JDAM-9018

8-channel Analog/
Thermocouple
Input Module

USER'S MANUAL (V1.0)

健昇科技股份有限公司

JS AUTOMATION CORP.

新北市汐止區中興路 100 號 6 樓
6F., No.100, Zhongxing Rd.,
Xizhi Dist., New Taipei City, Taiwan
TEL：+886-2-2647-6936
FAX：+886-2-2647-6940
http://www.automation.com.tw
http://www.automation-js.com/
E-mail：control.cards@automation.com.tw

Contents

1. Introduction..3
  1.1 Overview..3
  1.2 Module Compatibility..3
  1.3 Communication and Programming..3
  1.4 Software Configuration and Calibration..3
  1.5 Watchdog Timer..3
  1.6 Power Requirements..3
2. About the JDAM Analog Modules...4
  2.1 Outline of JDAM Analog modules..4
  2.2 Module Dimension..4
  2.3 JDAM Analog modules..5
  2.4 Block diagram of modules..6
  2.5 Wire connection..6
    2.5.1 JDAM-9018 wire connection...6
    2.5.2 JDAM9018 pin assignments..7
3. Installation..8
  3.1 Set up an JDAM network..8
  3.2 Host computer..8
  3.3 Power supply...9
  3.4 Communication Wiring..9
  3.5 JDAM Utility Software...9
  3.6 JDAM Isolated RS-232/RS485 Converter..9
  3.7 Initializing a Brand-New Module...10
  3.8 Initialization Procedure..10
  3.9 Install a New JDAM to a Existing Network..11
4. ASCII Command Set..12
  4.1 Introduction...12
  4.2 Format of JDAM Commands..12
  4.3 Calculate Checksum:...13
  4.4 Response of Commands..13
  4.5 Summary of Command Set..13
  4.6 Host Watchdog Command Sets..14
  4.7 General Command Sets..14
  4.8 Configuration Tables...15
    4.8.1 Baud rate setting (CC)...15
    4.8.2 Analog Input Type Setting (TT)...15
    4.8.3 Data Format Setting (FF)..15
    4.8.4 Analog Input Type and Data Format Table.......................................16
  4.9 Command description..17

4.9.1 %AANNTTCCFF Set Module Configuration 17
4.9.2 #AA Read analog data 18
4.9.3 #AAN Read analog input from channel N 19
4.9.4 $AA0 Span calibration 19
4.9.5 $AA1 Offset calibration 20
4.9.6 $AA2 Read configuration status 20
4.9.7 $AA5VV Enable/disable channels for multiplexing 21
4.9.8 $AA6 Read channel status 22
4.9.9 $AA7CiRrr Set channel type individually 23
4.9.10 $AA8Ci Read individual channel type 23
4.9.11 $AA9 Read CJC Offset 24
4.9.12 $AA9snnnn Set CJC Offset Value 24
4.9.13 $AAB Read channel burnout status 25
4.9.14 ~AACe Enable/disable cold junction compensation (CJC) 25
4.9.15 $AAF Read firmware version 26
4.9.16 $AAM Read module name 26
4.9.17 ~AAEV Enable/disable module calibration 27
4.9.18 ~AAO(name) Set module name 27
4.9.19 ~AA0 Read the host watchdog status 28
4.9.20 ~AA1 Reset the host watchdog time out status 28
4.9.21 ~AA2 Read the host watchdog time out value 29
4.9.22 ~AA3EVV Enables/disables the host watchdog 30
4.9.23 ~** Sends a “Host OK” 30
5. MODBUS RTU Command structure 31
5.1 MODBUS Function code introductions 31
5.2 MODBUS Address Mapping 32
5.2.1 Discrete coil address 32
5.2.2 Discrete input address 33
5.2.3 Input register address 34
5.2.4 Holding register address 35
5.3 MODBUS Engineering Data Format Table 36
5.4 MODBUS Hex 2’s complement Data Format Table 37
6. Calibration for JDAM9018 38

1. Introduction

1.1 Overview

The JDAM-*9000* analog modules is a set of intelligent sensor to computer interface modules containing built-in microprocessor. They provide data comparison, and digital communication functions. Some modules provide analog I/O lines for controlling and monitoring analog signals.

1.2 Module Compatibility

The *JDAM-9000* series are fully compatible to Advantech® ADAM-4000 series, ADlink® NµDAM-6000 series and ICP® I-7000 series by Command "~AA2X01V"

1.3 Communication and Programming

JDAM modules can connect to and communicate with all computers and terminals. They use RS-485 transmission standards, and communicate with ASCII format commands. All communications to and from the module are performed in ASCII, which means that JDAM modules can be programmed in virtually any high-level language.

Up to 256 JDAM modules may be connected to an RS-485 multi-drop network by using the JDAM RS-485 repeater, extending the maximum communication distance to 4,000 ft.

1.4 Software Configuration and Calibration

JDAM modules contain no pots or switches to set. By merely issuing a command from the host computer, you can change an analog input module to accept several ranges of voltage input.
Remote configuration can be done by using the command set's configuration and calibration commands. By storing configuration and calibration parameters in a nonvolatile EEPROM, modules are able to retain these parameters in case of power failure.

1.5 Watchdog Timer

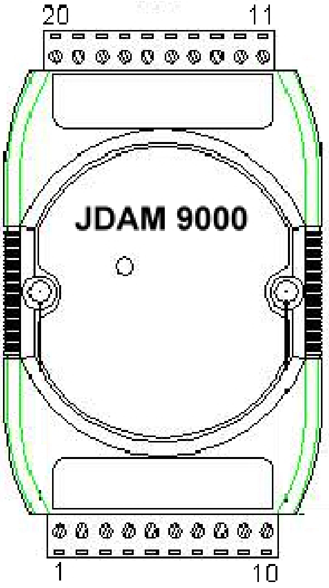
A watchdog timer supervisory function will automatically reset the JDAM modules in the event of system failure. Maintenance is thus simplified.

