

Contents

Chapter 1 ASCII Commands for EDAM-9000 Modules	3
1.1 About ASCII Commands	3
1.2 Syntax of ASCII	3
1.3 How to Send ASCII command with TCP/IP protocols.....	4
1.4 ASCII Command Set	6
Chapter 2 ASCII Command Description	10
2.1 \$aaM Read Module Name.....	10
2.2 \$aaF Read Firmware Version	11
2.3 \$aaID Read module ID number	12
2.4 %aaDHCPn Enable or disable DHCP	13
2.5 %aaSETUDPSTddhmmss Set module UDP active-send time interval.....	14
2.6 %aaSETUDPnpppppppp Set UDP Active/Event IP address	15
2.7 %aaSETUDPnR Enable UDP active-send message	16
2.8 %aaSETUDPnP Disable UDP active-send message.....	17
2.9 %aaSETUDPnRE Enable UDP event trigger	18
2.10 %aaSETUDPnPE Disable UDP event trigger	19
2.11 #aan Read Analog Input from Channel N	20
2.12 #aa Read Analog Input from All Channels	21
2.13 \$aa0 Span Calibration	22
2.14 \$aa1 Zero Calibration	23
2.15 \$aa6 Read Channel Enable/Disable Status.....	24
2.16 \$aa5mm Set Channel Enable/Disable Status	25
2.17 #aaMH Read Maximum Value	26
2.18 #aaMHn Read Maximum Value from channel N.....	27
2.19 #aaML Read Minimum Value	28
2.20 #aaMLn Read Minimum Value from channel N.....	29
2.21 \$aaCjAhs Set Alarm Mode.....	30
2.22 \$aaCjAh Read Alarm Mode	31
2.23 \$aaCjAhEs Enable/Disable Alarm	32
2.24 \$aaCjCh Clear Latch Alarm	33
2.25 \$aaCjAhCCn Set Alarm Connection	34
2.26 \$aaCjRhC Read Alarm Connection.....	35
2.27 \$aaCjAhU Set Alarm Limit	36
2.28 \$aaCjRhU Read Alarm Limit	37
2.29 \$aaCjS Read Alarm Status	38
2.30 \$aa3 Read cold junction temperature	39
2.31 \$aa9hhhhh Set CJ offset.....	40

2.32	\$aa9	Read CJ offset	41
2.33	\$aaAnntt	Set analog input type(range).....	42
2.34	\$aaAff	Set conversion rate of analog input module.....	43
2.35	\$aaBhh	Read analog input type	44
2.36	\$aa6	Read DI /DO Channel Status	46
2.37	\$aa7	Read DI latch status	47
2.38	#aa00dd	Write All Digital Output.....	48
2.39	#aa1n0d	Set Single Digital Output Channel	49
2.40	#aa2npppppppp	Write DO pulse counts	50

Chapter 1 ASCII Commands for EDAM-9000 Modules

1.1 About ASCII Commands

For users do not familiar to Modbus protocol, Inlog offers a function library as a protocol translator, integrating ASCII command into Modbus/TCP structure. Therefore, users familiar to ASCII command can access EDAM-9000 easily. Before explaining the structure of ASCII command packed with Modbus/TCP format. Lets see how to use an ASCII command and how many are available for your program.

EDAM9000 series also integrate ASCII command into UDP protocol with port 1025. User can simply send the Command of ASCII format through UDP protocol (such as `UPD_send(Dest_IP, "$01M")`)

1.2 Syntax of ASCII

Command Syntax: [delimiter character][address][channel][command][data][checksum][carriage return] Every command begins with a delimiter character.

There are two valid characters: \$ and # .The delimiter character is followed by a two-character address (hex-decimal) that specifies the target system. The two characters following the address specified the module and channel.

Depending on the command, an optional data segment may follow the command string. An optional two-character checksum may also be appended to the command string. Every command is terminated with a carriage return (cr).

The command set is divided into the following five categories:

- ♦ System Command Set
- ♦ Analog Input Command Set
- ♦ Analog Input Alarm Command Set
- ♦ Universal I/O Command Set
- ♦ Digital I/O Command Set

Every command set category starts with a command summary of the particular type of module, followed by datasheets that give detailed information about individual commands. Although commands in different subsections sometime share the same format, the effect they have on a certain module can be completely different than that of another. Therefore, the full command sets for each type of modules are listed along with a description of the effect the command has on the given module.

Note: All commands should be issued in UPPERCASE characters only!

