CB-7080 & CB-7080D
Counter/Timer

User's Guide

Document Revision 2 A
August 2005

MEASUREMENT COMPUTING™

Trademark and Copyright Information

Notice

Measurement Computing Corporation does not authorize any Measurement Computing Corporation product for use in life support systems and/or devices without prior written consent from Measurement Computing Corporation. Life support devices/systems are devices or systems that, a) are intended for surgical implantation into the body, or b) support or sustain life and whose failure to perform can be reasonably expected to result in injury. Measurement Computing Corporation products are not designed with the components required, and are not subject to the testing required to ensure a level of reliability suitable for the treatment and diagnosis of people.

Table of Contents

Introduction

CB-7000 is a family of network data acquisition and control modules. They provide A/D, D/A, DI/O, Timing/Counting and other functions. These modules can be remote controlled by a set of commands.

The common features of CB-7080 and CB-7080D include:

- Two independent 32-bit counters, counter 0 and counter 1.
- Input signals can be isolated or non-isolated.
- Programmable digital filter for isolated and non-isolated input.
- External gate control for isolated and non-isolated input.
- Programmable threshold value for non-isolated input.
- Programmable alarm output.
- Input frequency measurement up to 100 kHz.

The CB-7080D module identical to the CB-7080 with a 5-digit LED display added. The LEDs can display the counter value and input signal frequency without PC control.

Comparing the 7080 and 7080D

Table 1-1 summarizes the features of the CB7080 and CB7080D modules.

Table 1-1. Comparison between CB-7080 and CB-7080D

	CB-7080	**CB-7080D**
5-digit LED display	No	Yes
Response to LED command	No	Yes
Module name	programmable	programmable
Counter preset value	Yes (programmable)	Yes (programmable)
Alarm on counter 0 only	Yes (programmable)	Yes (programmable)
Alarm on counter 0 and 1	Yes (programmable)	Yes (programmable)
Channel 0 and channel 1 are both non-isolated (input mode 0, $AAB0)	Yes	Yes
Channel 0 and channel 1 are both isolated (input mode 1, $AAB1)	Yes	Yes
Channel 0 is non-isolated and channel 1 is isolated (input mode 2, $AAB2)	Yes	Yes
Channel 0 is isolated and channel 1 is non-isolated(input mode 3, $AAB3)	Yes	Yes
Input frequency	100 kHz max.	100 kHz max.
Default setting	4080 compatible ▪ High alarm on counter 0 and 1 ▪ Counter preset value: 0	4080D compatible ▪ High/High-High alarm on counter 0 ▪ Counter preset value: 0

Pin assignments

The pin names and locations on the CB-7080D module are shown in Figure 1-1.

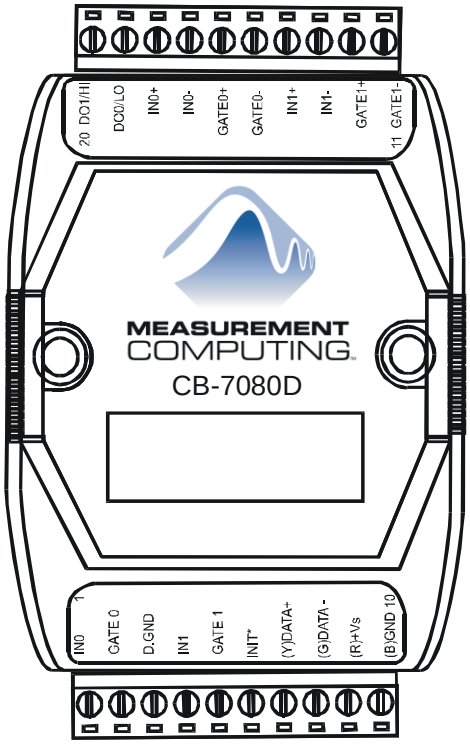

Figure 1-1. CB-7080D pin identification

Specifications

CB-7080	Counter/Frequency Module
CB-7080D	CB-7080 with LED Display
Counter Input Channels	Two independent 32 bit counters, counter 0 and 1.
Input signal	Isolated or non-isolated programmable
Isolation input levels:	
Logic level 0	+1 V max
Logic level 1	+3.5 V to +30 V
Isolation voltage	3750 V RMS
Non-isolation input threshold level:	Programmable
Logic level 0	0 to +5 V (default = 0.8 V)
Logic level 1	0 to +5 V (default = 2.4 V)
Maximum count	32 bit (4,294,967,295)
Programmable digital noise filter	2 µs to 65 ms
Alarming	alarm on counter 0 or counter 0 and 1, programmable
Counter preset value	Programmable
Display LED Indicator	5-digit read out, channel 0 or channel 1

Frequency measurement

Input frequency	1 Hz to 100 kHz max
Programmable built-in gate time	1.0 or 0.1 sec

Digital output

Capacity	2 channels, open-collector to 30 V, 30 mA max load
Power Dissipation	300 mW

Power

Power Requirements	+10V to 30V (non-regulated)