1.6 Power Requirements

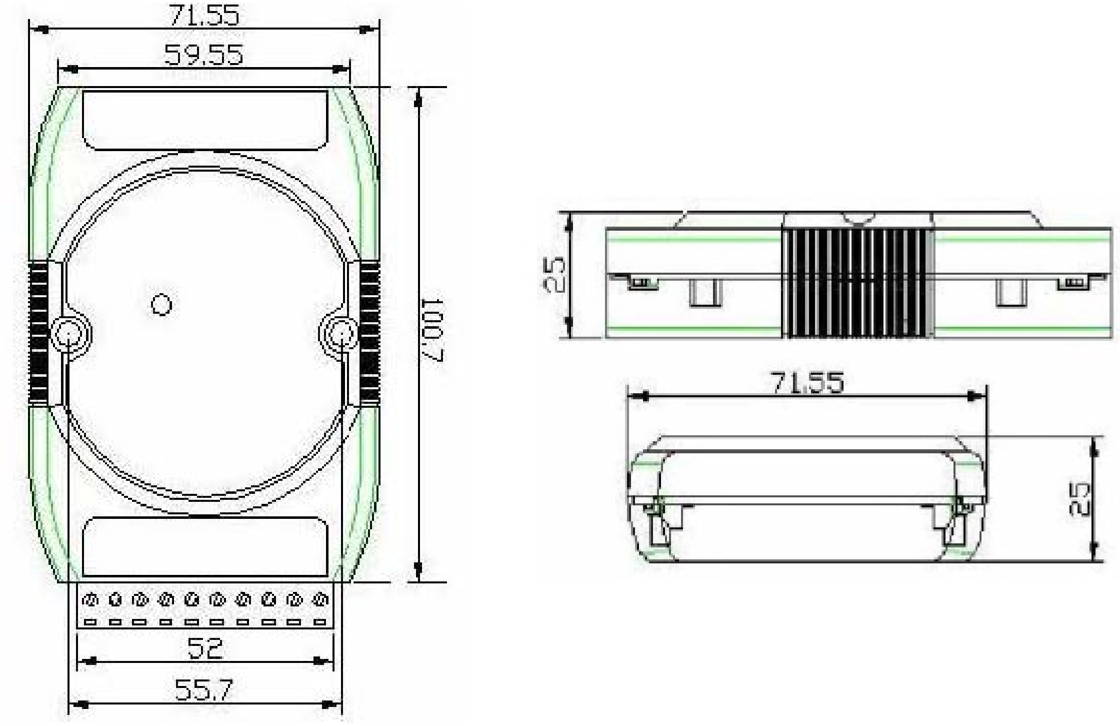
Although the modules are designed for standard industrial unregulated 24Vdc power supply , they accept any power unit that supplies power within the range of +10 to +30Vdc . The power supply ripple must be limited to 5V peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30Vdc .

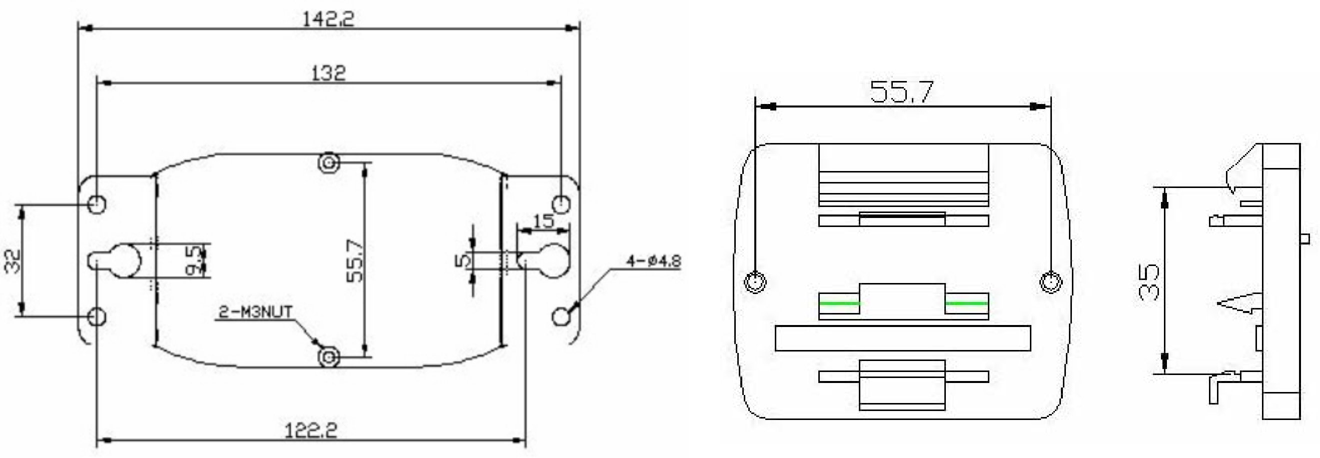
2. About the JDAM Analog Modules

2.1 Outline of JDAM Analog modules



2.2 Module Dimension



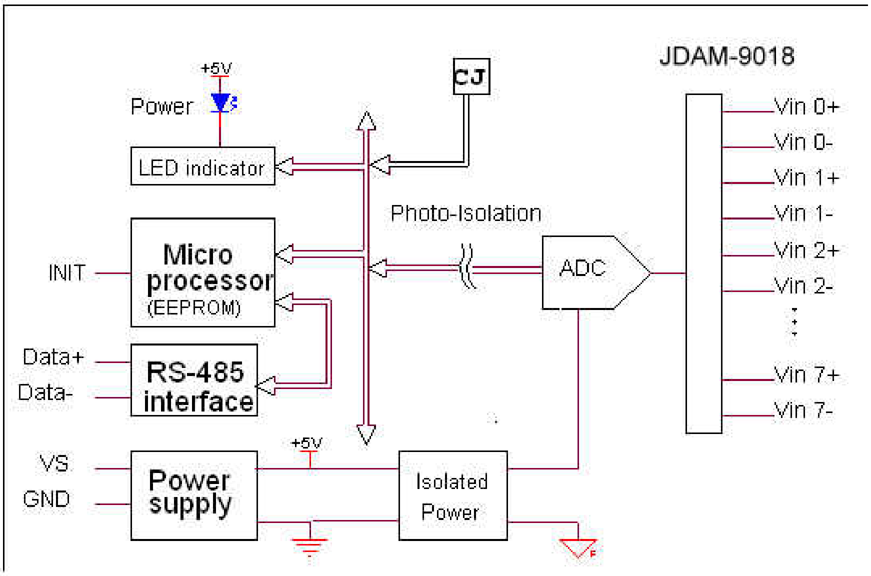


2.3 JDAM Analog modules

JDAM-9018 is a thermocouple input module with 8 input channels. Six of the eight channels are differential type and the other two are single ended type.

- Interface: RS-485, 2 wires
- Speed (bps): 1200, 2400, 4800, 9600, 19.2K, 38.4K , 115.2K
- Analog Input type: Differential input
- Analog Channels Numbers: 8
- Analog Resolution: 16 bits
- Unit Conversion: Thermocouple, mV, V or mA
- Thermocouple Type: J, K, T, E, R, S, B, N
- Sampling Rate :10 Samples/Second
- Bandwidth : 15.7 Hz
- Accuracy : ±0.1%
- Zero Drift : 0.5μV/°C
- Span Drift : 25ppm/°C
- CMR@50/60Hz : 150dB
- NMR@50/60Hz : 100dB
- Input Impedance : 20M Ohms
- Voltage Range: ±2.5V, ±1V,±500mV,±100mV, ±50mV, ±15mV
- Current Measurement: ±20mA (with external 125 ohms resistor)
- Power supply: +10V to +30V

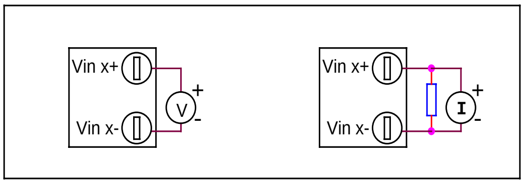
2.4 Block diagram of modules



2.5 Wire connection

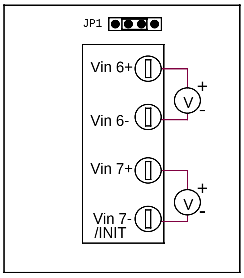
2.5.1 JDAM-9018 wire connection

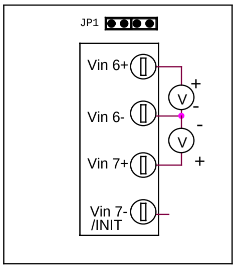
Differential analog input for channel 0 to channel 7