1.3 How to Send ASCII command with TCP/IP protocols

There are two ways to send ASCII command to EDAM9000 module

♦ UDP protocol with port 1025

Request: "\$01M" + chr(cr) ; just send ASCII command string ended with code 0x13 (cr)

Response: ASCII string ended with code 0x13 (cr)

Example:

```
Dim Asc_cmd ,Rep_cmd as string
```

```
Asc_cmd="$01M" & chr(13)
```

```
Winsock_UDP.SendData(Asc_cmd) 'Send ASCII command
```

```
.....
```

```
Winsock_UDP.GetData Rep_cmd vbByte ' Read Response string
```

♦ MODBUS/TCP protocol with port 502

The EDAM9000 module provides MODBUS/TCP function 17H to accept the ASCII command sent from the host. The format of request frame buffer as shown follows:

Framebuffer(0~3)=0	'always 0 for MODBUS/TCP standard
Framebuf(4) = 0	' always 0
Framebuf(5) = n	' number of following bytes
Framebuf(6) = 1	'Always 1 for EDAM9000
Framebuf(7) = &H17	'MODBUS/TCP function call 17H (used for ASCII command transaction)
Framebuf(8) = 0	'always 0
Framebuf(9) = 0	'always 0
Framebuf(10) = 0	'always 0
Framebuf(11) = 0	'always 0
Framebuf(12) = 0	'always 0
Framebuf(13) = 0	'always 0
Framebuf(14) = 0	'always 0
Framebuf(15) = 0	'always 0
Framebuf(16) = m	'number of ASCII command bytes
Framebuf(17+) = m	' ASCII command data

Example:

```
Dim Framebuf(22) ,Respbuf(30) as byte
```

```
Dim Asc_cmd as string
```

```
Asc_cmd="%01M" & chr(13)
```

```
Framebuffer(0)=0 'always 0 for MODBUS/TCP standard
```

Framebuffer(1)=0	'always 0 for MODBUS/TCP standard
Framebuffer(2)=0	'always 0 for MODBUS/TCP standard
Framebuffer(3)=0	'always 0 for MODBUS/TCP standard
Framebuf(4) = 0	' always 0
Framebuf(5) = 16	' number of following bytes=16 bytes
Framebuf(6) = 1	'Always 1 for EDAM9000
Framebuf(7) = &H17	'MODBUS/TCP function call 17H
Framebuf(8) = 0	'always 0
Framebuf(9) = 0	'always 0
Framebuf(10) = 0	'always 0
Framebuf(11) = 0	'always 0
Framebuf(12) = 0	'always 0
Framebuf(13) = 0	'always 0
Framebuf(14) = 0	'always 0
Framebuf(15) = 0	'always 0
Framebuf(16) =5	'number of ASCII command bytes=5 bytes
Framebuf(17) = Asc(Mid(Asc_cmd, 1, 1))	' 1's ASCII code "%"
Framebuf(18) = Asc(Mid(Asc_cmd, 2, 1))	' 1's ASCII code "0"
Framebuf(19) = Asc(Mid(Asc_cmd, 3, 1))	' 1's ASCII code "1"
Framebuf(20) = Asc(Mid(Asc_cmd, 4, 1))	' 1's ASCII code "M"
Framebuf(21) = Asc(Mid(Asc_cmd, 5, 1))	' 1's ASCII code chr(13)

MODBUS_TCP.SendData(Framebuf) 'Send ASCII command

.....

MODBUS_TCP.GetData Respbuf vbByte ' Read Response string

Where :

Respbuf(7)=8	'number of response ASCII string
Respbuf(8)='!	'ASCII code '!'
Respbuf(9)='0'	'ASCII code '0'
Respbuf(10)='1'	'ASCII code '1'
Respbuf(11)='9'	'ASCII code '9'
Respbuf(12)='0'	'ASCII code '0'
Respbuf(13)='5'	'ASCII code '5'
Respbuf(14)='2'	'ASCII code '2'
Respbuf(15)=13	'ASCII code 0x13 (cr)

1.4 ASCII Command Set

Command	Command Name	Description	Available Modules
---------	--------------	-------------	-------------------

