eDAM-8018
8-channel Thermocouple Input Module
User's manual

Web site: www.inlog.com.tw

Rev 1.0 Printed 2005/06//20 Taiwan

Table of Contents

Chapter 1 Introduction

1.1 Overview

The eDAM-*8000* 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 *eDAM-8000* 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

eDAM 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 eDAM modules can be programmed in virtually any high-level language.
Up to 256 eDAM modules may be connected to an RS-485 multi-drop network by using the eDAM RS-485 repeater, extending the maximum communication distance to 4,000 ft.

1.4 Software Configuration and Calibration

EDAM 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

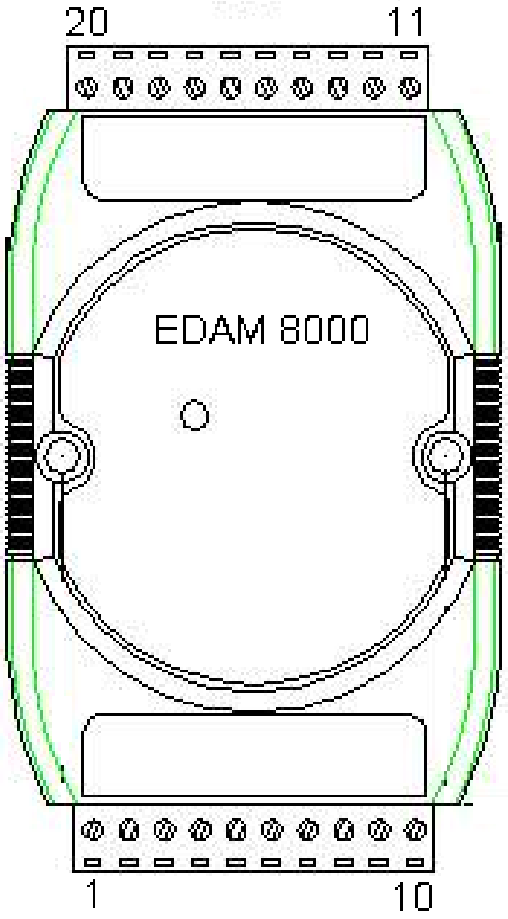
A watchdog timer supervisory function will automatically reset the eDAM modules in the event of system failure. Maintenance is thus simplified.

1.6 Power Requirements

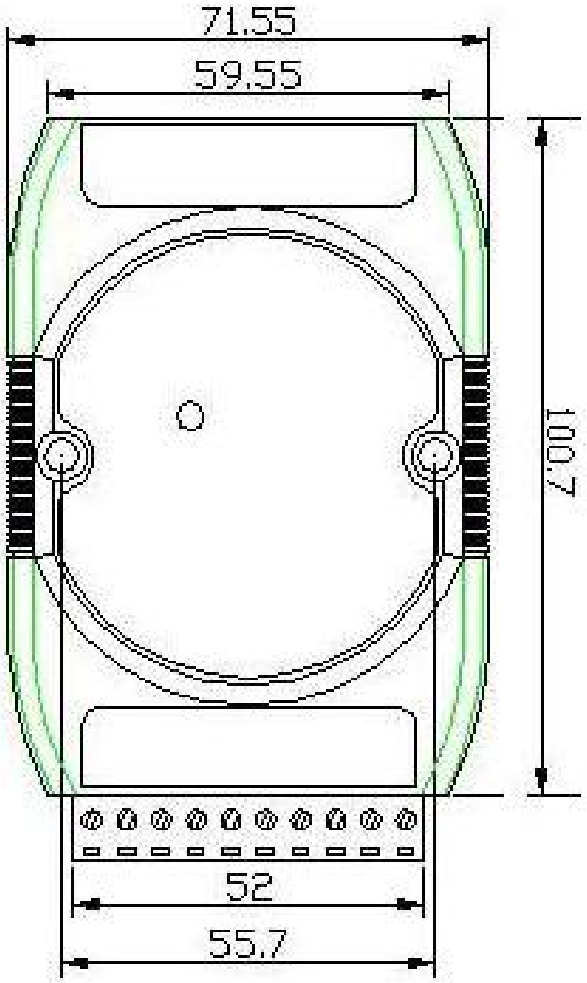
Although the modules are designed for standard industrial unregulated 24V DC power supply , they accept any power unit that supplies power within the range of +10 to +30 V DC . The power supply ripple must be limited to 5 V peak-to-peak, and the immediate ripple voltage should be maintained between +10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

