I-7088, I-7088D, M-7088 and M-7088D User Manual

I-7000 New Features

1. Internal Self Tuner
2. Multiple Baud Rates
3. Multiple Data Formats
4. Internal Dual WatchDog
5. True Distributed Control
6. High Speed & High Density I/O

Your Powerful Tools → **Create New Ideas** → **Create New Applications**

Warranty

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

Warning

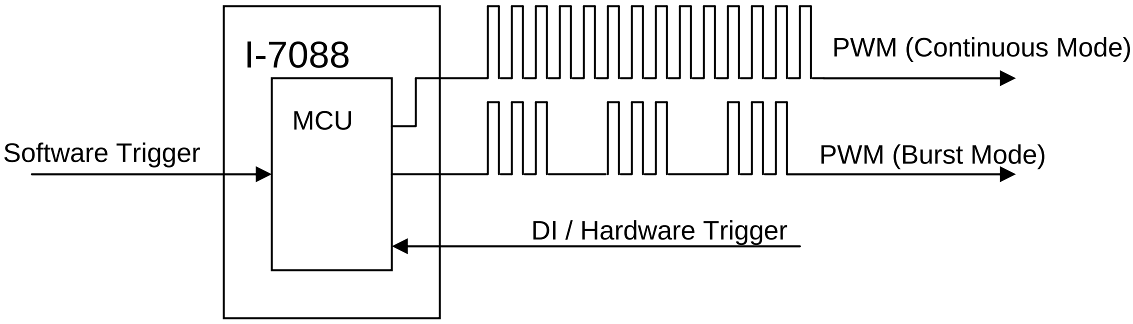
ICP DAS assume no liability for any damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademarks

Table of Contents

1. Introduction



The I-7088 has 8 PWM output channels and 8 counter inputs and can be used to develop powerful and cost effective analog control systems. PWM (Pulse Width Modulation) is a powerful technique for controlling analog circuits that uses digital outputs to generate a waveform with a variable Duty Cycle (the fraction of time that a system is in an "active" state) and frequency to control analog circuits, and can be used to control the position/speed of motors, control the brightness of lamps, or control the speed of fans, etc.

The I-7088 will also automatically save the counter value to EEPROM if the power supply is interrupted or lost. Refer to Section 1.7 for details.

Features

- Automatic hardware generation of PWM outputs without the need for software intervention.
- 1Hz ~ 500KHz PWM output frequency with 0.1%~99.9% duty cycle (Refer to Section 4.2).
- Software and hardware trigger mode for PWM output.
- Individual and synchronous PWM output. By using software trigger mode, you can set the configuration for all PWM channels then trigger them either individually or all at the same time.
- Burst mode PWM operation for standby.
- DI channel can be configured as either a simple digital input channel or a hardware trigger source for the PWM output.

Applications

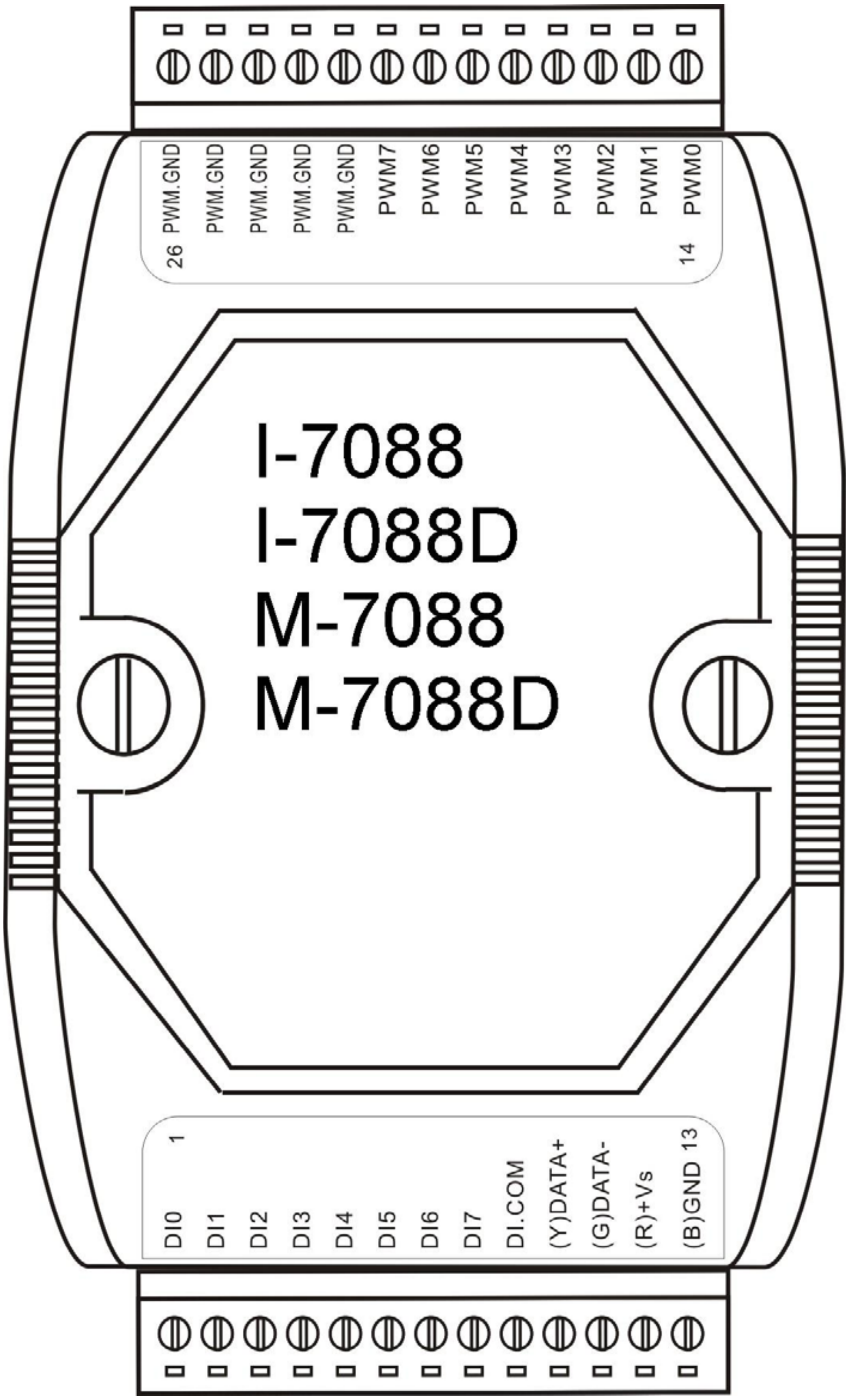
- Controlling the position/speed of motors
- Controlling the brightness of lamps
- Controlling the speed of fans

More Information

Refer to Chapter 1 of the “I-7000 Bus Converter User Manual” for more information regarding the following:

1.1. I-7000 Overview
1.2. I-7000 Related Documentation
1.3. I-7000 Common Features
1.4. I-7000 System Network Configuration
1.5. I-7000 Dimensions

1.1. Pin Assignments



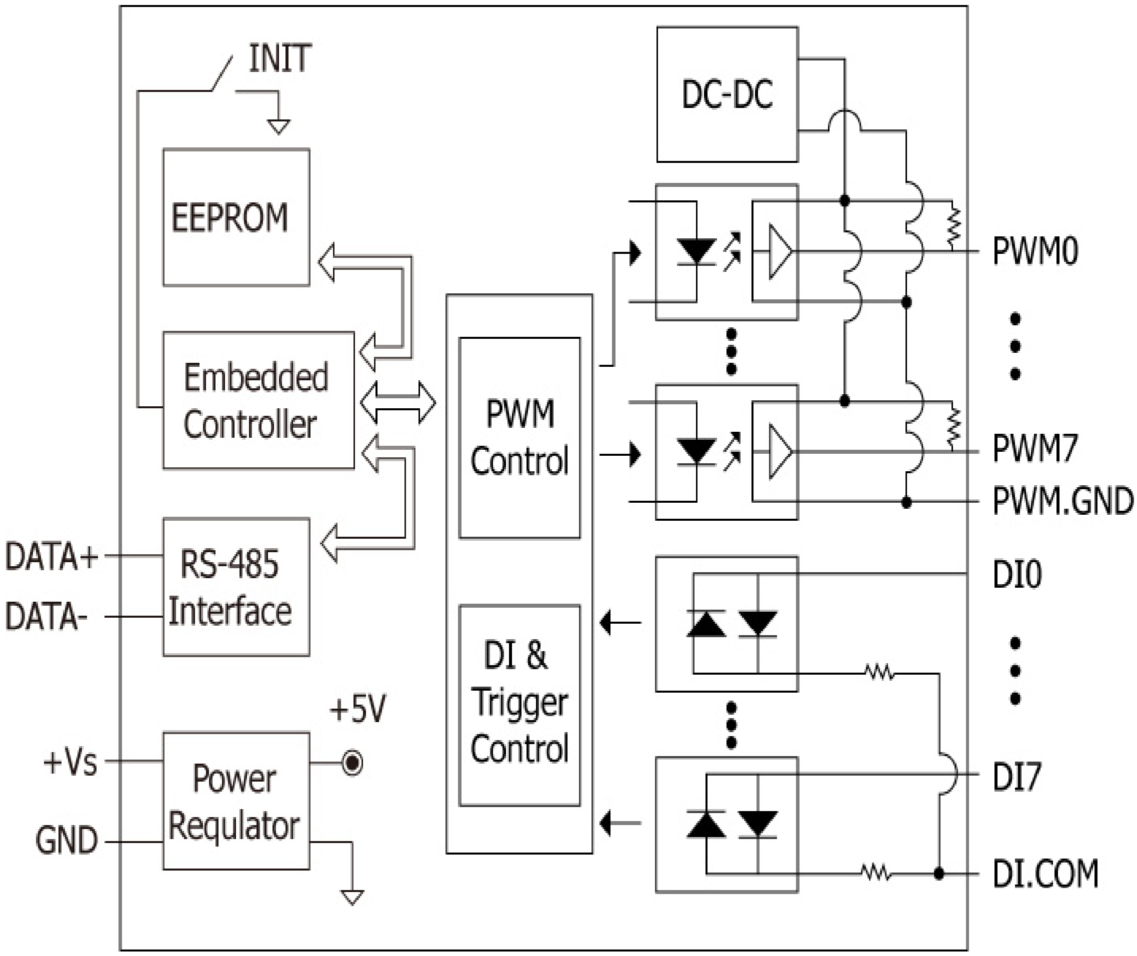
1.2. Specifications

PWM Output	
Channels	8
Type	TTL, Isolated
Max. Frequency	500 KHz
Duty Cycle	0.1%~99.9%
PWM Mode	Burst mode, Continuous mode
Burst Mode Counter	1~65535 counts
Trigger Start	Hardware or Software
ESD Protection	4 kV Contact for each terminal and 8 kV Air for random points
Isolation	2500 V_{DC}
Digital Input	
Channels	8
Type	Sink, Isolated
ON Voltage Level	+2.4 V~+5 V
OFF Voltage Level	+1 V Max.
Max. Frequency	1 MHz
Max. Counts	32bits (4,294,967,295)
Built-in Virtual Battery Backup for Counter Value	Yes
ESD Protection	4 kV Contact for each terminal and 8 kV Air for random point
Isolation	2500 V_{DC}
Interface	
Interface	RS-485
Format	N, 8, 1
Baud Rate	1200 ~ 115200bps
LED Display	

