I-87084W
Command Sets

ICP DAS CO., LTD.
Revision: 1.1
2014/05/21

Table of Contents

1.0 Default Settings

Default settings for the I-87084W modules are as follows:

- Protocol: DCON
- Module Address: 01
- Type: Type 50, up counter
- Baud Rate: 115200 bps

1.1 Configuration Tables

Baud Rate Setting (CC)

7	6	5	4	3	2	1	0
Data		Baud					

Key	Description
Baud	Baud Rate 03: 1200 04: 2400 05: 4800 06: 9600 07: 19200 08: 38400 09: 57600 0A: 115200
Data	Data Format 0: eight data bits, no parity, and one stop bit 1: eight data bits, no parity, and two stop bit 2: eight data bits, even parity, and one stop bit 3: eight data bits, odd parity, and one stop bit

Type Setting (TT)

Type Code	Input Type
50	Up counter
51	Frequency
54	Up/down counter
55	Pulse/direction counter
56	Quadrant counter

Data Format Setting (FF)

7	6	5	4	3	2	1	0
RD	CS	RD				DF	

Key	Description
DF	Data format 00: Engineering unit. The data consist of '+', 6 digits and one decimal point. This data format is only for the frequency type. 10: Hexadecimal
CS	Checksum setting 0: Disabled 1: Enabled
RD	Reserved

Note: The reserved bits should be zero.

Counter Type and Data Format Table

Type Code	Counter type	Data Format	Min	Max
50	Up	Hex	00000000	FFFFFFFF
54	Up/down	Hex	80000000	7FFFFFFF
55	Pulse/direction	Hex	80000000	7FFFFFFF
56	Quadrant	Hex	80000000	7FFFFFFF

1.2 Frequency Measurement

Frequency is usually measured by one of the following two methods.

1. by counting number of signal pulses during a known time interval, gate time
2. by counting number of pulses of a reference clock during the signal period

I-87084W uses the second method for the frequency measurement. It counts number of pulses of the reference clock during one signal period. The drawback of this method is that the resolution will be small when the frequency is high. M-7084 provides the high frequency measurement mode by counting 11 signal periods, instead of only one period, to increase resolution. The high frequency measurement mode can be enabled by @AAFHVV command. Refer to Section 2.29 for details. The side effects of the high frequency measurement mode are that it takes longer time and the counting may overflow when the signal frequency is very low. M-7084 can be set to automatic frequency measurement mode by @AAFAVV command. Refer to Section 2.27 for details. In automatic frequency measurement mode, the channel is switched to the high frequency measurement mode when the signal frequency is high and to the low frequency measurement mode when the signal frequency is low automatically.

2.0 DCON Protocol

All communication with I-87K I/O modules consists of commands generated by the host and responses transmitted by the I-87K I/O modules. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The ID is 01 by default and can be changed using a user command. All commands to the modules contain the ID address, meaning that only the addressed module will respond. The only exception to this is command ~** (Section 2.36), which is sent to all modules, but the modules do not reply to the command.

Command Format:

Leading Character	Module Address	Command	[CHKSUM]	CR

Response Format:

Leading Character	Module Address	Data	[CHKSUM]	CR

CHKSUM A 2-character checksum which is present when the checksum setting is enabled. See Sections 1.1 (Data Format Setting) and 2.1 for details.

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

Checksum Calculation:

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

Example:

Command string: $012(CR)

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

Response string: !01200600(CR)

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

Note:

All characters should be in upper case.

General Command Sets			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the Module Configuration	2.1
$AA2	!AANNTTCCFF	Reads the Module Configuration	2.6
$AA5	!AAS	Reads the Reset Status	2.11
$AAF	!AA(Data)	Reads the firmware Version	2.19
$AAI	!AAS	Reads the INIT Status	2.20
$AAM	!AA(Data)	Reads the Module Name	2.21
$AAP	!AASC	Reads the communication protocol	2.22
$AAPN	!AA	Sets the communication protocol	2.23
~AAI	!AA	Software INIT	2.41
~AAO(Name)	!AA	Sets the Module Name	2.42
~AARD	!AAVV	Reads the Response Delay Time	2.43
~AARDVV	!AA	Sets the Response Delay Time	2.44
~AATnn	!AA	Sets the Software INIT Timeout	2.45