Power consumption

CB-7080	2.0 W
CB-7080D	2.2 W

Functional block diagram

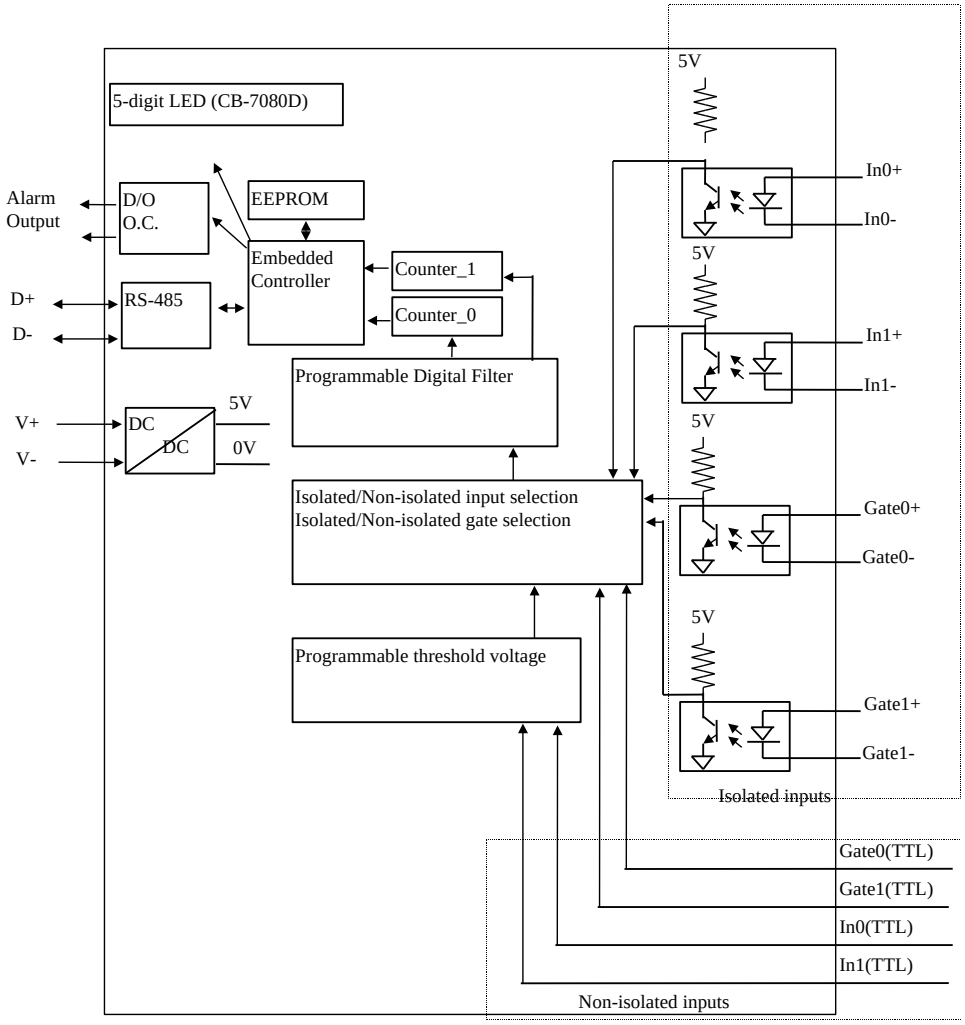

Figure 1-2. CB-7080D Block Diagram

Application wiring

Output drive to SSR or other load

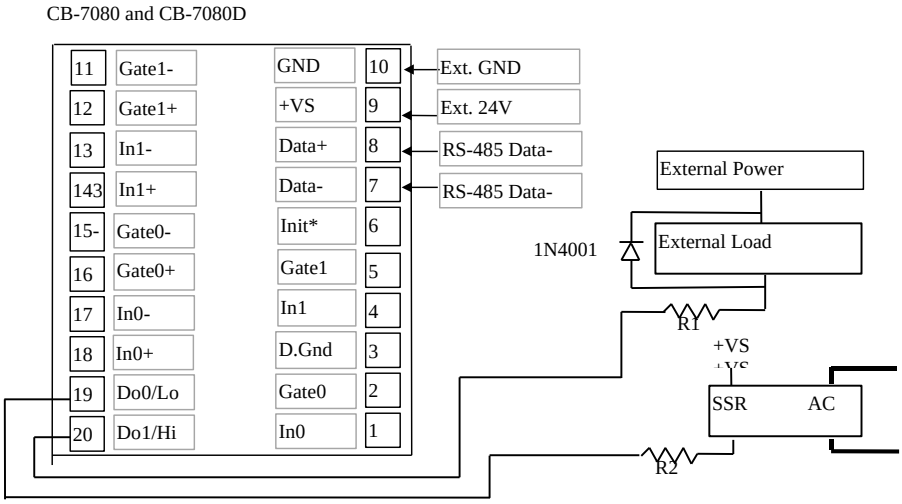

Figure 1-3. Output drive to SSR or other load

Note:

If the external load is resistive, the 1N4001 can be omitted. (transistor, lamp, resistor, etc.).

If the external load is inductive, the 1N4001 cannot be omitted. (relay coil, etc.).

Frequency Input

Use the $AABS command to select the isolated/non-isolated input.

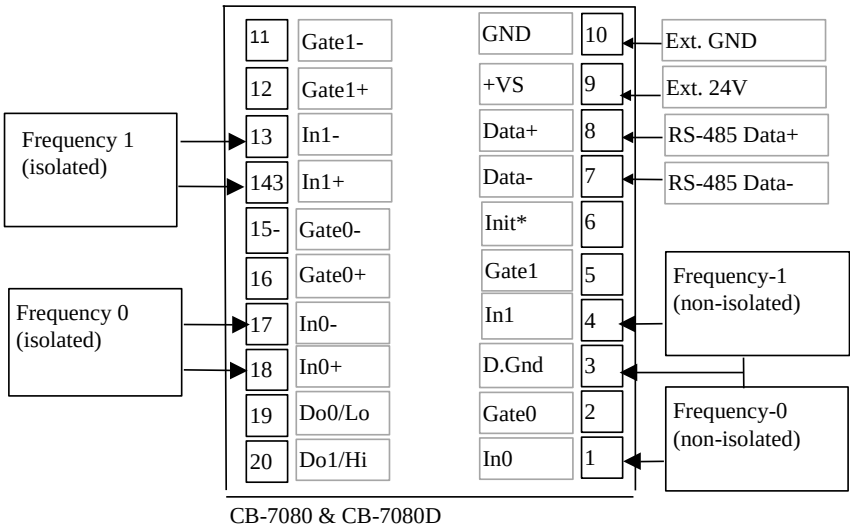

Figure 1-4. Frequency input

Counter input

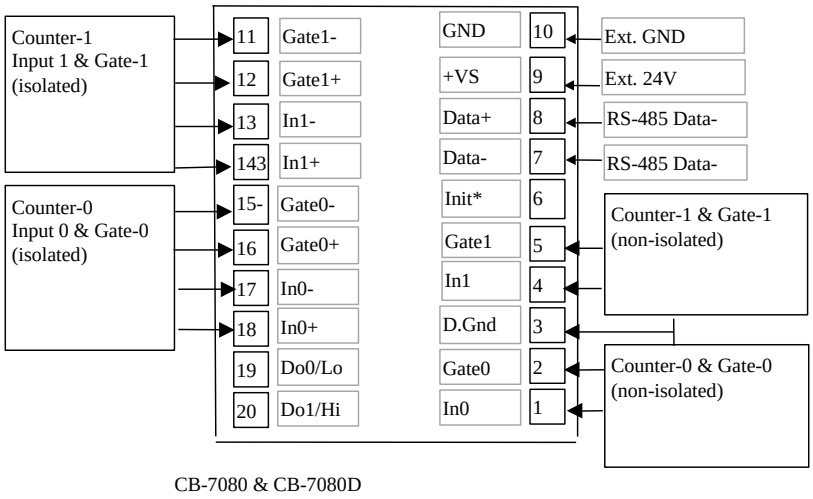

Figure 1-5. Counter input

Frequency input measurement

Perform the following procedure to measure the frequency input of each channel. Refer to Figure 1-4 for the wire connection.

1. Power on and run the test.exe application.

2. Press 2

3. Press $012[Enter] Receive=!01500600

4. Press 2

5. Press %0101510600[Enter] Receive=>!01

6. Press 2

7. Press $01B0[Enter] Receive=!01

8. Press 2

9. Press #010[Enter] Receive=>????????

10. Press 2

11. Press #011[Enter] Receive=>????????

- In step **3**: the status of CB-7080 is COUNTER mode.
- In step **5**: Change to frequency mode.
- In step **7**: Select non-isolated input.
- In step **9**: Frequency measurement of channel-0.
- In step **11**: Frequency measurement of channel-1.

Note

The command $01B1 that is referenced in step 7 can be used to select the isolated input. The commands $01B2 and $01B3 are used for the other selections.

Counter input measurement

Perform the following procedure to measure the counter input. Refer to Figure 1-5 for the wire connection.

1. Power on and run the **test.exe** application.
2. Press **2**
3. Press **$012**[Enter] Receive=!01500600
4. Press **2**
5. Press **$01B0**[Enter] Receive=!01
6. Press **2**
7. Press **#010**[Enter] Receive=>????????
8. Press **2**
9. Press **#011**[Enter] Receive=>????????

- In step 3: The status of CB-7080 is COUNTER mode
- In step 5: Select non-isolated input
- In step 7: Counter measurement of channel-0
- In step 9: Counter measurement of channel-1

Note:
The command $01B101B1 that is referenced in step 7 can be used to select the isolated input. The commands $01B2 and $01B3 are used for the other selections.

Default settings

CB-7080 and CB-7080D default settings are listed in Table 1-2.

