M-3578 for OMR-6080
Counter/Frequency
Input Module
## Contents

1. Introduction ........................................................................................................................................1-1
   1.1. About the OMR Counter/Frequency Modules .................................................................1-1
   1.2. Overview of OMR-6080 ........................................................................................................1-1
       - What is OMR-6080? ........................................................................................................1-1
       - Features of OMR-6080 .....................................................................................................1-2
       - Specifications of OMR-6080 .............................................................................................1-2
       - Pin Definitions of OMR-6080 ...........................................................................................1-4
       - A Look at OMR-6080 & Pin Assignment .......................................................................1-5
       - Functional Block Diagram of OMR-6080 .......................................................................1-6

2. Initialization & Installation ..................................................................................................................2-1
   2.1. Software Installation ..............................................................................................................2-1
   2.2. Initializing a Brand-New Module ..........................................................................................2-2
       - Objective of Initializing a Brand-New OMR .....................................................................2-2
       - Default State ......................................................................................................................2-2
       - Initialization Equipment ....................................................................................................2-3
       - Initialization Procedure ......................................................................................................2-3
       - Initialization Wiring ...........................................................................................................2-3
   2.3. Install a New OMR to a Existing Network .............................................................................2-4
       - Equipments for Install a New Module ..............................................................................2-4
       - Installing Procedures .........................................................................................................2-4
   2.4. Application Wiring for OMR-6080 ......................................................................................2-5
       - Non-isolated Input ...............................................................................................................2-5
       - Photo-isolated Input ...........................................................................................................2-5

3. Command Set ......................................................................................................................................3-1
   3.1. Command and Response .........................................................................................................3-1
       - Introduction ........................................................................................................................3-1
       - Document Conventions ......................................................................................................3-1
       - Format of OMR Commands ...............................................................................................3-2
       - Response of OMR Commands ............................................................................................3-3
   3.2. Summary of Command Set ......................................................................................................3-4
   3.3. Set Configuration .....................................................................................................................3-7
   3.4. Read Configuration ..................................................................................................................3-10
   3.5. Read Module Name ................................................................................................................3-11
   3.6. Read Firmware Version ........................................................................................................3-12
   3.7. Set Input Mode ......................................................................................................................3-13
   3.8. Read Input Mode .....................................................................................................................3-14
   3.9. Read Counter/Frequency Value in HEX Format ...............................................................3-15
   3.10. Read Counter/Frequency Value in DEC Format ...............................................................3-16
   3.11. Set Gate Mode .....................................................................................................................3-17
   3.12. Read Gate Mode ...................................................................................................................3-18
   3.13. Set Maximum Counter Value .............................................................................................3-19
   3.14. Read Maximum Counter Value ............................................................................................3-19
   3.15. Set Initial Count Value .........................................................................................................3-21
   3.16. Read Initial Count Value .....................................................................................................3-22
   3.17. Start/Stop Counter ...............................................................................................................3-23
   3.18. Read Start/Stop Counter Status .........................................................................................3-24
   3.19. Clear Counter .......................................................................................................................3-25
   3.20. Read then Clear Overflow Flag .........................................................................................3-26
   3.21. Enable/Disable Digital Filter ..............................................................................................3-28
   3.22. Read Filter Status ...............................................................................................................3-29
   3.23. Set Minimum Input Signal Width at High Level ...............................................................3-30
   3.24. Read Minimum Input Signal Width at High Level ............................................................3-31
   3.25. Set Minimum Input Signal Width at Low Level ...............................................................3-32
   3.26. Read Minimum Input Signal Width at Low Level ............................................................3-33
3. 27.  Set TTL Input High Trigger Level.....................................................3-34
3. 28.  Read TTL Input High Trigger Level...............................................3-35
3. 29.  Set TTL Input Low Trigger Level....................................................3-36
3. 30.  Read TTL Input Low Trigger Level..................................................3-37
3. 31.  Enable Alarm....................................................................................3-38
3. 32.  Disable Alarm....................................................................................3-39
3. 33.  Set Alarm Limit Value of Counter 0..................................................3-40
3. 34.  Set Alarm Limit Value of Counter 1..................................................3-41
3. 35.  Read Alarm Limit Value of Counter 0...............................................3-42
3. 36.  Read Alarm Limit Value of Counter 1...............................................3-43
3. 37.  Set Digital Output Values.................................................................3-44
3. 38.  Read Digital Output and Alarm Status..............................................3-45
3. 41.  Set Host Watchdog Timer & Safety Value........................................3-50
3. 42.  Read Host Watchdog Timer & Safety Value....................................3-51
3. 43.  Host is OK .......................................................................................3-52
1. Introduction

1.1. About the OMR Counter/Frequency Modules

The OMR provides a counter / frequency input module, which has two 32 bit counter input channels with built in programmable timer for frequency measure function.
OMR-6080: counter/frequency input module with digital output.

