response	Synchronized sampling	2.2
#AA	>(Data)	Reads AI data	2.3
#AAN	>(Data)	Reads the AI data of a special channel	2.4
#AAN(Data)	>	Sets the AO of a special channel	2.5
$AA0	!AA	Performs the AI span calibration	2.6
$AA1	!AA	Performs the AI zero calibration	2.7
$AA0N	!AA	Performs the AO zero calibration	2.8
$AA1N	!AA	Performs the AO span calibration	2.9
$AA2	!AANNTTCCFF	Reads the module configuration	2.10
$AA3NVV	!AA	Trims the AO calibration	2.11
$AA4	!AAS(Data)	Reads the synchronized data	2.12
$AA4N	!AA	Sets the AO power on value	2.13
$AA5	!AAS	Reads the module reset status	2.14
$AA5VV	!AA	Enables/Disables the AI channel	2.15
$AA6	!AAVV	Reads the AI enabled/disabled status	2.16
$AA6N	!AA(Data)	Last AO value readback	2.17
$AA7N	!AA(Data)	Reads the AO power on value	2.18
$AA7CiRrr	!AA	Sets the AI range configuration	2.19
$AA8N	!AA(Data)	Current AO value readback	2.20
$AA8Ci	!AACiRrr	Reads the AI range configuration	2.21
$AA9N	!AATTS	Reads AO configurations	2.22
$AA9NTS	!AA	Set AO configurations	2.23
$AAC	!AA	Clears the latched DI status	2.24
$AAD	!AAnn	Reads the DI counter status	2.25
$AADnn	!AA	Sets the DI counter status	2.26
$AAE	!AAnn	Reads the edge status of DI counter	2.27
$AAEnn	!AA	Sets the edge status of DI counter	2.28
$AAF	!AA(Data)	Reads the firmware version	2.29
$AAM	!AA(Data)	Reads the module name	2.30

$AALS	!(Data)	Reads the latched DI status	2.31
$AAS1	!AA	Reloads the default calibration parameters	2.32
~AA4N	!AA(Data)	Reads the AO safe value	2.39
~AA5N	!AA	Sets the AO safe value	2.40
~AAD	!AATT	Reads the DI/O configurations	2.44
~AADTT	!AA	Sets the DI/O configurations	2.45
~AAEV	!AA	Enables/Disables calibration	2.46
~AAO(Name)	!AA	Sets the module name	2.47
@AACECi	1AA	Clears DI counter	2.48
@AACH	!AA	Clears the AI high latches	2.49
@AACHi	!AA	Clears the AI high latch of a specific channel	2.50
@AACHCi	!AA	Clears the AI high latched alarm of a specific channel	2.51
@AACL	!AA	Clears the AI low latches	2.52
@AACLi	!AA	Clears the AI low latch of a specific channel	2.53
@AACLCi	!AA	Clears the AI low latched alarm of a specific channel	2.54
@AADI	!AAHHLL	Reads the AI alarm status	2.55
@AADHCi	!AA	Disables the AI high alarm of a specific channel	2.56
@AADLCi	!AA	Disables the AI low alarm of a specific channel	2.57
@AAHI(Data)CiT	!AA	Sets the the AI high alarm of a specific channel	2.58
@AALO(Data)CiT	!AA	Sets the AI low alarm of a specific channel	2.59
@AADODD	!AA	Sets digital output	2.60
@AARAOj	!AAHHLL	Reads the current alarms assocaited with a DO port	2.61
@AARECi	!AA(Data)	Reads the DI counter	2.62
@AARH	!AA(data)	Reads the high latches	2.63

@AARHi	!AA(data)	Reads the high latch of a specific channel	2.64
@AARHCi	!AA(data)S	Reads the high alarm of a specific channel	2.65
@AARL	!AA(data)	Reads the low latches	2.66
@AARLi	!AA(data)	Reads the low latch of a specific channel	2.67
@AARLCi	!AA(data)S	Reads the low alarm of a specific channel	2.68
@AAROOj	!AAHHLL	Reads the alarms assocaited with a DO port	2.69
Host Watchdog Command Sets			
Command	**Response**	**Description**	**Section**
~**	No Response	Host is OK	2.33
~AA0	!AASS	Reads the Host Watchdog status	2.34
~AA1	!AA	Resets the Host Watchdog status	2.35
~AA2	!AAETT	Reads the Host Watchdog timeout settings	2.36
~AA3ETT	!AA	Sets the Host Watchdog timeout settings	2.37
~AA4	!AA	Reads the DO power on/safe value	2.38
~AA5PPSS	!AA	Sets the DO power on/safe value directly	2.41
~AA6PN(Data)	!AA	Sets AO power on value directly of a special channel	2.42
~AA6SN(Data)	!AA	Sets AO safe value directly of a special channel	2.43

2.1. %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 1F)

NN The new address of the module in hexadecimal format (00 to 1F)

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

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

! Delimiter for a valid command

? Delimiter for an invalid command

AA The address of the module in hexadecimal format (00 to 1F)

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: %0202000A80 Response: !02

Sets the data format of module 02 to 90 (50Hz rejection) and returns a valid response.

Related Commands:

Section 2.10 $AA2

2.2. #**

Description:
This command allows every analog input module to read data from every input channels when the command is received and will store the data for later retrieval.

Syntax:
#[CHKSUM](CR)**
Delimiter character
**** Synchronized sampling command

Response:
There is no response with this command. To access the data, another command, $AA4, must be sent, see Section 2.12 for details.

Examples:
Command: #** No response
Send the synchronized sampling command.
Command: $014 Response:
>011+025.12+020.45+012.78+018.97+003.24+015.35+008.07+014.79
Send the command to read the synchronized data. The status byte of the response is 1, which means that is the first time the synchronized data has been read since the previous #** command.
Command: $014 Response:
>010+025.12+020.45+012.78+018.97+003.24+015.35+008.07+014.79
Send the command to read the synchronized data. The status byte of response is 0, which means that it is not the first time the synchronized data has been read since the previous #** command.

Related Commands:
Section 2.12 $AA4

2.3. #AA

Description:

This command is used to read the data from every analog input channel.

Syntax:

#AA[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be read (00 to 1F)

Response:

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

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

>	Delimiter character for a valid response
?	Delimiter character for an invalid response
(Data)	The data from every analog input channels, see Section 1.8 for the details of data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #01 Response:
>+025.12+020.45+012.78+018.97+003.24+015.35

Reads module 01 and receives the data in engineering format.