Table 1-2. Default Settings

Address	01
Baud rate	9600
Checksum	disabled
Data bits	1 start + 8 data + 1 stop (no parity)
Type	50 (counter input)
Alarm	▪ CB-7080: High alarm on counter 0 and counter 1 ▪ CB-7080D Hi/hi-high alarm on counter 0

Application notes

Counter/Frequency input mode selection

Select the counter/frequency input from an isolated or non-isolated signal. Channel 0 and channel 1 can be selected separately. Four different input modes are listed in the following table. These four input modes can be used in both the CB-7080 and the CB-7080D.

Table 1-3. Counter frequency input

Input Mode	Command	Channel 0	Channel 1
Input mode 0	$AAB0	Non-isolated	Non-isolated
Input mode 1	$AAB1	Isolated	Isolated
Input mode 2	$AAB2	Non-isolated	Isolated
Input mode 3	$AAB3	Isolated	Non-isolated

Counter alarm mode selection

There are no alarm functions in frequency mode (51). There are two counter alarm modes, **Alarm Mode 0** and **Alarm Mode 1**. These two alarm modes can be used in both CB-7080 and CB-7080D.

Alarm Mode 0 is used for two-channel applications, as follows:

- To select alarm mode 0: ~AAA0 (for both channels)
- To enable channel 0: @AAEA0
- To disable channel 0: @AADA0

Set the high alarm limit of channel 0: @AAPA(data)

- if (counter 0 >= alarm limit 0) → D/O 0 turn ON
- if (counter 0 < alarm limit 0) → D/O 0 turn OFF
- Enable channel 1: @AAEA1
- Disable channel 1: @AADA1

Set the high alarm limit of channel 1: @AASA(data)

- if (counter 1 >= alarm limit 1) → D/O 1 turn ON
- if (counter 1 < alarm limit 1) → D/O 1 turn OFF

Alarm mode 1 is used for single-channel applications, as follows:

- Select alarm mode 1: ~AAA1 (for channel 0 only)
- Enable channel 0: @AAEAT
- Disable channel 0: @AADA
- Clear latch alarm: @AACA
- Set high alarm limit: @AAPA(data)
- Set high-high alarm limit: @AASA(data)

Counter 0 < high alarm	OFF	OFF
high alarm <= counter 0, and counter 0 < high-high alarm	ON	OFF
High-high alarm <= counter 0	ON	ON

Note:
The high-high alarm must greater than the high-alarm.

Digital output application notes

The D/O0 and D/O1 can be used as D/O or an alarm output, as follows:

- D/O in the frequency mode.
- D/O in the counter mode and alarm disabled (by @AADA or @AADAN command).
- An alarm output in the counter mode and an alarm enable (by @AAEAT or @AAEAN command).

Table 1-4 Digital Alarm Configuration

	D/O 0	**D/O 1**
Frequency mode	D/O 0	D/O 1
Counter mode and alarm disabled	D/O 0	D/O 1
Counter mode and alarm enable (alarm mode 1, ~AAA1)	High-alarm on counter 0	High-high alarm on counter 0
Counter mode and alarm enable (alarm mode 0, ~AAA0 and @AAEA0)	Alarm on counter 0	D/O 1 or alarm on counter 1
Counter mode and alarm enable (alarm mode 0, ~AAA0 and @AAEA1)	D/O 0 or alarm on counter 0	alarm on counter 1

Programmable threshold voltage setting

The programmable threshold voltage is valid for a non-isolated input of counter mode (50) and frequency mode (51). The default settings are:

- TTL compatible
- Low trigger level = 0.8 volt
- High trigger level = 2.4 volts

You can change the high trigger level by the $AA1H (data) command. The low trigger can be changed by the $AA1L (data) command. The high trigger level must be greater than the low trigger level.

Digital filter setting

The digital filter is disabled in frequency mode (51). It is designed as a pulse-width filter for both high/low pulses, and is valid for both non-isolated and isolated inputs. The digital filter can be enabled or disabled. Key points of digital filter usage include:

1. Use $AABS to select the input signal.
2. Use $AA0H(data) to set the minimum width of high level.
3. Use $AA0L(data) to set the minimum width of low level.
4. Use $AA4S to enable/disable the digital filter (both channels).

If the high width of the input signal is smaller than the minimum high width of digital filter, the input signal is filtered out. Also, the low width of the input signal must be greater than the minimum low width of digital filter.

For example, if the width of the input signal is >1000 μs, set the digital filter at 900 μs. Therefore, all noise <900 μs is filtered out by the digital filter. These steps are given as follows:

- $AAB0
- $AA0H00900
- $AA0L00900
- $AA41

Gate control setting

The gate control is ignored in frequency mode (51). The gate control is disabled in counter mode (50) by default. To use a command to enable/disable the gate control:

- To use the $AAA0 command, the gate input must be low to enable the counter.
- To use the $AAA1 command, the gate input must be high to enable the counter.
- To use the $AAA2 command, the gate input is ignored. The counter is always enabled.

Preset value setting

The preset value is ignored in frequency mode (51). The counters go to their preset value in the first power-on state. The reset counter command, $AA6N, also forces the counters to go to their preset value. The default preset value is 0. Use the $AAPN(data) command to change the preset value. The key points include:

Table 1-5: Preset values

Factory default setting	Counter preset value is 0
Power on state	Counter 0/1 goes to preset value
$AA6N	Counter N goes to preset value
$AAPN(data)	Set preset value of counter N

Frequency input applications

The frequency mode type is indicated by entering **51**in the command.

Table 1-6. Frequency mode settings

Command	**Result - Frequency 0**	**Result - Frequency 1**
$AAB0 to set input mode 0 $AA1H(data) and $AA1L(data)	Non-isolated channel 0 and threshold value active	Non-isolated channel 1 and threshold value active
$AAB1 to set input mode 1 $AA1H(data) & $AA1L(data)	Isolated channel 0	Isolated channel 1
$AAB2 to set input mode 2 $AA1H(data) & $AA1L(data)	Non-isolated channel 0 and threshold value active	Isolated channel 1
$AAB3 to set input mode 3 $AA1H(data) & $AA1L(data)	Isolated channel 0	Non-isolated channel 1 and threshold value active

To measure frequency:

1. Use the $AA1H(data) and $AA1L(data) commands to set the threshold value, if the frequency is in a non-isolated input.
2. Use the $AAB? command to select the mode (this command will clear the current frequency first).
3. Use the #AA? command to perform frequency measurement.

Only four commands are important in frequency measurement mode:

$AAB?	Select mode
$AA1H(data)	Set high-level threshold value
$AA1L(data)	Set low-level threshold value
#AA?	Perform frequency measurement

The status-read-back commands are:

$AAB?	Mode read back
$AA1H	High_level threshold value read back
$AA1L(data)	Low_level threshold value read back

Counter input applications

The counter mode type is indicated by entering **50** in the command.

Table 1-7. Counter mode settings

	Counter 0	**Counter 1**
$AAB0 to set input mode 0 $AA1H(data) and $AA1L(data)	Non-isolated channel 0 and threshold value active	Non-isolated channel 1 and threshold value active
$AAB1 to set input mode 1 $AA1H(data) and $AA1L(data)	Isolated channel 0	Isolated channel 1
$AAB2 to set input mode 2 $AA1H(data) and $AA1L(data)	Non-isolated channel 0 and threshold value active	Isolated channel 1
$AAB3 to set input mode 3 $AA1H(data) and $AA1L(data)	Isolated channel 0	Non-isolated channel 1 and threshold value active

Note:
The threshold value commands $AA1H(data) and $AA1L(data) are effective to non-isolated inputs only.