Analog input mode for channel 6 can channel 7 can be selected by setting JP1 on the board

Differential input channel 6 and 7 Single-ended input channel 6 and 7





2.5.2 JDAM9018 pin assignments

pin	name	description
1	Vin5+	Differential positive input channel 5
2	Vin5-	Differential negative input channel 5
3	Vin6+	Differential/single-ended input channel 6
4	Vin6-/AGND*	Differential negative ground of channel 6 or AGND for single-ended input channel 6 & 7
5	Vin7+	Differential/single-ended input channel 7
6	Vin7-/INIT**	Differential negative ground of channel 7 or Initial state setting
7	DATA+	signal, positive
8	DATA-	signal, negative
9	+VS	+10V ~ +30Vdc
10	GND	Ground
11	Vin0+	Differential positive input channel 0
12	Vin0-	Differential negative input channel 0
13	Vin1+	Differential positive input channel 1
14	Vin1-	Differential negative input channel 1
15	Vin2+	Differential positive input channel 2
16	Vin2-	Differential negative input channel 2
17	Vin3+	Differential positive input channel 3
18	Vin3-	Differential negative input channel 3
19	Vin4+	Differential positive input channel 4
20	Vin4-	Differential negative input channel 4

* Negative input of channel 6 or common AGND of channel 6 and 7 depended on JP1 setting (see page 6)

** Negative input of channel 7 or INIT (Initial state setting) pin

3. Installation

This chapter provides guidelines to what is needed to set up and install an JDAM network. A quick hookup scheme is provided that lets you configure modules before they are installed in a network.

To help you to connect JDAM modules with sensor inputs, several wiring examples are provided. Finally, you will find at the end of this chapter a programming example using the JDAM command set.

Be sure to carefully plan the layout and configuration of your network before you start. Guidelines regarding layout are given in Appendix E: RS-485 Network.

NOTICE: Except for changing JDAM to other compatible modules, which have on-board switches for their baud rate setting, JDAM modules should not be opened. There is no need to open the JDAM modules: all configuration is done remotely and there are no user serviceable parts are inside. Opening the cover will therefore void the warranty.

3.1 Set up an JDAM network

The following list gives an overview of what is needed to setup, install and configure an JDAM environment.

- A host computer that can output ASCII characters with an RS-232C or RS-485 port.
- Power supply for the JDAM modules (+10 to +30Vdc)
- JDAM Series Utility software

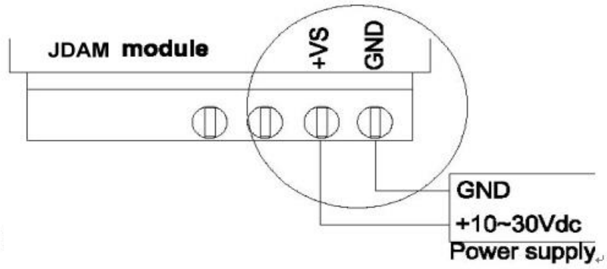
3.2 Host computer

Any computer or terminal that can output in ASCII format over either RS-232 or RS-485 can be connected as the host computer. When only RS-232 is available, an JDAM-9520 module (RS-232/RS-485 converter) is required to transform the host signals to the correct RS-485 protocol. The converter also provides opto-isolation and transformer-based isolation to protect your equipment.

For the ease of use in industrial environments the JDAM modules are designed to accept industry standard +24Vdc unregulated power. Operation is guaranteed when using any power supply between +10 and +30Vdc. Power ripples must be limited to 5 V peak to peak while the voltage in all cases must be maintained between +10 and +30Vdc . All power supply specifications are referenced at module connector. When modules are powered remotely, the effects of line voltage drops must be considered.

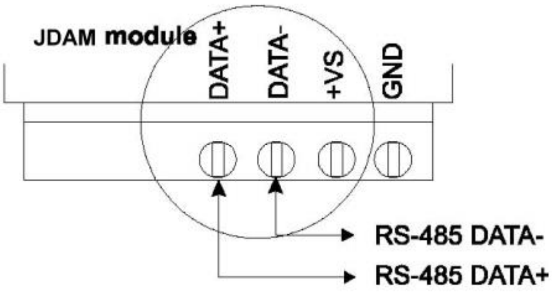
3.3 Power supply

All modules use on-board switching regulators to sustain good efficiency over the 10-30V input range, therefore we can assume that the actual current draw is inversely proportional to the line voltage. The following example shows how to calculate the required current that a power supply should be able to provide.



3.4 Communication Wiring

We recommend that shielded-twisted-pair cables that comply with the EIA RS-485 standard be used with the JDAM network to reduce interference.



3.5 JDAM Utility Software

A menu-driven utility program called “DOSJDAM.EXE” for DOS or “WINJDAM.EXE for Windows is provided for JDAM module configuration, monitoring and calibration. It also includes a terminal emulation program that lets you easily communicate through the JDAM command set

3.6 JDAM Isolated RS-232/RS485 Converter

When the host computer or terminal has only a RS-232 port, an JDAM-9520 Isolated RS-232/RS-485/422 converter connected to the host’s RS-232 port is required.

This module equips a “Auto baud rate detector” inside, therefore it can detect the baud rate and data format automatically and control the direction of RS-485 precisely

3.7 Initializing a Brand-New Module

All JDAM modules in a RS-485 network must have an unique address ID. Therefore, to configure the brand-new 9012/D, 9014/D, 9018 before using is necessary

- Factory default settings:
  - Address ID is 01
  - Baud rate is 9600 bps, check-sum disable
  - Analog input type: Type 08 (±10V)
  - 60Hz filter rejection mode
  - Normal operation mode (for 9012/D, 9018)
  - Six differential and 2 single-ended input mode (for 9018)
- INIT* State settings:

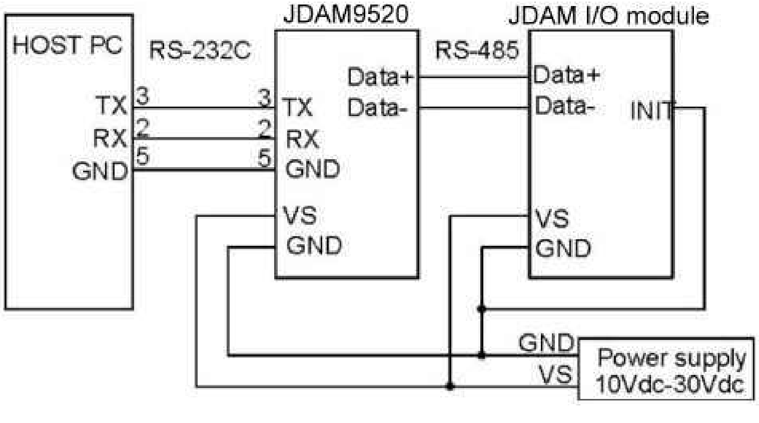
The JDAM I/O modules must be set at *INIT* State* when you want to change the default settings, such as the *ID address, baud rate, check-sum status* etc. All JDAM I/O modules have an special pin labeled as **INIT***. The module will be in *Default State* if the **INIT*** pin is shorted to ground when power ON. Under this state, the default configuration is set as following :

- Address ID is 00
- Baud rate is 9600 bps
- Check-sum disable

Therefore, the communication between host and the module will can be easily set as the same configuration, the initialization of a module will be possible no matter what configuration is set under operating state.