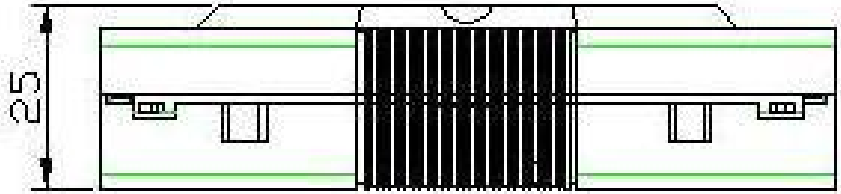
Chapter 2 About the eDAM Analog Modules

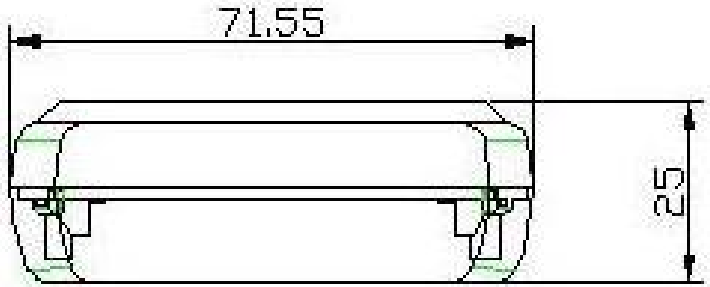
2.1 Outline of eDAM Analog modules

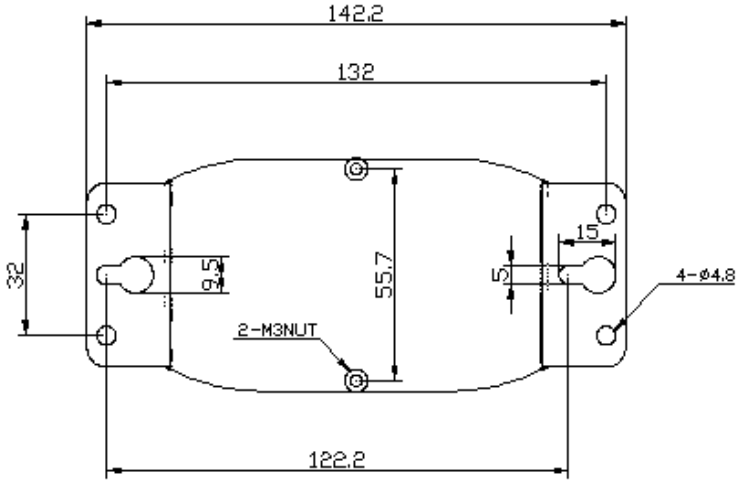


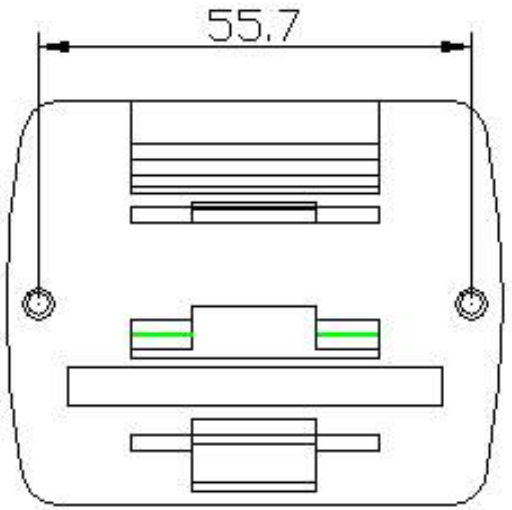
2.2 Module Dimension

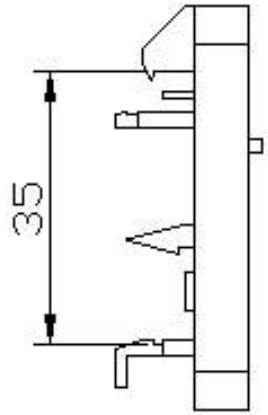



25


71.55
25


142.2
132
32
9.5
55.7
5
15
4-ø4.8
2-M3NUT
122.2


55.7


35

2.3 eDAM Analog modules

The eDAM provides a series of analog input or digital in/output modules to sense the analog and digital signal or to control the remote devices.

- eDAM-8018 : 8-channel Thermocouple Input Module

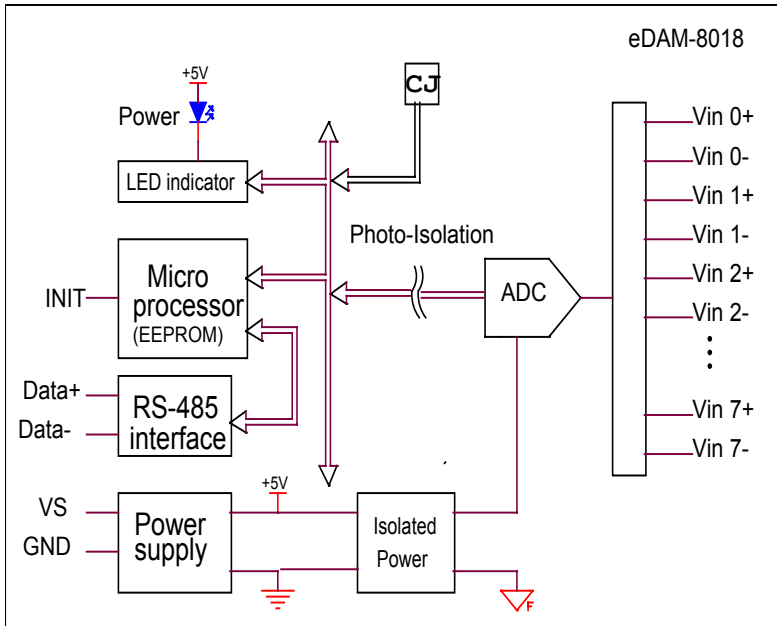
2.3.1 eDAM-8018 module

eDAM-8018 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.