Common commands

\$aaM	Read Module Name	Return the module name from a specified module	All modules
\$aaF	Read Firmware Version	Return the firmware version from a specified module	All modules
\$aaID	Read ID number	Return the ID number of the specified module	All modules
%aaDHCPn	Set DHCP	Enable or disable DHCP	All modules
%aaSETUDPSTd dhhmmss	Set Active time interval	Set module UDP active-send time interval	All modules
%aaSETUDPnpp pppppp	Set active/event IP	Set UDP active/event IP address	All modules
%aaSETUDPnR	Enable active send	Enable active-send message	All modules
%aaSETUDPnP	Disable active send	Disable active-send message	All modules
%aaSETUDPnRE	Enable event trigger	Enable event trigger	
%aaSETUDPnPE	Disable event trigger	Disable event trigger	

Analog commands

#aan	Read Analog Input from Channel N	Return the input value from the specified analog input channel	9015/9017/9019
#aa	Read Analog Input from all channels	Return the input values from all analog input channels	9015/9017/9019
\$aa0	Span Calibration	Calibrate the analog input module to correct the gain error	9015/9017/9019
\$aa1	Offset Calibration	Calibrate the analog input module to correct the offset error	9015/9017/9019
\$aa6	Read Channel Enable/Disable Status	Asks a specified module to return the Enable/Disable status of all analog input channels	9015/9017/9019
\$aa5mm	Set Channel Enable/Disable Status	Set Enable/Disable status for analog input channels	9015/9017/9019
#aaMH	Read all Max. Data	Read the maximum data from all analog input channels	9015/9017/9019
#aaMHn	Read single Max. Data	Read the maximum data from a specified analog input channel	9015/9017/9019
#aaML	Read all Min. Data	Read the minimum data from all analog input channels	9015/9017/9019
#aaMLn	Read single Min. Data	Read the minimum data from a specified analog input channel	9015/9017/9019
\$aaCjAhs	Set Alarm Mode	Set the High/Low alarm in either Momentary or Latching mode	9015/9017/9019
\$aaCjAh	Read Alarm Mode	Returns the alarm mode for the specified channels	9015/9017/9019
\$aaCjAhEs	Enable/Disable Alarm	Enables/Disables the high/low alarm of the specified channels	9015/9017/9019
\$aaCjCh	Clear Latch Alarm	Resets a latched alarm	9015/9017/9019
\$aaCjAhCCn	Set Alarm Connection	Connects the High/Low alarm of a specified input channel to interlock with a specified output channel	9017/9019
\$aaCjRhC	Read Alarm Connection	Returns the alarm configuration of a specified input channel	9017/9019
\$aaCjAhU	Set Alarm Limit	Sets the High/Low alarm limit value to a specified channel	9015/9017/9019
\$aaCjRhU	Read Alarm Limit	Returns the High/Low alarm limit value of the specified channel	9015/9017/9019

EDAM-9000 ASCII Command User's manual V1.1

\$aaCjS	Read Alarm Status	Reads whether an alarm occurred in the specified Channel	9015/9017/9019
\$aa3	Read cold junction	Return the Cold Junction temperature	9019
\$aa9hhhhh	Set CJ offset	Set Cold Junction temperature offset	9019
\$aa9	Read CJ offset	Return Cold Junction temperature offset	9019
\$aaAnntt	Set analog input type	Set analog input type(range)	9015/9017/9019
\$aaAff	Set conversion rate	Set conversion rate of analog input module	9015/9017/9019
\$aaBhh	Read input range	Return the input mode of the specified analog channel	9015/9017/9019

Digital I/O commands

\$aa6	Read Channels Status	Asks a specified input module to return the status of all channels	9050/9051/9052
\$aa7	Read DI latch status	Read DI latch status	9050/9051/9052
#aaDnd	Set Digital Output	Sets the status for the specified digital output channel	9050/9051/9052
#aabb	Write Digital Output	Writes specified values to either a single channel or all channels simultaneously	9050/9051/9052
#aa2hhhhhhh	Write DO pulse counts	Set output counts of the specified DO channel	9050/9051/9052

Chapter 2 ASCII Command Description

2.1 \$aaM Read Module Name

Description: Returns the module name from a specified module.

Syntax: \$aaM(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

M is the Module Name command.

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

Response: !aa60bb(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was in-valid.

aa (range 00-FF) represents the 2-character hexadecimal address of an EDAM-9000 module.

bb (range 00-FF) represents the 2-character model number of an EDAM-9000 module.

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

Example:

command: \$01M(cr)

Response: !016050(cr)

The command requests the system at address 01h to send its module name. The system at address 01h responds with module name 6050 indicating that there is an EDAM-9050 at address 01h.

2.2 \$aaF Read Firmware Version

Description: Returns the firmware version from a specified module.