3.8 Initialization Procedure

1. Power off the host computer and the installed JDAM-9520 to COM port of host computer.
2. Connect a brand new JDAM module with the RS-485. Set the module in Default State by shorting the INIT* pin to GND. Refer to Figure 4.1 for detailed wiring.
3. Power on the power supply for JDAM modules.
4. Use the JDAM utility to configure the address ID, baud rate, check-sum status and command sets of the module.



3.9 Install a New JDAM to a Existing Network

1. Equipments for Install a New Module
2. A existing JDAM network
3. New JDAM modules.
4. Power supply (+10 to +30Vdc)
5. Installing Procedures
6. Configure the new JDAM module according to the initialization procedure in section 3.7
7. The baud rate and check-sum status of the new module must be identity with the existing RS-485 network. The address ID must not be conflict with other JDAM modules on the network.
8. Power off the JDAM power supply of the existing RS-485 network.
9. Wire the power lines for the new JDAM with the existing network. Be careful about the signal polarity as wiring.
10. Wire the RS-485 data lines for the new JDAM with the existing network. Be careful about the signal polarity as wiring.
11. Wire to the input or output devices.
12. Power on the JDAM local power supply.
13. Use the JDAM utility to check entire network.

4. ASCII Command Set

4.1 Introduction

The JDAM command is composed by numbers of characteristics, including the leading code, address ID, the variables, the optional check-sum byte, and a carriage return to indicate the end of a command.

The host computer can only command only one JDAM module except those synchronized commands with wildcard address command “#**”. The JDAM may or may not give response to the command. The host should check the response to handshake with the modules.

4.2 Format of JDAM Commands

Syntax: (Leading code)(Addr)(Command)[Data] <Cksum><CR>

Every command begins with a delimiter character. There are five valid characters: a dollar sign $, a pound sign #, a percentage ,a wave sign ’~’ ,sign % and an at sign @.

The delimiter character is followed by a two-character address (hexadecimal) that specifies the target module. The actual two character command follows the address. Depending on the command, an optional data segment follows the command string. An optional two character checksum may be appended to the total string. Every commands is terminated by a carriage return (cr).

Conventions

Leading Code	The first characteristic of the JDAM command, such as %,$,#,~, @, ...etc(1- character)
Addr	Module’s address ID, the value is in the range of 00 – FF (Hex) 2- character
Command	Command codes or value of variables
Data	Data needed by some output command
Checksum	Checksum in brackets indicate optional parameter, only checksum is enable then this field is required (2- character)
<CR>	carriage return(0x0D)

Note:

1. all commands should be issued in ASCII uppercase characters.
   There is no spacing between characters.

4.3 Calculate Checksum:

1. Calculate ASCII sum of all characters of command (or response) string except the character return(cr)
2. Mask the sum of string with 0ffh
3. [Checksum]={(Leading code)+(addr)+(command)+[data]} MOD 0x100

Example:

Command string : $012(cr)

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

The checksum is B7h, and [CHK]="B7"

Command string with checksum=$012B7(cr)

Response string : !01400600(cr)

Sum of string='!'+'0'+'1'+'4'+'0'+'0'+'6'+'0'+'0'

=21h+30h+31h+34h+30h+30h+36h+30h+30h=1ACh

The checksum is ACh, and [CHK]="AC"

Response string with checksum=!01400600AC(cr)

4.4 Response of Commands

The response message depends on JDAM command. The response is also composed with several characteristics, including leading code, variables, and carriage return for ending. There are two kinds of leading code for response message, "!" or ">" means valid command and "?" means invalid. By checking the response message, user can monitor the command is valid or invalid.

But under the following conditions, there will have no response message.

- The specified address ID is not exist.
- Syntax error.
- Communication error
- Some special commands does not have response.

4.5 Summary of Command Set

There are four categories of JDAM commands. The first is the ***JDAM special command***s. The second is the ***general command***s, The third is the ***analog commands***., the forth is the ***digital commands*** and the last is ***linear mapping commands***. All the commands used in the JDAM analog input module are list in the following table.

4.6 Host Watchdog Command Sets

Command	Response	Description	Sec.
~**	no response	Host OK	4.9.23
~AA0	!AASS	Read Module Status	4.9.19
~AA1	!AA	Reset Module Status	4.9.20
~AA2	!AAVV	Read Host watchdog Timeout Value	4.9.21
~AA3EVV	!AA	Set Host Watchdog Timeout Value	4.9.22

4.7 General Command Sets

Command	Response	Description	Sec.
%AANNTTCCFF	!AA	Set Module Configuration	4.9.1
#AA	>(Data)	Read Analog Input	4.9.2
#AAN	>(Data)	Read Analog Input from channel N	4.9.3
$AA0	!AA	Perform Span Calibration	4.9.4
$AA1	!AA	Perform Zero Calibration	4.9.5
$AA2	!AATTCCFF	Read Configuration	4.9.6
$AA5VV	!AA	Set Channel Enable	4.9.7
$AA6	!AAVV	Read Channel Status	4.9.8
$AA7CiRrr	!AA	Set channel type individually	4.9.9
$AA8Ci	!AACiRrr	Read individual channel type	4.9.10
$AA9	!AA(Data)	Read CJC Offset	4.9.11
$AAB	!AA(Data)	Read channel burn out status	4.9.13
$AA9SNNNN	!AA	Set CJC Offset Value	4.9.12
~AACN	!AA	Enable/disable CJC	4.9.14
~AAEV	!AA	Enable/Disable Calibration	4.9.17
~AAO(Data)	!AA	Set Module Name	4.9.18

4.8 Configuration Tables

4.8.1 Baud rate setting (CC)

Code (CC)	03	04	05	06	07	08	09	0A
Baud rate	1200	2400	4800	9600	19200	38400	57600	115200

Note: The data bits are fixed at one start bit, eight data bits, no parity and one stop bit

4.8.2 Analog Input Type Setting (TT)

Input Range Type(Hex)	Input Range
00	No change
08	± 10 V
09	± 5 V
0A	± 1 V
0B	± 500 mV
0C	± 150 mV
0D	± 20 mA

4.8.3 Data Format Setting (FF)

7	6	5	4	3	2	1	0
FS	CS	reserved				DF	

Key	**Description**
DF	Data format 00: Engineer unit 01: % of FSR (full scale range) 10: 2’s complement hexadecimal
CS	Check sum 0: disabled 1: enabled
FS	Filter settings 0: 60 Hz rejection 1:50 Hz rejection