Counter/Frequency Command Sets			
Command	**Response**	**Description**	**Section**
#AA	>(Data)	Reads the Input Data of All Channels	2.2
#AAN	>(Data)	Reads the Input Data of the Specified Channel	2.3
$AA0N	!AA(Data)	Reads the Low Pass Filter Time of the Specified Channel	2.4
$AA0N(Data)	!AA	Sets the Low Pass Filter Time of the Specified Channel	2.5
$AA3N	!AA(Data)	Reads the Maximum Counter Value of the Specified Channel	2.7
$AA3N(Data)	!AA	Sets the Maximum Counter Value of the Specified Channel	2.8
$AA4	!AAVV	Reads the Low Pass Filter Channel Mask	2.9
$AA4VV	!AA	Sets the Low Pass Filter Channel Mask	2.10
$AA5VV	!AA	Starts/Stop Counting	2.12
$AA6	!AAVV	Reads the Start/Stop Counting Status	2.13

Command	Response	Description	Section
$AA6N	!AA	Clears the Counter Value of the Specified Channel	2.14
$AA7	!AAVV	Shows the Overflow Status of All Channels	2.15
$AA7VV	!AA	Clears the Overflow Status	2.16
$AA7CNRVV	!AA	Sets the Single Channel Type	2.17
$AA8CN	!AACNRVV	Reads the Channel Type of the Specified Channel	2.18
@AABB	!AAVV	Reads the Battery Backup Status	2.24
@AABBVV	!AA	Sets the Battery Backup Status	2.25
@AAFA	!AAVV	Reads the Auto Frequency Mode Status	2.26
@AAFAVV	!AA	Sets the Auto Frequency Mode Status	2.27
@AAFH	!AAVV	Reads the High Frequency Mode Status	2.28
@AAFHVV	!AA	Sets the High Frequency Mode Status	2.29
@AAFT	!AAVV	Reads the Frequency Measurement Timeout	2.30
@AAFTVV	!AA	Sets the Frequency Measurement Timeout	2.31
@AAGN	!AA(Data)	Reads the Preset Counter Value of the Specified Channel	2.32
@AAPN(Data)	!AA	Sets the Preset Counter Value of the Specified Channel e	2.33
@AASC	!AAVV	Reads the Stop Counting on Overflow Status	2.34
@AASCVV	!AA	Sets the Stop Counting on Overflow Status	2.35

Host Watchdog Command Sets			
Command	**Response**	**Description**	**Section**
~**	No Response	Host OK	2.36
~AA0	!AASS	Reads the Status	2.37
~AA1	!AA	Resets the Status	2.38
~AA2	!AAVV	Reads the Timeout Settings	2.39
~AA3EVV	!AA	Sets the Timeout Settings	2.40

2.1 %AANNTTCCFF

Description:

Sets the configuration of a module.

Syntax:

%AANNTTCCFF[CHKSUM](CR)

%	Delimiter character
AA	Address of the module to be configured in hexadecimal format (00 to FF)
NN	New address of the module in hexadecimal format (00 to FF)
TT	New type code. Not used by the I-87084W. It should be set to 00 for the I-87084W
CC	New Baud Rate code, see Section 1.1 for details. To change the Baud Rate, the module should be in INIT* mode.
FF	Used to set the data format and checksum (Section 1.1). To change the checksum setting, the module should be in INIT* mode.

Response:

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

! Delimiter for a valid command

? Delimiter for an invalid command. If the **Baud Rate** or **checksum** settings are changed without setting the module in INIT* mode, the module will return an invalid command.

AA Address of the module in hexadecimal format (00 to FF)

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

Examples:

Command: %0102000600 Response: !02

Changes the address of module 01 to 02. The module returns a valid response.

Command: %0101000A00 Response: ?01

Changes the Baud Rate of module 01 to 115200bps. The module returns an invalid command, because it is not in INIT* mode.

Command: %0101000A00 Response: !01

Changes the Baud Rate of module 01 to 115200bps and the module is in INIT* mode. The module returns a valid response.

Related Commands:

Section 2.6 $AA2, Section 2.41 ~AAI, Section 2.45 ~AATnn

Related Topics:

Section 1.1 Configuration Tables

Notes:

1. Changes to the address and data format settings take effect immediately after a valid command is received. Changes to the Baud Rate and checksum settings take effect on the next power-on reset.
2. For the I-87084W, changing the Baud Rate and checksum settings can be achieved using software only and is performed using the following commands:
   I. Send a ~AATnn command. See Section 2.45 for details.
   II. Send a ~AAI command. See Section 2.41 for details.
   III. Send a %AANNTTCCFF command.
   If the command is valid, the Baud Rate and checksum settings will be changed after the module responds with !AA.