1.2. Overview of OMR-6080

What is OMR-6080?

OMR-6080 is a counter / frequency input module. It has two 32-bit counter input channels with built in programmable timer for frequency measurement and supports both photo isolated and non-isolated input mode. The maximum counting value is 4,294,967,295 for counter input channel and the frequency-input range is from 1 Hz to 100 kHz. A programmable digital filter can be enable for both high and low level minimum signal width to reduce noise spike. Besides the programmable threshold for non-isolated input can further reject noise on the input signal level.

The module provides the counter comparator or the alarm function. The alarm limit of two counters can be set independently by programming. The alarm status can be send to digital output channels if this function is ON. The supervisor of a factory can 'see' or 'hear' the alarm if the digital output channel control a real alarm device. The two digital output channel can be set for general purpose used if the alarm is disable. For example, connecting relay devices to DO channels, the OMR-6080 can be used to control the high power devices.
Features of OMR-6080

• Two 32 bit counter / frequency input channel
• Two digital output channels of open collector type
• 5000 Vrms isolation voltage for isolated input mode
• External gate control for counter input
• Alarm function with alarm output
• Programmable digital filter for noise rejection
• Programmable threshold setting of trigger level for non-isolated input mode
• Programmable host watchdog timer for host failure protection
• Internal watchdog timer for device failure protection
• Easy programming by software
• Easy installation and wiring

Specifications of OMR-6080

✧ Interface
• Interface : RS-485, 2 wires
• Speed (bps) : 1200, 2400, 4800, 9600, 19.2K, 38.4K

✧ Counter Input
• Two independent 32 bit counters
• Input frequency: 100 kHz max.
• Input mode: Isolated or non-isolated
• Isolated input level:
  Logic level 0: +1V max.
  Logic level 1: +3.5V to +30V
• Isolation voltage: 5000 Vrms
• Non-isolated input level (programmable threshold):
  Logic level 0: 0 to +5V (default = 0.8V)
  Logic level 1: 0 to +5V (default = 2.4V)
• Input pulse width > 5 μsec.
• Programmable digital noise filter:
  4 μsec. to 1.02 msec.
• Alarm comparator on each counter
✧ **Frequency measurement Input**  
- Range: 1 Hz to 100 kHz  
- Programmable built in gate time: 0.1/1.0 sec.

✧ **Digital Output**  
- Channels: Two open collector to 30 V, 30 mA max. load

✧ **Watchdog Function**  
- Module internal watchdog timer: 150 ms  
- Power failure threshold: 4.65 V  
- Safety value: 2 digital output channels  
- Host programmable watchdog: 100 ms ~ 25.500 sec.

✧ **Power**  
- Power supply: +10V to +30V  
- Power consumption: 2.0W
### Pin Definitions of OMR-6080

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Signal Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IN0</td>
<td>Non-isolated input of counter 0</td>
</tr>
<tr>
<td>2</td>
<td>GATE0</td>
<td>External gate control of counter 0</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Ground for non-isolated input</td>
</tr>
<tr>
<td>4</td>
<td>IN1</td>
<td>Non-isolated input of counter 1</td>
</tr>
<tr>
<td>5</td>
<td>GATE1</td>
<td>External gate control of counter 1</td>
</tr>
<tr>
<td>6</td>
<td>DEFAULT*</td>
<td>Initial state setting</td>
</tr>
<tr>
<td>7</td>
<td>(Y) DATA+</td>
<td>RS-485 series signal, positive</td>
</tr>
<tr>
<td>8</td>
<td>(G) DATA-</td>
<td>RS-485 series signal, negative</td>
</tr>
<tr>
<td>9</td>
<td>(R) +Vs</td>
<td>Power supply, +10V~+30V</td>
</tr>
<tr>
<td>10</td>
<td>(B) GND</td>
<td>Ground</td>
</tr>
<tr>
<td>11</td>
<td>GATE1-</td>
<td>Differential negative external gate control of counter 1</td>
</tr>
<tr>
<td>12</td>
<td>GATE1+</td>
<td>Differential positive external gate control of counter 1</td>
</tr>
<tr>
<td>13</td>
<td>IN1-</td>
<td>Differential negative input of counter 1</td>
</tr>
<tr>
<td>14</td>
<td>IN1+</td>
<td>Differential positive input of counter 1</td>
</tr>
<tr>
<td>15</td>
<td>GATE0-</td>
<td>Differential negative external gate control of counter 0</td>
</tr>
<tr>
<td>16</td>
<td>GATE0+</td>
<td>Differential positive external gate control of counter 0</td>
</tr>
<tr>
<td>17</td>
<td>IN0-</td>
<td>Differential negative input of counter 0</td>
</tr>
<tr>
<td>18</td>
<td>IN0+</td>
<td>Differential positive input of counter 0</td>
</tr>
<tr>
<td>19</td>
<td>DO0</td>
<td>Digital output of channel 0 or counter 0 alarm output</td>
</tr>
<tr>
<td>20</td>
<td>DO1</td>
<td>Digital output of channel 1 or counter 1 alarm output</td>
</tr>
</tbody>
</table>
A Look at OMR-6080 & Pin Assignment