Configuration Code

Use the configuration codes listed in the following tables to set the baud rate, check the status of, and set the input type of a module.

Table 1-8. Baud Rate Configuration Code: CC

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

Table 1-9. Status Configuration Code: FF, 2-char (for all)

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	checksum 0=disabled 1=enabled	0	0	0	frequency gate time 0: 0.1 second 1: 1.0 second	0	0

Table 1-10. Input Type Configuration Code: TT

TT	**Input Range**
50	Counter
51	Frequency

Command Set

Overview

The commands in this section are grouped by function, and summarized as listed below.

Table	Function	Description
Table 2-1	General Command s	▪ Read a module's name, status, firmware number and init pin ▪ Read/write a module's configuration ▪ Read or enable the watchdog timer
Table 2-2	Frequency Command s	▪ Set the frequency input of a module
Table 2-3	General Counter Commands	▪ Set the counter input of a module
Table 2-4	Alarm Mode 0 Commands	▪ Used in two-channel applications to set a different alarm mode for each channel.
Table 2-5	Alarm Mode 1 Commands	▪ Used in two-channel application to set the same alarm mode for each channel.
Table 2-6	LED Commands	▪ Get or set LED configuration ▪ To send data to the LED

General syntax format

Commands are constructed with individual pieces of information represented as code. Each command is structured as follows:

- One-character *Delimiter* that indicates the type of command to execute.
- Two-character HEX module *address*, from 00 to FF
- *Function*, which indicates the task to execute.
- *Terminator* character, to indicate the end of the command.

This section lists the commands to use to perform specific functions. When writing a command, replace the generic address with the address of the module you want to perform the task on. For example, to *read the name of module 1*:

Select **$AAM**, the command used to *read the name of a module* (see page 35).

1. Replace the address (AA) with 01, to indicate module 1.
2. Terminate the command with a carriage return (cr).

This command would then be written as **$01M(cr)**.

Delimiter characters

The characters %, #, ~, %, $, @ are used as the leading character in a command. These characters indicate the type of command to execute.

Delimiter Character	Usage
%	Set the configuration of a module.
#	
~	Reset the data.
$	Get/set (read/write) module information.
[ ]	Characters inside [ ] indicate an optional task, such as [chk]
(cr)	Indicates the end of the command.
!	Always the leading character in the response to a command.

General Commands

Table 2-1. General Command Set

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set the module configuration	Page 2-4
#AAN	>(data)	Read the counter or frequency	Page 2-1
~**	No Response	Host OK	Page 2-2
~AA0	!AASS	Read the module status	Page 2-3
~AA1	!AA	Reset the module status	Page 2-4
~AA2	!AATT	Read the Host Watchdog Timer	Page 2-5
~AA3ETT	!AA	Enable the Host Watchdog Timer	Page 2-6
~AAO(name)	!AA	Set the module name	Page 2-8
$AA2	!AATTCCFF	Read the module configuration	Page 2-17
$AAF	!AA(data)	Read the firmware number	Page 2-33
$AAI	!AAS	Read the value of INIT* pin	Page 2-34
$AAM	!AA(data)	Read the module name	Page 2-35

Frequency commands

Table 2-2. Frequency Command Set

Command	Response	Description	Reference
$AAB	!AAS	Read the input mode	Page 2-31
$AABS	!AA	Set the input mode	Page 2-32
$AA1H	!AA(data)	Read the high trigger level	Page 2-13
$AA1H(data)	!AA	Set the high trigger level	Page 2-14
$AA1L	!AA(data)	Read the low trigger level	Page 2-15
$AA1L(data)	!AA	Set the low trigger level	Page 2-16

General counter commands

Table 2-3. General Counter Command Set

Command	Response	Description	Reference
~AAAS	!AA	Set the counter alarm mode	Page 2-7
$AA0H	!AA(data)	Read the minimum width of High	Page 2-9
$AA0H(data)	!AA	Set the minimum width of High	Page 2-10
$AA0L	!AA(data)	Read the minimum width of High	Page 2-11
$AA0L(data)	!AA	Set the minimum width of High	Page 2-16
$AA1H	!AA(data)	Read the high trigger level	Page 2-13
$AA1H(data)	!AA	Set the high trigger level	Page 2-14

Command	Response	Description	Reference
$AA1L	!AA(data)	Read the low trigger level	Page 2-15
$AA1L(data)	!AA	Set the low trigger level	Page 2-16
$AA3N	!AA(data)	Read the max. counter value	Page 2-18
$AA3N(data)	!AA	Set the max. counter value	Page 2-19
$AA4	!AAS	Read the filter status	Page 2-20
$AA4S	!AA	Set the filter status	Page 2-21
$AA5N	!AAS	Read the counter status	Page 2-22
$AA5NS	!AA	Set the counter status	Page 2-23
$AA6N	!AA	Reset the counter	Page 2-24
$AA7N	!AAS	Read the overflow status	Page 2-25
$AAG	!AAG	Read the gate mode	Page 2-29
$AAAG	!AA	Set the gate mode	Page 2-30
$AAB	!AAS	Read the input mode	Page 2-31
$AABS	!AA	Set the input mode	Page 2-32
@AADI	!AAS0D00	Read the D/O and alarm state	Page 2-36
@AADO0D	!AA	Set the D/O value	Page 2-38
@AAGN	!AA(data)	Read the preset value	Page 2-44
@AAPN(data)	!AA	Set the preset value	Page 2-45

Alarm mode 0 commands

Table 2-4. Alarm-mode 0 Command Set

Command	Response	Description	Reference
@AAEAN	!AA	Enable the alarm	Page 2-39
@AADAN	!AA	Disable the alarm	Page 2-43
@AAPA(data)	!AA	Set the counter 0 alarm value	Page 2-46
@AASA(data)	!AA	Set the counter 1 alarm value	Page 2-48
@AARP	!AA	Read the counter 0 alarm value	Page 2-50
@AARA	!AA	Read the counter 0 alarm value	Page 2-52

Alarm mode 1 commands

Table 2-5. Alarm-mode 1 Command Set

Command	Response	Description	Reference
@AAEAT	!AA	Enable the alarm	Page 2-40
@AACA	!AA	Clear the latch alarm	Page 2-41
@AADA	!AA	Disable the alarm	Page 2-42
@AAPA(data)	!AA	Set the Hi-alarm value	Page 2-46
@AASA(data)	!AA	Set the Hi-Hi-alarm value	Page 2-48
@AARP	!AA	Read the Hi-alarm value	Page 2-51
@AARA	!AA	Read the Hi-Hi-alarm value	Page 2-53

LED commands

Table 2-6. LED Command Set

Command	Response	Description	Reference
$AA8	!AAS	Read LED configuration	Page 2-26
$AA8V	!AA	Set LED configuration	Page 2-27
$AA9(data)	!AA	Send data to LED	Page 2-28

%AANNTTCCFF

Description: Set the configuration of a module.

Syntax: %AANNTTCCFF[chk](cr)

%	A delimiter character.
AA	Current 2-character HEX module address, from 00 to FF
NN	New 2-character HEX module address, from 00 to FF
TT	Input type code, refer to Table 1-10.
CC	Baud rate code, refer to Table 1-8.
FF	Status code, refer to Table 1-9.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response:	!AA[chk](cr)
Invalid response:	?AA[chk](cr)
No response	Syntax error, communication error or address error.
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The address of module 01 is configured to a new address 02, counter mode:

command: %0102500600(cr)

response: !02(cr)

Change to frequency mode:

command: %0202510600(cr)

response: !02(cr)

#AAN

Description: Read the counter or frequency value.