2.2 #AA

Description:
Reads the data from every input channels.

Syntax:
#AA[CHKSUM](CR)

#	Delimiter character
AA	Address of the module to be read (00 to FF)

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

>	Delimiter character for a valid response
?	Delimiter character for an invalid response
(Data)	Data from every input channels, see Section 1.1 for the details of data format.

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

Examples:
Command: #01 Response:
>0000123400005678000 09ABC0000DEF000001111000022 2200000333300004444

Reads module 01 and receives the data.

Related Commands:
Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:
Section 1.1 Configuration Tables

2.3 #AAN

Description:

Reads the input of channel N.

Syntax:

#AAN[CHKSUM](CR)

#	Delimiter character
AA	Address of the module to be read (00 to FF)
N	The channel to be read, zero based.

Response:

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

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

>	Delimiter character for a valid response
?	Delimiter character for an invalid response. An invalid command is returned if the specified channel is incorrect.
(Data)	Input data of the specified channel, see Section 1.1 for details of the data format.
AA	Address of the responding module (00 to FF)

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

Examples:

Command: #032 Response: >00001234

Reads data from channel 2 of module 03.

Command: #029 Response: ?02

Reads data from channel 9 of module 02. An error is returned because channel 9 is invalid.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.6 $AA2

Related Topics:

Section 1.1 Configuration Tables

2.4 $AA0N

Description:

Reads the low pass filter time of channel N.

Syntax:

$AA0N[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
0	Command to read the low pass filter time of a channel
N	Specifies the channel to be read (0 ~ 7)

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Data)	Low pass filter time is micro seconds, consisting of 5 decimal digits in the range 00001 ~ 32767.

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

Examples:

Command: $0103 Response: !0100010

Reads the channel 3 low pass filter time of module 01 and the module responds with 00010 which means 10 micro seconds.

Related Commands:

Section 2.5 $AA0N(data), Section 2.9 $AA4, Section 2.10 $AA4vv

Notes:

1. The low pass filter time should be the same for channel 0 and 1. When it is set to either one of the channel 0 or 1, both channels will be set to the same value.
2. The low pass filter time should be the same for channel 2 and 3. When it is set to either one of the channel 2 or 3, both channels will be set to the same value.
3. The low pass filter time should be the same for channel 4, 5, 6 and 7. When it is set to either one of the channel 4, 5, 6 or 7, all of the 4 channels will be set to the same value.

2.5 $AA0N(Data)

Description:

Sets the low pass filter time of a channel.

Syntax:

$AA0N(Data)[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
0	Command to read the low pass filter time of a channel
N	Specifies the channel to be read (0 ~ 7)
(Data)	Low pass filter time is micro seconds, consisting of 5 decimal digits in the range 00001 ~ 32767.

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $010300010 Response: !01

Sets the channel 3 low pass filter time of module 01 to 10 micro seconds and returns a valid response.

Related Commands:

Section 2.4 $AA0N, Section 2.9 $AA4, Section 2.10 $AA4vv

Notes:

1. The low pass filter time should be the same for channel 0 and 1. When it is set to either one of the channel 0 or 1, both channels will be set to the same value.
2. The low pass filter time should be the same for channel 2 and 3. When it is set to either one of the channel 2 or 3, both channels will be set to the same value.
3. The low pass filter time should be the same for channel 4, 5, 6 and 7. When it is set to either one of the channel 4, 5, 6 or 7, all of the 4 channels will be set to the same value.

2.6 $AA2

Description:

Reads the module configuration.

Syntax:

$AA2[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
2	Command to read the module configuration

Response:

Valid Command: **!AATTCCFF[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
TT	Type code of the module, should be 00 for I-87084W.
CC	Baud Rate code of the module, see Section 1.1 for details.
FF	Checksum and counter update direction settings of the module, see Section 1.1 for details.

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

Examples:

Command: $012 Response: !01000600

Reads the configuration of module 01.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.1 Configuration Tables

2.7 $AA3N

Description:

Reads the maximum counter value of channel N.

Syntax:

$AA3N[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
3	Command to read the maximum counter value of a channel
N	Specifies the channel to be read (0 ~ 7)

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Data)	The maximum counter value in eight hexadecimal digits

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

Examples:

Command: $0132 Response: !01FFFFFFFF

Reads the channel 2 maximum counter value of module 01 and the module responds with FFFFFFFF.