Specifications of eDAM-8018

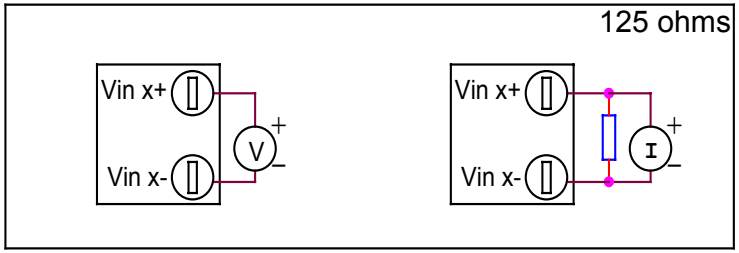
- Interface: RS-485, 2 wires
- Speed (bps): 1200, 2400, 4800, 9600, 19.2K, 38.4K , 115.2K
- Analog Input type: Differential input
- Analog Channels Numbers: 8
- Analog Resolution: 16 bits
- Unit Conversion: Thermocouple, mV, V or mA
- Thermocouple Type: J, K, T, E, R, S, B, N, C
- Sampling Rate :10 Samples/Second
- 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 125W resistor)
- Power supply: +10V to +30V

2.4 Block diagram of modules



2.5 Wire connection

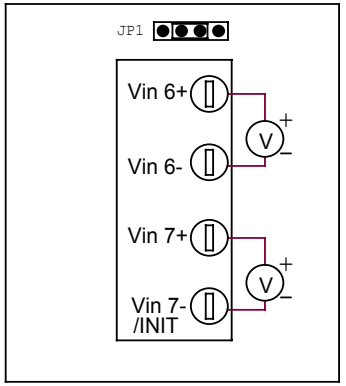
2.5.1 EDAM-8018 wire connection

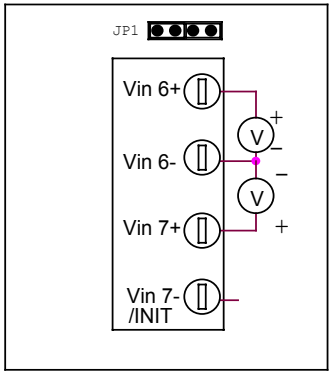


Differential analog input for channel 0 to channel 7

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





2.6 EDAM8018 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 16)

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

Chapter 3 Installation

This chapter provides guidelines to what is needed to set up and install an eDAM network. A quick hookup scheme is provided that lets you configure modules before they are installed in a network. To help you to connect eDAM modules with sensor inputs, several wiring examples are provided. Finally, you will find at the end of this chapter a programming example using the eDAM 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 eDAM to other compatible modules, which have on-board switches for their baud rate setting, eDAM modules should not be opened. There is no need to open the eDAM 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 eDAM network

The following list gives an overview of what is needed to setup, install and configure an eDAM environment.
A host computer that can output ASCII characters with an RS-232C or RS-485 port.
Power supply for the eDAM modules (+10 to +30 V DC)
eDAM Series Utility software

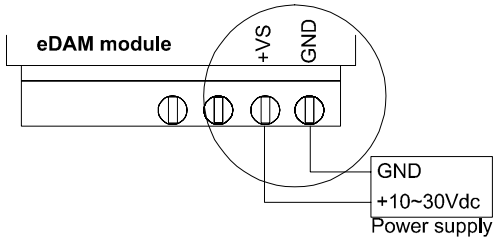
3.2 Host computer

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 eDAM-8520 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 eDAM modules are designed to accept industry standard +24 VDC unregulated power. Operation is guaranteed when using any power supply between +10 and +30 VDC. Power ripples must be limited to 5 V peak to peak while the voltage in all cases must be maintained between +10 and +30 VDC . All power supply specifications are referenced at module connector. When modules are powered remotely, the effects of line voltage drops must be considered.

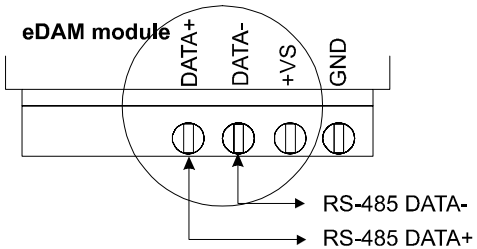
3.3 Power supply

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 eDAM network to reduce interference.



3.5 eDAM Utility Software

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

3.6 eDAM Isolated RS-232/RS485 Converter

When the host computer or terminal has only a RS-232 port, an eDAM-8520 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 eDAM modules in a RS-485 network must have an *unique* address ID. Therefore, to configure the brand-new 8012/D, 8014/D, 8017 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 8012/D, 8017)
  Six differential and 2 single-ended input mode (for 8017)
- INIT* State settings:
  The eDAM 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 eDAM 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 eDAM-8520 to COM port of host computer.
2. Connect a brand new eDAM 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 eDAM modules.
4. Use the eDAM utility to configure the address ID, baud rate, check-sum status and command sets of the module.

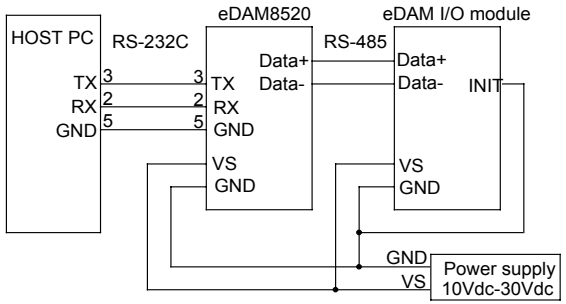


Figure 4.1

3.9 Install a New eDAM to a Existing Network

1. Equipments for Install a New Module
2. A existing eDAM network
3. New eDAM modules.
4. Power supply (+10 to +30 VDC)

Installing Procedures

1. Configure the new eDAM module according to the initialization procedure in section 3.7
2. 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 eDAM modules on the network.
3. Power off the eDAM power supply of the existing RS-485 network.
4. Wire the power lines for the new eDAM with the existing network. Be careful about the signal polarity as wiring.
5. Wire the RS-485 data lines for the new eDAM with the existing network. Be careful about the signal polarity as wiring.
6. Wire to the input or output devices.
7. Power on the eDAM local power supply.
8. Use the eDAM utility to check entire network.