Power Input
+10V ~ +30V

Power Regulator & Filter
+ 5V

GND

Watchdog/Power Failure Supervisor

Data +
Data -

RS-485 Rec/Drv

Micro Processor

EEPROM
Config Data Safe Value

Counter 0
Counter 1
2-bits
Digital Output
1-bit
Digital Input

DO0
DO1
Default* Pin

Programmable Digital Noise Filter

PHTO/TTL Input Select and GATE Control

Programmable Threshold Voltage

CH0+
CH0-

CH1+
CH1-

GATE0+
GATE0-

GATE1+
GATE1-

CH1 (TTL)
GATE0 (TTL)
GATE1 (TTL)
Functional Block Diagram of OMR-6080

<table>
<thead>
<tr>
<th>CODE</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>COUNTER</td>
</tr>
<tr>
<td>51</td>
<td>FREQUENCY</td>
</tr>
</tbody>
</table>
2. Initialization & Installation

2.1. Software Installation

1. If you had installed “OMR Administration” then skip other steps.
2. Backup your software diskette
3. Insert “OMR Administration” diskette into floppy drive A:
4. Change drive to A:
5. Installation command syntax

   INSTALL drive:
   Drive name is C to Z.

   Example 1: install to drive C:
   A:\> INSTALL C:

   Example 2: install to drive F:
   A:\> INSTALL F:

6. OMR Administration Utility will be installed in the directory C:\OMR
2. 2. Initializing a Brand-New Module

**Objective of Initializing a Brand-New OMR**

All OMR modules except OMR-6520 and OMR-6510, in an RS-485 network must have a *unique* address ID, however, every brand-new OMR has a factory default setting as following:

- Address ID is 01.
- Baud rate is 9600 bps
- Check-sum disable
- Host Watchdog timer is disable

Therefore, to configure the brand-new OMR before using is necessary, otherwise the address ID will be conflict with others modules because the ID of new modules are identity. The baud rate may also be changed according to user's requirements.

The following sections show how to initialize a brand-new module, which is applicable for initializing OMR-6080.

**Default State**

The OMR I/O modules must be set at *Default State* when you want to change the default settings, such as the baud rate and check-sum status etc. All OMR I/O modules have an special pin labeled as `DEFAULT*`. The module will be in *Default State* if the `DEFAULT*` pin is shorted to ground when power ON. Under this state, the default configuration is set as following:

- Address ID is 00.
- Baud rate is 9600 bps.
- Check-sum is disable.

Therefore, the communication between host and the module can be easily set as the same default configuration, the initialization of a module will be possible no matter what configuration is set under operating state.
Initialization Equipment

- Host computer with an RS-232 port.
- An installed RS-485 module (OMR-6520) with 9600 baud rate.
- The brand new OMR module
- Power supply (+10 to +30 VDC) for OMR modules
- Administration utility software

Initialization Procedure

1. Power off the host computer and the installed OMR-6520. Be sure of the baud rate of the OMR-6520 is 9600 bps.
2. Connect a brand new OMR module with the RS-485. Set the module in Default State by shorting the DEFAULT* pin. Refer to Figure 2.1 for detailed wiring.
3. Power on the host computer.
4. Power on the power supply for OMR modules.
5. Use the OMR Administration utility to configure the address ID, Baud rate and check-sum status of the module.

Initialization Wiring

![Diagram of initialization wiring](image)

Figure 2-1 Layout for Initialization the OMR module
## 2.3. Install a New OMR to a Existing Network

### Equipments for Install a New Module
- A existing OMR network
- New OMR modules.
- Power supply (+10 to +30 V<sub>DC</sub>).