Related Commands:

Section 2.8 $AA3N(Data)

Note:

The maximum counter value is only available to the channels that are set to up counter type (type code 50).

2.8 $AA3N(Data)

Description:

Sets the maximum counter value of a channel.

Syntax:

$AA3N(Data)[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
3	Command to set the maximum counter value of a channel
N	Specifies the channel to be read (0 ~ 7)
(Data)	The maximum counter value in eight hexadecimal digits

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $0132F0000000 Response: !01

Sets the channel 2 maximum counter value of module 01 to 0xF0000000 and returns a valid response.

Related Commands:

Section 2.7 $AA3N

Note:

The maximum counter value is only available to the channels that are set to up counter type (type code 50).

2.9 $AA4

Description:

Reads the enabled/disabled low pass filter status of each channel.

Syntax:

$AA4[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
4	Command to read the low pass filter status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the low pass filter is enable for the channel and 0 means that the low pass filter is disabled for the channel.

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

Examples:

Command: $014 Response: !013A

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

Related Commands:

Section 2.10 $AA4VV

2.10 $AA4VV

Description:

Specifies the channel(s) which low pass filter should be enabled.

Syntax:

$AA4VV[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
4	Command to enable/disable the low pass filter
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the low pass filter is enable for the channel and 0 means that the low pass filter is disabled for the channel.

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $0143A Response: !01

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

Related Commands:

Section 2.9 $AA4

2.11 $AA5

Description:

Reads the reset status of a module.

Syntax:

$AA5[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
5	Command to read the module reset status

Response:

Valid Command: **!AAS[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
S	Reset status of the module 1: This is the first time the command has been sent since the module was powered on. 0: This is not the first time the command has been sent since the module was powered on, which denotes that there has been no module reset since the last $AA5 command was sent.

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

Examples:

Command: $015 Response: !011

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

Command: $015 Response: !010

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

2.12 $AA5VV

Description:

Specifies the channel(s) to count.

Syntax:

$AA5VV(VV)[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
5	Command to start counting
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the channel is counting and 0 means that the channel is not counting.

Response:

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

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

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

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

Examples:

Command: $0153A Response: !01

Sets channels 1, 3, 4, and 5 to count and all other channels not counting for module 01. The module returns a valid response.

Command: $016 Response: !013A

Reads the counting status of module 01 and returns a response of 3A, meaning that channels 1, 3, 4, and 5 are counting and all other channels are not counting.

Related Commands:

Section 2.13 $AA6

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.13 $AA6

Description:

Reads the start/stop counting status of each channel.

Syntax:

$AA6[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
6	Command to read the channel counting status

Response:

Valid Response: **!AAVV(VV)[CHKSUM](CR)**

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the channel is counting and 0 means that the channel is not counting.

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

Examples:

Command: $0153A Response: !01

Sets channels 1, 3, 4, and 5 to count and all other channels not counting for module 01. The module returns a valid response.

Command: $016 Response: !013A

Reads the counting status of module 01 and returns a response of 3A, meaning that channels 1, 3, 4, and 5 are counting and all other channels are not counting.

Related Commands:

Section 2.12 $AA5VV

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.14 $AA6N

Description:

Sets the counter value of a channel to preset value.

Syntax:

$AA6N[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
6	Command to clear the counter value
N	Specifies the channel to be cleared (0 ~ 7)

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $0161 Response: !01

Clears the counter value of channel 1 of module 01 and the module returns a valid response.

Related Commands:

Section 2.15 $AA7

Notes:

1. The overflow status is cleared by the command, too.
2. The command is not available to the channels that are set to type code 51, frequency measurement.

2.15 $AA7

Description:

Reads the counter overflow status of each channel.

Syntax:

$AA7[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
7	Command to read the counter overflow status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. For type code 50, when the bit is 1 it means that the counter of the channel is overflowed and 0 means that the counter of the channel is not overflowed. For type code 54 ~ 56, the overflow status is represented by two bits.

00: no over/underflow
01: overflow
10: underflow
11: not applicable

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

Examples:

Command: $017 Response: !013A

Reads the counter overflow status for module 01 and returns a response of 3A, meaning that counters of channels 1, 3, 4, and 5 are overflowed.

Related Commands:

Section 2.16 $AA7VV

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.16 $AA7VV

Description:

Specifies the channels which counter overflow status are to be cleared.

Syntax:

$AA7VV[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
7	Command to clear the counter overflow status
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the counter overflow status of the channel should be cleared.