Chapter 4 Command Set

4.1 Introduction

The eDAM 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 eDAM module except those synchronized commands with wildcard address command "#**". The eDAM may or may not give response to the command. The host should check the response to handshake with the modules.

4.2 Format of eDAM 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 eDAM 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.

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)

4.3 Response of Commands

The response message depends on eDAM 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.

4.4 Summary of Command Set

There are four categories of eDAM commands. The first is the ***eDAM 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 eDAM analog input module are list in the following table.

4.4.1 eDAM Special commands

Command	Syntax	Modules	page
Set brand compatible	~AA2X01V	All modules	31
Read current brand setting	~AA2X02	All modules	32

4.4.2 Host Watchdog Command Sets

Command	Response	Description	Page
~**	no response	Host OK	
~AA0	!AASS	Read Module Status	
~AA1	!AA	Reset Module Status	
~AA2	!AAVV	Read Host watchdog Timeout Value	
~AA3EVV	!AA	Set Host Watchdog Timeout Value	

4.4.3 General Command Sets

Command	Response	Description	Page
%AANNTTCCFF	!AA	Set Module Configuration	33
# AA	>(Data)	Read Analog Input	40
# AAN	>(Data)	Read Analog Input from channel N	41
$ AA0	!AA	Perform Span Calibration	42
$ AA1	!AA	Perform Zero Calibration	43
$ AA2	!AATTCCFF	Read Configuration	44
$ AA3	> (Data)	Read CJC Temperature	45
$ AA5VV	!AA	Set Channel Enable	46
$ AA6	!AAVV	Read Channel Status	47
$ AA9SNNNN	!AA	Set CJC Offset Value	48
$ AAF	!AA(Data)	Read Firmware Version	*49*
$ AAM	!AA(Data)	Read Module Name	51
~AAC	!AAN	Read the CJC status	52
~AACN	!AA	Enable/disable CJC	53
~AAEV	!AA	Enable/Disable Calibration	61
~AAI	!AA	Enable soft INIT function	62
~AAO(Data)	!AA	Set Module Name	63
~AATnn	!AA	Set soft INIT time-out value	

4.5 Set brand compatible

Modules:	All eDAM modules	
Description:	Set compatible to other brand	
Command:	~AA2X01V[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	2X	eDAM exclusive code
	01	Set Compatible command.
	V	Brand ID 0= eDAM , 1=ADAM 2=NuDAM, 3=I-7000
	CHK	Check sum
	(cr)	Carriage return
Response:	!AA[CHK](cr)	Valid Command
	?AA[CHK](cr)	Invalid Command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	CHK	Check sum
	(cr)	Carriage return and then reboot module

Note: Module will be set to default states after this command issued

Example: Set eDAM-8018 module with ID=02 to command compatible to NuDAM-6018

Command: ~022X012(cr)

Response: !02((cr)

Example: Set eDAM-8018 module with ID=02 to command compatible to I-7018

Command: ~022X013(cr)

Response: !02(cr)

4.6 Read current brand setting

Modules:	All eDAM modules	
Description:	Read current brand ID setting	
Command:	~AA2X02[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	2X	eDAM exclusive code
	02	Read Brand ID com..
	CHK	Check sum
	(cr)	Carriage return
Response:	!AAV[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	V	Brand ID
	CHK	Check sum
	(cr)	Carriage return

Example: Read current brand ID of eDAM-8018 module with ID=02

Command: ~022X02(cr)

Response: !022((cr) // Compatible to uDAM-6018

4.7 Set Module configuration

Modules:	8017,8018	
Description:	Set module configuration	
Command:	%AANNTTCCFF[CHK](cr)	
Syntax:	%	Command leading code
	AA	Module address ID (00 to FF)
	NN	New eDAM address ID (00 to FF)
	TT	Analog input range (See *)
	CC	Set new baud rate of module (See **)
	FF	Data format (See ***)
	CHK	Check sum
	(cr)	Carriage return
Response:	!AA[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	New Module address ID
	CHK	Check sum
	(cr)	Carriage return

***Analog Input type and range (TT)**

Type code	range	Modules
08	±10 V	8012/D,8017,8014/D
09	±5 V	8012/D,8017,8014/D
0A	±1 V	8012/D,8017,8014/D
0B	±500 mV	8012/D,8017,8014/D
0C	±150 mV	8012/D,8017,8014/D
0D	±20 mA	8012/D,8017, 8014/D (Required 125Ω current conversion resistor.)

**** Baud Rate settings (CC)**

code	03	04	05	06	07	08	09	0A
baud rate	1200	2400	4800	9600	19200	38400	57600	115200

*** :**Data format settings (FF)**

Bit	7	6	5	4	3	2	1	0

Bit7: =0 for 60 Hz (default)
=1 for 50 Hz
Bit6: =1 Enable checksum
=0 Disable checksum (default setting)
Bit5: =0 for normal operation mode (Default setting)
=1 for fast operation mode (8017 only)
Bit4~bit2: No used
Bit1~bit0:=00 Engineer unit format (default setting)
=01 Percent format
=11 2's complement Hex format

Input types and data format table

Code	Range	Format	+F.S.	zero	-F.S
00	-15~+15 mV	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~+50 mV	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~+100 mV	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~+500 mV	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~+1 V	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~+2.5 V	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~+20 mA	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

Code	Range	Format	+F.S.	zero	-F.S
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
16	Type C T/C -270~2320°C	Engineer unit	+2320.0	+0000.0	-0000.0
		% of F.S.R	+100.00	+000.00	-100.00
		2's complement	7FFF	0000	0000

Note:

It's needed to short the INIT* pin to ground while changing baud rate and/or enable/disable checksum (see following examples) or using " **~AAI**" soft INIT command to change baud rate and/or checksum without shorting INIT* pin to ground (see)

Example 1: Change ID address from 01 to 03 (Assume current baud rate is 9600 and checksum disabled)

Command: %0103080600(cr)
Response: !03(cr)
response new module ID address 03 (change ID address only)

Example 2: Change baud rate from 9600 to 19200(Assume current ID is 03, baud rate is 9600, and checksum disabled).