Syntax: #AAN[chk](cr)

#	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	Channel-0 of counter or frequency
N=1	Channel-1 of counter or frequency
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	[chk](data)(cr)
invalid response	No Response
no response	Syntax error, communication error or address error.
>	A delimiter character indicating a valid response
(data)	8-character data (in HEX format)
[chk]	2-character checksum. If checksum is disabled, no [chk]

(cr)=0x0D

Examples:

Counter − 0=0x1E=30 (in decimal):

command:	$012(cr)
response:	!01500600(cr)
command:	#010(cr)
response:	>0000001E(cr)

Frequency − 1=0x1E Hz = 30 Hz (in decimal):

command:	$022(cr)
response:	!02510600(cr)
command:	#021(cr)
response:	>0000001E(cr)

~**

Description: The host sends this command to tell all modules “Host is OK”.

Syntax: ~**[chk](cr)

~	A delimiter character
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

No response

Examples:

command	~**(cr)
response	No Response

~AA0

Description: Reads the module status. The module status will be latched until ~AA1 command is sent. If the host watchdog is enabled and the host is down, (no ~** command received), the module status will be set to 4. If the module status=4, all output commands are ignored.

Syntax: ~AA0[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AASS[chk](cr)
Invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
SS	2-character HEX status value:
	Bit_0, Bit_1 = reserved
	Bit_2 = 0: OK,
	1: host watchdog time-out
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Status of module 01 is OK:

Command:	~010(cr)
Response:	0100(cr)

Module status=04; host watchdog timed-out; HOST is down now:

Command:	~020(cr)
Response:	0204(cr)

~AA1

Description: Resets the module status. The module status will be latched until ~AA1 command is sent. If the module status=4, all output commands will be ignored. Read the module status first to verify that the module status is 0. If the module status is not 0, only a ~AA1 command can clear the module status.

Syntax: ~AA1[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
invalid response	?AA[chk](cr)
No response:	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Module status=0x04 → host is down:

command:	~010(cr)
response:	!0104(cr)

Output command is ignored:

command:	@01DO00(cr)
response:	!(cr)

Clear module status:

command:	~011(cr)
response:	!01(cr)

Module status=0x00:

command:	~010(cr)
response:	!0100(cr)

Output command is OK:

command:	@01DO00(cr)
response:	>(cr)

~AA2

Description: Reads the status and timer value of host watchdog. When the host watchdog is enabled, the host must send ~** command to all modules before the timer times-out. When the ~** command is received, the host watchdog timer is reset and starts counting down. Use ~AA3ETT to enable/disable/setting the host watchdog timer.

Syntax: ~AA2[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	AASTT[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	Host watchdog is disabled
S=1	Host watchdog is enabled
TT	2-character HEX value, from 00 to FF, unit=0.1 second
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Host watchdog timer of module 01 is disabled

command:	~012(cr)
response:	!01000(cr)

Host watchdog timer of module 02 is enabled and time-out
time = 0.1 x 10 = 1 second

command:	~022(cr)
response:	!0210A(cr)

~AA3ETT

Description: Enable/disable the timer value of host watchdog. The host watchdog timer is a software host watchdog. When the software host watchdog is enabled, the host must send ~** command to all modules before the timer times-out. When the ~** command is received, the host watchdog timer is reset and restarted. Use ~AA2 to read the host watchdog status and value.

Syntax: ~AA3ETT[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
E	0 is disabled and 1 is enabled
TT	2-character HEX value, from 00 to FF, unit=0.1 second
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Examples:

Disable host watchdog timer of module 01

command	~013000(cr)
response	!01(cr)

Host watchdog timer of module 02 is enabled and time-out time = 0.1 x 10 = 1 second.

command	~02310A(cr)
response	!02(cr)

~AAAS

Description: Sets counter alarm mode. Refer to "Counter alarm mode selection" on page 1-7 for more information.

Syntax: ~AAAS[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
S=0	alarm mode 0.
S=1	alarm mode 1.
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Set alarm mode=0.

command:	~01A0(cr)
response:	!01(cr)

Set alarm mode=1.

command:	~02A1(cr)
response:	!02(cr)

~AAO(name)

Description: Sets the module name.

Syntax: ~AAO(name)[chk](cr)

~	A delimiter character
AA	2-character HEX module address, from 00 to FF
(name)	4-character/5-character module name
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Change module name from 7080 to 8080

command:	$01M(cr)
response:	!017080(cr)
command:	~01O8080(cr)
response:	!01(cr)

Change module name from 7080D to 8080D

command:	$01M(cr)
response:	!017080D(cr)
command:	~01O8080D(cr)
response:	!01(cr)

Note
This command was designed for OEM/ODM users. You can use it to change the module name for other purposes.

$AA0H

Description: Reads the minimum input signal width at high level. Refer to "Digital filter setting" for more information.

Syntax: $AA0H[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	5-character decimal value for min. width at high level. The unit is microseconds (μs) and the range can be from 2 μs to 65535 μs.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Min. width = 10 μs

command	$010H(cr)
response	!0100010(cr)

Min. width = 1000 μs = 1 ms

command	$020H(cr)
response	!0201000(cr)

$AA0H(data)

Description: Sets the minimum input signal width at high level. Refer to "Digital filter setting" for more information.

Syntax: $AA0H(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
(data)	5-character decimal value for minimum width at high level. The unit is microseconds (μs) and the range can be from 2 μs to 65535 μs.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Min. width = 10 μs

command:	$010H00010(cr)
response:	!01(cr)

Min. width = 1000 μs = 1 ms

command:	$020H01000(cr)
response:	!02(cr)

$AA0L

Description: Read the minimum input signal width at low level. Refer to "Digital filter setting" for more information.

Syntax: $AA0L[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	5-character decimal value for min. width at low level. The unit is microseconds and the range can be from 2 μs to 65535 μs.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Min. width=20 μs

command:	$010H(cr)
response:	!0100020(cr)

Min. width=2000 μs = 2 ms

command:	$020H(cr)
response:	!0202000(cr)

$AA0L(data)

Description: Sets the minimum input signal width at low level. Refer to "Digital filter setting" for more information.

Syntax: $AA0H(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
(data)	5-character decimal value for min. width at low level. The unit is microseconds (μs), and the range can be from 2 μs to 65535 μs.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Min. width = 20 μs

command:	$010H00020(cr)
response:	!01(cr)

Min. width = 2000 μs = 2 ms

command:	$020H02000(cr)
response:	!02(cr)

$AA1H

Description: Reads the high trigger level of non-isolated input. Refer to "Programmable threshold voltage setting" on page 1-8 for more information.

Syntax: $AA1H[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	2-character decimal value for high trigger level. The unit is 0.1 volt and the range can be from 0.0 to 5.0 volts.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

High trigger level=2.4 volts:

command:$011H(cr)

response: !0124(cr)

High trigger level=3.0 volts:

command: $021H(cr)

response:!0230(cr)

$AA1H(data)

Description: Sets the high trigger level of non-isolated inputs. Refer to "Programmable threshold voltage setting" for more information.