1 LED as Power/Communication Indicator	
Dimensions	
72mm x 122mm x 35mm (W x L x H)	
Power	
Input Voltage Range	10 ~ 30 V_{DC}
Power Consumption	2.4 W (max.)
Power Reverse Polarity Protection	Yes
+/- 4 kV ESD , +/- 4 kV EFT and +/- 3 kV Surge Protection	Yes

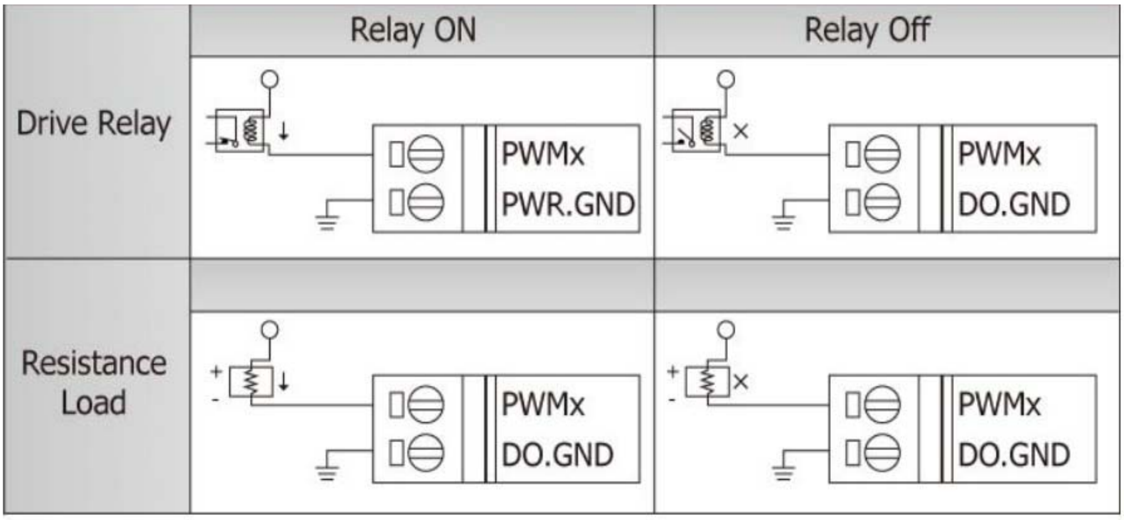
Environment	
Operating Temperature	-25 ~ 75°C
Storage Temperature	-40 ~ 85°C
Humidity	5 ~ 95%, non-condensing

1.3. Block Diagram

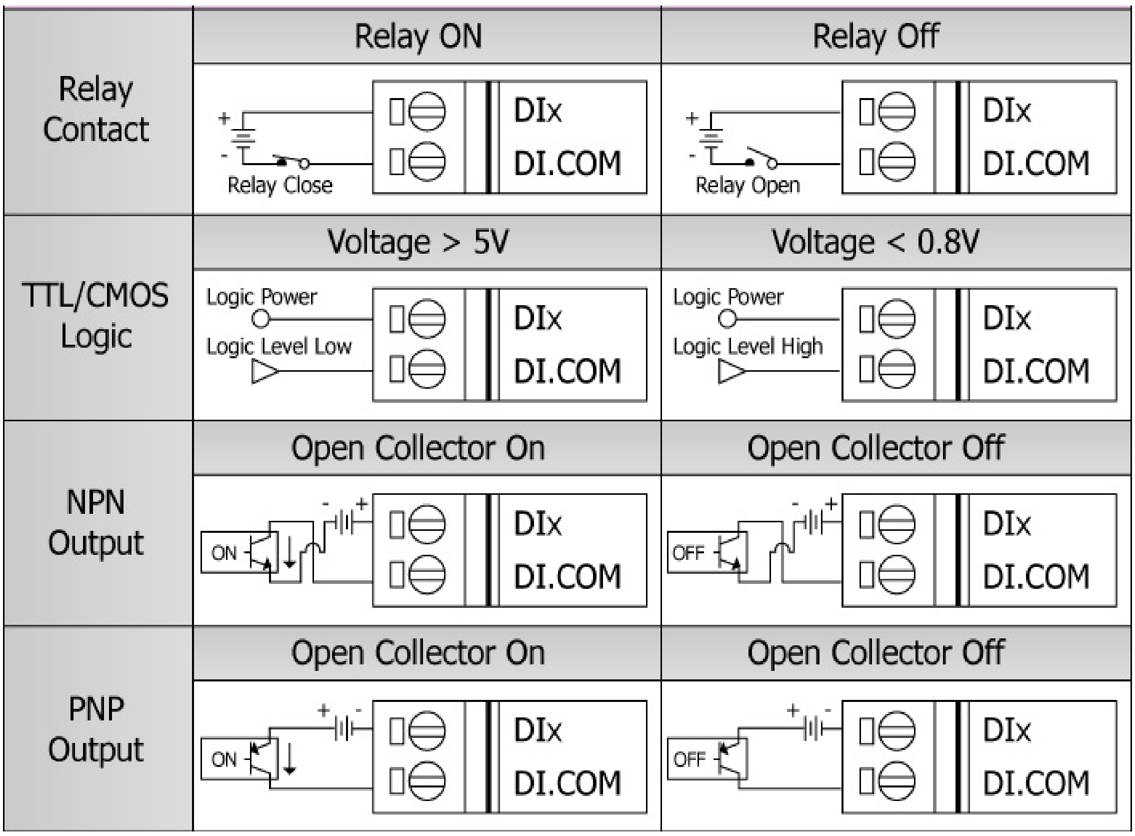


1.4. Application Wiring

1.4.1. PWM Wiring Connections



1.4.2. DI/Counter Wiring Connections



1.5. Quick Start

Refer to http://www.icpdas.com/download/7000/manual.htm and use the "DCON Utility" to control the module. Otherwise, use "DCON Utility -> Terminal -> Command Line" and follow the commands shown below.

DI Status and Counter

1. Type @01DI[Enter]	→ Receive => !01xx01
2. Type $01500[Enter]	→ Receive => !01
3. Type @01P200000000[Enter]	→ Receive => !01
4. Type $0132FFFFFFFF[Enter]	→ Receive => !01
5. Type $0162[Enter]	→ Receive => !01
6. Type #012[Enter]	→ Receive => >00000000
7. Type $01504[Enter]	→ Receive => !01
8. Type #012[Enter]	→ Receive => >xxxxxxxx

- Step 1: Read the DI status channel 0 = 1, channel 1 = 0, etc.
- Step 2: Disable the DI counter of channel 2
- Step 3: Set the DI preset counter value (00000000) of channel 2
- Step 4: Set the DI max. counter value (FFFFFFFF) of channel 2
- Step 5: Reset the DI counter of channel 2
- Step 6: Read the DI counter value (00000000) of channel 2
- Step 7: Enable the DI counter of channel 2
- Step 8: Read the DI counter value (xxxxxxxx) of channel 2

PWM Output

1. Type $01C0F100000[Enter] → Receive => !01100000
2. Type $01C0D50.0[Enter] → Receive => !0150.0
3. Type $01C0M1[Enter] → Receive => !01
4. Type @01DO01[Enter] → Receive => !01

- Step 1: Set the frequency of PWM channel 0 to 100 KHz
- Step 2: Set the duty cycle of PWM channel 0 to 50.0%
- Step 3: Set PWM channel 0 to continuous mode
- Step 4: Start the output of PWM channel 0

1.6. Default Settings

The default settings are as follows:

- Address = 01
- Baud Rate = 9600
- Checksum disabled
- Data = 1 Start + 8 Data + 1 Stop (no parity)
- PWM Frequency = 10 KHz
- PWM Duty Cycle = 50%
- PWM Steps = 1 (Continuous Type)

1.7. Configuration Tables

Baud Rate Setting (CC)

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

Bits 7:6	Description
00	No parity and one stop bit
01	No parity and two stop bits
10	Even parity and one stop bit
11	Odd parity and one stop bit

Configuration Code Table (TT)

TT	Input Range
50	Counter
52	Virtual Battery Backup

Note: For type 52, the count value will continue from the last power-off value.

Data Format Settings (FF)

7	6	5	4	3	2	1	0
0	CS	Reserved					

Key	Description
CS	Checksum setting 0: Disabled 1: Enabled

Note: Reserved bits should be zero.

2. DCON Protocol

All communication with I-7000 modules consists of commands generated by the host and responses transmitted by the I-7000 module. 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 sent to a module contain the ID address, meaning that only the addressed module will respond. The only exception to this is command ~** (Section 2.49), which is sent to all modules, but, in this case, the modules do not reply to the command.