4.8.4 Analog Input Type and Data Format Table

Code	Range	Format	+F.S.	zero	-F.S
00	-15~+15mV	Engineer unit	+15.000	+00.000	-15.000
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
01	-50~+50mV	Engineer unit	+50.000	+00.000	-50.000
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
02	-100~+100mV	Engineer unit	+100.00	+000.00	-100.00
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
03	-500~+500mV	Engineer unit	+500.00	+000.00	-500.00
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
04	-1~+1V	Engineer unit	+1.0000	+0.0000	-1.0000
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
05	-2.5~+2.5V	Engineer unit	+2.5000	+0.0000	-2.5000
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
06	-20~+20mA	Engineer unit	+20.000	+00.000	-20.000
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	8000
OE	Type J T/C -210~760°C	Engineer unit	+760.00	+00.000	-210.00
		% of F.S.R	+100.00	+000.00	-027.63
		2’s complement	7FFF	0000	DCA2
0F	Type K T/C -270~1372°C	Engineer unit	+1372.0	+00.000	-0270.0
		% of F.S.R	+100.00	+000.00	-019.68
		2’s complement	7FFF	0000	E6D0
10	Type T T/C -270~400°C	Engineer unit	+400.00	+00.000	-270.00
		%of F.S.R	+100.00	+000.00	-067.50
		2’s complement	7FFF	0000	DCA2
11	Type E T/C -270~1000°C	Engineer unit	+1000.0	+0000.0	-0270.0
		% of F.S.R	+100.00	+000.00	-027.00
		2’s complement	7FFF	0000	DD71
12	Type R T/C 0~1768°C	Engineer unit	+1768.0	+00.000	-0000.0
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	0000
13	Type S T/C 0~1768°C	Engineer unit	+1768.0	+00.000	-0000.0
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	0000
14	Type B T/C 0~1820°C	Engineer unit	+1820.0	+00.000	-0000.0
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	0000
15	Type N T/C -270~1300°C	Engineer unit	+1300.0	+0000.0	-0270.0
		% of F.S.R	+100.00	+000.00	-020.77
		2’s complement	7FFF	0000	E56B
		% of F.S.R	+100.00	+000.00	-100.00
		2’s complement	7FFF	0000	F54D

4.9 Command description

4.9.1 %AANNTTCCFF Set Module Configuration

Description set the configuration of the module at address AA.

Syntax **%AANNTTCCFF (cr)**

% is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

NN new module address (00~FF)

TT represents the type code. Type code determines the input range. If TT=FF the type of all channels keep no change

CC represents the baud rate code.

FF is a hexadecimal number that equals the 8-bit parameter that represents the data format, checksum status and integration time. Bits 2 to 5 are not used, and are set to 0.

(cr) is the terminating character, carriage return (0Dh).

(***Please refer to Sec-4.8 to see TT, CC and FF parameter definition***)

Response **!AA (cr)** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

(cr) is the terminating character, carriage return (0Dh).

Note:), if TT >0,this command will set all channels to have the same type code (TT)

(Please refer to $AA7CiRii command in Sec. 4.9.9)

Example Command: **%0203080602(cr)**

Response:**!02(cr)**

Result: new module address=03

Analog input type code=08(-10V~+10V) for all channels

baud rate=06 (9600)

data format =02 (2's complement hexadecimal)

4.9.2 #AA Read analog data

Description The command will return the input value from a specified (AA) module in the currently configured data format.

Syntax **#AA(cr)**

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

(cr) is the terminating character, carriage return (0Dh).

Response **>(data)(cr)** if the command is valid or **?AA (cr)** if the command is invalid

There is no response if the module detects a syntax error or communication error.

> is a delimiter character.

(data) is the input value in the configured data format of the module.

(cr) is the terminating character, carriage return (0Dh).

Example Command: **#21(cr)**

Response: **>+7.2111+7.2567+7.3125+7.1000+7.4712+7.2555+7.1234+7.5678(cr)**

The command response the analog input module at address 21h for its input values of all channels.

The analog input module responds with channels from 0 to 7 with +7.2111 volts, +7.2567 volts, +7.3125 volts, +7.1000 volts, +7.4712 volts, +7.2555 volts, +7.1234 volts and +7.5678 volts.

Example Command: **#DE(cr)**

Response: **>FF5DE4323212AE33233345663E000FF03(cr)**

The analog input module at address DEh has an input value of **FF5DE4323212AE33233345663E000FF03**. (The configured data format of the analog input module is two’s complement)

4.9.3 #AAN Read analog input from channel N

Description The command will return the input value from one of the eight channels of a specified (AA) module in the currently configured data format.

Syntax **#AAN(cr)**

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.

N identifies the channel you want to read. The value can range from 0 to 7

(cr) is the terminating character, carriage return (0Dh).

Response **(data)(cr)** if the command is valid or **?AA (cr)** if the command is invalid

There is no response if the module detects a syntax error or communication error.

> is a delimiter character.

(data) is the input value of the channel number N. Data consists of a + or - sign followed by five decimal digits with a fixed decimal point.

(cr) is the terminating character, carriage return (0Dh).

Example Command: **#120(cr)**

Response: **>+1.4567(cr)**

The command requests the analog input module at address 12h to return the input value of channel 0.

The analog input module responds that the input value of channel 0 is equal to +1.4567 volts.

4.9.4 $AA0 Span calibration

Description Calibrates an analog input module to correct for gain errors.

Syntax **$AA0(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the module which is to be calibrated.

0 is the Span Calibration command.

(cr) is the terminating character, carriage return (0Dh).

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the module.

(cr) represents terminating character, carriage return (0Dh).

In order to successfully calibrate an analog input module’s input range, a proper calibration input signal should be connected to the analog input module before and during the calibration.

4.9.5 $AA1 Offset calibration

Description Calibrates an analog input module to correct for offset errors.

Syntax **$AA1(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the module you want to calibrate.

1 is the Offset Calibration command.

(cr) is the terminating character, carriage return (0Dh).

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the module.

(cr) represents terminating character, carriage return (0Dh).

4.9.6 $AA2 Read configuration status

Description Requests the return of the configuration data from the module at address AA.

Syntax **$AA2(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

2 is the Configuration Status command.

(cr) is the terminating character, carriage return (0Dh).

Response **!AATTCCFF(cr)** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

TT the type of channel 0 **(Please refer to $AA8Ci command in Sec.** 4.9.10**)**.

CC represents the baud rate code.

FF is a hexadecimal number that equals the 8-bit parameter that represents the data format, checksum status and integration time.Bits 2 to 5 are not used, and are set to 0.

(cr) is the terminating character, carriage return (0Dh).

(Please refer to Sec-4.8 to see TT, CC and FF parameter definition)

Example Command: **$452(cr)**

Response: **!45050600(cr)**

The command asks the analog input module at address 45h to send its configuration data.The analog input module at address 45h responds with an input range of 2.5 volts, a baud rate of 9600 bps, an integration time of 50 ms (60 Hz), engineering units are the currently configured data format, and no checksum function or checksum generation.

4.9.7 $AA5VV Enable/disable channels for multiplexing

Description Enables/disables multiplexing simultaneously for separate channels of a specified input module.