Command: #02 Response:
>4C532628E2D683A20F2ADBA1

Reads module 02 and receives the data in hexadecimal format.

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

Reads module 03 and the data is under range.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.19 $AA7CiRrr

2.4. #AAN

Description:
This command is read the analog input of channel N.

Syntax:
#AAN[CHKSUM](CR)
Delimiter character
AA The address of the module to be read (00 to 1F)
N The channel to be read, zero based.

Response:
Valid Response: **>(Data)[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
> Delimiter character for a valid response
? Delimiter character for an invalid response. An invalid command is returned if the specified channel is incorrect.
(Data) Analog input data of the specified channel, see Section 1.8 for details of the data format.
AA The address of the responding module (00 to 1F)

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 return a valid response.
Command: #029 Response: ?02
Reads data from channel 9 of module 02 and returns an invalid response because channel 9 is invalid.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.10 $AA2

2.5. #AAN(Data)

Description:
This command is used to set analog output of channel N.

Syntax:
#AAN(Data)[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be set (00 to 1F)
N	The channel to be set, zero based.
(Data)	The analog output value, see the Section 1.8 for details.

Response:

>	Delimiter character for a valid response
?	Delimiter character for the (Data) is out of range, and the output will go to the closest value in the setting of module's range
!	Delimiter character for the module's host Watch Dog flag is set, and the output command will be ignored and the output is set to Safe value.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $0190300 Response: !01

Reads the configuration of module 01 and returns output type 0 to 20mA and output change immediate.

Command: #010+05.000 Response: >

Outputs the module 01 value 5.0mA of the channel 0 and returns a valid response.

Command: #010+25.000 Response: ?

Outputs the module 01 value 25.0mA of the channel 0 and returns an invalid response that means the output value is over range.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.10 $AA2

2.6. $AA0

Description:
This command is used to perform a analog input span calibration.

Syntax:
$AA0[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be calibrated (00 to 1F)
0 The command for the span calibration

Response:
Valid Response: **!AA[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $010 Response: !01

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

Command: $020 Response: ?02

Performs a span calibration on module 02 and returns an invalid command because the “enable calibration” command was not sent in advance.

Related Commands:

Section 2.7 $AA1, Section 2.46 ~AAEV

Related Topics:

Section 1.10.1 Calibration

Notes:

The “enable calibration” command, ~AAEV, must be sent before this command is used, see Section 1.10.1 for details.

2.7. $AA1

Description:
This command is used to perform a analog input zero calibration.

Syntax:
$AA1[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be set (00 to 1F)
1 The command for the zero calibration

Response:
Valid Response: **!AA[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $011 Response: !01

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

Command: $021 Response: ?02

Performs a zero calibration on module 02 and returns an invalid command because the “enable calibration” command was not sent in advance.

Related Commands:

Section 2.6 $AA0, Section 2.46 ~AAEV

Related Topics:

Section 1.10.1 Calibration

Notes:

The “enable calibration” command, ~AAEV, must be sent before this command is used, see Section 1.10.1 for details.

2.8. $AA0N

Description:
The command is used to perform the analog output offset calibration of channel N.

Syntax:
$AA1[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be calibrated (00 to 1F)
0	Command to perform the offset calibration
N	The channel to be set, 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 1F)

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: $0101 Response: !01

Perform the analog output channel 1 zero calibration of module 01 and returns a valid response.

Related Commands:
Section 2.9 $AA1N, Section 2.11 $AA3NVV

2.9. $AA1N

Description:
This command is used to perform the analog output span calibration of channel N.

Syntax:
$AA1N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be calibrated (00 to 1F)
1	Command to perform the span calibration
N	The channel to be set, 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 1F)

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:$0112 Response: !01
Perform the analog output channel 2 span calibration of module 01 and returns a valid response.

Related Commands:
Section 2.8 $AA0N, Section 2.11 $AA3NVV

2.10. $AA2

Description:
This command is used to read the module configuration.

Syntax:
$AA2[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
2	Command to read the module configuration

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	The address of the responding module (00 to 1F)
TT	00 (Reserved).
CC	0A (Reserved).
FF	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.

Examples:

Command: $012 Response: !01000A00
Reads the configuration of module 01.

Command: $022 Response: !02000A02
Reads the configuration of module 02.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.8 Configuration Tables

2.11. $AA3NVV

Description:
This command is used to trim calibration of channel N.

Syntax:
$AA3NVV[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to 1F)
3	Command to trim the module calibration
N	The channel to be set, zero based
VV	Two hexadecimal digits to present the trim calibration value. 00 to 5F to increase 0 to 95 counts, and FF to A1 to decrease 1 to 95 counts.

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	The address of the responding module (00 to 1F)

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: $01301F Response: !01
Trims module 01 output to increase 31 counts and returns a valid response.

Related Commands:
Section 2.8 $AA0N, Section 2.9 $AA1N

2.12. $AA4

Description:
This command is used to read synchronized data that was obtained from the last #** command.

Syntax:
$AA4[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
4	The command for read synchronized

Response:
Valid Command: **!AAS(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 1F)
S	Status of the synchronized data 1: first reading 0: not the first reading
(Data)	Synchronized data. See Section 1.8 for details of data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #** No response

Sends the synchronized sampling command.

Command: $014 Response:
>011+00.000+00.100+01.000+10.000-01.000+05.000

Reads the synchronized data of module 01 and returns the synchronized data and sets the status byte to 1 to indicate that this is the first time the synchronized data has been read.

Related Commands:

Section 2.2 #**

2.13. $AA4N

Description:
This command is used to set the analog output power-on value for channel N.

Syntax:
$AA4N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to 1F)
4	The command for setting power-on value, store the current output value as power-on value
N	The channel to be set, 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 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: #012+00.000 Response: >
Sets channel 2 output 0.0 of module 01 and returns a valid response.
Command: $0142 Response: !01
Sets channel 2 power-on value of module 01and returns a valid response. The power-on value of channel 2 is set to 0.0 immediately.

Related Commands:
Section 2.5 #AAN(Data)

2.14. $AA5

Description:
This command is used to read the reset status.

Syntax:
$AA5[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read (00 to 1F)
5 Command to read the reset status

Response:
Valid Command: **!AAS[CHKSUM](CR)**
Invalid Command: **?AA[CHKSUM](CR)**
! Delimiter character for a valid command
? Delimiter character for an invalid command
AA The address of the responding module (00 to 1F)
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: $015 Response: !011

Reads the reset status of module 01. The response show that is the first time the $AA5 command has been sent since the module was powered on.