Because that the baud rate is changed from 9600 to 19200, the following procedures should be done before sending this command

1. Power off the module
2. Short INIT* pin to GROUND (see Appendix C)
3. Power on the module
4. send command string
5. Command: %0003080700(cr)
6. Response: !03(cr)
7. response module ID address 03
8. Power off module
9. Open INIT* pin and power on module again

Example 3: Enable checksum(Assume current ID is 03, baud rate is 9600 and checksum disabled).

Because that the checksum is changed from disable to enable, the following procedures should be done before sending this command

1. Power off the module
2. Short INIT* pin to GROUND (see Appendix C)
3. Power on the module
4. send command string
5. Command: %0003080640(cr)
6. Response: !03(cr)
7. response module ID address 03
8. Power off module
9. Open INIT* pin and power on module again (checksum enabled)

Example 4: Change baud rate from 9600 to 19200 and enable checksum (Assume current ID is 03, baud rate is 9600 and checksum disabled).

Because that both the baud rate and checksum is changed , the following procedures should be done before sending this command

1. Power off the module
2. Short INIT* pin to GROUND (see Appendix C)
3. Power on the module
4. send command string
5. Command: %0003080740(cr)
6. Response: !03(cr)
7. response module ID address 03
8. Power off module
9. Open INIT* pin and power on module again (Baud rate changed to 19200 and checksum enabled)

It is recommended to use the setup utility to configure the module (see section 3.7 and 3.8)

Related topics: $AA2

4.8 Read analog data

Modules:	For,8017,8018	
Description:	Read the ANALOG input value	
Command:	#AA[CHK](cr)	
Syntax:	#	Command leading code
	AA	Module address ID (00 to FF)
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	>(data)[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	>	Delimiter for valid command
	?	Delimiter for invalid command
	(data)	Analog input data(see *)
	CHK	Check sum
	(cr)	Carriage return

If analog data of eDAM-8018 module be read by using this command, data of all channels are responded as follows:
>(chan.0 data) (chan.1 data) …… (chan.7 data) [CHK](cr)

Example 5: Read analog input data from eDAM8018 at addr.=05

Command: #05(cr) Response:
+02.645-01.001+03.023+00.321+08.123-03.333+09.210-06.000(cr)

Related command: $AA4

4.9 Read data from channel N

Modules:	For 8017,8018		
Description:	Read the analog input value of a specified AD channel from an analog input module		
Command:	#AAN[CHK](cr)		
Syntax:	#	Command leading code	
	AA	Module address ID (00 to FF)	
	N	Command for reading analog input value	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	>(data)[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	>	Delimiter for valid command	
	?	Delimiter for invalid command	
	(data)	Analog input data	
	CHK	Check sum	
	(cr)	Carriage return	

Example 6: Read the analog input channel 1 of AD module at address 06 in the network. (Data format is engineering unit)

User command: #061<CR>
Response: >+1.6888<CR>

Related command: $AA

4.10 Perform Span calibration

Modules:	For 88017,8018		
Description:	To correct the gain errors of AD converter by using the span calibration.		
Command:	$AA0[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	0	Command for span calibration	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	!AA[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID (00 to FF)	
	CHK	Check sum	
	(cr)	Carriage return	

Note: To perform the calibration, a proper input signal should be connected to the analog input module. Different input range have different input voltage, detail refer Appendix B **"Calibratio**n".

Example 7: Perform span calibration of module with address=06

Command: $060<CR>
Response: !06<CR>

4.11 Perform Offset calibration

Modules:	For 8017,8018		
Description:	To correct the offset errors of AD converter by using the offset calibration		
Command:	$AA1[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	1	Command for offset calibration	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	!AA[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID (00 to FF)	
	CHK	Check sum	
	(cr)	Carriage return	

Note:
To perform the calibration, a proper input signal should be connected to the analog input module. Different input range have different input voltage, detail refer Appendix B **"Calibratio**n".

Example 8: Perform offset calibration of module with address=06

Command: $061<CR>
Response: !06<CR>

4.12 Read Configuration

Modules:	For eDAM modules		
Description:	Read module configuration		
Command:	$AA2[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	2	Command for reading configuration	
	CHK	Check sum	
	(cr)	Carriage return	
Response:	!AATTCCFF[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID	
	TT	Analog input type and range (see sec.4.7)	
	CC	Baud rate (see sec.4.7)	
	FF	Data format of module (see sec.4.7)	
	CHK	Check sum	
	(cr)	Carriage return	

Example 9: Read configuration of module with ID address=05

Command: $052(cr)
Response: !05080600(cr)
Read address ID=05 module configuration
08=Analog input range ±10 V
06=9600 baud rate
00=no checksum,

Related command: $AA2

4.13 Read CJC temperature

Modules:	For 8018 only	
Description:	Read CJC temperature.	
Command:	$AA3[CHK](cr)	
Syntax:	$	Command leading code
	AA	Module address ID (00 to FF)
	3	Command for reading CJC temp.
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	>AA(data) [CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	>	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	data	CJC temperature in degree Celsius, including a sign byte, '+' or '-', and followed by 5 decimal digital with fixed decimal point in tenths of a degree
	CHK	Check sum
	(cr)	Carriage return