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $0173A Response: !01

Clears the counter overflow status of channels 1, 3, 4, and 5 for module 01 and returns a valid response.

Related Commands:

Section 2.15 $AA7

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.17 $AA7CNRVV

Description:

Sets the type code of a channel.

Syntax:

$AA7CNRVV[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be set (00 to FF)
7	Command to set the channel range code
CN	N specifies the input channel to be set (0-7)
RVV	VV represents the type code of the channel to be set. Refer to the Analog Input Type Setting table in Section 1.1 for details.

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response or invalid type code

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

Examples:

Command: $017C0R50 Response: !01

Sets the type code for channel 0 of module 01 to be 50 (up counting) and the module returns a valid response.

Command: $037C1R30 Response: ?03

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

Related Commands:

Section 2.18 $AA8CN

Related Topics:

Section 1.1 Configuration Tables

Note:

If the type code of a channel is set to one of 54 ~ 56, then the adjacent channel in the same pair will be set to the same type code.

2.18 $AA8CN

Description:

Reads the type code information of a channel.

Syntax:

$AA8CN[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
8	Command to read the type code of a channel
CN	N specifies which channel to access for the type code information (N=0-7).

Response:

Valid Response: !AACNRVV[CHKSUM](CR)

Invalid Response: ?AA[CHKSUM](CR)

!	Delimiter character for a valid response
?	Delimiter character for an invalid response or invalid channel
AA	Address of the responding module (00 to FF)
CN	N specifies which input channel to access to retrieve the type code information.
RVV	VV represents the type code of the specified input channel. Refer to the Analog Input Type Setting table in Section 1.1 for details.

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

Examples:

Command: $018C0 Response: !01C0R50

Reads the channel 0 input range of module 01 and returns 50 (up counting).

Related Commands:

Section 2.17 $AA7CNRVV

Related Topics:

Section 1.1 Configuration Tables

2.19 $AAF

Description:

Reads the firmware version of a module.

Syntax:

$AAF[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
F	Command to read the firmware version

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Data)	Firmware version string of the module

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

Examples:

Command: $01F Response: !01A2.0

Reads the firmware version of module 01, and shows that it is version A2.0.

Command: $02F Response: !02B1.1

Reads the firmware version of module 02, and shows that it is version B1.1.

2.20 $AAI

Description:
Reads the INIT status.

Syntax:
$AAI[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
I	Command to read the INIT status

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
S	INIT status: 0= the INIT pin is shorted to GND. 1= the INIT pin is not shorted to GND. 2= Update CJC temperature once only.

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

Examples:

Command: $01I Response: !010

Reads the INIT status of module 01 and the module responds with 0, meaning that the INIT pin is shorted to GND.

2.21 $AAM

Description:
Reads the name of a module.

Syntax:
$AAM[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
M	Command to read the module name

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Name)	Name string of the module

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

Examples:
Command: $01M Response: !0187084

Reads the module name of module 01 and returns the name “87084”.

Related Commands:
Section 2.23 ~AAO(Name)

2.22 $AAP

Description:

Reads the communication protocol information.

Syntax:

$AAP[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
P	Command to read the communication protocol

Response:

Valid Response: !AASC[CHKSUM](CR)

Invalid Response: ?AA[CHKSUM](CR)

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)
S	The protocols supported by the module 0: only DCON protocol is supported 1: both the DCON and Modbus RTU protocols are supported
C	The current protocol that is saved in the EEPROM that will be used at the next power-on reset 0: the protocol set in the EEPROM is DCON 1: the protocol set in the EEPROM is Modbus RTU

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

Examples:

Command: $01P Response: !0110

Reads the communication protocol of module 01 and returns a response of 10 meaning that it supports both the DCON and Modbus RTU protocols and the protocol that will be used at the next power-on reset is DCON.

Related Commands:

Section 2.23 $AAPN

Note:

The command is not supported by the I-87084W.

2.23 $AAPN

Description:

Sets the communication protocol.

Syntax:

$AAPN[CHKSUM](CR)

$	Delimiter character
AA	Address of the module to be read (00 to FF)
P	Command to set the communication protocol
N	0: DCON protocol 1: Modbus RTU protocols Before using this command, the rear slide switch must be in the INIT position. The new protocol is saved in the EEPROM and will be effective after the next power-on reset.

Response:

Valid Response: !AASC[CHKSUM](CR)

Invalid Response: ?AA[CHKSUM](CR)

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)

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

Examples:

Command: $01P1 Response: ?01

Sets the communication protocol of module 01 to Modbus RTU and returns an invalid response because the module is not in INIT mode.