### Installing Procedures
1. Configure the new OMR module according to the initialization procedures in section 2.2.
2. The baud rate and check-sum status of the new module must be identity with the existing RS-485 network. The address ID must not be conflict with other OMR modules on the network.
3. Power off the OMR power supply of the existing RS-485 network.
4. Power off the host computer.
5. Wire the power lines for the new OMR with the existing network. Be careful about the signal polarity as wiring.
6. Wire the RS-485 data lines for the new OMR with the existing network. Be careful about the signal polarity as wiring.
7. Wire to the input or output devices. Refer to section 2.4 for illustrations.
8. Power on the host computer.
9. Power on the OMR local power supply.
10. Use the OMR administration utility to check entire network.
2.4. Application Wiring for OMR-6080

Non-isolated Input

Counter Input
Gate Control
GND

Photo-isolated Input

IN1+
IN1-
GATE1+
GATE1-
11

+ Counter Input
- Gate Control

IN0 1
+GATE0
D.GND
3. Command Set

3.1. Command and Response

Introduction

The OMR command is composed by numbers of characteristics, including the leading code, address ID, the variables, the optional check-sum bytes, and a carriage return to indicate the end of a command. The host computer can only command only one OMR module except those synchronized commands with wildcard address "***". The OMR may or may not give response to the command. The host should check the response to handshake with the modules.

Document Conventions

The following syntax conventions describes the OMR commands in this manual.

<table>
<thead>
<tr>
<th>(Leading Code)</th>
<th>Leading Code is the first characteristic of the OMR command. All OMR commands need a command leading code, such as %,$,#,@,...etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Addr)</td>
<td>Module’s address ID, the value is in the range of 00 - FF (Hex).</td>
</tr>
<tr>
<td>(Command Variable)</td>
<td>Command codes or value of variables.</td>
</tr>
<tr>
<td>[Data]</td>
<td>Some commands need additional data.</td>
</tr>
<tr>
<td>[Checksum]</td>
<td>Checksum in brackets indicate optional parameter, only checksum is enable then this field is required.</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Identifies a control code character, such as &lt;CR&gt; for carriage return, its value is 0x0D.</td>
</tr>
</tbody>
</table>
Format of OMR Commands

When checksum is enable then [Checksum] is needed, it is 2-character. Both command and response must append the checksum characters.

How to calculate checksum value?

[Checksum] = ((LeadingCode)+(Addr)+(Command)+(Data)) MOD 0x100

Example 1: checksum is disable

User Command : $012<CR>
Response : 01400600<CR>

$ : LeadingCode
01 : Address
2 : Command (Read Configuration)
<CR> : Carriage return 0x0D

Example 2: checksum is enable

User Command : $012B7<CR>
Response : 01400600AC<CR>

$ : LeadingCode
01 : Address
2 : Command (Read Configuration)
B7 : Checksum value
<CR> : Carriage return 0x0D

\['$' = 0x24 \]
\['0' = 0x30 \]
\['1' = 0x31 \]
\['2' = 0x32 \]

B7 = ( 0x24 + 0x30 + 0x31 + 0x32 ) MOD 0x100

\['!' = 0x24 \]
\[0' = 0x30 \]
\['1' = 0x31 \]
\['4' = 0x34 \]
\[6' = 0x36 \]

AC = ( 0x24 + 0x30 + 0x31 + 0x34 + 0x30 + 0x30 + 0x36 + 0x30 + 0x30 ) MOD 0x100

Note : 1. There is no spacing between the command words and the checksum characters.
2. Every command follows a <CR> carriage return for ending.
3. The checksum characters are optional.
Response of OMR Commands

The response message depends on versatile OMR command. The response is composed with a few characteristics, including leading code, variables, and carriage return for ending. There are two categories of leading code for response message, "!" or ">" means valid command and "?" means invalid. By checking the response message, user can monitor the command is valid or not.

Note: Under the following conditions, there will have no response message.
1. The specified address ID is not exist.
2. Syntax error.
3. Communication error.
4. Some special commands do not have response.
3.2. Summary of Command Set

There are three categories of OMR commands. The first is the **general commands**, including set configuration command, read configuration, reset, read module's name or firmware version, etc. Every OMR can respond to the general commands. The second is the **functional commands**, which depends on functions of each module. Not every module can execute all function commands. The third is the **special commands** including functions about the programmable watchdog timer, safe values, and the programmable leading code. All the commands used in the OMR analog input module are list in the following table.