Syntax: $AA1H(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
(data)	2-character decimal value for high trigger level. The unit is 0.1 volt, and the range can be from 0.0 to 5.0V.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum,
if checksum is disabled	no [chk]
(cr)	0x0D

Examples:

High trigger level=2.4 volts

command:	$011H24(cr)
response:	!01(cr)

High trigger level=3.0 volts

command:	$021H30(cr)
response:	!02(cr)

Note:
The default value for the high trigger level is 2.4V

$AA1L

Description: Read the Low trigger level of non-isolated input. Refer to "Programmable threshold voltage setting" for more information.

Syntax: $AA1L[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum
if checksum is disabled	no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	2-character decimal value for low trigger level. The unit is 0.1V, and the range can be from 0.0 to 5.0V.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Low trigger level=0.8 volts:

command:	$011L(cr)
response:	!0108(cr)

Low trigger level=1.0 volts:

command:	$021L(cr)
response:	!0210(cr)

$AA1L(data)

Description: Set the low trigger level of non-isolated input. Refer to "Programmable threshold voltage setting" for more information.

Syntax: $AA1L(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
(data)	2-character decimal value for low trigger level. The unit is 0.1V and the range can be from 0.0 to 5.0V
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Low trigger level=0.8 volt:

command:	$011L08(cr)
response:	!01(cr)

Low trigger level=1.0 volt:

command:	$021L10(cr)
response:	!02(cr)

Note:
The default value for the low trigger level is 0.8V.

$AA2

Description: Read the configuration of module.

Syntax: $AA2[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response:	!AATTCCFF[chk](cr),
Invalid response:	?AA[chk](cr)
no response:	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
TT	Refer to Table 1-10.
CC	Refer to Table 1-8.
FF	Refer to Table 1-9.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Address=01, counter, 9600 BPS, checksum disabled:

command:	$012(cr)
response:	!01500600(cr)

Address=02, frequency, 19200 BPS, checksum disabled:

command:	$022(cr)
response:	!02510700(cr)

Note

If you use the %AANNTTCCFF command to change module configuration, the new configuration code is stored into EEPROM immediately.

The configuration code includes the module address, module type, baud rate code, checksum enable/disable code, calibration code, power-on value and safe value.

The EEPROM data of a CB-7080 can be read an infinite number of times, and can be written about 100,000 times maximum. Therefore, you should not frequently change the configuration code for testing.

The $AA2 command is used to read EEPROM data only. You can send this command to CB-7080 module without limit.

$AA3N

Description: Read the maximum counter value.

Syntax: $AA3N[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	channel-0 of counter or frequency
N=1	channel-1 of counter or frequency
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk].
(cr)	0x0D

Examples:

Counter-0 from preset value to FFFFh

command:	$0130(cr)
response:	!010000FFFF(cr)

Counter-1 from preset value to FFFFFFFFh

command:	$0131(cr)
response:	!01FFFFFFFF(cr)

$AA3N(data)

Description: Set the maximum counter value.

Syntax: $AA3N(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	channel-0 of counter or frequency
N=1	channel-1 of counter or frequency
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Counter-0 from preset value to FFFFh:

command:	$01300000FFFF(cr)
response:	!01(cr)

Counter-1 from preset value to FFFFFFFFh:

command:	$0131FFFFFFFF(cr)
response:	!01(cr)

$AA4

Description: Read the status of digital filter. Refer to "Digital filter setting" for more information.

Syntax: $AA4[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	digital filter is disabled
S= 1	digital filter is enable
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Digital filter is disabled:

command:	$014(cr)
response:	!010(cr)

Digital filter is enabled:

command:	$024(cr)
response:	!021(cr)

$AA4S

Description: Set the filter status. Refer to "Digital filter setting" for more information.

Syntax: $AA4S[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
S=0	digital filter is disabled
S=1	digital filter is enabled
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Disable the digital filter:

command:	$0140(cr)
response	!01(cr)

Enable the digital filter:

command	$0241(cr)
response	!02(cr)

$AA5N

Description: Read the counter status

Syntax: $AA5N[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	counter 0
N=1	counter 1
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	counter is stopped (is disabled)
S= 1	counter is started (enabled)
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Stop Counter 0:

command	$0150(cr)
response	!010(cr)

Start Counter 1:

command	$0151(cr)
response	!011(cr)

$AA5NS

Description: Set the counter status

Syntax: $AA5NS[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	counter 0
N=1	counter 1
S=0	stop counter
S=1	start counter
[chk	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Stop counter 0:

command:	$01500(cr)
response:	!01(cr)

Start counter 1:

command:	$01511(cr)
response:	!01(cr)

$AA6N

Description: Reset counter 0 or counter 1 to the preset value and clear the overflow flag. Refer to "Preset value setting" for more information.

Syntax: $AA6N[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	Counter 0
N=1	Counter 1
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]

(cr)=0x0D

Examples:

Reads preset value of Counter 0.

command:	@01G0(cr)
response:	!0100000000(cr)

Preset value=0xABCD:

command:	$0160(cr)
response:	!01(cr)

Reset counter 1 to preset value 0xABCD:

command	@01G1(cr)
response	010000ABCD(cr)
command:	$0161(cr)
response:	!01(cr)

$AA7N

Description: Read the overflow flag of counter. You can use the $AA6S command to reset the counter and clear the overflow flag.

Syntax: $AA7N[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	Counter 0
N=1	Counter 1
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	no overflow
S=1	overflowed
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Counter 0 has overflowed. Clear the overflow flag:

command:	$0170(cr)
response:	!011(cr)
command:	$0160(cr)
response:	!01(cr)

Counter 1 is OK:

command:	$0171(cr)
response:	!010(cr)

$AA8

Description: Read the LED configuration.

Syntax: $AA8[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS[chk](cr)
Invalid response	?AA[chk](cr)
No response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	Show counter/frequency channel 0
S=1	Show counter/frequency channel 1
S=2	HOST control
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

LED shows the value of channel 0:

command:	$018(cr)
response:	!010(cr)

LED shows the value of channel 1:

command:	$028(cr)
response:	!021(cr)

HOST controls the LED display:

command:	$038(cr)
response:	!032 (cr)

$AA8V

Description: Select LED Configuration.

Syntax: AA8V[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
V=0	LED shows counter/frequency channel 0
V=1	LED shows counter/frequency channel 1
V=2	HOST control LED
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

LED shows channel 1:

command:	$0181(cr)
response:	!01(cr)

HOST will control LED:

command:	$0282(cr)
response:	!02(cr)
command:	$029040.00(cr)
response:	!02(cr)

$AA9(data)

Description: Send data to the LED display.

Syntax: $AA9(data)[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
9	The function to execute.
(data)	5 decimal digit + 1 decimal point
	Maximum = 99999.
	Minimum = 0.0000
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Show max. = 99999:

command:	$01999999.(cr)
response:	!01(cr)

Show min. = 0.0000:

command:	$0290.0000(cr):
response:	!02(cr)

Show display = 12.345:

command:	$03912.345(cr)
response:	!03(cr)

$AAG

Description: Read gate control mode. Refer to "Gate control setting" for more information.

Syntax: $AAG[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
G=0	Gate is low active
G=1	Gate is high active
G=2	Gate is disabled
[chk]	character checksum, if checksum is disabled, no [chk]
(cr)	x0D

Response:

Valid response	!AAG[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
G=0	Gate is low active
G=1	Gate is high active
G=2	Gate is disabled
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Module 1 gate is active low:

command:	$01A(cr)
response:	!010(cr)

Module 2 gate is active high:

command:	$02A(cr)
response:	!021(cr)

Module 3 gate is disabled (always active):

command:	$03A(cr)
response:	!032 (cr)

$AAAG

Description: Set gate control mode. Refer to "Gate control setting" for more information.