Command Format:

Leading Character	Module Address	Command	[CHKSUM]	CR

Response Format:

Leading Character	Module Address	Data	[CHKSUM]	CR

CHKSUM A 2-character checksum that is present when the checksum setting is enabled. See Sections 1.7 and 2.1 for details.

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

Checksum Calculation:

Calculate the ASCII code sum of all the characters in the command/response string, except for the carriage return character (CR).

The checksum is equal to the sum masked by 0ffh.

Example:

Command string: $012(CR)

The sum of the string = “$”+”0”+”1”+”2” = 24h+30h+31h+32h = B7h

Therefore the checksum is B7h, and so CHKSUM = “B7”

The command string with the checksum = $012B7(CR)

Response string: !01200600(CR)

The sum of the string = “!”+”0”+”1”+”2”+”0”+”0”+”6”+”0”+”0” = 21h+30h+31h+32h+30h+30h+36h+30h+30h = 1AAh

Therefore the checksum is AAh, and so CHKSUM = “AA”

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 configuration of the module	2.1
$AA2	!AANNTTCCFF	Reads the configuration of the module	2.6
$AA5	!AAS	Reads reset status of the module	2.9
$AAF	!AA(Data)	Reads the firmware version	2.32
$AAI	!AAS	Reads the status of the INIT switch	2.33

$AAM	!AA(Data)	Reads the module name	2.34
$AAP	!AASC	Reads the communication protocol	2.35
$AAPN	!AA	Sets the communication protocol	2.36
~AAO(Name)	!AA	Sets the module name	2.46
~AARD	!AATT	Reads the response delay time	2.47
~AARDTT	!AA	Sets the response delay time	2.48

PWM Command Sets			
Command	Response	Description	Section
#AA	>(Data)	Reads the count	2.2
#AAN	>(Data)	Reads the count of a specific channel	2.3
#AA1cDD	>	Sets the output for a specific PWM channel	2.4
#AAAcDD	>	Sets the output for a specific PWM channel	2.5
$AA3N	!AA	Reads the max. counter value of a specific channel	2.7
$AA3N(data)	!AA	Sets the max. counter value for a specific channel	2.8
$AA5VV	!AA	Sets the counter status for a specific channel	2.10
$AA6	!AASS	Reads the counter status	2.11
$AA6N	!AA	Resets the counter of a specific channel	2.12
$AA6NN	!AA	Resets the counter of a specific channel	2.13
$AA7N	!AAS	Reads the status of the overflow for a specific channel	2.14
$AAB	!AAS	Reads the power-down count	2.18

$AABR	!AA	Clears the power-down count	2.19
$AACnD	!AA(data)	Reads the duty cycle value of a specific channel	2.20
$AACnD(data)	!AA	Sets the duty cycle value for a specific channel	2.21
$AACnF	!AA(data)	Reads the frequency value of a specific channel	2.22
$AACnF(data)	!AA	Sets the frequency value for a specific channel	2.23
$AACnM	!AAS	Reads the continuous mode status of a specific channel	2.24
$AACnMS	!AA	Sets the continuous mode for a specific channel	2.25
$AACnP	!AA(data)	Reads the PWM step value of a specific channel	2.26
$AACnP(data)	!AA	Sets the PWM step value for a specific channel	2.27
$AACnT	!AAS	Reads the hardware trigger configuration of a specific channel	2.28
$AACnTS	!AA	Sets the hardware trigger configuration for a specific channel	2.29
$AACnN	!AAS	Reads the status of the PWM synchronization of a specific channel	2.30
$AACnNS	!AA	Sets the PWM synchronization for a specific channel	2.31
$AAR	!AA	Resets the PWM	2.37
$AAW	!AA	Saves the PWM configuration	2.38
$AAYS	!AA	Starts the PWM synchronization	2.39
@AADODD	!AA	Sets the status of the PWM output port	2.40

@AADI	!AAOO	Reads the status of the PWM output port and the DI	2.41
@AAGn	!AA(data)	Reads the preset count value of a specific channel	2.42
@AAPN(data)	!AA	Sets the preset count value for a specific channel	2.43
~AAD	!AASS	Reads the miscellaneous settings	2.44
~AADVV	!AA	Sets the miscellaneous settings	2.45

Host Watchdog Command Sets			
Command	Response	Description	Section
~**	No Response	The Host is OK	2.49
~AA0	!AASS	Reads the status of the Host Watchdog	2.50
~AA1	!AA	Resets the status of the Host Watchdog	2.51
~AA2	!AAETT	Reads the Host Watchdog timeout settings	2.52
~AA3ETT	!AA	Sets the Host Watchdog timeout settings	2.53
~AAI	!AA	Sets the Software INIT	2.54
~AATnn	!AA	Sets the Software INIT timeout value	2.55

LED Command Sets			
Command	Response	Description	Section
$AA8	!AAS	Reads the LED configuration	2.15
$AA8V	!!AA	Sets the LED configuration	2.16
$AA9(data)	!!AA	Sends the data to the LED	2.17

2.1. %AANNTTCCFF

Description:

This command is used to set the configuration of a module.

Syntax:

%AANNTTCCFF[CHKSUM](CR)

%	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
NN	The new address of the module in hexadecimal format (00 to FF)
TT	The new Type Code, see Section 1.7 for details
CC	The new Baud Rate code, see Section 1.7 for details. For the I-7088, the rear slide switch must be moved to the INIT position in order to change the Baud Rate settings. See Section A.1 for details.
FF	The command used to set the checksum, and the input range settings (Section 1.7). For the I-7088, the rear slide switch must be moved to the INIT position in order to change the checksum setting. See Section A.1 for details.

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 switching the rear slide switch to the INIT position, the module will return an invalid command.)
AA	The 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:

%0102500600

Response:

!02

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

Command:

%0202520600

Response:

!02

Sets the type of module 02 to 52 (Virtual Battery Backup) and the module returns a valid response.

Command:

%0202520A00

Response:

?02

Changes the Baud Rate of module 02 to 115200bps and the module returns an invalid response because it is not in INIT mode.

Command:

%0202520A00

Response:

!01

Changes the Baud Rate of module 02 to 115200bps and the module is in INIT mode. The module returns a valid response.

Related Commands:

Section 2.6 $AA2, Section 2.54 ~AAI, Section 2.55 ~AATnn

Related Topics

Section 1.7 Configuration Tables, Section A.1 INIT Pin Operation

Notes:

1. Changes to the address, Type Code 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-7088, changing the Baud Rate and checksum settings can only be achieved using software and are performed by using the following commands:
   I. Send an ~AATnn command. See Section 2.55 for details.
   II. Send an ~AAI command. See Section 2.54 for details.
   III. Send an %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:

This command is used to read the DI count.

Syntax:

#AA[CHKSUM](CR)

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

Response:

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

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

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	The count data from all DI channels
AA	The 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:

#01

Response:

\>00000008000000090000000A000000B000000C00000
0D000000E0000000F

Reads module 01 and returns the count of DI channel 0 (8), channel 1 (9), etc.

Related Commands:

Section 2.3 #AAN

2.3. #AAN

Description:

This command is used to read the count of a specific channel.

Syntax:

#AAN[CHKSUM](CR)

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

Response:

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

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

>	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
(Data)	The DI count of the specified channel
AA	The 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:

>00000008

Reads data from channel 2 of module 03 and returns a valid response.

Command:

#029

Response:

?02

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

Related Commands:

Section 2.2 #AA

2.4. #AA1cDD

Description:

This command is used to set the status of a specific PWM channel.

Syntax:

#AA1cDD[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be set (00 to FF)
	The command to set the status of the PWM
c	Specifies the channel to be set
DD	00: Sets the PWM output port to off 01: Sets the PWM output port to on

Response:

Valid Command: >[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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:

#011201

Response:

>

Sets the output of PWM channel 2 to on and returns a valid response.

Related Commands:

Section 2.5 #AAAcDD, Section 2.40 @AADODD

Note:

This command is the same as the #AAAcDD command.

2.5. #AAAcDD

Description:

This command is used to set the status of a specific PWM channel.

Syntax:

#AAAcDD[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be set (00 to FF)
A	The command to set the status of the PWM
c	Specifies the channel to be set
DD	00: Sets the PWM output port to off 01: Sets the PWM output port to on

Response:

Valid Command: **>[CHKSUM](CR)**

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

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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:

#01A201

Response:

>

Sets the output of PWM channel 2 to on and returns a valid response.

Related Commands:

Section 2.4 #AA1cDD, Section 2.40 @AADODD

Note:

This command is the same as the #AA1cDD command.

2.6. $AA2

Description:

This command is used to read the configuration of a module.

Syntax:

$AA2[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
2	The 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	The address of the responding module (00 to FF**)**
TT	The Type Code of the module, see Section 1.7 for details
CC	The Baud Rate code of the module, see Section 1.7 for details
FF	The checksum settings and the input range settings of the module, see Section 1.7 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:

!01500600

Reads the configuration of module 01 and returns a valid response.

Command:

$022

Response:

!02520600

Reads the configuration of module 02 and returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.7 Configuration Tables

2.7. $AA3N

Description:

This command is used to read the maximum counter value for a specific channel.

Syntax:

$AA3N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
3	The command to read the maximum counter value
N	The channel to be read, zero based

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
(Data)	8 hexadecimal digits (00000001 to FFFFFFFF) representing the maximum counter value

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

Examples:

Command:

$0130

Response:

!01FFFFFFFF

Reads the maximum counter value of channel 0 at address 01, returns a value of 4294967295.

Related Commands:

Section 2.8 $AA3N(Data)

2.8. $AA3N(Data)

Description:

This command is used to set the maximum counter value for a specific channel.

Syntax:

$AA3N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
3	The command to set the maximum counter value
N	The channel to be set, zero based
(Data)	8 hexadecimal digits (00000001 to FFFFFFFF) representing the maximum counter value

Response:

Valid Command: !AA [CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

$030FFFFFFFF

Response:

!03

Sets the maximum counter value of counter 0 at address 03 to 4294967295, and returns a response indicating that the command was successful.