Syntax **$AA5VV(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

5 is the Enable/disable Channels command.

VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4~7, and the second word represents the status of channel 0~3. Value 0 means the channel is disabled, value 1 means the channel is enabled.

(cr) is the terminating character, carriage return (0Dh).

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid.

ere is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(cr) is the terminating character, carriage return (0Dh).

Example Command: **$00581(cr)**

sponse: **!00(cr)**

exadecimal 8 equals binary 1000, which enables channel 7 and disables channels 4, 5, and 6. Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channel 1, 2, and 3.

4.9.8 $AA6 Read channel status

Description Asks a specified input module to return the status of all channels.

Syntax **$AA6(cr)**

AA (range 00-FF) represents the 2-character hexadecimal address of the module of which the channel status you want to send. The channel status defines whether a channel is enabled or disabled.

6 is the Read Channel Status command.

(cr) is the terminating character, carriage return (0Dh).

Response **!AAVV(cr)** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4-7, and the second word represents the status of channel 0-3. Value 0 means the channel is disabled, value 1 means the channel is enabled.

(cr) is the terminating character, carriage return (0Dh).

Example Command: **$026(cr)**

Response: **!02FF(cr)**

The command asks the analog input module at address 02 to send the status of it input channels.

The analog input module at address 02 responds that all its multiplex channels are enabled (FF equals 1111 and 1111).

4.9.9 $AA7CiRrr Set channel type individually

Description The command set channel type individually.

Syntax **$AA7CiRrr**(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

7C is the Set channel type command.

i channel number

rr channel type code

(cr) is the terminating character, carriage return (ODh).

Response **!AA** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(cr) is the terminating character, carriage return

Examples Command: **$017C3R08(cr)**

Response: **!01(cr)**

Set type code 08 (+/-10V) to channel 3.

$AA8Ci Read individual channel type

Description The command read individual channel type.

Syntax **$AA8Ci** (cr)

4.9.10 $AA8Ci Read individual channel type

Description The command read individual channel type.

Syntax **$AA8Ci (cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

8C is the read channel type command.

i channel number

(cr) is the terminating character, carriage return (ODh).

Response **!AACiRrr** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

i channel number(0~7)

rr type of channel **i**

(cr) is the terminating character, carriage return

Examples Command: **$018C3**(cr)

Response: **!01C3R08(cr)**

The type code of channel 3 is 08 (+/-10V).

4.9.11 $AA9 Read CJC Offset

Description The command read cold junction offset.

Syntax **$AA9(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

9 is the read cold junction offset command.

(cr) is the terminating character, carriage return (ODh).

Response **!AAsnnnn** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

s sign of cold junction offset

nnnn cold junction offset in 0.01C unit

(cr) is the terminating character, carriage return

Examples Command: **$019(cr)**

Response: **!01+0010(cr)**

The cold junction offset is +0010(Hex)*0.01=+0.16C.

4.9.12 $AA9snnnn Set CJC Offset Value

Description The command set cold junction offset.

Syntax **$AA9snnnn(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

9 is the set cold junction offset command.

S ign of cold junction offset

nnnn cold junction offset (Hex) in 0.01C unit

(cr) is the terminating character, carriage return (ODh).

Response **!AA** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(cr) is the terminating character, carriage return

Examples Command: **$019+0010(cr)**

Response: **!01(cr)**

Set cold junction offset to +0010(Hex)*0.01=+0.16C.

4.9.13 $AAB Read channel burnout status

Description Read channel burn out status

Syntax **$AAB(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

B is the Channel Diagnose command.

(cr) is the terminating character, carriage return (0Dh).

Response **!AANN(cr)** if the command is valid when it applied with.

?AA(cr) if an invalid command was issued.

There is no response if the module detects a syntax error or communication error.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the module.

NN (range 00-FF) is a hexadecimal number that equals the 8-bit parameter, representing the status of analog input channels. Bit value 0 means normal status; and bit value 1 means channel open wiring.

(cr) is the terminating character, carriage return (0Dh)

Examples Command: **$01B(cr)**

Response: **!0101(cr)**

Channel 0 is open wiring and channel 1~7 are all normal.

4.9.14 ~AACe Enable/disable cold junction compensation (CJC)

Description The command enable/disable cold junction compensation.

Syntax **$AACe(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

C is the enable/disable CJC command.

e e=0 disable CJC, e=1 enable CJC

(cr) is the terminating character, carriage return (ODh).

Response **!AA** if the command is valid or **?AA (cr)** if the command is invalid.

There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(cr) is the terminating character, carriage return

Examples Command: **$01C1(cr)**

Response: **!01(cr)**

Enable burnout detection for all channels

4.9.15 $AAF Read firmware version

Description The command requests the module at address AA to return the version code of its firmware.

Syntax **$AAF(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

F is the Read Firmware Version command.

(cr) is the terminating character, carriage return (ODh).

Response **!AA(Version)(cr)** if the command is valid or **?AA (cr)** if the command is invalid. There is no response if the module detects a syntax error or communication error.

! is a delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(Version) is the version code of the module's firmware at address AA.

(cr) is the terminating character, carriage return

4.9.16 $AAM Read module name

Description Requests the analog output module at address AA to return its name

Syntax **$AAM(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address that you ant to access.

M is the Read Module Name command.

(cr) is the terminating character, carriage return (ODh)

Response **!AA(Module Name)(cr)** if the command is valid or **?AA (cr)** if the command is invalid. There is no response if the module detects a syntax or communication error.

! is a delimiter character indicating that a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of module.

(Module Name) is the name of the module at address AA.

(cr) is the terminating character, carriage return

4.9.17 ~AAEV Enable/disable module calibration

Description Enable/Disable module calibration.

Syntax **~AAEV(cr)**

~ delimiter character

AA address of the module

E command to enable/disable calibration

V 1: enable calibration , 0:disable calibration

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response

? delimiter character for an invalid response

AA address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error

Examples Command: **$010(cr)**

Response: **?01(cr)**

Send the command to perform a span calibration on module 01. It returns an invalid response because the "enable calibration command" was not sent in advance

Command: **~01E1(cr)**

Response: **!01(cr)**

Enable calibration on module 1

Command: **$010(cr)**

Response: **!01(cr)**

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

4.9.18 ~AAO(name) Set module name

Description Set module name

Syntax **~AAO(name)(cr)**

~ delimiter character

AA address of the module to be set (00 to FF)

O command to set the module name (new name of the module (max. 6 characters).

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response

? delimiter character for an invalid response

AA address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error

4.9.19 ~AA0 Read the host watchdog status

Description Reads the host watchdog status of a module.

Syntax **~AA0(cr)**

~ delimiter character

AA address of the module to be read (00 to FF)

0 command to reads the host watchdog status.

Response: **AASS(cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response

? delimiter character for an invalid response

AA address of the responding module (00 to FF)

SS two hexadecimal digits that represent the host watchdog status,

where:

Bit 7: 0 indicates that the host watchdog is disabled and

=1 indicates the host watchdog is enabled,

Bit 2: =1 indicates watchdog time out has occurred.