Syntax: \$aaF(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

F is the Firmware Version command.

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

Response: !aa(version)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an EDAM-9000 module.

(version) represents the firmware version of the module.

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

Example: command: \$01F(cr)

response: !01 1.01(cr)

The command requests the system at address 01h to send its firmware version. The system responds with firmware version 1.01.

2.3 \$aaID Read module ID number

Description: Returns the ID number from a specified module.

Syntax: \$aaID(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

ID is the ID command.

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

Response: !aann(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an EDAM-9000 module.

nn represents the ID number of the module.

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

Example: command: \$01ID(cr)

response: !010A(cr)

The command requests the system at address 01h to send its ID number. The system responds with ID number 10(0AH).

2.4 %aaDHCPn Enable or disable DHCP

Description: enable or disable DHCP function for the specific module.

Syntax: %aaDHCPn(cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

DHCP represents the DHCP setting command.

n represents Enable/disable DHCP . n=0 represents disable, ff=1 represents enable

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01DHCP1(cr)

response: !01

The command enables DHCP function of the module at address 01.

2.5 %aaSETUDPSTddhhmmss Set module UDP active-send time interval

Description: Set module UDP active-send time interval. When reach active time interval, the module will automatically send an UDP message to specific IP with port 5168.

Syntax: %aaSETUDPSTddhhmmss(cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDPST represents set active-send time interval command.

dd represents days (0~1F)

hh represents hours(0~17)

mm represents minutes(0~3B)

ss represents seconds(0~3B)

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDPST010F1220(cr)

response: !01

The command set active-send time interval to 01 day, 15(0FH) hours, 18(12H) minutes, 32(20H) seconds.

2.6 %aaSETUDPnpppppppp Set UDP Active/Event IP address

Description: Set module UDP active/event destination IP address. When reach active time interval, the module will automatically send an UDP message to specific active/event destination IP with port 5168.

Syntax: %aaSETUDPnpppppppp(cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDP represents set active/Event IP command.

n represents n'th destination IP address

pppppppp represents IP address (8 bytes)

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDP2C0A80078(cr)

response: !01

The command set active/event IP address of 2'th destination device to be C0A80078=192.168.0.120.

PS:

See *struct_StreamData* in "EDAM9K_DataStructure.pdf" to understand the content of the active-send message

2.7 %aaSETUDPnR Enable UDP active-send message

Description: Enable module UDP active-send message. When reach active time interval, the module will automatically send an UDP message to specific active/event destination IP with port 5168.

Syntax: %aaSETUDPnR(cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDPxR represents enable active-send message.

n represents n'th destination IP address

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDP3R(cr)

response: !01

The command enable active-send message to 3'th destination IP device

PS:

- See **struct_StreamData** in "EDAM9K_DataStructure.pdf" to understand the content of the active-send message
- See **struct_AlarmInfo** in "EDAM9K_DataStructure.pdf" to understand the content of the alarm event message

2.8 %aaSETUDPnP Disable UDP active-send message

Description: Disable module UDP active-send message.

Syntax: %aaSETUDPnP(cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDPxP represents disable active-send message.

n represents n'th destination IP address

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDP3P(cr)

response: !01

The command disable active-send message to 3'th destination IP device.

2.9 %aaSETUDPnRE Enable UDP event trigger

Description: Enable module UDP event trigger. When alarm event occurred in the module, the module will send alarm event message to destination IP device with port 5168.

Syntax: %aaSETUDPnRE (cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDPxRE represents enable event trigger.

n represents n'th destination IP address

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDP3RE (cr)

response: !01

The command enables the module to send alarm event message to 3'th destination IP device.

PS:

See **struct_AlarmInfo** in "EDAM9K_DataStructure.pdf" to understand the content of the alarm event message

2.10 %aaSETUDPnPE Disable UDP event trigger

Description: disable module UDP event trigger.

Syntax: %aaSETUDPnPE (cr)

% is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

SETUDPxPE represents disable event trigger.

n represents n'th destination IP address

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: %01SETUDP3PE (cr)

response: !01

The command disables the module to send alarm event message to 3'th destination IP device.

2.11 #aan Read Analog Input from Channel N

Description: Returns the input data from a specified analog input channel in a specified module.

Syntax: #aan(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of the EDAM-9000 (Always 01)

n (range 0-8) represents the specific channel you want to read the input data.

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

Response: >(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

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

Example: command: #012(cr)

response: >+10.000

Channel 2 of the EDAM-9000 analog module at address 01h responds with an input value +10.000.