Command: #015 Response: !010

Reads the reset status of module 01. The response show that there has been no module reset since last $AA5 command was sent.

2.15. $AA5VV

Description:
This command is used to specify the channel(s) to be enabled.

Syntax:
$AA5VVVV[CHKSUM](CR)

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response. An invalid command is returned if an attempt is made to enable a channel that is not present.
AA	The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $0153A Response: !01

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

Command: $016 Response: !013A

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

Related Commands:

Section 2.16 $AA6

2.16. $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 1F)
6	Command to read the channel status

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

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

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $0153A Response: !01

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

Command: $016 Response: !013A

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

Related Commands:

Section 2.15 $AA5VV

2.17. $AA6N

Description:
This command is used to read the analog output requisition of channel N.

Syntax:
$AA6N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
6	Command to read the last output command value
N	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 or invalid type code
AA	The address of the responding module (00 to 1F)
(Data)	The last output command value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: #011+10.000 Response: !01
Sets the channel 1 output +10.000 of module 01 and returns a valid response.

Command: $0161 Response: !01+10.000
Reads the channel 1 the last output value and returns +10.000

Related Commands:
Section 2.5 #AAN(Data), Section 2.20 $AA8N

2.18. $AA7N

Description:
This command is used to read power-on value of channel N.

Syntax:
$AA7N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
7	Command to read power-on value
N	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 a invalid command
AA	The address of responding module (00 to 1F)
(Data)	The last output command value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $0170 Response: !01+10.000
Reads the channel 0 power-on value of module 01, return +10.000

Related Commands:
Section 2.13 $AA4N

2.19. $AA7CiRrr

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

Syntax:
$AA7CiRrr[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to 1F)
7C	Command to set the channel range code
i	The channel to be set, zero based
Rrr	rr represents the type code of the channel to be set. Refer to the Analog Input Type Setting table in Section 1.8 for details.

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response or invalid type code
AA	The address of the responding module (00 to 1F)

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: $017C0R08 Response: !01

Sets the type code for channel 0 of module 01 to be 08 (-10~+10V) and the module returns a valid response.

Command: $037C1R09 Response: ?03

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

Related Commands:

Section 2.21 $AA8Ci

2.20. $AA8N

Description:
This command is used to read the current analog output value of channel N.

Syntax:
$AA8N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
8	Command to read current output value
N	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 a invalid command
AA	The address of the responding module (00 to 1F)
(Data)	The current output value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $0180 Response: !01+01.000
Reads channel 0 current value of module 01, return +01.000

Related Commands:
Section 2.5 #AAN(Data), Section 2.17 $AA6N

2.21. $AA8Ci

Description:
This command is used to read the type code information of a channel.

Syntax:
$AA8Ci[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
8	Command to read the type code of a channel
Ci	Specifies which channel to access for the type code information

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response or invalid channel
AA	Address of the responding module (00 to 1F)
Ci	Specifies which input channel to access to retrieve the type code information.
Rrr	Represents the type code of the specified input channel. Refer to the Analog Input Type Setting table in 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.

Examples:

Command: $018C0 Response: !01C0R09

Reads the channel 0 input range of module 01 and returns 09 (-5~+5V).

Related Commands:

Section 2.19 $AA7CiRrr

Related Topics:

Section 1.8 Configuration Tables

2.22. $AA9N

Description:
This command is used to read analog output configuration of channel N.

Syntax:
$AA9N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
9	Command to read analog output configuration
N	The channel to be read, zero based

Response:
Valid Command: **!AATS[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 1F)
T	Analog output type. Refer to the Analog Input Type Setting table in Section 1.9 for details.
S	Analog output slew rate. Refer to the Analog Input Type Setting table in Section 1.9 for details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $0190 Response: !01330
Reads the channel 0 analog output configuration of module 01 and returns +/-10V output and change immediate

Related Commands:
Section 2.23 $AA9NTS

2.23. $AA9NTS

Description:
This command is used to set analog output configuration of channel N.

Syntax:
$AA9NTS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to 1F)
9	Command for setting analog output configuration
N	The channel to be set, zero based
T	Analog output type. Refer to the Analog Input Type Setting table in Section 1.9 for details.
S	Analog output slew rate. Refer to the Analog Input Type Setting table in Section 1.9 for details.

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 1F)

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: $019131 Response: !01
Sets the channel 1 analog output configuration of module 01 -10 to 10V output range and slew rate 0.625V/Second and returns a valid response.

Related Commands:
Section 2.22 $AA9N

2.24. $AAC

Description:
This command is used to clear digital input/output latch.

Syntax:
$AAC[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be clear (00 to 1F)
C Command to clear digital input/output latch

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 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01C Response: !01
Clears address 01 latched data and returns a valid response.

Related Commands:
Section 2.31 $AALS

2.25. $AAD

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

Syntax:
$AAD[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
D	Command to read the counter status of the channel

Response:
Valid Command: **!AAnn[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 1F)
nn	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.

Examples:

Command: $01D3A Response: !01

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

Command: $01D Response: !013A

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

Related Commands:

Section 2.26 $AADnn

2.26. $AADnn

Description:
This command is used to specify the counters of channel to be enabled.

Syntax:
$AADnn[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to 1F)
D	The command to set the counters of channel to enabled
nn	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 1F)

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: $01D3A Response: !01

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

Command: $01D Response: !013A

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

Related Commands:

Section 2.25 $AAD

2.27. $AAE

Description:
This command is used to read the rising/falling edges of each channel.

Syntax:
$AAE[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
E	The command to read the edge status of the channel

Response:
Valid Command: **!AAnn[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 1F)
nn	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 counter of the channel is rising edge, and 1 denotes that the counter of the channel is falling edge.

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: $01E3A Response: !01

The counters of channel 1, 3, 4, and 5 are falling edge and all other channels are rising edge module 01. The module returns a valid response.

Command: $01E Response: !013A

Reads the counter status of module 01 and returns a response of 3A, which denotes that counters of channel 1, 3, 4, and 5 are falling edge and all other channels are rising edge.

Related Commands:

Section 2.28 $AAEnn

2.28. $AAEnn

Description:
This command is used to specify the counters of channel to be rising/falling edge.