The host watchdog status is stored in EEPROM and can only be reset using the ~AA1 command.

4.9.20 ~AA1 Reset the host watchdog time out status

Description: Reset the host watchdog time out status of a module.

Syntax: **~AA1(cr)**

~ delimiter character A

AA address of the module to be set (00 to FF)

1 command to reset the host watchdog time out status

Response: **!AA (cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response module w

? delimiter character for an invalid response le (00 to FF)

AA address of the responding module

There will be no response if the command syntax is incorrect, there is a communication error

4.9.21 ~AA2 Read the host watchdog time out value

Description Reads the host watchdog time out value

Syntax **~AA2(cr)**

~ delimiter character

AA address of the module to be read (00 to FF)

2 command to read the host watchdog time out value.

Response **!AAEVV(cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response

? delimiter character for an invalid response

AA address of the responding module

E 1: the host watchdog is enabled

0: the host watchdog is disabled V

VV two hexadecimal digits to represent the value in tenths of a second,

There will be no response if the command syntax is incorrect, there is a communication error

Examples Command: **~012(cr)**

Response: **!011FF(cr)**

Reads the host watchdog time out value of module 01 and returns FF, meaning that the host watchdog is enabled and watchdog timeout value is FF(hex)=255.5 seconds

4.9.22 ~AA3EVV Enables/disables the host watchdog

Description Enables/disables the host watchdog and set the host watchdog time out value of a module.

Syntax **~AA3EVV(cr)**

~ delimiter character

AA address of the module to be read (00 to FF)

3 command to set watchdog time out value

E 1: enable host watchdog

0: disable host watchdog

VV two hexadecimal digits to represent timeout value in tenths of second, for example, 01 means 0.1 seconds and FF means 25.5 seconds

Response **!AA(cr)** if the command is valid or **?AA (cr)** if the command is invalid

! delimiter character for a valid response

? delimiter character for an invalid response

AA address of the responding module

There will be no response if the command syntax is incorrect, there is a communication error

Examples Command**: ~013164(c**r)

Response: **!01(cr)**

Enable host watchdog of module 01 and sets e host watchdog time out value to 10.0 seconds.

4.9.23 ~** Sends a "Host OK"

Description Informs all modules that the host is OK.

Syntax **~**(cr)**

~ delimiter character

****** Host OK command

Response no response

5. MODBUS RTU Command structure

JDAM-9018 system accepts a command/response form with the host computer. When systems are not transmitting they are in listen mode. The host issues a command to a system with a specified address and waits a certain amount of time for the system to respond. If no response arrives, a time-out aborts the sequence and returns control to the host. This chapter explains the structure of the commands with Modbus RTU protocol, and guides to use these command sets to implement user's programs.

5.1 MODBUS Function code introductions

Code (Hex)	**Name**	**Usage**
01	Read discrete coil	Read channel burnout status
02	Read discrete input	Read channel burnout status
03	Read Holding Registers	Read 16-bit register. Used to read integer or floating point process data.
04	Read Input Registers	
05	Write single coil	
06	Preset Single Register	Write data in 16-bit integer format
0F	Write multiple coils	
10	Preset Multiple Registers	Write multiple data in 16-bit integer format

5.2 MODBUS Address Mapping

5.2.1 Discrete coil address

Discrete coil addresses are available for MODBUS function code 0x01, 0x05, 0x0F

Address	**Channel**	**Item**	**Attribute**
00201	0	AD channel active status(*)	R/W
00202	1	AD channel active status(*)	R/W
00203	2	AD channel active status(*)	R/W
00204	3	AD channel active status(*)	R/W
00205	4	AD channel active status(*)	R/W
00206	5	AD channel active status(*)	R/W
00207	6	AD channel active status(*)	R/W
00208	7	AD channel active status(*)	R/W
00269		Set MODBUS data format	R/W (**) 1=Hex 2's format 0=engineering format
00281		Enable/disable Burnout detect	R/W 0=disable,1=enable(***)

(*): active status=1 the channel is active (open wire) , =0 the channel is inactive

(**): 1. MODBUS data format of analog input value is 2's complement format or engineering format

2. Factory default: engineering format

(***) : this address is valid only for JDAM9018 with firmware version M6.3A or later

5.2.2 Discrete input address

Discrete input addresses are available for MODBUS function code 0x02

Address	Channel	Item	Attribute
10201	0	AD channel active status(*)	R
10202	1	AD channel active status(*)	R
10203	2	AD channel active status(*)	R
10204	3	AD channel active status(*)	R
10205	4	AD channel active status(*)	R
10206	5	AD channel active status(*)	R
10207	6	AD channel active status(*)	R
10208	7	AD channel active status(*)	R
10269		Set MODBUS data format	R/W 1=Hex 2's format 0=engineering format
10281		Enable/disable Burnout detect	R 0=disable,1=enable (*)

(*) : this address is valid only for JDAM9018 with firmware version M6.3A or later

5.2.3 Input register address

Input register addresses are available for MODBUS function code 0x04

Address	Channel	Item	Attribute	Memo
30001	0	Analog input Value	R	
30002	1	Analog input Value	R	
30003	2	Analog input Value	R	
30004	3	Analog input Value	R	
30005	4	Analog input Value	R	
30006	5	Analog input Value	R	
30007	6	Analog input Value	R	
30008	7	Analog input Value	R	
30129		Cold junction temperature	R	
30201	0	Input type Code	R	0x00~0x06 ,0x0e~0x16
30202	1	Input type Code	R	0x00~0x06 ,0x0e~0x16
30203	2	Input type Code	R	0x00~0x06 ,0x0e~0x16
30204	3	Input type Code	R	0x00~0x06 ,0x0e~0x16
30205	4	Input type Code	R	0x00~0x06 ,0x0e~0x16
30206	5	Input type Code	R	0x00~0x06 ,0x0e~0x16
30207	6	Input type Code	R	0x00~0x06 ,0x0e~0x16
30208	7	Input type Code	R	0x00~0x06 ,0x0e~0x16
30211		Module Name 1	R	0x9018
30212		Module Name 2	R	0x9000
30213		Version 1	R	
30214		Version 2	R	
30221		Channel Enable	R	0x00~0xFF
30269		Set MODBUS data format	R	0x0001=Hex 2's format 0x0000=engineering format
30271		Read channel 0 CJC offset	R	2's complement in 0.01C increment
30281		Burnout status	R	0x0000~0x00FF (*)
30291	0	Channel 0 CJC offset	R	2's complement in 0.01C increment
30292	1	Channel 1 CJC offset	R	2's complement in 0.01C increment (*)
30292	2	Channel 2 CJC offset	R	2's complement in 0.01C increment (*)
30294	3	Channel 3 CJC offset	R	2's complement in 0.01C increment (*)
30295	4	Channel 4 CJC offset	R	2's complement in 0.01C increment (*)
30296	5	Channel 5 CJC offset	R	2's complement in 0.01C increment (*)
30297	6	Channel 6 CJC offset	R	2's complement in 0.01C increment (*)
30298	7	Channel 7 CJC offset	R	2's complement in 0.01C increment (*)