Related Commands:

Section 2.7 $AA3N

2.9. $AA5

Description:

This command is used to read the reset status of a module.

Syntax:

$AA5[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
5	The command to read the reset status of the module

Response:

Valid Command: !AAS[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

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

Examples:

Command:

$015

Response:

!011

Reads the reset status of module 01. The response 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.

2.10. $AA5VV

Description:

This command is used to specify the channel number of the DI counter to be enabled.

Syntax:

$AA5VV[CHKSUM](CR)

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

Response:

Valid Command: !AA [CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The 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

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

Related Commands:

Section 2.11 $AA6

2.11. $AA6

Description:

This command is used to read the status of the DI counter.

Syntax:

$AA6[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
6	The command to read the status of the DI counter

Response:

Valid Command: !AAVV[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

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

Examples:

Command:

$016

Response:

!013A

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

Related Commands:

Section 2.10 $AA5VV

2.12. $AA6N

Description:

This command is used to reset the counter of a specific channel.

Syntax:

$AA6N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be reset (00 to FF)
6	The command to reset the counter
N	Specifies the channel to be reset, zero based

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The 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:

$0160

Response:

!01

Resets the counter 0 of module 01 to the preset value and returns a valid response indicating that the command was successful.

Related Commands:

Section 2.42 @AAGN, Section 2.43 @AAPN(Data)

2.13. $AA6NN

Description:

This command is used to reset the DI counter of a specific channel.

Syntax:

$AA6N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be reset (00 to FF**)**
6	The command to reset the DI counter
NN	A two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds channel 1, etc. When the bit is 0, it means that the channel is inactive, and 1 means that the channel has been reset.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The 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:

$01601

Response:

!01

Resets the counter 0 of module 01 to the preset value and returns a valid response indicating that the command was successful.

Related Commands:

Section 2.42 @AAGN, Section 2.43 @AAPN(Data), Section 2.11 $AA6

2.14. $AA7N

Description:

This command is used to read the status of the overflow flag for a specific channel.

Syntax:

$AA7N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF**)**
7	The command to read the status of the overflow flag
N	Specifies the channel to be read, zero based

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The address of the responding module (00 to FF)
S	The overflow flag of channel N 0: The counter has not exceeded the maximum counter value and the overflow flag has been cleared. 1: The counter has exceeded the maximum counter value and the overflow flag has been set.

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

Examples:

Command:

$0170

Response:

!010

Reads the status of the overflow flag for counter 0 of module 01 and returns a response indicating that the counter has not been exceeded.

Related Commands:

Section 2.7 $AA3N, Section 2.8 $AA3N(Data), Section 2.12 $AA6N, Section 2.13 $AA6NN

2.15. $AA8

Description:

This command is used to read the configuration of the LED.

Syntax:

$AA8[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF**)**
8	The command to read the configuration of the LED

Response:

Valid Command: !AAS[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The address of the responding module (00 to FF)
S	0~7: Shows the count of channels 0~7 8: Rotates the count of channels 0~7 9: Shows the host control

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:

$018

Response:

!010

Read the configuration of the LED and returns a response indicating the LED is showing the count for DI channel 0.

Related Commands:

Section 2.16 $AA8V, Section 2.17 $AA9(Data)

2.16. $AA8V

Description:

This command is used to set the configuration of the LED.

Syntax:

$AA8[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF) The command to set the configuration of the LED
V	0~7: Shows the count of channels 0~7 8: Rotates mode 9: Host control mode

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The 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:

$0181

Response:

!01

Sets the LED to show the count for DI channel 1 and returns a valid response.

Related Commands:

Section 2.15 $AA8V, Section 2.17 $AA9(Data)

2.17. $AA9(Data)

Description:

This command is used to send data to the LED display.

Syntax:

$AA8[CHKSUM](CR)

$	Delimiter character
AA	The address of the module where the data is to be sent (00 to FF) The command to send data to the LED display
(Data)	5 decimal digits + 1 decimal point (Max. = 99999. , Min. = 0.0000)

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The 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:

$0199999.

Response:

!01

Displays “99999.” when the configuration LED is set to Host Control mode and returns a valid response.

Related Commands:

Section 2.15 $AA8V, Section 2.16 $AA8V

2.18. $AAB

Description:

This command is used to read the power-down count.

Syntax:

$AAB[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
B	The command to read the power-down count

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
(Data)	2 hexadecimal digits (00 to FF) representing the count data

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:

$01B

Response:

!0110

Reads the power-down count for module 01 and returns a response indicating that has happened 16 times.

Related Commands:

Section 2.19 $AABR

2.19. $AABR

Description:

This command is used to clear the power-down count.

Syntax:

$AABR[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
BR	The command to clear the power-down count

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

$01B

Response:

!0110

Reads the power-down count of module 01 and returns a response indicating that has happened 16 times.

Command:

$01BR

Response:

!01

Clears the power-down count of module 01 and returns a valid response indicating that the command was successful.

Command:

$01B

Response:

!0100

Reads the power-down count of module 01 and returns a response indicating that a power-down event has never occurred.

Related Commands:

Section 2.18 $AAB

2.20. $AACnD

Description:

This command is used to read the duty cycle value of a specific channel.

Syntax:

$AACnD[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
Cn	Specifies the channel to be read
D	The command to read the duty cycle value

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The address of the responding module (00 to FF)
(Data)	The duty cycle value for the specified channel (00.1 to 99.9)

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:

$01C0D

Response:

!0150.0

Reads the duty cycle value for PWM channel 0 of module 01 and returns a value of 50%.

Command:

$01C1D

Response:

!0133.3

Reads the duty cycle value for PWM channel 1 of module 01 and returns a value of 33.3%.

Related Commands:

Section 2.21 $AACnD(Data)

2.21. $AACnD(Data)

Description:

This command is used to set the duty cycle value for a specific channel.

Syntax:

$AACnD[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
D	The command to set the duty cycle value
(Data)	The duty cycle value for the specified channel (00.1 to 99.9)

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to F**F)**
(Data)	The actual duty cycle value for the specified channel (00.1 to 99.9)

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:

$01C0D50.0

Response:

!0150.0

Sets the duty cycle value for PWM channel 0 of module 01 to 50% and returns the true output value of 50%.

Command:

$01C1D33.4

Response:

!0133.3

Sets the duty cycle of PWM channel 1 of module 01 to 33.4% and returns the true output value of 33.3%.

Related Commands:

Section 2.20 $AACnD

2.22. $AACnF

Description:

This command is used to read the frequency value of a specific channel.

Syntax:

$AACnF[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
Cn	Specifies the channel to be read
F	The command to read the frequency value

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect.)
AA	The address of the responding module (00 to FF)
(Data)	The actual frequency value for the specified channel (000001 to 500000)

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:

$01C0F

Response:

!01500000

Reads the frequency value for PWM channel 0 of module 01 and returns a value of 500 KHz.

Command:

$01C2F

Response:

!01000001

Reads the frequency value for PWM channel 2 of module 01 and returns a value of 1 Hz.

Related Commands:

Section 2.23 $AACnF(Data)

2.23. $AACnF(Data)

Description:

This command is used to set the frequency value for a specific channel.

Syntax:

$AACnF(Data)[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
F	The command to set the frequency value
(Data)	The frequency value for the specified channel (000001 to 500000)

Response:

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

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to F**F)**
(Data)	The actual frequency value for the specified channel (000001 to 500000)

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:

$01C0F500000

Response:

!01500000

Sets the frequency value for PWM channel 0 of module 01 to 500 KHz and returns the actual frequency of 500 KHz. The duty cycle value will be automatically set to 50.0%.

Command:

$01C2F340000

Response:

!01333333

Sets the frequency value for PWM channel 2 of module 01 to 340 KHz and returns the actual frequency of 333333 Hz. The duty cycle value will be automatically set to 33.3%.

Related Commands:

Section 2.22 $AACnF

Note:

After using the $AACnF(Data) command, the duty cycle value will be automatically reset to 50.0%.

2.24. $AACnM

Description:

This command is used to read the continuous mode status of a specific channel.

Syntax:

$AACnM[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
Cn	Specifies the channel to be read
M	The command to read the continuous mode

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
S	0: PWM continuous mode is disabled 1: PWM continuous mode is enabled

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

Examples:

Command:

$01C0M

Response:

!010

Reads the PWM continuous mode of channel 0 and returns a response indicating that it is disabled.

Command:

$01C1M

Response:

!011

Reads the PWM continuous mode of channel 1 and returns a response indicating that it is enabled.

Related Commands:

Section 2.25 $AACnMS, Section 2.26 $AACnP, Section 2.27 $AACnP(Data)

2.25. $AACnMS

Description:

This command is used to set the continuous mode for a specific channel.

Syntax:

$AACnMS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
M	The command to set the continuous mode
S	0: Disables the PWM continuous mode 1: Enables the PWM continuous mode (If the PWM continuous mode is enabled, the step value for PWM will be automatically set to 1)

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

$01C0M1

Response:

!01

Sets the PWM continuous mode of channel 0 to enabled and returns a valid response. The PWM step value will be automatically set to 1.

Command:

$01C1M0

Response:

!01

Sets the PWM continuous mode of channel 1 to disabled and returns a valid response. The PWM step value will not be affected.

Related Commands:

Section 2.24 $AACnM, Section 2.26 $AACnP, Section 2.27 $AACnP(Data)

2.26. $AACnP

Description:

This command is used to read the PWM step value of a specific channel.

Syntax:

$AACnP[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF**)**
Cn	Specifies the channel to be read
P	The command to read the PWM step value

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
(Data)	The PWM step value (0001 to FFFF)

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:

$01C0P

Response:

!01001A

Reads the PWM step value for channel 0 and returns a value of 26 steps.

Command:

$01C1P

Response:

!011000

Reads the PWM step value for channel 1 and returns a value of 4096 steps.