Syntax:
$AAEnn[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to 1F)
E	The command to set the counters of channel to enabled
nn	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 rising edge, and 1 denotes that the channel is falling edge.

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 1F)

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: $01E3A Response: !01

The counters of channel 1, 3, 4, and 5 are falling edge and all other channels are rising edge on module 01. The module returns a valid response.

Command: $01E Response: !013A

Reads the counter status of module 01 and returns a response “3A”, which denotes that channels 1, 3, 4, and 5 are falling edge and all other channels are rising edge.

Related Commands:

Section 2.27 $AAE

2.29. $AAF

Description:
This command is used to read the firmware version.

Syntax:
$AAF[CHKSUM](CR)
$ Delimiter character
AA The address of the module to be read (00 to 1F)
F Command to read the firmware version

Response:
Valid Response: **!AA(Data)[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)
(Data) A string indicating the firmware version of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01F Response: !01A2.0
Reads the firmware version of module 01 and shows that it is version A2.0.

2.30. $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 1F)
M Command to read the module name

Response:
Valid Response: !AA(Name)[CHKSUM](CR)
Invalid Response: ?AA[CHKSUM](CR)
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)
(Name) A string showing the name of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: $01M Response: !01Z2026
Reads the module name of module 01 and shows the name “Z2026”.

Related Commands:
Section 2.47 ~AAO(Name)

2.31. $AALS

Description:
This command is used to read digital input/output latch.

Syntax:
$AALS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to 1F)
L	Command to read latched digital input
S	0 = select latch low status 1 = select latch high status

Response:
Valid Command: **!(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 1F)
(Data)	Status of the latched digital output/input channels (a four digit hexadecimal value followed by 00)

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: $01L1 Response: !030100
Reads address 01 latch-high data and shows that digital output channel 0 and 1 and digital input channel 0 are latched high.

2.32. $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 to be set (00 to 1F)
S1	Command to reload the factory default calibration parameters

Response:

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

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

Related Topics:
Section 1.10 Calibration

2.33. ~**

Description:
Informs all modules that the host is OK

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

Response:
No response.

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

Related Commands:
Section 2.34 ~AA0, Section 2.35 ~AA1, Section 2.36 ~AA2, Section 2.37 ~AA3ETT, Section 2.39 ~AA4N, Section 2.41 ~AA5PPSS

2.34. ~AA0

Description:
This command is used to read the host watchdog status

Syntax:
~AA0[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be read (00 to 1F)
0 Command to read the module status

Response:
Valid Response: **!AASS[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)
SS Two hexadecimal digits that represent the host watchdog status, where:
Bit 2: 0 indicates that no host watchdog time out has occurred and 1 indicates that a host watchdog time out has occurred.
The host watchdog status is stored in EEPROM and can only be reset using the ~AA1 command.
Bit 7: 0 indicates that the host watchdog is disabled and 1 indicates the host watchdog 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: ~010 Response: !0100

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

Command: ~020 Response: !0204

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

Related Commands:

Section 2.33 ~**, Section 2.35 ~AA1, Section 2.36 ~AA2, Section 2.37 ~AA3ETT

2.35. ~AA1

Description:
This command is used to reset the host watchdog time out status

Syntax:
~AA1[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set (00 to 1F)
1 Command to reset the host watchdog time out status

Response:
Valid Response: **!AA[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:
Command: ~010 Response: !0104
Reads the host watchdog status of module 01 and shows that a host watchdog time out has occurred.
Command: ~011 Response: !01
Resets the host watchdog time out status of module 01 and returns a valid response.
Command: ~010 Response: !0100
Reads the host watchdog status of module 01 and shows that no host watchdog time out has occurred.

Related Commands:
Section 2.33 ~**, Section 2.34 ~AA0, Section 2.36 ~AA2, Section 2.37 ~AA3EVV

2.36. ~AA2

Description:
This command is used to read the host watchdog time out value of a module.

Syntax:
~AA2[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to 1F)
2	Command to read the host watchdog time out value

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	The address of the responding module (00 to 1F)
E	0: the host watchdog is disabled 1: the host watchdog is enabled
VV	Two hexadecimal digits to represent the time out value in tenths of a second, for example, 01 means 0.1 seconds and FF means 25.5 seconds.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~012 Response: !011FF

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

Related Commands:

Section 2.33 ~**, Section 2.34 ~AA0, Section 2.35 ~AA1, Section 2.37 ~AA3ETT

2.37. ~AA3ETT

Description:
This command is used to enable/disable the host watchdog and set the host watchdog time out value

Syntax:
~AA3EVV[CHKSUM](CR)

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~013164 Response: !01

Enables the host watchdog of module 01 and sets the time out value to be 10.0 seconds. The module returns a valid response.

Command: ~012 Response: !01164

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

Related Commands:

Section 2.33 ~**, Section 2.34 ~AA0, Section 2.35 ~AA1, Section 2.36 ~AA2

2.38. ~AA4

Description:
This command is used to read the digital output power-on value and safe value.

Syntax:
~AA5PPSS[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to 1F)
4	Command to read the digital output power-on value and safe value

Response:

Valid command: **!AAPPSS[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 1F)
PP	Two hexadecimal digits to represent the digital output power on value
SS	Two hexadecimal digits to represent the digital output safe value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~0150102 Response: !01

Sets the digital output power on value to be 01 and the safe value to be 02 and returns a valid response.

Command: ~014 Response: !010102

Reads the digital output power on value and safe value of module 01 and returns 0102, which denotes that the digital output power on value is 01 and safe value is 02.

Related Commands:

Section 2.41 ~AA5PPSS

Notes:

Both the power-on value and safe value have no effect on the digital output that are associated with alarm outputs.

2.39. ~AA4N

Description:
This command is used to read analog output safe value of channel N

Syntax:
~AA4N[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to 1F)
4	Command to read the analog output safe value
N	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 1F)
(Data)	Analog output value, see Section 1.8 for 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: $0190 Response: !01+06.000
Reads the channel 0 analog output safe value of module 01 and returns +6.000V.

Related Command:
Section 2.40 ~AA5N

2.40. ~AA5N

Description:
This command is used to set analog output safe value.

Syntax:
~AA5N[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set (00 to 1F)
5 Command to set analog output safe value
N The channel to be, 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 1F)

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: ~0151 Response: !01
Sets the current channel 1 analog output to be the safe value and returns a valid response.