Example 10: Read CJC temperature at address ID=03

Command: $013<cr>
Response: >+0028.5<cr> CJC temperature is +28.5℃

Related command: $AA9SNNNN

4.14 Enable/disable channel for multiplexing

Modules:	For 8017,8018	
Description:	Enable/Disable multiplexing simultaneously for individual channel.	
Command:	$AA5VV[CHK](cr)	
Syntax:	$	Command leading code
	AA	Module address ID (00 to FF)
	5	Command for reading digital I/O status
	VV	bit 3~0 of 1st character control channel 7-4 bit 3~0 of 2nd character control channel 3-0 bit value 0: Disable channel bit value 1: Enable channel
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID (00 to FF)
	CHK	Check sum
	(cr)	Carriage return

Example 1: Enable channel 3 and channel 6, the other channels are all disable of eDAM-8018.

Command: $06548<cr>
'48' is 01001000 that means enable channel 3 and channel 6, the other channels are all disable.
Response: !06<cr>
Related command:

4.15 Read channel status

Modules:	For eDAM 8017 only		
Description:	Read the enable/disable status the channels of eDAM-8017		
Command:	$AA6[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	6	Command for reading channel status	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	!AAVV[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID (00 to FF)	
	VV	Channel status (See sec 4.14)	
	CHK	Check sum	
	(cr)	Carriage return	

Example 1: Read channel status of eDAM-8017 with address=06.

Command: $066<CR>
Response: !0648<CR>

4 is equals binary 0100 that means enable channel 6 and disable channel 7, 5, 4.
8 is equals binary 1000 that means enable channel 3 and disable channel 2, 1, 0.

Related command:

4.16 Set CJC offset value

Modules:	For eDAM 8018 only		
Description:	Set Cold junction offset of eDAM-8018		
Command:	$AA9SNNNN[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	9	Command for setting CJC offset value	
	SNNNN	CJC offset value including a sign and 4 hexadecimal digits from –1000 to +1000, each count is 0.01℃	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	!AA[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID (00 to FF)	
	CHK	Check sum	
	(cr)	Carriage return	

Example 1: Set Address 01 CJC offset to increase 16 counts(+0.16 ℃).

Command: $019+0010<CR>
Response: !01<CR>

Related command: $AA3

4.17 Read firmware version

Modules:	For eDAM DIO modules		
Description:	Read module's firmware version.		
Command:	$AAF[CHK](cr)		
Syntax:	$	Command leading code	
	AA	Module address ID (00 to FF)	
	F	Command for reading firmware version.	
	CHK	Check sum	
	(cr)	Carriage return	
Response: (see Note)	!AA(data)[CHK](cr)		Valid command
	?AA[CHK](cr)		Invalid command
	!	Delimiter for valid command	
	?	Delimiter for invalid command	
	AA	Module address ID	
	data	Module's firmware version.	
	CHK	Check sum	
	(cr)	Carriage return	

Example 2: Read firmware version of module address ID=30

Command: $30F<CR>

Response: !30A1.04<CR>

! Command is valid., Address ID=30, Firmware Version=A1.04

4.18 Reset module

Modules:	All eDAM modules	
Description:	Reset all existing eDAM modules	
Command:	$AARS[CHK](cr)	
Syntax:	$	Command leading code
	AA	Module address ID (00 to FF)
	RS	Reset command
	CHK	Check sum
	(cr)	Carriage return
Response:	No response	

Note: Reset command will reset module to default settings.
This command has no response from module

Example 3: Example: Reset module with ID address is 02

Command: $02RS(cr)

Response: No response

Related command: $AAM

4.19 Read module name

Modules:	For eDAM DIO modules	
Description:	Read module‘s name	
Command:	$AAM[CHK](cr)	
Syntax:	$	Command leading code
	AA	Module address ID (00 to FF)
	M	Command for reading module’s name
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA(data)[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	data	Module‘s name
	CHK	Check sum
	(cr)	Carriage return

Example 4: Read module’s name of address ID=30

Command: $30M<CR>

Response: !308014<CR>

! Command is valid., Address ID=30, module’s name=8014

Related command: $AARS

4.20 Read CJC status

Modules:	For eDAM 8018 only	
Description:	Read Cold junction status of eDAM-8018	
Command:	~AAC[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	C	Command for reading CJC status
	CHK	Check sum
	(cr)	Carriage return
Response:	!AAN[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID (00 to FF)
	N	0: CJC disabled 1: CJC enabled
	CHK	Check sum
	(cr)	Carriage return

Example 1: Read Address 01 CJC status and response CJC enabled

Command: ~01C<CR>

Response: !011<CR>

Related command: ~AACN

4.21 Enable/disable CJC

Modules:	For eDAM 8018 only	
Description:	Enable/disbale Cold junction of eDAM-8018	
Command:	~AACN[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	C	Command for reading CJC status
	N	0: disable CJC 1: enable CJC
	CHK	Check sum
	(cr)	Carriage return
Response:	!AAN[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID (00 to FF)
	CHK	Check sum
	(cr)	Carriage return

Example 1: Enable Address 01 CJC

Command: ~01C1<CR>

Response: !01<CR>

Related command: ~AAC

4.22 Host OK

Modules:	For all eDAM modules	
Description:	Host send this command to all modules for send the information “Host OK”	
Command:	~**[CHK](cr)	
Syntax:	~	Command leading code
	**	For all modules
	CHK	Check sum
	(cr)	Carriage return
Response:	No response	

Note:

When host watchdog timer is enable, host computer must send this command to all module before timeout otherwise **“Host watchdog timer enabled”** module‘s output value will go to safety state output value.