Related Commands:

Section 2.24 $AACnM, Section 2.25 $AACnMS, Section 2.27 $AACnP(Data)

2.27. $AACnP(Data)

Description:

This command is used to set the PWM step value for a specific channel.

Syntax:

$AACnP(Data)[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
P	The command to set the PWM step value
(Data)	The PWM steps (0001 to FFFF) (When set to more than 1 step, the PWM continuous mode will be automatically set to disabled)

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:

$01C0P0001

Response:

!01

Sets the PWM step value for channel 0 to 1 and returns a valid response.

Command:

$01C1P001A

Response:

!01

Sets the PWM step value for channel 1 to 4096 steps and returns a valid response. The PWM continuous mode for channel 1 will be automatically set to disabled.

Related Commands:

Section 2.24 $AACnM, Section 2.25 $AACnMS, Section 2.26 $AACnP

2.28. $AACnT

Description:

This command is used to read the status of the PWM hardware trigger of a specific channel.

Syntax:

$AACnT[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
Cn	Specifies the channel to be read
T	The command to read the PWM hardware trigger

Response:

Valid Command: !AAS[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
S	0: The hardware trigger is disabled 1: The trigger start is enabled 2: The trigger stop is enabled

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

Examples:

Command:

$01C0T

Response:

!011

Reads the status of the hardware trigger for PWM channel 0 and returns a response indicating that the PWM channel 0 trigger will start when the rising edge of the DI is received.

Command:

$01C1T

Response:

!010

Reads the status of the hardware trigger for PWM channel 1 and returns a response indicating that PWM channel 1 will not be affected when the rising edge of the DI is received.

Related Commands:

Section 2.29 $AACnTS

2.29. $AACnTS

Description:

This command is used to set the hardware trigger for a specific channel.

Syntax:

$AACnTS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
T	The command to set the PWM hardware trigger
S	0: Disables the hardware trigger 1: Enables the trigger start 2: Enables the trigger stop

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The 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:

$01C0T2

Response:

!01

Sets the status of the hardware trigger for PWM channel 0 to trigger stop and returns a valid response. When the rising edge of the DI is received, the status of the PWM will be set to trigger stop.

Command:

$01C1T0

Response:

!010

Sets the status of the hardware trigger for PWM channel 1 to disabled and returns a valid response. The PWM channel 1will not be affected when the rising edge of the DI is received.

Related Commands:

Section 2.28 $AACnT

2.30. $AACnN

Description:

This is command is used to read the PWM synchronization status of a specific channel.

Syntax:

$AACnN[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
Cn	Specifies the channel to be read
N	The command to read the PWM synchronization status

Response:

Valid Command: !AAS[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
S	0: PWM synchronization is disabled 1: PWM synchronization is enabled

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

Examples:

Command:

$01C0N

Response:

!011

Reads the synchronization status of PWM channel 0 and returns a response indicating that it is enabled.

Command:

$01C1N

Response:

!010

Reads the synchronization status of PWM channel 1 and return a response indicating that it is disabled.

Related Commands:

Section 2.31 $AACnNS, Section 2.39 $AAYS

2.31. $AACnNS

Description:

This command is used to set the PWM synchronization status for a specific channel.

Syntax:

$AACnN[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
Cn	Specifies the channel to be set
N	The command to set the PWM synchronization
S	0: Disables the PWM synchronization 1: Enables the PWM synchronization

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The 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:

$01C0N1

Response:

!01

Sets the synchronization status for PWM channel 0 to enabled and returns a valid response.

Command:

$01C1N0

Response:

!01

Sets the synchronization status for PWM channel 1 to disabled and returns a valid response.

Related Commands:

Section 2.30 $AACnN, Section 2.39 $AAYS

2.32. $AAF

Description:

This command is used to read the firmware version of a module.

Syntax:

$AAF[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
F	The 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	The address of the responding module (00 to FF)
(Data)	The firmware version of the module as a string value

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

Examples:

Command:

$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.

2.33. $AAI

Description:

This command is used to read the status of the INIT switch of a module.

Syntax:

$AAI[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
I	The command to read the status of the INIT switch of the module

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
S	0: The INIT switch is in the INIT position 1: The INIT switch is in the Normal position

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 status of the INIT switch of module 01. The response shows that the INIT switch is in the INIT position.

2.34. $AAM

Description:

This command is used to read the name of a module.

Syntax:

$AAM[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
M	The 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	The address of the responding module (00 to FF)
(Name)	The name of the module as a string value

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

Examples:

Command:

$01M

Response:

!017088

Reads the name of module 01 and returns the name “7088”.

Related Commands:

Section 2.46 ~AAO(Name)

2.35. $AAP

Description:

This command is used to read which communication protocol is supported and being used by the module.

Syntax:

$AAM[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read (00 to FF)
P	The command to read the communication protocol

Response:

Valid Command: !AASC[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid comman**d**
AA	The address of the responding module (00 to FF)
S	0: Only the DCON protocol is supported 1: Both the DCON and Modbus RTU protocols are supported
C	0: The protocol set in EEPROM is DCON 1: The protocol set in 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 which communication protocol is being used by 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 2.36 $AAPN

2.36. $AAPN

Description:

This command is used to set the communication protocol.

Syntax:

$AAM[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set (00 to FF)
P	The command to set the communication protocol
N	0: DCON 1: Modbus RTU

Before using this command, the rear slide switch must be in the INIT position, see Section A.1 for details. The new protocol is saved in the EEPROM and will be effective after the next power-on reset.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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 for 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 for module 01 to Modbus RTU and returns a valid response.

Related Commands:

Section 2.35 $AAP

2.37. $AAR

Description:

This command is used to reset the PWM.

Syntax:

$AAR[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be reset (00 to FF)
R	The command to reset the PWM

Response:

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

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

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

$01R

Response:

!01

Resets the PWM and stops all of the outputs, and returns a valid response.

2.38. $AAW

Description:

This command is used to save the PWM configuration.

Syntax:

$AAW[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be accessed (00 to FF)
W	The command to save the PWM configuration

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

$01W

Response:

!01

Saves the PWM configuration for all channels into the EEPROM and returns a valid response. After the next power on, the PWM configuration will be automatically loaded from the EEPROM without giving any notification.

2.39. $AAYS

Description:

This command is used to start the PWM synchronization.

Syntax:

$AAYS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be accessed (00 to FF)
Y	The command to set the PWM synchronization
S	0: Stops the PWM synchronization 1: Starts the PWM synchronization

Response:

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

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

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

$01Y1

Response:

!01

Starts the PWM output that has been set to synchronized and returns a valid response.

Command:

$01Y0

Response:

!01

Stops the PWM output that has been set to synchronized and returns a valid response.

2.40. @AADODD

Description:

This command is used to set the status of the PWM output port.

Syntax:

@AADODD[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set (00 to FF)
DO	The command to set the status of the PWM output port
DD	A two-digit hexadecimal value, where bit 0 corresponds to PWM channel 0, and bit 1 corresponds to PWM channel 1, etc. When the bit is 0, it denotes that the PWM output port is off, and 1 denotes that the PWM output port is on.

Response:

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

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

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

@01DO33

Response:

!01

Sets the PWM output port for channel 0 to on, channel 1 to on, channel 2 to off, channel 3 to off, channel 4 to on, and channel 5 to on, and the module returns a valid response.

Related Commands:

Section 2.41 @AADI

Note:

When a Host Watchdog timeout occurs, the module will return an invalid response for this command and the PWM value that was sent is ignored.

2.41. @AADI

Description:

This command is used to read the status of the PWM output port and the digital input port.

Syntax:

@AADI[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to FF)
DI	The command to read the status of the PWM and digital input

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
OO	A two-digit hexadecimal value, where bit 0 corresponds to PWM channel 0, and bit 1 corresponds to PWM channel 1, etc. When the bit is 0, it denotes that the PWM is inactive and 1 denotes that the PWM is active.
II	A two-digit hexadecimal value, where bit 0 corresponds to DI channel 0, and bit 1 corresponds to DI channel 1, etc. When the bit is 0, it denotes that the DI is inactive and 1 denotes that the DI is active.

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

Examples:

Command:

@01DI

Response:

!0101F0

Reads the status of the PWM and DI and returns a response indicating that PWM channel 0 is active and the others are inactive, and DI channels 4, 5, 6 and 7 are active and the others are inactive.

Related Commands:

Section 2.40 @AADODD, Section2.44 ~AAD, Section 2.45 ~AADVV

2.42. @AAGN

Description:

This command is used to read the preset count value of a specific channel.

Syntax:

@AAGN[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read (00 to FF**)**
G	The command to read the preset value of the DI counter
N	Specifies the channel to be read, zero based

Response:

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

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command (An invalid command is returned if the specified channel is incorrect)
AA	The address of the responding module (00 to FF)
(Data)	8 hexadecimal digits (00000000 to FFFFFFFE) representing the preset count value

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

Examples:

Command:

@01G0

Response:

!0100000000

Reads the preset count value for counter 0 of module 01 and returns a response indicating that the preset value is 0.

Related Commands:

Section 2.43 @AAPN(Data)

2.43. @AAPN(Data)

Description:

This command is used to set the preset count value of a specific channel.

Syntax:

@AAPN(Data)[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set (00 to FF)
P	The command to set the preset value of the DI counter
N	Specifies the channel to be set, zero based.
(Data)	8 hexadecimal digits (00000000 to FFFFFFFE) representing the preset count value

Response:

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

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

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

@01P000000000

Response:

!01

Sets the preset count value for counter 0 of module 01 to 0 and returns a response indicating that the command was successful.

Related Commands:

Section 2.42 @AAGN

2.44. ~AAD

Description:

This command is used to read the miscellaneous settings.

Syntax:

~AAD[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to FF)
D	The command to read the miscellaneous settings

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
VV	A two-digit hexadecimal value, where bit 0 corresponds to the active status of the DI, as indicated below. The other bits are reserved. 0: Input value 1 for non-signal or low voltage Input value 0 for high voltage 1: Input value 1 for high voltage Input value 0 for non-signal or low voltage

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

Examples:

Command:

$01D

Response:

!0101

Reads the miscellaneous settings of module 01 and returns a value of 01.