Related Command:
Section 2.38 ~AA4N, Section 2.5 #AAN(Data)

2.41. ~AA5PPSS

Description:
This command is used to set the digital output power on value and safe value.

Syntax:
~AA5PPSS[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to 1F)
5	Command to set digital output power-on value and safe value
PP	Two hexadecimal digits to represent the digital output power-on value
SS	Two hexadecimal digits to represent the digital output safe value

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 1F)

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: ~0150102 Response: !01

Sets the digital output power-on value to be 01 and the safe value to be 02 and returns a valid response.

Command: ~014 Response: !010102

Reads the digital output power-on value and safe value of module 01 and returns 0102, which denotes that the digital output power-on value is 01 and safe value is 02.

Related Commands:

Section 2.38 ~AA4

Notes:

Both the power-on value and safe value have no effect on the digital output that are associated with alarm outputs.

2.42. ~AA6PN(Data)

Description:
This command is used to set the analog output power on value of a special channel.

Syntax:
~AA6PN(Data)[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to 1F)
6P	Command to set analog output power-on value
N	The channel to be, zero based
(Data)	The analog output value, see the Section 1.8 for details.

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 1F)

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: $0190300 Response: !01

Reads the configuration of module 01 and returns output type 0 to 20mA and output change immediate.

Command: ~016P0+05.000 Response: !01

Sets the channel 0 power-on value 5.0mA of the module 01 and returns a valid response.

Command: ~016P1+25.000 Response: ?01

Sets the channel 1 power-on value 25mA of module 01 and returns an invalid response that means the output value is over range.

Related Commands:

Section 2.39 $AA4N, Section2.8 $AA7N

2.43. ~AA6SN(Data)

Description:
This command is used to set the analog output safe value of a special channel.

Syntax:
~AA6SN(Data)[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to 1F)
6P	Command to set analog output safe value
N	The channel to be, zero based
(Data)	The analog output value, see the Section 1.8 for details.

Response:

Valid command:	**!AA[CHKSUM](CR)**
Invalid command:	**?AA[CHKSUM](CR)**
!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 1F)

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: $0190300 Response: !01

Reads the configuration of module 01 and returns output type 0 to 20mA and output change immediate.

Command: ~016S0+05.000 Response: !01

Sets the channel 0 safe value 5.0mA of the module 01 and returns a valid response.

Command: ~016S1+25.000 Response: ?01

Sets the channel 1 safe value 25mA of module 01 and returns an invalid response that means the output value is over range.

Related Commands:

Section 2.39 ~AA4N, Section2.40 ~AA5N

2.44. ~AAD

Description:
This command is used to read the DI/O configurations.

Syntax:
~AAD[CHKSUM](CR)
~ Delimiter character
AA Address of the module to be read (00 to 1F)
D Command to read the DI/O configurations

Response:
Valid Response: **!AATT[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)
TT Two hexadecimal digits that represent the miscellaneous setting as follows

7	6	5	4	3	2	1	0
Reserved						OA	IA

Key	Description
OA	DO active state 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IA	DI active state 0: input value 1 for non-signal or the low voltage Input value 0 for the high voltage 1: input value 0 for non-signal or the low voltage Input value 1 for the high voltage

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: ~01D Response: !0101

Reads the miscellaneous settings of module 01 and returns 01, which denotes that digital output channels are inactive mode.

Related Commands:

Section 2.5 ~AADTT

2.45. ~AADTT

Description:
This command is used to set DI/O configurations.

Syntax:
~AAEV[CHKSUM](CR)

~ Delimiter character
AA Address of the module to be set (00 to 1F)
D Command to set DI/O configurations
TT Two hexadecimal digits that represent the miscellaneous setting as follows

7	6	5	4	3	2	1	0
Reserved						OA	IA

Key	Description
OA	DO active state 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IA	DI active state 0: input value 1 for non-signal or the low voltage Input value 0 for the high voltage 1: input value 0 for non-signal or the low voltage Input value 1 for the high voltage

Response:
Valid Response: **!AA[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)

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: ~01D01 Response: !01
Sets the miscellaneous settings of module 01 to 01 and returns a valid response.

Related Commands:
Section 2.44 ~AAD

2.46. ~AAEV

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

Syntax:
~AAEV[CHKSUM](CR)

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

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	The address of the responding module (00 to 1F)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $010 Response: ?01

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

Command: ~01E1 Response: !01

Enables calibration on module 01 and returns a valid response.

Command: $010 Response: !01

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

Related Commands:

Section 2.6 $AA0, Section 2.7 $AA1

Related Topics:

Section 1.10.1 Calibration

2.47. ~AAO(Data)

Description:
This command is used to set the module name

Syntax:
~AAO(Name)[CHKSUM](CR)
~ Delimiter character
AA The address of the module to be set (00 to 1F)
O Command to set the module name
(Name) New name of the module (max. 6 characters).

Response:
Valid Response: **!AA[CHKSUM](CR)**
Invalid Response: **?AA[CHKSUM](CR)**
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA The address of the responding module (00 to 1F)

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: ~01OZ2026 Response: !01
Sets the name of module 01 to be “Z2026” and returns a valid response.
Command: $01M Response: !01Z2026
Reads the name of module 01 and returns “Z2026”.

Related Commands:
Section 2.30 $AAM

2.48. @AACECi

Description:
This command is used to reset the counter of a special channel.

Syntax:
@AACECi[CHKSUM](CR)
@ Delimiter character
AA The address of the module to be set (00 to 1F)
CE Command to reset the counter
Ci i specifies the channel to be reset, 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 1F)

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: @01CEC1 Response: !01
Resets the counter 1 of module 01 to the preset value and returns a valid response.

Related Commands:
Section 2.62 @AARECi

2.49. @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 cleared (00 to 1F)
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 1F)

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: @01RH0 Response: !01+05.000
Reads the high latch value of channel 0 and returns +05.000.
Command: @01CH Response: !01
Clears the high latch value of channel 0 and returns a valid response.
Command: @01RH0 Response: !01+00.000
Reads the high latch value of channel 0 and returns +00.000.

Related Commands:
Section 2.50 @AACHi, Section 2.63 @AARH, Section 2.64 @AARHi

2.50. @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 cleared (00 to 1F)
CH 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 1F)

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: @01RH1 Response: !01+06.000
Reads the high latch value of channel 1 and returns +06.000.
Command: @01CH1 Response: !01
Clears the high latch value of channel 1 and returns a valid response.
Command: @01RH1 Response: !01+00.000
Reads the high latch value of channel 1 and returns +00.000.