2.12 #aa Read Analog Input from All Channels

Description: Returns the input data from all analog input channels in a specified module.

Syntax: #aa(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

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

Response: >(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

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

Note: The latest data returned is the Average value of the preset channels in this module.

Example: command: #01(cr)

response: >+10.000+10.000+10.000+10.000+10.000+10.000+10.000+10.000+10.000

2.13 \$aa0 Span Calibration

Description: Calibrates a specified module to correct for gain errors

Syntax: \$aa0(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

0 represents the span calibration command.

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Note: 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 process.

2.14 \$aa1 Zero Calibration

Description: Calibrates a specified module to correct for offset errors

Syntax: \$aa1(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

1 represents the zero calibration command.

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Note: 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 process.

2.15 \$aa6 Read Channel Enable/Disable Status

Description: Asks a specified module to return the Enable/Disable status of all analog input channels

Syntax: \$aa6(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

6 is the read channels status command.

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

Response: !aamm(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

mm are two hexadecimal values. Each value is interpreted as 4 bits. The first 4-bit value represents the status of channels 7-4, the second 4 bits represents the status of channels 3-0. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled.

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

Example: command: \$016(cr)

response: !01FF(cr)

The command asks the specific module at address 01h to send Enable/Disable status of all analog input channels. The analog input module responds that all its channels are enabled (FF equals 1111 and 1111).

2.16 \$aa5mm Set Channel Enable/Disable Status

Description: Set Enable/Disable status for all analog input channels

Syntax: \$aa5mm(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address . (Always 01)

5 identifies the enable/disable channels command.

mm (range 00-FF) are two hexadecimal characters. Each character is interpreted as 4 bits. The first 4-bit value represents the status of channels 7-4; the second 4-bit value represents the status of channels 3-0. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled.

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: \$01581(cr)

response: !01(cr)

The command enables/disables channels of the analog input module at address 01h. Hexadecimal 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 channels 1, 2 and 3.

2.17 #aaMH Read Maximum Value

Description: Read the maximum values from all analog input channels in a specified analog module

Syntax: #aaMH(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

MH represents the read maximum value command.

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

Response: >(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #01MH(cr)

response:>+10.000+10.000+10.000+10.000+10.000+10.000 +10.000+10.000+10.000

The command asks the specific module at address 01h to send historic maximum value from all analog input channels.

Note: The latest data returned is the Average value of the preset channels in this module.

2.18 #aaMHn Read Maximum Value from channel N

Description: Read the maximum value from a specific channel in a specified module

Syntax: #aaMHn(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

MH represents the read maximum value command.

n (range 0-8) represents the specific channel you want to read the input data.

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

Response: >(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #01MH2(cr)

response: >+10.000

The command asks the specific module at address 01h to send historic maximum value from analog input channel 2.

2.19 #aaML Read Minimum Value

Description: Read the minimum values from all analog input channels in a specified module

Syntax #aaML(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

ML represents the read minimum value command.

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

Response: >(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #01ML(cr)

response:>+10.000+10.000+10.000+10.000+10.000+10.000+10.000+10.000+10.000

The command asks the specific module at address 01h to send historic minimum value from all AI channels.

Note: The latest data returned is the Average value of the preset channels in this module.

2.20 #aaMLn Read Minimum Value from channel N

Description: Read the minimum value from a specific analog input channel in a specified module

Syntax: #aaMLn(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

ML represents the read minimum value command.

n (range 0-8) represents the specific channel you want to read the input data.

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

Response: >(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #01ML3(cr)

response: >+10.000

The command asks the specific module at address 01h to send historic minimum value from analog input channel 3.

2.21 \$aaCjAhs Set Alarm Mode

Description: Sets the High/Low alarm of the specified input channel in the addressed EDAM-9000 module to either Latching or Momentary mode.

Syntax: \$aaCjAhs(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired channel j (j : 0 to 7).

A is the Set Alarm Mode command.

h indicates alarm types (H = High alarm, L = Low alarm)

s indicates alarm modes (M = Momentary mode,L = Latching mode)

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

Response: !aa(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding EDAM-9000 module.

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

Example: command: \$01C1AHL(cr)

response: !01(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to set its High alarm in Latching mode. The module confirms that the command has been received.

2.22 \$aaCjAh Read Alarm Mode

Description: Returns the alarm mode for the specified channel in the specified EDAM-9000 module.

Syntax: \$aaCjAh(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired channel j (j : 0 to 7).