4.23 Read module status

Modules:	For all eDAM modules	
Description:	Read watchdog timeout status	
Command:	~AA0[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	0	Command for reading timeout status
	CHK	Check sum
	(cr)	Carriage return
Response:	! AASS[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	SS	SS=00 - watchdog timeout is cleared SS=04 - watchdog timeout is set
	CHK	Check sum
	(cr)	Carriage return

Note:

1. the watchdog timeout status will be stored in EEPROM of the module and can only be cleared by issuing ~AA1 command (see ~AA1 and ~AA3EVV commands)
2. When the module's watchdog timeout value is reached, this command will be responded with SS=04 otherwise SS=00

Example:

Command: ~010<cr> Response: !0104

The host watchdog timeout status is set

4.24 Reset module status

Modules:	For all eDAM modules	
Description:	Reset watchdog timeout status	
Command:	~AA1[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	1	Command for resetting watchdog timeout status
	CHK	Check sum
	(cr)	Carriage return
Response:	! AA [CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	CHK	Check sum
	(cr)	Carriage return

Note:

1. The module's watch dog status will be cleared after this command issued
2. (reference to ~AA3EVV command)

4.25 Read host watchdog timeout value

Modules:	For all eDAM modules	
Description:	Read host watchdog timeout value	
Command:	~AA2[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	2	Command for reading watchdog timeout value
	CHK	Check sum
	(cr)	Carriage return
Response:	! AAEVV[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	E	Host watchdog enable/disable status E=1 – Enabled E=0 – Disabled
	VV	Timeout value in Hex format from 01 to FF=25.5 seconds (one unit is 0.1 sec)
	CHK	Check sum
	(cr)	Carriage return

(also see sec 4.26)

4.26 Set host watchdog timeout value

Modules:	For all eDAM modules	
Description:	Set host watchdog timeout value	
Command:	~AA3EVV[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	3	Command for setting watchdog timeout value
	E	1= enable, 0= disable Host watchdog
	VV	Timeout value (01~FF, each for 0.1 second)
	CHK	Check sum
	(cr)	Carriage return
Response:	! AA [CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	CHK	Check sum
	(cr)	Carriage return

Note:
If host watchdog timer is enabled, the host should send *Host OK* (see section 4.22) command periodically within Timeout value to refresh the timer, otherwise the module will be forced to safety state

Set module (ID=04) to have watchdog timeout value 10.0 seconds and enable host watchdog
Command: ~043164<cr>
Set watchdog timeout value 10.0 sec and enable host watchdog
Response: !04<cr> Valid command

Example 2: Read watchdog timeout value form module (ID=04)

Command: ~042<cr>
Read watchdog timeout value
Response: !04164
Watchdog timeout value=10.0 seconds, and host watchdog is enabled

Example 3: Reset watchdog timer

Command: ~**<cr>
Read host watchdog timer

Stop sending any command string to modules for at least 10.0 seconds. The LED on the module will go to flash. The flash LED indicates the host watchdog is timeout and timeout status is set

Example 4: Read watchdog timeout status

Command: ~040<cr>
Read module (ID=04) watchdog timeout status
Response: !0404<cr>
Timeout status is set

Example 5: Read watchdog timeout value form module (ID=04)

Command: ~042<cr>
Read watchdog timeout value
Response: !04164
Watchdog timeout value=10.0 seconds, and host watchdog is **enabled**

Example 6: Reset watchdog timeout status

Command: ~041<cr>
Reset watchdog timeout status
Response: !04<cr>
Watchdog timeout is cleared and LED stop flashing, and host watchdog is disabled

Example 7: Read watchdog timeout status

Command: ~040<cr>
Read module (ID=04) watchdog timeout status
Response: !0400<cr>
Timeout status is cleared

4.27 Enable/disable calibration

Modules:	For 8012,8012D,8014,8014D,8017	
Description:	Enable or disable Span calibration.	
Command:	~AAEV[CHK](cr)	
Syntax:	!	Command leading code
	AA	Module address ID (00 to FF)
	E	Enable/disable calibration command
	V	0=Disable span calibration 1=Enable span calibration
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID (00 to FF)
	CHK	Check sum
	(cr)	Carriage return

Note: send enable calibration command before performing the calibration,

Example 8: Perform span calibration of module with address=06

Command: $06E1<CR> // Enable calibration
Response: !06<CR>
Command: $060<CR> // perform span calibration
Response: !06<CR>

4.28 Enable soft INIT function

Modules:	For 8018 only	
Description:	Enable soft INIT function	
Command:	~AAI[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	I	Enable soft INIT function
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	CHK	Check sum
	(cr)	Carriage return

Note:
This command is used to enable modification of Baud Rate and Check Sum settings without shorting INIT pin

Example 9: Enable Address 06 software INIT function

Command: ~06I<CR> // Enable software INIT function
Response: !06<CR>

Related command: ~AATnn and %AANNTTCCFF

4.29 Set module name

Modules:	For all eDAM modules	
Description:	Set new module name.	
Command:	~AAO(data)[CHK](cr)	
Syntax:	$	Command leading code
	AA	Module address ID (00 to FF)
	O	Command for setting new name
	(data)	Module name, Max. 6 characters
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA [CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	AA	Module address ID
	CHK	Check sum
	(cr)	Carriage return