Related Commands:
Section 2.49 @AACH, Section 2.63 @AARH, Section 2.64 @AARHi

2.51. @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 cleared (00 to 1F)
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 1F)

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: @01CHC0 Response: !01
Clears the high alarm status of channel 0 and returns a valid response.

Related Commands:
Section 2.49 @AACH, Section 2.63 @AARH, Section 2.64 @AARHi

2.52. @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 cleared (00 to 1F)
CL 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 1F)

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: @01RL0 Response: !01-05.000
Reads the low latch value of channel 0 and returns -05.000.
Command: @01CL Response: !01
Clears the low latch value of channel 0 and returns a valid response.
Command: @01RL0 Response: !01+00.000
Reads the low latch value of channel 0 and returns +00.000.

Related Commands:
Section 2.53 @AACLi, Section 2.66 @AARL, Section 2.67 @AARLi

2.53. @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 cleared (00 to 1F)
CL 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 1F)

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: @01RL1 Response: !01-06.000
Reads the low latch value of channel 1 and returns -06.000.
Command: @01CL1 Response: !01
Clears the low latch value of channel 1 and returns a valid response.
Command: @01RL1 Response: !01+00.000
Reads the low latch value of channel 1 and returns +00.000.

Related Commands:
Section 2.52 @AACL, Section 2.66 @AARL, Section 2.67 @AARLi

2.54. @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 cleared (00 to 1F)
CLC	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 1F)

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: @01CHC7 Response: !01
Clears the low alarm status of channel 7 and returns an valid response.

Related Commands:
Section 2.52 @AACL, Section 2.66 @AARL, Section 2.67 @AARLi

2.55. @AADI

Description:
This command is used to read the digital input and digital output status.

Syntax:
@AADI [CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to 1F)
DI	Command to read the DI/O 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 1F)
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 DO has not output, and 1 denotes that DO has output.
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 DI has not input, and 1 denotes that DI has input.

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: @01DI Response: !010102

Reads the DI/O status of module 01 and returns a response indicating that DO on channel 0 has in the output and DI on channel 1 has in the input.

Related Commands:

Section 2.56 @AADHCi, Section 2.57 @AADLCi, Section 2.58 @AAHI(data)CiT, Section 2.59 @AALO(data)CiT

2.56. @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 1F)
DH	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 1F)

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: @01DHC0 Response: !01
Disables the channel 0 high alarm of module 01 and returns a valid response.

Related Commands:
Section 2.55 @AADI

2.57. @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 1F)
DL	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 1F)

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: @01DLC5 Response: !01
Disables the low alarm of channel 5 and returns a valid response.

Related Commands:
Section 2.55 @AADI

2.58. @AAHI(Data)CiTOj

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

Syntax:
@AAHI(data)CiTOj [CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set (00 to 1F)
HI	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 the details.
Ci	The channel to be set, zero based
T	The alarm type: M: Momentary alarm L: Latched alarm
Oj	j specifies the digital output port to be used for the alarm ouput

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 1F)

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: @01HI+09.000C0MO1 Response: !01

Sets the channel 0 high alarm limit is +09.000, the type is momentary and digital output channel 1 is the high alarm output and returns a valid response.

Command: @01RHC0 Response: !01+09.000MO1

Reads the alarm status and returns a response indicating that the channel 0 high alarm limit is +09.000, the type is momentary and digital output channel 1 is the high alarm output.

Related Commands:

Section 2.65 @AARHCi

2.59. @AALO(Data)CiTOj

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 1F)
LO	Command to set the low alarm
(data)	The low alarm limit, which should be consistent with the data format. Refer to Section 1.8 for the details.
Ci	The channel to be set, zero based
T	The alarm type: M: Momentary alarm L: Latched alarm
Oj	j specifies the digital output port to be used for the alarm ouput

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 1F)

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: @01LO-03.000C1LO0 Response: !01

Sets the channel 1 low alarm limit is -03.000, the type is latched and the channel 0 digital output is the low alarm output and returns a valid response.

Command: @01RLC1 Response: !010-03.000LO0

Reads the alarm status and returns a response indicating that the channel 1 low alarm limit is -03.000, the type is latched and channel 0 digital output is the low alarm output.

Related Commands:

Section 2.68 @AARLCi

2.60. @AADODD

Description:
This command is used to set digital output status

Syntax:
@AADODD[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set (00 to 1F)
DO	Command to set the digital output ports
DD	A two-digit hexadecimal value, where bit 0 corresponds to DO0, bit 1 corresponds to DO1, etc. When the bit is 1, it denotes that the digital output port is on, and 0 denotes that the digital output port is off.

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 1F)

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: @01DO01 Response: !01

Sets DO0 to on, DO1 to off and returns a valid response.

Related Commands:

Section 2.55 @AADI

Notes:

1. If the digital output port is already set to be an alarm output port, then the value written to the port is ignored.

2. When a host watchdog timeout occurs, the module will respond with an invalid command for this command and the DO value that was sent is ignored.

2.61. @AARAOj

Description:
This command is used to read current alarm associated with a digital output port

Syntax:
@AARAOj[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to 1F)
RA	Command to read the currently activated alarms associated with a digital output port.
Oj	j specifies the digital output port

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 1F)
HH	A two-digital hexadecimal value to represent the currently activated high alarms associated with the digital output port, where bio 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, it denotes that there is an activated high alarm associated with the channel. When the bit is 0, it denotes that there are no activated high alarms associated with the channel.
LL	A two-digital hexadecimal value to respresent the currently activated low alarms associated with the digital output port, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, it denotes that there is an activated low alarm associated with the channel. When the bit is 0, it denotes that there are no activated low alarms associated with the channel.

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: @01RAO1 Response: !011122

Reads the currently activated alarms associated with the DO1 of module 01, The module responds with 1122, which denotes that there is an activated high alarm associated with channels 0 and 4, and an activated low alarm associated with channels 1 and 5.

Related Commands:

Section 2.69 @AAROOj

2.62. @AARECi

Description:
This command is used to read the count of a special channel

Syntax:
@AARECi[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to 1F)
RE	Command to read the currently activated alarms associated with a digital output port.
Ci	i specifies 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 1F)
(Data)	The DI count of the specified channel

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: @01REC0 Response: !0100000008

Reads data from channel 0 of module 01 and returns the count 00000008 of CH0.