A is the Read Alarm Mode command.

h indicates the alarm types (H = High alarm,L = Low alarm)

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

Response: !aas(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding EDAM-9000 module.

s indicates alarm modes (M = Momentary mode, L = Latching mode)

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

Example: command: \$01C1AL(cr)

response: !01M(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to return its Low alarm mode. The system responds that it is in Momentary mode.

2.23 \$aaCjAhEs Enable/Disable Alarm

Description: Enables/Disables the High/Low alarm of the specified input channel in the addressed EDAM-9000 module

Syntax: \$aaCjAhEs(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired channel j (j : 0 to 7).

AhEs is the Set Alarm Mode command.

h indicates alarm type (H = High alarm, L = Low alarm)

s indicates alarm enable/disable (E = Enable, D = Disable)

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

Response: !aa(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding EDAM-9000 module.

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

Example: command: \$01C1ALEE(cr)

response: !01(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to enable its Low alarm function. The module confirms that its Low alarm function has been enabled.

Note: An analog input module requires a maximum of 2 seconds after it receives an Enable/Disable Alarm command to let the setting take effect. During this interval, the module can not be addressed to perform any other actions.

2.24 \$aaCjCh Clear Latch Alarm

Description: Sets the High/Low alarm to OFF (no alarm) for the specified input channel in the addressed EDAM-9000 module

Syntax: \$aaCjCh(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired channel j (j : 0 to 7).

Ch is the Clear Latch Alarm command.

h indicates alarm type (H = High alarm, L = Low alarm)

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

Response: !aa(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal Modbus network address of the corresponding EDAM-9000 module

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

Example: command: \$01C1CL(cr)

response: !01(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to set its Low alarm state to OFF. The system confirms it has done so accordingly.

2.25 \$aaCjAhCCn Set Alarm Connection

Description: Connects the High/Low alarm of the specified input channel to interlock the specified digital output in the addressed EDAM-9000 module

Syntax: \$aaCjAhCCn(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired analog input channel j (j : 0 to 7).

AhC is the Set Alarm Connection command.

h indicates alarm type (H = High alarm, L = Low alarm)

Cn identifies the desired digital output channel n (n : 0 to 1). To disconnect the digital output, n should be set as ~*

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

Response: !aa(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal Modbus network address of the corresponding EDAM-9000 module.

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

Example: command: \$01C1ALCC0(cr)

response: !01(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to connect its Low alarm to the digital output of channel 0 in the specific module. The system confirms it has done so accordingly.

2.26 \$aaCjRhC Read Alarm Connection

Description: Returns the High/Low alarm limit output connection of a specified input channel in the addressed module

Syntax: \$aaCjRhC(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module. (Always 01)

Cj identifies the desired analog input channel j (j : 0 to 7).

RhC is the Read Alarm Connection command.

h indicates alarm type (H = High alarm, L = Low alarm)

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

Response: !aaCn(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal Modbus network address of the corresponding EDAM-9000 module.

Cn identifies the desired digital output channel n (n : 0 to 1) whether interlock with the alarm of the specific analog input channel. If the values of n are "*", the analog input has no connection with a digital output point.

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

Example: command: \$01C1RLC(cr)

response: !01C0(cr)

Channel 1 of the EDAM-9000 module at address 01h is instructed to read its Low alarm output connection. The system responds that the Low alarm output connects to the digital output at channel 0 in the specific module.

2.27 \$aaCjAhU Set Alarm Limit

Description: Sets the High/Low alarm limit value for the specified input channel of a specified EDAM-9000 module.

Syntax: \$aaCjAhU(data)(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired analog input channel j (j : 0 to 7).

AhU is the Set Alarm Limit command.

h indicates alarm type (H = High alarm, L = Low alarm)

(data) represents the desired alarm limit setting. The format is always in engineering units.

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

Response: !aa(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal Modbus network address of the corresponding EDAM-9000 module.

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

Example: command: \$01C1AHU+080.00(cr)

response: !01(cr)

The high alarm limit of the channel 1 in the specific module at address 01h is been set +80. The system confirms the command has been received.

Note: An analog input module requires a maximum of 2 seconds after it receives a Set Alarm Limit command to let the settings take effect. During this interval, the module cannot be addressed to perform any other actions.

2.28 \$aaCjRhU Read Alarm Limit

Description: Returns the High/Low alarm limit value for the specified input channel in the addressed EDAM-9000 module

Syntax: \$aaCjRhU(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired analog input channel j (j : 0 to 7).