Syntax: $AAAG[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
G=0	Gate is low active
G=1	Gate is high active
G=2	Gate is disabled
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
G=0	Gate is low active
G=1	Gate is high active
G=2	Gate is disabled
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Module 1 gate is active low:

command:	$01A0(cr)
response:	!01(cr)

Module 2 gate is active high:

command:	$02A1(cr)
response:	!02(cr)

Module 3 gate is disabled (always active):

command:	$03A2(cr)
response:	!03(cr)

$AAB

Description: Read input mode. Refer to "Counter/Frequency input mode selection" on page 1-6 for more information.

Syntax: $AAB[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
B	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAB[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	channel 0 is non-isolated, channel 1 is non-isolated.
S=1	channel 0 is isolated, channel 1 is isolated.
S=2	channel 0 is non-isolated, channel 1 is isolated.
S=3	channel 0 is isolated, channel 1 is non-isolated.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Counter/frequency channel 0 is non-isolated, channel 1 is non-isolated:

command:	$01B(cr)
response:	!010(cr)

Counter/frequency channel 0 is isolated, channel 1 is isolated:

command:	$02B(cr)
response:	!021(cr)

Counter/frequency channel 0 is non-isolated, channel 1 is isolated:

command:	$03B(cr)
response:	!032(cr)

$AABS

Description: Set input mode. Refer to "Counter/Frequency input mode selection".

Syntax: $AABS[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
B	
S=0	channel 0 is non-isolated, channel 1 is non-isolated.
S=1	channel 0 is isolated, channel 1 is isolated.
S=2	channel 0 is non-isolated, channel 1 is isolated.
S=3	channel 0 is isolated, channel 1 is non-isolated.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Counter/frequency channel 0 is non-isolated, channel 1 is non-isolated:

command:	$01B0(cr)
response:	!01(cr)

Counter/frequency channel 0 is isolated, channel 1 is isolated:

command:	$02B1(cr)
response:	!02(cr)

Counter/frequency channel 0 is non-isolated, channel 1 is isolated:

command:	$03B2(cr)
response:	!03(cr)

$AAF

Description: Read the version number of firmware.

Syntax: $AAF[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
F	
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
data	5-character for version number
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Version A2.0:

command:	$01F(cr)
response:	!01A2.0(cr)

Version A3.0:

command:	$02F(cr)
response:	!02A3.0(cr)

$AAI

Description: Read the value of *INIT pin.

Syntax: $AAI[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
I	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
S=0	INIT* pin is connected to GND pin
S=1	INIT* pin is open
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

INIT* pin is connected to GND pin:

command:	$01I(cr)
response:	!010(cr)

INIT* pin is open:

command:	$02I(cr)
response:	!021(cr)

Note
Refer to "INIT* pin operation principle" for more information on INIT*.

$AAM

Description: Read the module name.

Syntax: $AAM[chk](cr)

$	A delimiter character
AA	2-character HEX module address, from 00 to FF
M	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
data	4-character for module name
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Module name of 01 is 7080:

command:	$01M(cr)
response:	!017080(cr)

Module name of 02 is 7080D:

command:	$02M(cr)
response:	!027080D(cr)

@AADI

Description: Read the status of the D/O and the alarm. Refer to "Counter alarm mode selection", "Digital output application notes", and "D/O operation principles" for more information.

Syntax: @AADI[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
D=0	D/O0=D/O1=OFF
D=1	D/O0=ON, D/O1=OFF
D=2	D/O0=OFF, D/O1=ON
D=3	D/O0=D/O1=ON
I	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AAS0D00[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
D=0	D/O0=D/O1=OFF
D=1	D/O0=ON, D/O1=OFF
D=2	D/O0=OFF, D/O1=ON
D=3	D/O0=D/O1=ON

Alarm mode 0:

S=0	counter 0 alarm=disabled, counter 1 alarm=disabled
S=1	counter 0 alarm=enabled, counter 1 alarm=disabled
S=2	counter 0 alarm=disabled, counter 1 alarm=enabled
S=3	counter 0 alarm=enabled, counter 1 alarm=enabled

Alarm mode 1:

S=0	counter 0 alarm=disabled
S=1	counter 0 alarm=enabled and MOMENTARY mode
S=2	counter 0 alarm=enabled and LATCH mode
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Alarm disabled. D/O0=D/O1=OFF:

command: @01DI(cr)

response: !0100000(cr)

Alarm enabled. D/O0=ON. D/O1=OFF:

command: @02DI(cr)

response: !0230100(cr)

@AADO0D

Description: Set digital output.

Syntax: @AADO0D[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
DO0	
D=0	D/O0=D/O1=OFF
D=1	D/O0=ON, D/O1=OFF
D=2	D/O0=OFF, D/O1=ON
D=3	D/O0=D/O1=ON
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
alarm is enabled	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Examples:

Turn all D/O OFF:

command:	@01DO00(cr)
response:	!01(cr)

Turn D/O ON; Turn D/O 1 OFF:

command:	@02DO01(cr)
response:	!02(cr)

Note:

If the alarm is enabled, D/O 0 and D/O 1 will always be controlled by the module. Therefore the following D/O conditions will exist:

- Power-on value is changed to hi/lo condition immediately.

- The @AADO0D command is ignored.

@AAEAN

Description: Enable the counter alarm (for alarm-mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AAEAN[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
EA	
N=0	Enable counter 0
N=1	Enable counter 1
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Enable counter 0:

command:	@01EA0(cr)
response:	!01(cr)

Enable counter 1:

command:	@01EA1(cr)
response:	!02(cr)

@AAEAT

Description: Enable the counter alarm (for alarm-mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AAEAT[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
T=M	momentary alarm, T=L → latch alarm
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Latch alarm for module 1:

command:	@01EAL(cr)
response:	!01(cr)

Momentary alarm for module 2:

command:	@02EAM(cr)
response:	!02(cr)

Note:
If the alarm is enabled, D/O 0 and D/O 1 will always be controlled by the module. Therefore the following D/O conditions will exist:

- Power-on value is changed to hi/lo condition immediately.

- The @AADO0D command is ignored.

@AACA

Description: Clear latch alarm (for alarm-mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AACA[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
CA	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Clear the latch alarm for module 1:

command:	@01CA(cr)
response:	!01(cr)

Clear the latch alarm for module 2:

command:	@02CA(cr)
response:	!02(cr)

@AADA

Description: Disable alarm (for alarm-mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AADA[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
DA	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Disable the alarm for module 1:

command:	@01DA(cr)
response:	!01(cr)

Disable the alarm for module 2:

command:	@02DA(cr)
response:	!02(cr)

@AADAN

Description: Disable alarm (for alarm-mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AADAN[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
N=0	disable counter 0
N= 1	disable counter 1
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

Disable the alarm for counter 0:

command:	@01DA0(cr)
response:	!01(cr)

Disable the alarm for counter 1:

command:	@02DA1(cr)
response:	!02(cr)

@AAGN

Description: Read the preset value of counter. The $AA6 command can reset counter to the preset value. Refer to "Preset value setting" for more information.