Example 10: Set new module name at address ID=30

Command: ~30O4012<CR>

Set new name 4012 to the module at address ID=30

Response: !30<CR>

! Command is valid.,

Related command:

4.30 Set the soft INIT time out value

Modules:	For 8018 only	
Description:	Set the soft INIT time out value	
Command:	~AATnn[CHK](cr)	
Syntax:	~	Command leading code
	AA	Module address ID (00 to FF)
	T	Enable soft INIT function
	nn	nn= Two hexadecimal digits representing time out value in seconds. The max. time out value is 60 seconds
	CHK	Check sum
	(cr)	Carriage return
Response: (see Note)	!AA[CHK](cr)	Valid command
	?AA[CHK](cr)	Invalid command
	!	Delimiter for valid command
	?	Delimiter for invalid command
	CHK	Check sum
	(cr)	Carriage return

Note:

When changing Baud Rate and/or Checksum without altering INIT* pin, ~AAI,%AANNTTCCFF commands should be sent consecutively and the time interval between two commands should be less than soft INIT time out value.

If soft INIT time out valu=0, then the Baud rate and/or checksum can not be changed by using ~AAI,%AANNTTCCFF soft commands.

The power on default value = 0 seconds

Example 1: Enable Address 06 software INIT* function

Command: ~06I<CR> // Enable software INIT* function
Response: !06<CR>
Command: ~0606000700<CR> // change baud rate
Response: ?06<CR> // attempts to change Baud
// rate/without first altering INIT*
// pin or the soft INIT time out
// value=0

Example 2: Enable Address 06 software INIT* function

Command: ~06T10<CR> // soft INIT* time out value=10
seconds
Response: !06<CR>
Command: ~06I<CR> // Enable software INIT* function
Response: !06<CR>
Command: ~0606000700<CR> //change baud rate
Response: !06<CR> // Baud rate changed successfully

Related command: ~AATnn and %AANNTTCCFF

Appendix A Data Format and Input range

4.31 Data Format of Analog Input Modules

There are three types of data format used in analog input modules.

1. Engineering units.
2. Percent of FSR (Full Scale Range).
3. Two's complements hexadecimal.

4.31.1 Engineering Units

Example 3: Input Range is ±5 V
Input is -1.37 Volts
engineering units: **-1.3700**<CR>

Example 4: Input Range is ±10 V
Input is +3.653 Volts
engineering units: **+03.653**<CR>

4.31.2 Percent of FSR (Full Scale Range)

Example 5: Input Range is ±5 V
Input is +1 Volts
% of FSR: +020.00<CR> (+(20/100) x 5 V) = +1 V

Example 6: Input Range is ±10 V
Input is +4 Volts
% of FSR: +040.00<CR> (+(40/100) x 10 V) = +4 V

4.31.3 Two's Complement Hexadecimal

Example 7: Input Range is ±5 V
Input is +1 Volts
Two's complement hexadecimal: 1999<CR>
((1/5) x 32768) = 6553.6 = 1999H

Example 8: Input Range is ±5 V

Input is -2 Volts

Two's complement hexadecimal: CD27<CR>

((-2/5) x 32768) = -13107.2 = CD27H

Example 9: Input Range is ±10 V

Input is +4 Volt

Two's complement hexadecimal: 3333<CR>

((4/10) x 32768) = 13107.2 = 3333H

Appendix B Calibration

4.32 Calibration for eDAM8018

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

The span calibration is used to calibrate the full scale output when the input is full scale voltage

- Calibration procedures
  1. Apply zero voltage to channel 0 of analog module (refer to sec 2.5)
  2. Issues configuration command with type=08 (refer to sec 4.7)
  3. Issues enable calibration command
  4. Issues zero offset calibration command five times
  5. Apply span voltage to channel 0 of analog module
  6. Issues span calibration command five times
  7. Repeat procedure 1 to procedure 6 two times

Type code	08	09	0A	0B	0C	0D
Zero input	0V	0V	0V	0mV	0mV	0mA
Span input	+10V	+5V	+1V	+500mV	+150mV	+20mA

Table 7-1: 8012/D,8014/D,8017 Calibration voltages

Note:

While using calibration type 0D to calibrate 8012/D, 8017 module, an external shunt resistor 125 ohms should be connected to channel 0 of module

Appendix C INIT* pin operation

The “INIT* mode” has two purposes, one for reading module current configuration, and another for configuring the module baud rate and checksum

■ Reading module current configuration

Each eDAM module has a built-in EEPROM which is used to store the configuration information such as address ID, type, baud rate etc..

If the user unfortunately forget the configuration of the module. User may use a special mode called “INIT* mode” to resolve the problem

When the module is set to “INIT*” mode”, the default settings are ID=00, baud rate=9600, and checksum = disable

The following steps show you how to enable INIT* mode and read the current configuration

,

Power off the module

Connect the “INIT*” pin to GND pin

Power on the module

Send command $002<cr> in 9600 baud rate to read the current configuration stored in the EEPROM

Power off the module again

Open “INIT*” pin to force the module to normal mode

■

Configuring the module baud rate and checksum

The module should be set to “INIT* mode”, While changing baud rate and/or checksum state by sending “Set module configuration” command (see section 4.7)

The following steps show you how to enable INIT* mode and change baud rate and/or checksum state

Power off the module

Connect the “INIT*” pin to GND pin

Power on the module

Send command %AANNTTCCFF in 9600 baud rate to set baud rate and/or checksum state (*ID should be set to 00 in “INIT* mode”*)

Power off the module again

Open “INIT*” pin to force the module to normal mode