Related Commands:

Section 2.45 ~AADVV

2.45. ~AADVV

Description:

This command is used to set the miscellaneous settings.

Syntax:

~AADVV[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
D	The command to set the miscellaneous settings
VV	A two-digit hexadecimal value, where bit 0 corresponds to the active status of the DI, as indicated below. The other bits are reserved. 0: Input value 1 for non-signal or low voltage Input value 0 for high voltage 1: Input value 1 for high voltage Input value 0 for non-signal or low voltage

Response:

Valid Command: !AAVV[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

$01D01

Response:

!01

Sets the miscellaneous settings of module 01 and returns a valid response.

Related Commands:

Section 2.44 ~AAD

2.46. ~AAO(Name)

Description:

This command is used to set the name of a module.

Syntax:

~AAO(Name)[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
O	The command to set the name of the module
(Name)	The 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	The 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:

~01O7088

Response:

!01

Sets the name of module 01 to “7088” and returns a valid response.

Command:

$01M

Response:

!017088

Reads the name of module 01 and returns the name “7088”.

Related Commands:

Section 2.34 $AAM

2.47. ~AARD

Description:

This command is used to read the response delay time.

Syntax:

~AARD[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to FF)
RD	The command to read the response delay tim**e**

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
TT	Two hexadecimal digits to represent the response time value in milliseconds. The value must be less than or equal to 1E. For example, 01 denotes 1 millisecond and 1A denotes 26 milliseconds.
AA	The 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:

~01RD10

Response:

!01

Sets the response time to 16 milliseconds and returns a valid response.

Command:

~01RD

Response:

!0110

Reads the response time and returns a value of 16 milliseconds. The response will be sent after 16 milliseconds have elapsed.

Related Commands:

Section 2.48 ~AARDTT

2.48. ~AARDTT

Description:

This command is used to set the response delay time.

Syntax:

~AARDTT[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
RD	The command to set response time
TT	Two hexadecimal digits to represent the response time value in milliseconds. The value must be less than or equal to 1E. For example, 01 denotes 1 millisecond and 1A denotes 26 milliseconds.

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

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

~01RD10

Response:

!01

Sets the response time to 16 milliseconds and returns a valid response.

Command:

~01RD

Response:

!0110

Reads the response time and returns a value of 16 milliseconds. The response will be sent after 16 milliseconds have elapsed.

Related Commands:

Section 2.47 ~AARD

2.49. ~**

Description:

This command is used to inform all modules that the Host is OK.

Syntax:

~[CHKSUM](CR)**

~	Delimiter character
**	The “Host OK” command

Response:

No response.

Examples:

Command:

~**

No response

Sends a “Host OK” command to all modules.

Related Commands:

Section 2.50 ~AA0, Section 2.51 ~AA1, Section 2.52 ~AA2, Section 2.53 ~AA3ETT

Related Topics:

Section A.2 Dual Watchdog Operation

2.50. ~AA0

Description:

This command is used to read the status of a module's Host Watchdog.

Syntax:

~AA0[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to FF)
0	The command to read the status of the Host Watchdog

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
SS	Two hexadecimal digits that represent the status of the Host Watchdog, where: Bit 2: 0 indicates that no Host Watchdog timeout has occurred, and 1 indicates that a Host Watchdog timeout has occurred. Bit 7: 0 indicates that the host watchdog is disabled, and 1 indicates that the Host Watchdog is enabled. The status of the Host Watchdog 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 status of the Host Watchdog 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 status of the Host Watchdog of module 02 and returns 04, meaning that a Host Watchdog timeout has occurred.

Related Commands:

Section 2.49 ~**, Section 2.51 ~AA1, Section 2.52 ~AA2, Sec 2.53 ~AA3ETT

Related Topics:

Section A.2 Dual Watchdog Operation

2.51. ~AA1

Description:

This command is used to reset the timeout status of a module's Host Watchdog.

Syntax:

~AA1[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be reset (00 to FF)
1	The command to reset the timeout status of the Host Watchdog

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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 status of the Host Watchdog of module 01 and shows that a Host Watchdog timeout has occurred.

Command:

~011

Response:

!01

Resets the timeout status of the Host Watchdog of module 01 and returns a valid response.

Command:

~010

Response:

!0100

Reads the status of the Host Watchdog of module 01 and shows that no Host Watchdog timeout has occurred.

Related Commands:

Section 2.49 ~**, Section 2.50 ~AA0, Section 2.52~AA2, Section 2.53~AA3ETT

Related Topics:

Section A.2 Dual Watchdog Operation

2.52. ~AA2

Description:

This command is used to read the timeout value of a module's Host Watchdog.

Syntax:

~AA2[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read (00 to FF)
2	The command to read the Host Watchdog timeout value

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to FF)
E	0: The Host Watchdog is disabled 1: The Host Watchdog is enabled
TT	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 2.49 ~**, Section 2.50 ~AA0, Section 2.51 ~AA1, Section 2.53 ~AA3ETT

Related Topics:

Section A.2 Dual Watchdog Operation

2.53. ~AA3ETT

Description:

This command is used to enable/disable a module's Host Watchdog and set the timeout value of the Host Watchdog.

Syntax:

~AA3ETT[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
3	The command to set the Host Watchdog
E	0: Disables the Host Watchdog 1: Enables the Host Watchdog
TT	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	The 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 2.49 ~**, Section 2.50 ~AA0, Section 2.51 ~AA1, Section 2.52 ~AA2

Related Topics:

Section A.2 Dual Watchdog Operation

Note:

When a Host Watchdog timeout occurs, the Host Watchdog is disabled and all PWM outputs are stopped. The ~AA3EVV command should be sent again to re-enable the Host Watchdog.

2.54. ~AAI

Description:

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

Syntax:

~AAI[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
I	The command to set the software INIT

Response:

Valid Command: !AA[CHKSUM](CR)

InValid Command: ?AA[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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 software INIT of module 01 and returns a valid response.

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.55 ~AATnn

Related Topics:

Section A.1 INIT Mode

Note:

The ~AATnn command should be sent prior to sending this command, see Section 2.55 for details.

2.55. ~AATnn

Description:

This command is used to set the timeout value for the software INIT.

Syntax:

~AATnn[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be set (00 to FF)
T	The command to set the timeout value for the software INIT.
nn	Two hexadecimal digits representing the timeout value in seconds. The max. timeout value is 60 seconds. When changing the Baud Rate and checksum settings without altering the position of 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 software INIT timeout value. If the software INIT timeout value is 0, then the Baud Rate and checksum settings cannot be changed using software only. The power-on reset value of the Software INIT timeout is 0.

Response:

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

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

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the responding module (00 to 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 software INIT of module 01 and returns a valid response.

Command:

%0101500700

Response:

?01

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

Command:

~01T10

Response:

!01

Sets the software INIT timeout value of module 01 to 16 seconds and returns a valid response.

Command:

~01I

Response:

!01

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

Command:

%0101500700

Response:

!01

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

Related Commands:

Section 2.1 %AANNTTCCFF, Section 2.54 ~AAI

Related Topics:

Section A.1 INIT Mode

Note:

It is recommended that the software INIT timeout value is reset to 0 once any changes to the Baud Rate and checksum settings have been completed in order to ensure that these settings are not inadvertently modified.

3. Modbus RTU Protocol

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

M-7000 series modules support the Modbus RTU protocol. The communication Baud Rates range from 1200bps to 115200bps. The number of data bits is fixed to 8. The following Modbus functions are supported:

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

If the function specified in the message is not supported, then the module responds as follows:

Error Responses

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

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

3.1. 02 (0x02) Read PWM Status

This function code is used to read the PWM status of a module.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02 ~ 03	Starting channel	2 Bytes	0x00 to 0x07, where 0x00 corresponds to channel 0, 0x01 corresponds to PWM channel 1, etc.
04 ~ 05	Number of input channels	2 Bytes	N, 1 to 8; (Starting channel + N) This should be less than or equal to 0x08

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x02
02	Byte count	1 Byte	1
03	Input channel data	1 Byte	A bit corresponds to a channel. When the bit is 1, it denotes that the channel is outputting PWM signals. If the bit is 0 it denotes that the channel is not set to use PWM output.

Error Response

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

3.2. 04 (0x04) Read DI Count

This function code is used to read the count of the analog input channels.

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x04
02 ~ 03	Starting channel	2 Bytes	0x00 to 0x0F, where 0x00 corresponds to the low word of channel 0, 0x01 corresponds to the high word of channel 0, etc
04 ~ 05	Number of input channels (N)	2 Bytes	N, 1 to 8; (Starting channel + N) should be less than or equal to 0x08

Response

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

Error Response

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

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

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

Sub-function Code	Description	Section
00 (0x00)	Reads the module name	3.3.1
04 (0x04)	Sets the module address	3.3.2
05 (0x05)	Reads the communication settings	3.3.3
06 (0x06)	Sets the communication settings	3.3.4
32 (0x20)	Reads the firmware version	3.3.5
41 (0x29)	Reads the miscellaneous settings	3.3.6
42 (0x2A)	Writes the miscellaneous settings	3.3.7

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

Error Response

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

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

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

Request

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

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x00
03 ~ 06	Module name	4 Bytes	0x00 0x70 0x88 0x00

Error Response

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

3.3.2. Sub-function 04 (0x04) Set module address

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x04
03	New address	1 Byte	1 to 247
04 ~ 06	Reserved	3 Bytes	0x00 0x00 0x00

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x04
03	Set address result	1 Byte	0: OK Others: error
04 ~ 06	Reserved	3 Bytes	0x00 0x00 0x00

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: The new address is out of range, reserved bytes should be filled with zero, or an incorrect number of bytes were received

3.3.3. Sub-function 05 (0x05) Read communication settings

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x05
03	Reserved	1 Byte	0x00

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x05
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0x03 ~ 0x0A, Baud Rate code, see Section 1.10 for details.
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0x00: No parity, 1 stop bit 0x01: No parity, 2 stop bits 0x02: Even parity, 1 stop bit 0x03: Odd parity, 1 stop bit Reserved for other modules or firmware versions and should be zero
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: DCON protocol 1: Modbus RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Note: This information is the data saved in the EEPROM and will be used for the next power-on reset. It is not the currently used settings.