Syntax: @AAGN[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
G	
N=0	read counter 0
N= 1	read counter 1
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled → no [chk]
(cr)	0x0D

Examples:

The preset value of counter 0 is 0000FFFF:

command:	@01G0(cr)
response:	!010000FFFF(cr)

The preset value of counter 1 is 00000000:

command:	@02G1(cr)
response:	!0200000000(cr)

@AAPN(data)

Description: Set the preset value of counter. The $AA6 command can reset counter to preset value. Refer to "Preset value setting" for more information.

Syntax: @AAPN(data)[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
P	
N=0	Counter 0
N=1	Counter 1
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The preset value of counter 0 is FFFF0000h:

command:	@01P0FFFF0000(cr)
response:	!01(cr)

The preset value of counter 1 is 0000FFFFh:

command:	@02P10000FFFF(cr)
response:	!02(cr)

@AAPA(data)

Description: Set the alarm limit of counter 0 (for alarm mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AAPA(data)[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
PA	
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The alarm limit of counter − 0 is FFFF0000h:

command:	@01PAFFFF0000(cr)
response:	!01(cr)

The alarm limit of counter 0 is 0000FFFFh:

command:	@02PA0000FFFF(cr)
response:	!02(cr)

@AAPA(data)

Description: Set the Hi-alarm limit of counter 0 (for alarm mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AAPA(data)[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF

PA

(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The Hi-alarm limit of counter 0 is FFFF0000h:

command:	@01PAFFFF0000(cr)
response:	!01(cr)

The Hi-alarm limit of counter 0 is 0000FFFFh:

command:	@02PA0000FFFF(cr)
response:	!02(cr)

@AASA(data)

Description: Set alarm limit of counter-1(for alarm-mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AASA(data)[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
SA	
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The alarm limit of counter 1 is FFFF0000h:

command:	@01SAFFFF0000(cr)
response:	!01(cr)

The alarm limit of counter 1 is 0000FFFFh:

command:	@02SA0000FFFF(cr)
response:	!02(cr)

@AASA(data)

Description: Set the Hi-Hi alarm limit of counter 0 (for alarm-mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AASA(data)[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
SA	
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The Hi-Hi alarm limit of counter 0 is FFFF0000h:

command:	@01SAFFFF0000(cr)
response:	!01(cr)

The Hi-Hi alarm limit of counter 0 is 0000FFFFh:

command:	@02SA0000FFFF(cr)
response:	!02(cr)

@AARP

Description: Read the alarm limit of counter 0 (for alarm-mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AARP[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF

RP

[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The alarm limit of counter 0 is FFFF0000h:

command:	@01RP(cr)
response:	!01FFFF0000(cr)

The alarm limit of counter 0 is 0000FFFFh:

command:	@02RP(cr)
response:	!020000FFFF(cr)

@AARP

Description: Read the Hi-alarm limit of counter 0 (for alarm-mode 1). Refer to "Counter alarm mode selection" for more information.

Syntax: @AARP[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
RP	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The Hi-alarm limit of counter 0 is FFFF0000h:

command:	@01RP(cr)
response:	!01FFFF0000(cr)

The Hi-alarm limit of counter 0 is 0000FFFFh:

command:	@02RP(cr)
response:	!020000FFFF(cr)

@AARA

Description: Read the alarm limit of counter-1 (for alarm-mode 0). Refer to "Counter alarm mode selection" for more information.

Syntax: @AARA[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
RA	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA[chk](cr)
Invalid response	?AA[chk](cr)
no response	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The alarm limit of counter 1 is FFFF0000h:

command:	@01RA(cr)
response:	!01FFFF0000(cr)

The alarm limit of counter 1 is 0000FFFFh:

command:	@02RA(cr)
response:	!020000FFFF(cr)

@AARA

Description: Read the Hi-Hi-alarm limit of counter 0 (for alarm-mode 1). Refer to Counter alarm mode selection for more information.

Syntax: @AARA[chk](cr)

@	A delimiter character
AA	2-character HEX module address, from 00 to FF
RA	
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Response:

Valid response	!AA(data)[chk](cr)
invalid response	?AA[chk](cr)
no response:	Syntax error, communication error or address error
!	A delimiter character indicating a valid response
?	A delimiter character indicating an invalid response
AA	2-character HEX module address
(data)	8-character HEX value.
[chk]	2-character checksum. If checksum is disabled, no [chk]
(cr)	0x0D

Examples:

The Hi-Hi-alarm limit of counter 0 is FFFF0000h:

command:	@01RA(cr)
response:	!01FFFF0000(cr)

The Hi-Hi-alarm limit of counter 0 is 0000FFFFh;

command:	@02RA(cr)
response:	!020000FFFF(cr)

Operating principle and application notes

INIT*_pin operation principle

All CB-7000 modules contain an EEPROM to store configuration information. To determine the status of the CB-7080 modules, connect the INIT*_pin to the GND_pin and apply power to the module. The CB-7080 modules will go to the factory default setting without changing the EEPROM data. The factory default setting is as follows:

Address	= 00
baud rate	= 9600
checksum	= DISABLED
data format	= 1 start + 8 data bits + 1 stop bit

To auto-configure the CB-7080/7080D module according to the EEPROM data:

Disconnect the INIT*_pin and GND_pin.

To find the EEPROM configuration data easily in the default setting:

1. Power-down the module, and connect the INIT*_pin to GND_pin.
2. Power-up the module.
3. Send command string $002[0x0D] to the module.

   The module returns EPROM data.
4. Record the EEPROM data.
5. Power-down the module, and disconnect the INIT*_pin from the GND pin.
6. Power-up the module.

D/O operation principles

Refer to Digital output application notes on page 1-7 for more information.

- The D/O outputs of CB-7080 and CB-7080D modules are OFF on the first power-up.
- The D/O outputs are changed to the desired state if the “@AADO0D” command is received. Then, all D/Os will stay in the same states until the next “@AADO0D” command.
- If the host watchdog is active, D/Os do not change, and the module status is set to 04. If the host computer sends “@AADO0D” to those modules now, those modules will ignore this command and return “!” as warning information. The host must use “~AA1” command to clear the module status to 0, then the CB-7080 and CB-7080D module will accept the “@AADO0D” again.
- If a D/O output is configured as an alarm output, the module controls the ON/OFF state automatically. The “@AADO0D” command is ignored in this condition.

EC Declaration of Conformity

We, Measurement Computing Corporation, declare under sole responsibility that the product:

Part Numbers	*Description*
CB-7080	2 channel 32-bit counter/frequency input module
CB-7080D	2 channel 32-bit counter/frequency input module with LED display

to which this declaration relates, meets the essential requirements, is in conformity with, and CE marking has been applied according to the relevant EC Directives listed below using the relevant section of the following EC standards and other normative documents:

- EU EMC Directive 89/336/EEC: Essential requirements relating to electromagnetic compatibility.
- EU 55022 Class B (1995): Radiated and conducted emission requirements for information technology equipment.
- EN 50082-1 (1997): EC generic immunity requirements.
- EN 61000-4-2 (1995): Electrostatic discharge immunity.
- EN 61000-4-3 (1997) ENV 50204 (1996): RF Immunity.
- EN 61000-4-4 (1995): Electric fast transient burst immunity.

We hereby declare that the equipment specified conforms to the above Directives and Standards.

Carl Haapaoja, Vice-President of Design Verification

Measurement Computing Corporation
10 Commerce Way
Suite 1008
Norton, Massachusetts 02766
(508) 946-5100
Fax: (508) 946-9500
E-mail: info@mccdaq.com
www.mccdaq.com