Command: $01P1 Response: !01

Sets the communication protocol of module 01 to Modbus RTU and returns a valid response.

Related Commands:

Section 2.22 $AAP

Note:

The command is not supported by the I-87084W.

2.24 @AABB

Description:

Reads the battery backup counter status of each channel.

Syntax:

@AABB[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
BB	Command to read the battery backup counter status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

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

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

Examples:

Command: @01BB Response: !013A

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

Related Commands:

Section 2.25 @AABBVV

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.25 @AABBVV

Description:

Specifies the channels which battery backup counter function should be enabled.

Syntax:

@AABBVV[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
BB	Command to enable the battery backup counter function
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the battery backup counter is enabled for the channel and 0 means that the battery backup counter is disabled for the channel.

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01BB3A Response: !01

Sets the battery backup counter function of channels 1, 3, 4, and 5 for module 01 to be enabled and returns a valid response.

Related Commands:

Section 2.24 @AABB

Note:

The command is not available to the channels that are set to type code 51, frequency measurement.

2.26 @AAFA

Description:

Reads the automatic frequency mode status of each channel.

Syntax:

@AAFA[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FA	Command to read the automatic frequency mode status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

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

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

Examples:

Command: @01FA Response: !013A

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

Related Commands:

Section 2.27 @AAFAVV

Related Topics:

Section 1.2 Frequency Measurement

Note:

The command is only available to the channels that are set to type code 51, frequency measurement.

2.27 @AAFAVV

Description:

Specifies the channels which automatic frequency mode is to be enabled.

Syntax:

@AAFAVV[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FA	Command to enable the automatic frequency mode
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the automatic frequency mode is enabled for the channel and 0 means that the automatic frequency mode is disabled for the channel.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01FA3A Response: !01

Enables the automatic frequency mode of channels 1, 3, 4, and 5 for module 01 and returns a valid response.

Related Commands:

Section 2.26 @AAFA

Related Topics:

Section 1.2 Frequency Measurement

Note:

The command is only available to the channels that are set to type code 51, frequency measurement.

2.28 @AAFH

Description:

Reads the high frequency mode status of each channel.

Syntax:

@AAFH[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FH	Command to read the high frequency mode status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

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

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

Examples:

Command: @01FH Response: !013A

Reads the high frequency mode status for module 01 and returns a response of 3A, meaning that battery backup counter for channels 1, 3, 4, and 5 are enabled and disabled for all other channels.

Related Commands:

Section 2.29 @AAFHVV

Related Topics:

Section 1.2 Frequency Measurement

Note:

The command is only available to the channels that are set to type code 51, frequency measurement.

2.29 @AAFHVV

Description:

Specifies the channels which high frequency mode is enabled.

Syntax:

@AAFH[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FH	Command to enable the high frequency mode
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the high frequency mode is enabled for the channel and 0 means that the high frequency mode is disabled for the channel.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01FH3A Response: !01

Enables the high frequency mode of channels 1, 3, 4, and 5 for module 01 and returns a valid response.

Related Commands:

Section 2.28 @AAFH

Related Topics:

Section 1.2 Frequency Measurement

Note:

The command is only available to the channels that are set to type code 51, frequency measurement.

2.30 @AAFT

Description:

Reads the frequency measurement timeout setting.

Syntax:

@AAFT[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FT	Command to read the frequency measurement timeout setting

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	Two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

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

Examples:

Command: @01FT Response: !010A

Reads the frequency measurement timeout for module 01 and returns a response of 0A, meaning that frequency measurement timeout value is 1 second.

Related Commands:

Section 2.31 @AAFTVV

Related Topics:

Section 1.2 Frequency Measurement

Note:

1. The command is only available to the channels that are set to type code 51, frequency measurement.
2. If there is no input pulse detected during the frequency measurement time out, then the frequency of the channel is set to 0 Hz.

2.31 @AAFTVV

Description:

Sets the frequency measurement time out value of a module.

Syntax:

@AAFTVV[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
FT	Command to set the frequency measurement time out
VV	Two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01FT0A Response: !01

Sets the frequency measurement timeout for module 01 to 1 second and returns a valid response.

Related Commands:

Section 2.30 @AAFT

Related Topics:

Section 1.2 Frequency Measurement

Note:

1. The command is only available to the channels that are set to type code 51, frequency measurement.
2. If there is no input pulse detected during the frequency measurement time out, then the frequency of the channel is set to 0 Hz.