Related Commands:
Section 2.48 @AACECi

2.63. @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 1F)
RH 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 1F)
(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.

Examples:
Command: @01RH Response:
!01+08.000+00.000+00.000+00.000+00.000+00.000+00.000+00.000

Reads the high latch values of module 01 and returns the data in engineering format.

Related Commands:
Section 2.49 @AACH, Section 2.50 @AACHi, Section 2.64 @AARHi

2.64. @AARHi

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

Syntax:
@AARHi [CHKSUM](CR)
@ Delimiter character
AA The address of the module to be read (00 to 1F)
RH 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 1F)
(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.

Examples:
Command: @01RH0 Response: !01+08.000
Reads the high latch value of channel 0 and returns the data +08.000 in engineering format.

Related Commands:
Section 2.49 @AACH, Section 2.50 @AACHi, Section 2.63 @AARH

2.65. @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 1F)
RH	Command to read the high alarm status
Ci	The channel to be read, zero based

Response:
Valid command: **!AA(data)SOj[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 1F)
(data)	The high 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
Oj	j specifies the digital output port

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: @01RHC0 Response: !01+08.0002O0

Reads the high alarm status of channel 0 and returns a response indicating that the high alarm limit is +08.000, the type is latched and high alarm output is channel 0 digital output.

Related Commands:

Section 2.56 @AADHCi, Section 2.58 @AAHI(data)CiTOj

2.66. @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 1F)
RL 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 1F)
(Data) The low latch values of all 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.

Examples:
Command: @01RL Response: !01
-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:
Section 2.52 @AACL, Section 2.53 @AACLi, Section 2.67 @AARLi

2.67. @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 1F)
RL 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 1F)
(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.

Examples:
Command: @01RL0 Response: !01-02.000
Reads the low latch value of channel 0 and returns the data -02.000 in engineering format.

Related Commands:
Section 2.52 @AACL, Section 2.53 @AACLi, Section 2.66 @AARL

2.68. @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 1F)
RL	Command to read the low alarm status
Ci	The channel to be read, zero based

Response:

Valid command: **!AA(data)SOj[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 1F)
(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
Oj	j specifies the digital output port

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: @01RLC0 Response:
!01-03.0001O1

Reads the low alarm status of channel 0 and returns a response indicating that the high alarm limit is -03.000, the type is momentary and channel 1 digital output is low alarm output.

Related Commands:

Section 2.59 @AALO(data)CiTOj , Section 2.57 @AADLCi

2.69. @AAROOj

Description:
This command is used to read the alarms associated with a digital output port.

Syntax:
@AARAOj[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to 1F)
RO	Command to read the alarms associated with a digital output port.
Oj	j specifies the digital output port.

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 1F)
HH	A two-digit hexadecimal value to represent the high alarms associated with the digital output port, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, it denotes that the high alarm of the channel is set. When the bit is 0, it denotes that the high alarm of the channel is disabled.
LL	A two-digit hexadecimal value to represent the low alarms associated with the digital output port, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, it denotes that the low alarm of the channel is set. When the bit is 0, it denotes that the low alarm of the channel is disabled.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: @01ROO1 Response: !011122

Reads the alarms associated with the DO1 of module 01. The module responds with 1122 meaning that the high alarms on channels 0 and 4 and the low alarms on channels 1 and 5 are associated with the DO1.

Related Commands:

Section 2.61 @AARAOj

3.Modbus RTU Protocol

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

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

Error Responses

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

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

3.1. 02 (0x02) Read Input Status

This function code is used to read the wire opening status of a module. (Supports types 0x7 and 0x1A only)

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02 ~ 03	Starting channel	2 Bytes	0x80 to 0x83, where 0x80 corresponds to channel 0, 0x81 corresponds to channel 1, etc.
04 ~ 05	Number of input channels	2 Bytes	N, 1 to 8; (Starting channel + N)

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02	Byte count	1 Byte	1
03	Input channel data	1 Byte	A bit corresponds to a channel. When the bit is 1, it denotes that the channel is either over-range or under-range. If the bit is 0 it denotes that the channel is normal.

Error Response

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

3.2. 04 (0x04) Read Input Channels

This function code is used to read from contiguous analog input channels.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x04
02 ~ 03	Starting channel	2 Bytes	0 to 3
04 ~ 05	Number of input channels (N)	2 Bytes	1 to 4

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x04
02	Byte count	1 Byte	2 x N
03 ~	Input channel data	2 x N Bytes	Data in 2’s complement hex format or engineering format.

Error Response

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

3.3. 70 (0x46) Read/Write Module Settings

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

Sub-function Code	Description	Section
00 (0x00)	Reads the module name	3.3.1
07 (0x07)	Reads the type code	3.3.2
08 (0x08)	Sets the type code	3.3.3
32 (0x20)	Reads the firmware version	3.3.4
37 (0x25)	Reads the channel enabled/disabled status of a channel	3.3.5
38 (0x26)	Sets the channel to enabled/disabled	3.3.6
41 (0x29)	Reads the miscellaneous settings	3.3.7
42 (0x2A)	Writes the miscellaneous settings	3.3.8

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

Error Response

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

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x00

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x00
03 ~ 06	Module name	4 Bytes	0x5A 0x20 0x26 0x00

Error Response

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

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x07
03	Reserved	1 Bytes	0x00
04	Channel	1 Byte	0x00 ~ 0x03

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x07
03	Type code	1 Byte	Type code, see Section 1.8 for details.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: Reserved bytes should be filled with zero, the channel is out of range for ZB-2026 series, or an incorrect number of bytes were received

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x08
03	Reserved	1 Byte	0x00
04	Channel	1 Byte	0x00 ~ 0x03
05	Type code	1 Byte	Type code, see Section 1.8 for details.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x08
03	Type code	1 Byte	0: OK Others: error

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: The type code is out of range, the channel is out of range for ZB-2026 series, reserved bytes should be filled with zero, or an incorrect number of bytes were received

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x20

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x20
03	Major version	1 Byte	0x00 ~ 0xFF
04	Minor version	1 Byte	0x00 ~ 0xFF
05	Reserved	1 Byte	0x00
06	Build version	1 Byte	0x00 ~ 0xFF

Error Response

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

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x25

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x25
03	Enabled/disabled status	1 Byte	0x00 ~ 0xFF, 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

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

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x26
03	Enabled/disabled settings	1 Byte	0x00 ~ 0x0F, 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

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x26
03	Enabled/disabled settings	1 Byte	0: OK Others: error.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: The enabled/disabled settings are out of range, or an incorrect number of bytes were received

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x29

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x29
03	Miscellaneous settings	1 Byte	Data format, see Section 1.8 for details.