. (*) : this address is valid only for JDAM9018 with firmware version M6.4A or later

5.2.4 Holding register address

Holding register addresses are available for MODBUS function code 0x03, 0x06, 0x10

Address	Channel	Item	Attribute	Memo
40001	0	Analog input Value	R	
40002	1	Analog input Value	R	
40003	2	Analog input Value	R	
40004	3	Analog input Value	R	
40005	4	Analog input Value	R	
40006	5	Analog input Value	R	
40007	6	Analog input Value	R	
40008	7	Analog input Value	R	
40201	0	Input type Code	R/W	0x00~0x06 ,0x0e~0x16
40202	1	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40203	2	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40204	3	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40205	4	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40206	5	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40207	6	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40208	7	Input type Code	R /W	0x00~0x06 ,0x0e~0x16
40211		Module Name 1	R	0x9018
40212		Module Name 2	R	0x9018
40213		Version 1	R	
40214		Version 2	R	
40221	0~7	Channel Enable	R /W	0x00~0xFF
40269		Set MODBUS data format	R /W	0x0001=Hex 2's format 0x0000=engineering format
40271		Set all channels to have the same CJC offset	R/W	2's complement in 0.01C increment
40281	All channels	Burnout status	R	0x0000~0x00FF (*)
40291	0	Channel 0 CJC offset	R/W	2's complement in 0.01C increment (*)
40292	1	Channel 1 CJC offset	R/W	2's complement in 0.01C increment (*)
40292	2	Channel 2 CJC offset	R/W	2's complement in 0.01C increment (*)
40294	3	Channel 3 CJC offset	R/W	2's complement in 0.01C increment (*)
40295	4	Channel 4 CJC offset	R/W	2's complement in 0.01C increment (*)
40296	5	Channel 5 CJC offset	R/W	2's complement in 0.01C increment (*)
40297	6	Channel 6 CJC offset	R/W	2's complement in 0.01C increment (*)
40298	7	Channel 7 CJC offset	R/W	2's complement in 0.01C increment (*)

(*) : this address is valid only for JDAM9018 with firmware version M6.4A or later

5.3 MODBUS Engineering Data Format Table

Type Code	Input Type	Min.	Max.	Formula
00	-15 mV ~ +15 mV	-15000	15000	Volt=(MODBUS data) /1000 (mV)
01	-50 mV ~ + 50 mV	-5000	5000	Volt=(MODBUS data) /100 (mV)
02	-100 mV ~ +100 mV	-10000	10000	Volt=(MODBUS data) /100 (mV)
03	-500 mV ~ +500 mV	-5000	5000	Volt=(MODBUS data) /10 (mV)
04	-1 V ~ +1 V	-10000	10000	Volt=(MODBUS data) /10000 (V)
05	-2.5 V ~ +2.5 V	-25000	25000	Volt=(MODBUS data) /10000 (V)
06	-20 mA ~ +20 mA	-20000	20000	Volt=(MODBUS data) /1000 (mA)
0E	Type J Thermocouple -210°C to 760°C	-2100	7600	Temp.=(MODBUS data) /10 (°C)
0F	Type K Thermocouple -270°C to 1372°C	-2700	13720	
10	Type T Thermocouple -270°C to 400°C	-2700	4000	
11	Type E Thermocouple -270°C to 1000°C	-2700	10000	
12	Type R Thermocouple 0°C to 1768°C	0	17680	
13	Type S Thermocouple 0°C to 1768°C	0	17680	
14	Type B Thermocouple 00°C to 1820°C	0	18200	
15	Type N Thermocouple -270°C to 1300°C	-270	13000	

Example: Assume type of channel 2 is +/-10V and MODBUS data=0x2030(Hex)=8240(Dec)
The voltage of channel 2 is 8240/1000=8.24V

Example: Assume type of channel 1 is +/-500mV and MODBUS data=0xEF1B(Hex)=-4325(Dec)
The voltage of channel 2 is -4235/10=423.5mV

Example: Assume type of channel 1 is +/-20mA and MODBUS data=0x3B84(Hex)=15236(Dec)
The current of channel 2 is 15236/1000=15.236mA

5.4 MODBUS Hex 2's complement Data Format Table

Type Code	Input Type	Min.	Max.	Formula
00	± 15 mV	8000	7FFF	Volt=(MODBUS data *15)/0x7FFF (mV)
01	± 50 mV	8000	7FFF	Volt=(MODBUS data *50)/0x7FFF (mV)
02	± 100 mV	8000	7FFF	Volt=(MODBUS data *100)/0x7FFF (mV)
03	± 500 mV	8000	7FFF	Volt=(MODBUS data *500)/0x7FFF (mV)
04	± 1 V	8000	7FFF	Volt=(MODBUS data *1)/0x7FFF (V)
05	± 2.5 V	8000	7FFF	Volt=(MODBUS data *2.5)/0x7FFF (V)
06	± 20 mA	8000	7FFF	Volt=(MODBUS data *20)/0x7FFF (mA)
0E	Type J Thermocouple -210°C to 760°C	DCA2	7FFF	Temp.=(MODBUS data*760) /0x7FFF (°C)
0F	Type K Thermocouple -270°C to 1372°C	E6D0	7FFF	Temp.=(MODBUS data*1372) /0x7FFF (°C)
10	Type T Thermocouple -270°C to 400°C	A99A	7FFF	Temp.=(MODBUS data*400) /0x7FFF (°C)
11	Type E Thermocouple -270°C to 1000°C	DD71	7FFF	Temp.=(MODBUS data*1000) /0x7FFF (°C)
12	Type R Thermocouple 0°C to 1768°C	0000	7FFF	Temp.=(MODBUS data*1768) /0x7FFF (°C)
13	Type S Thermocouple 0°C to 1768°C	0000	7FFF	Temp.=(MODBUS data*1768) /0x7FFF (°C)
14	Type B Thermocouple 0°C to 1820°C	0000	7FFF	Temp.=(MODBUS data*1820) /0x7FFF (°C)
15	Type N Thermocouple -270°C to 1300°C	E56B	7FFF	Temp.=(MODBUS data*1300) /0x7FFF (°C)

Example: Assume type of channel 2 is +/-10V and MODBUS data=0x2030(Hex)=8240(Dec)
The voltage of channel 2 is (8240*10)/32767=2.514V

Example: Assume type of channel 1 is +/-500mV and MODBUS data=0xEF1B(Hex)=-4325(Dec)
The voltage of channel 2 is (-4235*500)/32767=-64.622mV

Example: Assume type of channel 1 is +/-20mA and MODBUS data=0x3B84(Hex)=15236(Dec)
The current of channel 2 is (15236*20)/32767=9.299mA

6. Calibration for JDAM9018

The offset calibration is used to calibrate output offset when the input voltage is 0V

- Calibration procedures
  - Apply zero voltage to channel 0 of analog module
  - Issues configuration command with type=00~06
  - Issues enable calibration command
  - Issues zero offset calibration command

Code	Type and range
0	+/-15mV
1	+/-50mV
2	+/-100mV
3	+/-500mV
4	+/-1V
5	+/-2.5V
6	+/-20mA