2.32 @AAGN

Description:

Reads the preset counter value of channel N.

Syntax:

@AAGN[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be set (00 to FF)
G	Command to read the preset counter value of a channel
N	Specifies the channel to be read (0 ~ 7)

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
(Data)	The preset counter value in eight hexadecimal digits

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

Examples:

Command: @01G2 Response: !0100000000

Reads the channel 2 maximum counter value of module 01 and the module responds with 00000000.

Related Commands:

Section 2.33 @AAGN(Data)

Note:

The preset counter value is only available to the channels that are set to up counter type (type code 50).

2.33 @AAGN(Data)

Description:

Sets the preset counter value of a channel.

Syntax:

@AAGN(Data)[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be set (00 to FF)
G	Command to set the preset counter value of a channel
N	Specifies the channel to be read (0 ~ 7)
(Data)	The preset counter value in eight hexadecimal digits

Response:

Valid command: **!AA(Data)[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01G2F0000000 Response: !01

Sets the channel 2 preset counter value of module 01 to 0xF0000000 and returns a valid response.

Related Commands:

Section 2.32 @AAGN

Note:

The preset counter value is only available to the channels that are set to up counter type (type code 50).

2.34 @AASC

Description:

Reads the stop counting on overflow status of each channel.

Syntax:

@AASC[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
SC	Command to read the stop counting on overflow status

Response:

Valid Command: **!AAVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the stop counting on overflow is enabled for the channel and 0 means that the stop counting on overflow is disabled for the channel.

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

Examples:

Command: @01SC Response: !013A

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

Related Commands:

Section 2.35 @AASCVV

Note:

The command is only available to the channels that are set to type code 50, up counter measurement.

2.35 @AASCVV

Description:

Specifies the channels which stop counting on overflow are enabled.

Syntax:

@AASCVV[CHKSUM](CR)

@	Delimiter character
AA	Address of the module to be read (00 to FF)
SC	Command to enable the stop counting on overflow
VV	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it means that the stop counting on overflow is enabled for the channel and 0 means that the stop counting on overflow is disabled for the channel.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: @01SC3A Response: !01

Enables the stop counting on overflow of channels 1, 3, 4, and 5 for module 01 and returns a valid response.

Related Commands:

Section 2.34 @AASC

Note:

The command is only available to the channels that are set to type code 50, up counter measurement.

2.36 ~**

Description:
Informs all modules that the host is OK.

Syntax:
~[CHKSUM](CR)**

~ Delimiter character

** Host OK command

Response:
No response.

Examples:
Command: ~** No response

Sends a “Host OK” command to all modules.

Related Commands:
Section 2.37 ~AA0, Section 2.38 ~AA1, Section 2.39 ~AA2, Section 2.40 ~AA3EVV

2.37 ~AA0

Description:

Reads the host watchdog status of a module.

Syntax:

~AA0[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
0	Command to read the module status

Response:

Valid command: **!AASS[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
SS	Two hexadecimal digits that represent the host watchdog status, where: Bit 7: 0 indicates that the host watchdog is disabled, and 1 indicates that the host watchdog is enabled, Bit 2: 1 indicates that a host watchdog timeout has occurred, and 0 indicates that no host watchdog timeout has occurred. The host watchdog status is stored in EEPROM and can only be reset by using the ~AA1 command.

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

Examples:

Command: ~010 Response: !0100

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

Command: ~020 Response: !0204

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

Related Commands:

Section 2.36 ~**, Section 2.38 ~AA1, Section 2.39 ~AA2, Section 2.40 ~AA3EVV

2.38 ~AA1

Description:

Resets the host watchdog timeout status of a module.

Syntax:

~AA1[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be reset (00 to FF)
1	Command to reset the host watchdog timeout status

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~010 Response: !0104

Reads the host watchdog status of module 01 and shows that a host watchdog timeout has occurred.

Command: ~011 Response: !01

Resets the host watchdog timeout status of module 01 and returns a valid response.

Command: ~010 Response: !0100

Reads the host watchdog status of module 01 and shows that no host watchdog timeout has occurred.

Related Commands:

Section 2.36 ~**, Section 2.37 ~AA0, Section 2.39 ~AA2, Section 2.40 ~AA3EVV

2.39 ~AA2

Description:

Reads the host watchdog timeout value of a module.