Note: Reserved fields are filled with zeros.

Error Response

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

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

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x2A
03	Miscellaneous settings	1 Byte	Data format, see Section 1.8 for details.

Note: The reserved fields are filled with zeros.

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x2A
03	Miscellaneous settings	1 Byte	0: OK Others: error

Error Response

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

3.4. Address Mappings

The address mappings are as follows:

Address	Description	Attribute
00001 ~ 00002	Digital output	R/W
00065 ~ 00066	Digital input latch high	R/W
00073 ~ 00074	Digital output latch high	R/W
00097 ~ 00098	Digital input latch low	R/W
00105 ~ 00106	Digital output latch low	R/W
00129 ~ 00130	Digital output safe value	R/W
00161 ~ 00162	Digital output power-on value	R/W
00193 ~ 00194	Digital input edge status 0: falling edge 1: rising edge	R/W
00225 ~ 00226	Digital input counter status 0: disable 1: enable	
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
00264	Digital input/output latch high and latch	W

	low, write 1 to clear channels 0 to 1	
00266	Digital input counter status, write 1 to clear channels 0 to 1	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
00271	The filter format 0: Normal 1: Fast	R/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
00278	Digital input active 0: Normal 1: Inverse	R/W
00279	Digital output active 0: Normal 1: Inverse	R/W
00280	The high latch of channels 0 to 3, write 1 to clear all	W
00281	The low latch of channels 0 to 3, write 1 to clear all	W
00284	Enable/disable calibration 0: disable 1: enable	R/W
00513 ~ 00516	The high latch of channels 0 to 3, write 1 to clear	W
00545 ~ 00548	The low latch of channels 0 to 3, write 1 to clear	W
00577 ~	Enable/disable the high alarm of channels 0	R/W

00580	to 3 0: disable 1: enable	
00609 ~ 00612	Enable/disable the low alarm of channels 0 to 3 0: disable 1: enable	R/W
00641 ~ 00644	The high alarm mode of channels 0 to 3 0: momentary 1: latch	R/W
00673 ~ 00676	The low alarm mode of channels 0 to 3 0: momentary 1: latch	R/W
00705 ~ 00708	The high alarm status of channels 0 to 3	R/W
00737 ~ 00740	The low alarm status of channels 0 to 3	R/W
00769 ~ 00770	Digital input counter of channels 0 to 1, write 1 to clear	W
10033 ~ 10034	Digital input status of channels 0 to 1	R
30001 ~ 30004	The analog input value of channels 0 to 3	R
30065 ~ 30066	The analog output current readback	R
30129 ~ 30130	The digital input counter of channels 0 to 1	R
30513 ~ 30516	The high latch value of channels 0 to 3	R
30545 ~ 30548	The low latch value of channels 0 to 3	R
30705 ~ 30706	The current alarms associated with a digital output port	R
30737 ~ 30738	The alarms associated with a digital output port	R
40033 ~	The analog output of channels 0 to 1	R/W

40034		
40097 ~ 40098	The analog output safe value of channels 0 to 1	R/W
40193 ~ 40194	The analog output power-on value of channels 0 to 1	R/W
40257 ~ 40260	The type code of channels 0 to 3	R/W
40289 ~ 40290	The slew rate of channel 0 to 1	R/W
40417 ~ 40418	The analog output type code of channels 0 to 1	R/W
40481 ~ 40482	The firmware version	R
40483 ~ 40484	The module name	R
40485	The module address, valid range: 0x1 ~ 0x1F	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
40673 ~ 40674	Trim the alanog output of channels 0 to 1	W
40769 ~ 40772	The analog input calibration 0x5A45: zxero 0x5350: span	W
40801 ~ 40802	The analog output calibration 0x5A45: zero 0x5350: span	W

Notes:
1. The command to load factory calibration parameters takes about 3 seconds to be processed. The next command should not be sent before this time has elapsed.

3.5. Engineering Data Format Table

The Modbus protocol supports engineering data format, and the table is as follows.

Type Code	Analog Input Type	-F.S.	+F.S.
07	+4 to +20 mA	4000	20000
08*1	-10 to +10 V	-10000	10000
09*1	-5 to +5 V	-5000	5000
0A*1	-1 to +1 V	-10000	10000
0B*1	-500 to +500 mV	-5000	5000
0C*1	-150 to +150 mV	-15000	15000
0D	-20 to +20 mA	-20000	20000
1A	0 to +20 mA	0	20000

The under range value is –32768 and the over range value is +32767. Refer to Section 1.8 for details regarding hex data format.

4. Troubleshooting

If you are having difficulty using the ZB-2026 series modules, here are some suggestions that may help. If you cannot find the answers you need in these guides, contact ICP DAS Product Support. Contact information is located in Section 1.10.

4.1. Communicating with the module

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

- Ensure that the supplied power is within the range of +10 to +30 V DC. If the supplied power is correct, then the power LED should be on.
- When the module receives a command, the power LED will be set to "off". The power LED will again be shown as "on" after the module responds. This method can be used to check whether the module has received a command sent from the host.
- If possible, use another device that is known to be functional to check whether the host can communicate with the device through the same ZigBee network.
- If the host is a PC with a Windows operating system installed, then execute the DCON Utility to determine whether the module can be found. The DCON Utility can be downloaded from the ICP DAS website http://www.icpdas.com. The DCON Utility documentation can be found in the "G**etting Started For I-7000 Series Modules**" manual.
- Set the module to "INIT mode" and attempt to communicate with the module using the following settings: address 00 and DCON protocol. See Section 1.7 for details.

4.2. Reading Data

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

- Ensure that the type code and data format settings are correct. The type code is set by using the $AA7CiRrr command, see Section 2.19 for details. The data format is set by using the %AANNTTCCFF command. For the Modbus RTU protocol, the type code is set by using sub-function 08h of the function 46h.
- If the voltage read by the module is incorrect, then it may be because the calibration parameters stored in the non-volatile memory are corrupted. You can calibrate the module by yourself, but be sure to read Section 1.8 for details before performing any calibration. Use the $AAS1 command to reload the factory calibration parameters, see Section 2.32 for details.

5. Appendix

5.1. 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 module Watchdog reset 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.

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