RhU is the Read Alarm Limit command.

h indicates alarm type (H = High alarm, L = Low alarm)

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

Response: !aa(data)(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal Modbus network address of the corresponding EDAM-9000 module.

(data) represents the desired alarm limit setting. The format is always in engineering units.

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

Example: command: \$01C1RHU(cr)

response: !01+2.0500(cr)

Channel 1 of the EDAM-9000 module at address 01h is configured to accept 5V input. The command instructs the system to return the High alarm limit value for that channel. The system responds that the High alarm limit value in the desired channel is 2.0500 V.

2.29 \$aaCjS Read Alarm Status

Description: Reads whether an alarm occurred to the specified input channel in the specified EDAM-9000 module

Syntax: \$aaCjS(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module(Always 01)

Cj identifies the desired analog input channel j (j : 0 to 7).

S is the Read Alarm Status command.

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

Response: !aahl(cr) if the command was valid

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal address Modbus of the corresponding EDAM-9000 module.

h represents the status of High alarm. "1" means the High alarm occurred, '0" means it did not occur. l represents the status of Low alarm. ~1" means the Low alarm occurred, ~0" means it did not occur.

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

Example: command: \$01C1S(cr)

response: !0101(cr)

The command asks the module at address 01h to return its alarm status for channel 1. The system responds that a High alarm has not occurred, but the Low alarm has occurred.

2.30 \$aa3 Read cold junction temperature

Description: Return the Cold Junction temperature of EDAM-9019

Syntax: \$aa3(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

3 is the command to read cold junction temperature.

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

Response: >(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

(data) a 8-character hexadecimal value representing the cold junction temperature.

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

Example: command: \$013(cr)

response: >+00017.5(cr)

The command asks the specific module at address 01h to return the cold junction temperature of specified module. The response is +17.5C

2.31 \$aa9hhhhh Set CJ offset

Description: Set Cold Junction temperature offset of EDAM-9019

Syntax: \$aa19hhhhh(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

9 is the command to set cold junction temperature offset.

hhhhh is the offset value times by 80 (5-character hexadecimal format)

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

aa represents the 2-character hexadecimal address Modbus of the corresponding EDAM-9000 module.

? delimiter indicating the command was invalid.

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

Example: command: \$019000A0(cr)

response: !01(cr)

This example need to set cold junction offset to 2C , then the actual ASCII value should be $2 * 80 = 160$ (hex=000A0). Hence the complete ASCII command string is \$019000A0(cr)

2.32 \$aa9 Read CJ offset

Description: Return Cold Junction temperature offset of EDAM-9019

Syntax: \$aa9(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

9 is the command to read cold junction temperature offset.

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

Response: >(data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

(data) a 8-character hexadecimal value representing the cold junction temperature offset.

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

Example: command: \$019(cr)

response: >+00005.5(cr)

The command asks the specific module at address 01h to return the cold junction temperature offset of specified module. The response is +5.5C

2.33 \$aaAnntt Set analog input type(range)

Description: Set the analog input type (range) in EDAM-9000 analog input module.

Syntax: \$aaAnntt (cr)

\$ is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

A represents the analog input setting command.

nn (range 00-FF) represents the specific channel you want to set the input type.

tt (range 0-16) represents the type you want to set to the specific channel

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! Delimiter indicating a valid command was received.

? Delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: \$01A030D(cr)

response: !01

The command set analog input channel 3 to type 0D (0~20mA) for the specific analog input module at address 01h.

2.34 \$aaAff Set conversion rate of analog input module

Description: Set conversion rate of the analog input module.

Syntax: \$aaAff(cr)

\$ is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

A represents the analog input setting command.

ff represents the conversion rate . ff=60 represents 60Hz, ff=50 represents 50Hz

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: \$01A60(cr)

response: !01

The command set conversion rate to 60Hz for the specific analog input module at address 01.

2.35 \$aaBhh Read analog input type

Description: Return the input type of the specified analog channel 9015/9017/9019

Syntax: \$aaBhh(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

hh is the analog input channel number represents the 2-character in hexadecimal format.

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

Response: !aann(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

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

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (always 01) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module.

nn a 2-character hexadecimal value representing the type of the analog input channel.