Syntax:

~AA2[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
2	Command to read the host watchdog timeout value

Response:

Valid command : **!AAEVV[CHKSUM](CR)**

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

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

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

Examples:

Command: ~012 Response: !011FF

Reads the host watchdog timeout value of module 01 and returns FF, which denotes that the host watchdog is enabled and the host watchdog timeout value is 25.5 seconds.

Related Commands:

Section 2.36 ~**, Section 2.37 ~AA0, Section 2.38 ~AA1, Section 2.40 ~AA3EVV

2.40 ~AA3EVV

Description:

Enables/disables the host watchdog and sets the host watchdog timeout value of a module.

Syntax:

~AA3EVV[CHKSUM](CR)

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

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~013164 Response: !01

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

Command: ~012 Response: !01164

Reads the host watchdog timeout value of module 01. The module returns 164, which denotes that the host watchdog is enabled and the host watchdog timeout value is 10.0 seconds.

Related Commands:

Section 2.36 ~**, Section 2.37 ~AA0, Section 2.38 ~AA1, Section 2.39 ~AA2

Notes:

When a host watchdog timeout occurs, the host watchdog is disabled. The ~AA3EVV command should be sent again to reenable the host watchdog.

2.41 ~AAI

Description:

The Soft INIT command is used to enable modification of the Baud Rate and checksum settings using software only.

Syntax:

~AAI[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
I	Command to set the Soft INIT

Response:

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

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~01I Response: !01

Sets the soft INIT of module 01 and returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.45 ~AATnn

Note:
The ~AATnn command should be sent prior to sending this command, see Section 2.45 for details.

2.42 ~AAO(Name)

Description:

Sets the name of a module.

Syntax:

~AAO(Name)[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
O	Command to set the module name
(Name)	New name of the module (max. 6 characters).

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~01O87084N Response: !01

Sets the name of module 01 to be “87084N” and returns a valid response.

Command: $01M Response: !0187084N

Reads the name of module 01 and returns the name “87084N”.

Related Commands:

Section 2.21 $AAM

2.43 ~AARD

Description:

Reads the response delay time value of a module.

Syntax:

~AARD[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be read (00 to FF)
RD	Command to read the response delay time value

Response:

Valid command : **!AAEVV[CHKSUM](CR)**

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)
VV	Two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30ms. The max allowable value is 30 (1Eh).

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

Examples:

Command: ~01RD Response: !0102

Reads the response delay time value of module 01 and returns 02, which denotes that the response delay time value is 2ms.

Related Commands:

Section 2.44 ~AARDVV

2.44 ~AARDVV

Description:

Sets the response delay time value of a module.

Syntax:

~AARDVV[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
RD	Command to set the response delay time
VV	Two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30s. The max allowable value is 30 (1Eh).

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~01RD06 Response: !01

Sets the response delay time value to 6ms. The module returns a valid response.

Command: ~01RD Response: !0106

Reads the response delay time value of module 01. The module returns 06, which denotes that the rsponse delay time value is 6ms.

Related Commands:

Section 2.43 ~AARD

2.45 ~AATnn

Description:

Sets the soft INIT time out value.

Syntax:

~AATnn[CHKSUM](CR)

~	Delimiter character
AA	Address of the module to be set (00 to FF)
T	Command to set the soft INIT time out value
nn	Two hexadecimal digits representing the time out value in seconds. The max time out value is 60 seconds. When changing the Baud Rate or checksum settings without altering the INIT* pin, the ~AAI and %AANNTTCCFF commands should be sent consecutively and the time interval between the two commands should be less than the soft INIT time out. If the soft INIT time out is 0, then the Baud Rate and checksum settings cannot be changed using software only. The power on reset value of the soft INIT time out is 0.

Response:

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

!	Delimiter character for a valid response
?	Delimiter character for an invalid response
AA	Address of the responding module (00 to FF)

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

Examples:

Command: ~01I Response: !01

Sets the soft INIT of module 01 and returns a valid response.

Command: %0101000700 Response: ?01

Attempts to change the Baud Rate of module 01 to 19200 without first altering the INIT * pin. The module returns an invalid response because the soft INIT time out value is 0.

Command: ~01T10 Response: !01

Sets the soft INIT time out value of module 01 to 16 seconds and returns a valid response.

Command: ~01I Response: !01

Sets the soft INIT of module 01 and returns a valid response.

Command: %0101000700 Response: !01

Changes the Baud Rate of module 01 to 19200 without first altering INIT * pin. The module returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.41 ~AAI

Note:

It is recommended that the soft INIT time out value is reset to 0 once any changes to Baud Rate and checksum settings have been completed.