Error Response

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

3.3.4. Sub-function 06 (0x06) Set communication settings

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0x03 ~ 0x0A, Baud Rate code, see Section 1.10 for details.
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0x00: No parity, 1 stop bit 0x01: No parity, 2 stop bits 0x02: Even parity, 1 stop bit 0x03: Odd parity, 1 stop bit Reserved for other modules or firmware versions and should be zero
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: DCON protocol 1: Modbus RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0: OK Others: error
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0: OK Others: error
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: OK Others: error
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Note: The new Baud Rate and protocol will be effective after the next power-on reset.

Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: The Baud Rate or mode is out of range, reserved bytes should be filled with zeros, or an incorrect number of bytes were received

3.3.5. Sub-function 32 (0x20) Read firmware version

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

Request

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

Response

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

Error Response

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

3.3.6. Sub-function 41 (0x29) Read miscellaneous

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

Request

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

Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x29
03	Miscellaneous settings	1 Byte	The checksum settings and the input range settings of the module, see Section 1.7 for details

Note: The reserved fields are filled with zeros.

Error Response

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

3.3.7. Sub-function 42 (0x2A) Write miscellaneous settings

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

Request

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	0x46
02	Sub-function code	1 Byte	0x2A
03	Miscellaneous settings	1 Byte	The checksum settings and the input range settings of the module, see Section 1.7 for details

Note: Reserved fields are filled with zeros.

Response

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

Error Response

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

3.4. M-7088 Address Mappings

Address	Description	Attribute
00001 ~ 00008	PWM status 0: Stopped 1: Started	R/W
00065 ~ 00072	Overflow of DI counter, write 1 to clear	R/W
00225 ~ 00232	DI counter status 0: Disabled 1: Enabled	R/W
00257	Protocol 0: DCON 1: Modbus RTU	R/W
00260	Modbus Host Watchdog mode 0: The same as I-7000 series modules 1: The AO and DO command will clear Host Watchdog timeout status	R/W
00261	Host Watchdog 0: Disabled 1: Enabled	R/W
00266	Clear all DI counters	W
00270	Host Watchdog timeout status, write 1 to clear the Host Watchdog timeout status	R/W
00278	DI active 0: Inverse 1: Normal	R/W
00289	Save all PWM configurations into EEPROM, write 1 to save	W
00865 ~ 00872	PWM mode 0: Burst mode 1: Continuous mode	R/W

00897 ~ 00904	PWM trigger status 0: Disabled 1: Enabled	R/W
Address	Description	Attribute
00929 ~ 00936	PWM trigger start 0: Trigger stop 1: Trigger start	R/W
00961 ~ 00968	PWM synchronized	R/W
10033 ~ 10040	DI status	R
10273	Reset status 0: Not the first read after power-on 1: First read after power-on	R
30001 ~ 30016	DI count 30001=low word of channel 0, 30002=high word of channel 0, etc.	R
30769 ~ 30776	PWM burst count Condition: PWM mode = burst, PWM status = stop	R
30481	Firmware version (low word)	R
30482	Firmware version (high word)	R
30483	Module name (low word)	R
30484	Module name (high word)	R
40065 ~ 40080	Max DI count value 40065=low word of channel 0, 40066=high word of channel 0, etc.	R/W
40097 ~ 40112	Preset value of DI count 40097=low word of channel 0, 40104=high word of channel 0, etc.	R/W
40485	The module address, valid range: 1 ~ 247	R/W
40705 ~ 40712	PWM duty cycle	R/W

<table>
<tr><td>40737 ~
40752</td><td>PWM frequency
40737=low word of channel 0,
40738=high word of channel 0, etc.</td><td>R/W</td></tr>
<tr><td>40801 ~
40808</td><td>PWM burst steps</td><td>R/W</td></tr>
<tr><td>40486</td><td>Bits 5:0
Baud Rate, 0x03 ~ 0x0A
<table>
<tr><td>Code</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td></tr>
<tr><td>Baud</td><td>1200</td><td>2400</td><td>4800</td><td>9600</td></tr>
<tr><td>Code</td><td>0x07</td><td>0x08</td><td>0x09</td><td>0x0A</td></tr>
<tr><td>Baud</td><td>19200</td><td>38400</td><td>57600</td><td>115200</td></tr>
</table>
Bits 7:6
00: No parity, 1 stop bit
01: No parity, 2 stop bits
10: Even parity, 1 stop bit
11: Odd parity, 1 stop bit</td><td>R/W</td></tr>
<tr><td>40488</td><td>Modbus response delay time in ms, valid range: 0 ~ 30</td><td>R/W</td></tr>
<tr><td>40489</td><td>Host Watchdog timeout value, 0 ~ 255, in 0.1s</td><td>R/W</td></tr>
<tr><td>40495</td><td>LED configuration</td><td>R/W</td></tr>
<tr><td>40496</td><td>LED data for host mode</td><td>R/W</td></tr>
<tr><td>40498</td><td>Power-down count</td><td>R/W</td></tr>
</table>

4. Operation Principles & Application Notes

4.1. INIT* pin Operation Principles

All I-7000 modules contain an EEPROM that can be used to store configuration information. Consequently, it is difficult for the user to determine the status of the I-7000 module. If the INIT* pin is connected to the GND pin while powering on the module, the modules will enter the factory default settings without changing the EEPROM data. The factory default settings are as follows:

Address	= 00
Baud Rate	= 9600
Checksum	= Disabled
Data Format	= 1 Start + 8 Data + 1 stop

If the INIT* pin is disconnected from the GND pin, the I-7000 module will be automatically configured according to the EEPROM data. It is easy to determine the EEPROM configuration data in the default settings using the following steps:

Step 1: Power off the module and connect the INIT* pin to the GND pin

Step 2: Power on the module

Step 3: Send the command string $002[0x0D] to the module. The module will respond with the EEPROM data.

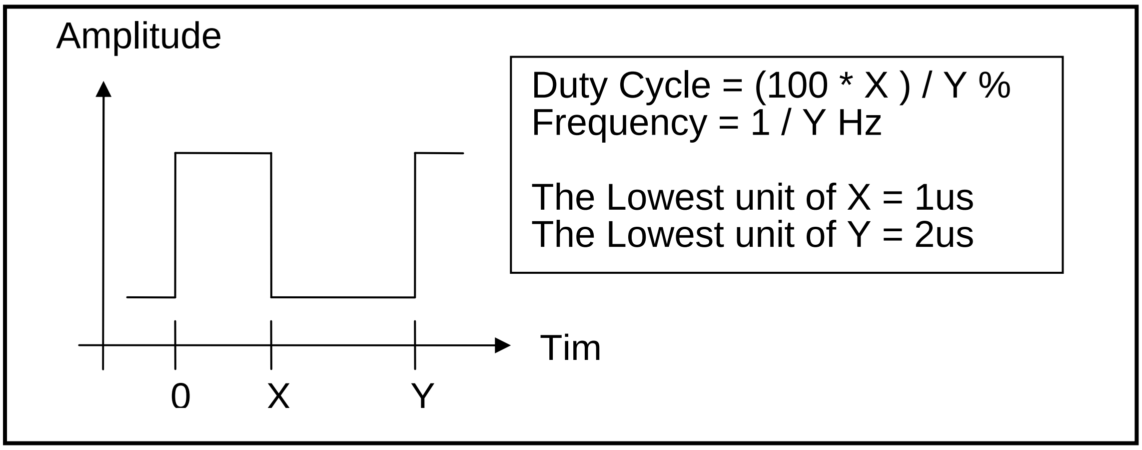
Step 4: Record the EEPROM data for this I-7000 module

Step 5: Power off the module and disconnect the INIT*_pin from the GND pin

Step 6: Power on the module

4.2. PWM Operation Principle

1. The PWM output modules will be turned OFF after being powered on for the first time.
2. If the Host Watchdog is active, the PWM output modules will stop automatically and cannot be re-started. The module status will then be set to 04 meaning that. If the host computer then sends n "@AADO" command to those modules, the command will be ignored and an invalid command delimiter character "?" will be sent in response as warning information. An "~AA1" command can be sent by the host to clear the module status to 0 and then the module will accept "@AADO" commands again.
3. The PWM configuration will be loaded from EEPROM after being powered on for the first time.
4. After modifying the PWM configuration, use the "$AAW" command to save all PWM configurations in the EEPROM so that they will be loaded when the module is next powered on.
5. The limit of the frequency and the duty cycle is:



(1) If the frequency is set to 1 ~ 1000Hz, the duty cycle can be set to 00.1% ~ 99.9%.

(2) If the frequency is set to 1001 ~ 10000Hz, the duty cycle can be set to 1% ~ 99%.

(3) Otherwise, the frequency and the duty cycle is not complete.

Examples:

If the frequency is 500000Hz (Supports a duty cycle of 50% only)

If the frequency is 333333Hz (Supports duty cycles of 33.3% and 66.6% only)

If the frequency is 400000Hz (Modifies the frequency to 333333Hz and supports duty cycles of 33.3% and 66.6% only)

Appendix

A.1. INIT Mode

Each I-7000 and M-7000 module has a built-in EEPROM that can be used to store configuration information, such as the module address, Type Code, and Baud Rate, etc. Occasionally, the configuration of a module may be forgotten and there are no visual indications of the configuration of the module. It is difficult to communicate with the module when the configuration of the module is unknown. To help avoid this problem, the I-7000 and M-7000 series has a special mode called “INIT mode”. When the module is powered on in “INIT mode” the configuration of the module is reset to the default settings shown below, allowing it to be operated as normal.