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

Example: command: \$01B01(cr)

response: !0108(cr)

The first 2-character portion of the response (exclude the "!" character) indicates the address of the EDAM-9000 module. The second 2-character portion of the response is the type of channel (For each analog module, the type number is different, ref to Figure 2-1 Analog input types)

Code(Hex)	Type
0x07	4-20mA
0x08	+/-10V
0x09	+/-5V
0x0a	+/-1V
0x0b	+/-500mV
0x0c	+/-150mV
0x0d	0-20mA
0x0e	J type -8824 uV ~ 69536 uV,
0x0f	K type -5891 uV ~ 54807 uV
0x10	T type -5603 uV ~ 20869 uV
0x11	E type -9835 uV ~ 76373 uV
0x12	R type -0000 uV ~ 21101 uV
0x13	S type -0000 uV ~ 18693 uV
0x14	B type -0000 uV ~ 13820 uV
0x20	IEC Pt100 -50C ~ 150C
0x21	IEC Pt100 0C ~ 100C
0x22	IEC Pt100 0C ~ 200C
0x23	IEC Pt100 0C ~ 400C
0x24	IEC Pt100 -200C ~ 200C
0x25	JIS Pt100 -50C ~ 150C
0x26	JIS Pt100 0C ~ 100C
0x27	JIS Pt100 0C ~ 200C
0x28	JIS Pt100 0C ~ 400C
0x29	JIS Pt100 -200C ~ 200C
0x2A	Pt1000 -40C ~ 160C
0x2B	BALCO500 -30C ~ 120C
0x2C	Ni -80C ~ 100C
0x2D	Ni 0C ~ 100C

Figure 2-1 Analog input types

2.36 \$aa6 Read DI /DO Channel Status

Description: This command requests that the specified EDAM-9000 module return the status of its digital input and digital output channels

Syntax: \$aa6(cr)

\$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

6 is the Digital Data In command.

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

Response: !aa0(Do data)(Di data)(Di data)(Di data)(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an EDAM-9000 module.

(D0 data)a 1-character hexadecimal value representing the values of the digital output module.

(Di data)a 2-character hexadecimal value representing the values of the digital input module.

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

Example: command: \$016(cr)

response: !0100A35D(cr)

0A: the status of digital output channels.

0A=(00001010) DO channels 1/3 =ON, DO Channel 0/2/4/5/6/7=OFF

35D: the status of digital input channels.

35D=(001101011101) DI channels 0/2/3/4/6/8/9 =Active state, DI Channel 1/5/7=Inactive

2.37 \$aa7 Read DI latch status

Description: Read DI latch status.

Syntax: \$aa7(cr)

\$ is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

7 represents read DI latch status command.

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: \$017(cr)

response: !010003

The command read DI latch status= 0003, DI #0 latched, DI #1 latched, and DI #2 ~ DI #15 no latched

2.38 #aa00dd Write All Digital Output

Description: This command sets all digital output channels to the specific EDAM-9000 module.

Syntax: #aa00nn(data)(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address (Always 01)

00 represents Writing to all channels (write a byte) command

dd represents the data be written to digital output

Response: >(cr) if the command was valid.

?aa(cr) if an invalid command has been issued.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

> delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

a (range 00-FF) represents the 2-character hexadecimal Modbus network address of a module that is responding.

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

Example:

command: #010033(cr)

response: >!(cr)

An output byte with value 33h (00110011) is sent to the digital output module at address 01h.

The Output channel 0/1/4/5 = ON ,Output channel 2/3/6/7 = OFF

2.39 #aa1n0d Set Single Digital Output Channel

Description: Set the digital output status of EDAM-9000 digital output module.

Syntax: #aa1n0d(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address (Always 01)

n (range 0-F) represents the specific channel you want to set the output status.

d (range 0-1) represents the status you want to set to the specific channel

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! Delimiter indicating a valid command was received.

? Delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #011201(cr)

response: !01

The command set digital channel 2 "ON" status for the specific module at address 01h.

Example: command: #011200(cr)

response: !01

The command set digital channel 2 "OFF" status for the specific module at address 01h.

2.40 #aa2npppppppp Write DO pulse counts

Description: Generates DO pulse output.

Syntax: #aa2nmmmmmmm(cr)

is a delimiter character.

aa (range 00-2D) represents the 2-character hexadecimal Modbus address (Always 01)

2 represents Generates DO pulse output command.

n represents DO channel n

pppppppp represents pulse counts (8 digits)

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

Response: !aa(cr) if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the address does not exist.

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal Modbus address of an EDAM-9000 module.

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

Example: command: #aa230000001F(cr)

response: !01

The command force the DO channel #3 to output 31(1FH) pulses