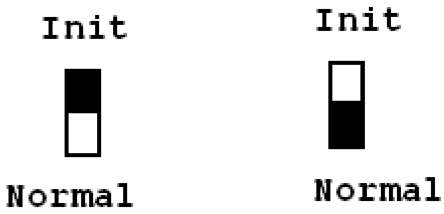
1. Address: 00
2. Baud Rate: 9600 bps
3. No checksum
4. Protocol: DCON

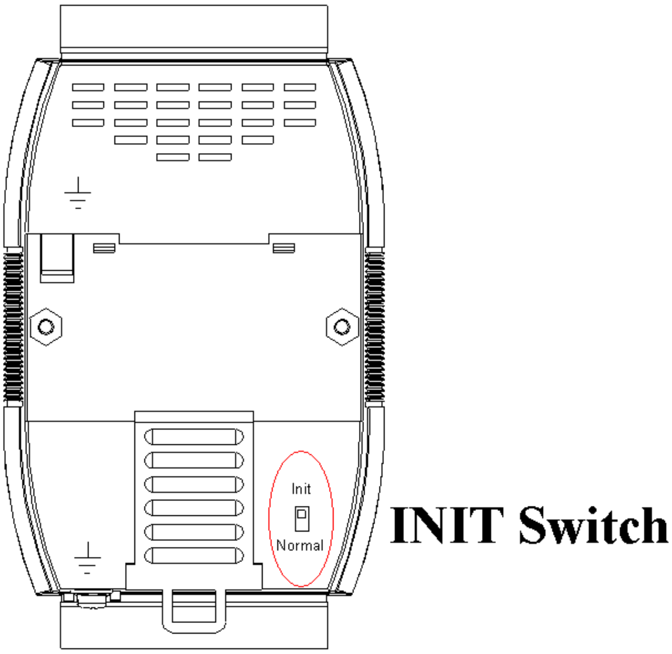
The configuration information stored in the EEPROM is not changed and can be read by sending the $002(CR) command at 9600bps.

There are other commands that require the module to be in INIT mode. They are:

1. %AANNTTCCFF, which is used when changing the Baud Rate and checksum settings. See Section 2.1 for details.
2. $AAPN, see Section 2.36 for details.

Originally, INIT mode was accessed by connecting the INIT* terminal to the GND terminal. New I-7000 and M-7000 modules have an INIT switch located on the rear of the module to allow easier access to INIT mode. For these modules, INIT mode is accessed by sliding the INIT switch to the Init position, as shown below.





A.2. Dual Watchdog Operation

Dual Watchdog = Module Watchdog + Host Watchdog

The Module Watchdog is a hardware reset circuit that monitors the operating status of the module. While working in harsh or noisy environments, the module may be shut down by external signals. The reset circuit allows the module to work continuously without disruption.

The Host Watchdog is a software function that monitors the operating status of the host. Its purpose is to prevent problems due to network/communication errors or host malfunctions. When a Host Watchdog timeout occurs, the module will reset all outputs to a safe state in order to prevent any erroneous operations of the controlled target.

I-7000 and M-7000 series modules include an internal Dual Watchdog, making the control system more reliable and stable.

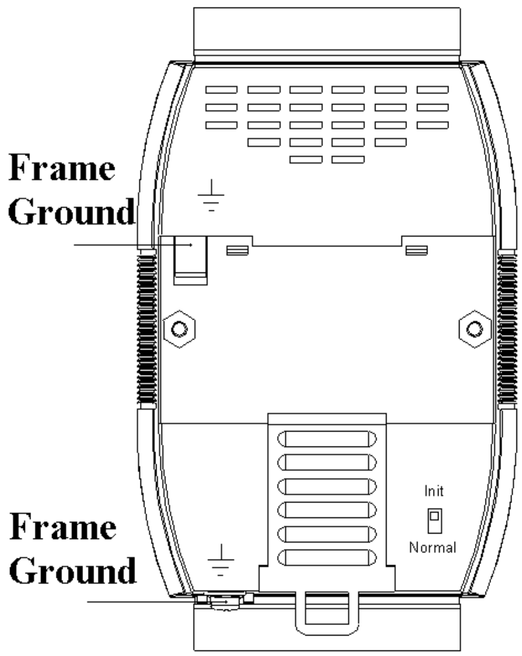
For more information regarding the Dual Watchdog, please refer to Chapter 5 of the "Getting Started for I-7000 Series Modules" manual that can be downloaded from the ICP DAS website http://www.icpdas.com.

A.3. Frame Ground

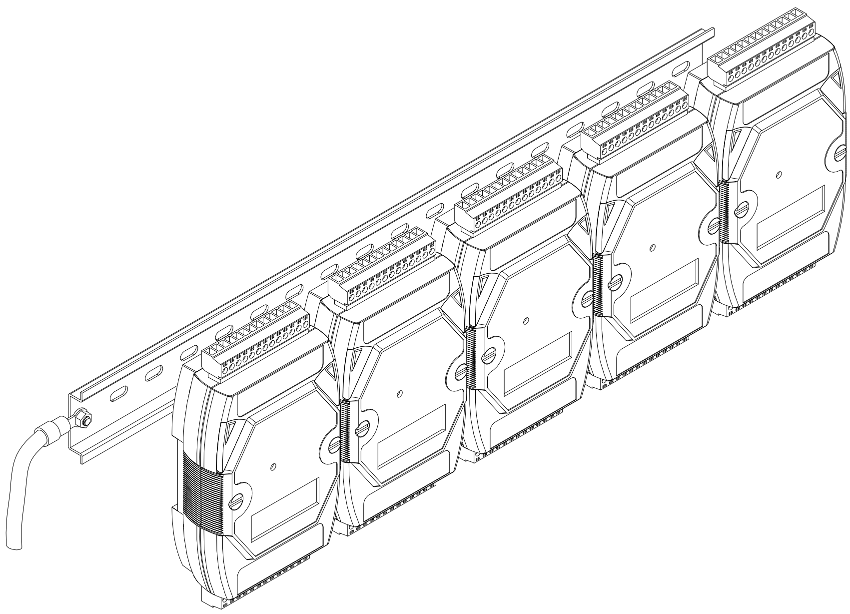
Electronic circuits are constantly vulnerable to Electro-Static Discharge (ESD), which become worse in a continental climate area. Some I-7000 and M-7000 modules feature a new design for the frame ground, which provides a path for bypassing ESD, allowing enhanced static protection (ESD) capabilities and ensures that the module is more reliable.

Either of the following options will provide a better protection for the module:

1. If the module is DIN-rail mounted, connect the DIN-rail to the earth ground. This is because the DIN-rail is in contact with the upper frame ground, as shown in the figure below.
2. Alternatively, connect the lower frame ground terminal to a wire and connect the wire to the earth ground, as shown in the figure below.

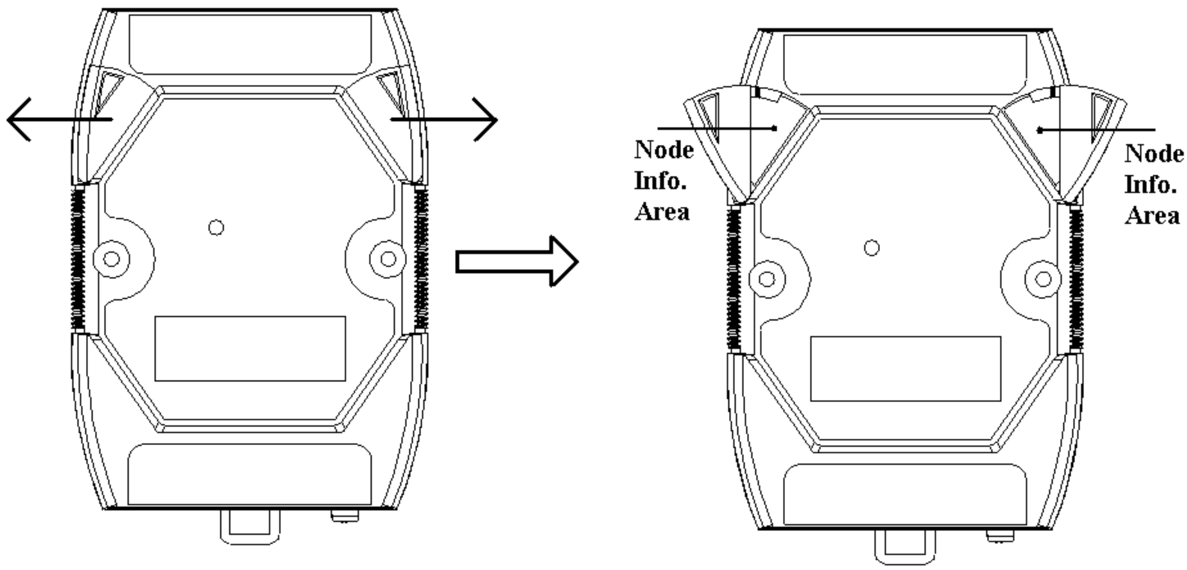


New DIN-rail models are available that can easily be connected to the earth ground. Each rail is made of stainless steel, which is stronger than those made of aluminum. There is a screw at one end and a ring terminal is included, as shown in the figure below. Refer to Section 1.12.1 for more information about the new DIN-rail models.



A.4. Node Information Area

Each I-7000 and M-7000 module has a built-in EEPROM to store configuration information, such as the module address, Type Code, and Baud Rate, etc. One minor drawback is that there are no visual indications of the configuration of the module. New I-7000 and M-7000 modules include node information areas that are protected by a cover, as shown below, and can be used to make a written record of the node information, such as module address, and Baud Rate, etc. To access the node information areas, first slide the covers outward, as shown in the figure below.



A.5. Reset Status

The reset status of a module is set when the module is powered on or when the module is reset by the Module Watchdog, and is cleared after the responding to the first $AA5 command. This can be used to check whether the module has recently been reset. When the response from the $AA5 command indicates that the reset status has been cleared, it means that the module has not been reset since the last $AA5 command was sent. When the response from the $AA5 command indicates that the reset status is set and it is not the first time an $AA5 command has been sent, it means that the module has been reset and the digital output value has been changed to the power-on value.