<table>
<thead>
<tr>
<th>Command Set of OMR 6080</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
</tr>
<tr>
<td><strong>Configuration, Counter Input &amp; Display Commands</strong></td>
</tr>
<tr>
<td>Set Configuration</td>
</tr>
<tr>
<td>Read Configuration</td>
</tr>
<tr>
<td>Read Module Name</td>
</tr>
<tr>
<td>Read Firmware Version</td>
</tr>
<tr>
<td>Set Input Signal Mode</td>
</tr>
<tr>
<td>Read Input Signal Mode</td>
</tr>
<tr>
<td>Read Counter/Frequency</td>
</tr>
<tr>
<td>Value in Hexadecimal</td>
</tr>
<tr>
<td>Read Counter/Frequency</td>
</tr>
<tr>
<td>Value in Decimal</td>
</tr>
<tr>
<td>Command</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td><strong>Counter Setup Commands</strong></td>
</tr>
<tr>
<td>Set Gate Mode</td>
</tr>
<tr>
<td>Read Gate Mode</td>
</tr>
<tr>
<td>Set Maximum Counter Value</td>
</tr>
<tr>
<td>Read Maximum Counter Value</td>
</tr>
<tr>
<td>Set Initial Count Value</td>
</tr>
<tr>
<td>Read Initial Count Value</td>
</tr>
<tr>
<td>Start/Stop Counter</td>
</tr>
<tr>
<td>Read Counter Start/Stop Status</td>
</tr>
<tr>
<td>Clear Counter</td>
</tr>
<tr>
<td>Read then Clear the Overflow Flag</td>
</tr>
<tr>
<td><strong>Digital Filter &amp; Programmable Threshold Commands</strong></td>
</tr>
<tr>
<td>Enable/Disable Digital Filter</td>
</tr>
<tr>
<td>Read Filter Status</td>
</tr>
<tr>
<td>Set Minimum Input Signal Width at High Level</td>
</tr>
<tr>
<td>Read Minimum Input Signal Width at High Level</td>
</tr>
<tr>
<td>Set Minimum Input Signal Width at Low Level</td>
</tr>
<tr>
<td>Read Minimum Input Signal Width at Low Level</td>
</tr>
<tr>
<td>Set TTL Input High Trigger Level</td>
</tr>
<tr>
<td>Read TTL Input High Trigger Level</td>
</tr>
<tr>
<td>Set TTL Input Low Trigger Level</td>
</tr>
<tr>
<td>Read TTL Input Low Trigger Level</td>
</tr>
<tr>
<td>Command</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>Digital Output &amp; Alarm Commands</strong></td>
</tr>
<tr>
<td>Enable Alarm</td>
</tr>
<tr>
<td>Disable Alarm</td>
</tr>
<tr>
<td>Set Alarm Limit Value of Counter 0</td>
</tr>
<tr>
<td>Set Alarm Limit Value of Counter 1</td>
</tr>
<tr>
<td>Read Alarm Limit Value of Counter 0</td>
</tr>
<tr>
<td>Read Alarm Limit Value of Counter 1</td>
</tr>
<tr>
<td>Set Digital Output Values</td>
</tr>
<tr>
<td>Read Digital Output and Alarm Status</td>
</tr>
<tr>
<td><strong>Special Commands</strong></td>
</tr>
<tr>
<td>Read Command Leading Code Setting</td>
</tr>
<tr>
<td>Change Command Leading Code Setting</td>
</tr>
<tr>
<td>Set Host Watchdog / Safety Value</td>
</tr>
<tr>
<td>Read Host WatchDog / Safe Value</td>
</tr>
<tr>
<td>Host is OK</td>
</tr>
</tbody>
</table>
3. 3. Set Configuration

@Description
Configure the basic setting about address ID, baud rate, and checksum.

@Syntax
\%(OldAddr)(NewAddr)(TypeCode)(BaudRate)(CheckSumFlag)<CR>

% Command leading code. (1-character)
(OldAddr) OMR module original address ID. The default address ID of a brand new module is 01. The value range of address ID is 00 to FF in hexadecimal. (2-character)
(NewAddr) New address ID, if you don’t want to change address ID, let new address ID equals to the old one. (2-character)
(TypeCode) Type Code represents the input mode. (2-character)
50: counter input mode
51: frequency input mode
(BaudRate) Communication baud rate, refer to Table 3-1 for details. (2-character)
(CheckSumFlag) Define check-sum status and frequency gate time, refer to Table 3-2 for details. (2-character)

@Response
!(Addr)<CR> or
?(Addr)<CR>

(Addr) Address ID.
! Command is valid.
?
Command is invalid. Invalid parameter values, When you wanted to change the setting without grounding the DEFAULT* pin.
Note: When you want to change the checksum or baud rate then the DEFAULT* pin should be grounded at first.

@Example

User command: %0130500600<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>01</td>
<td>(OldAddr)</td>
<td>Original address ID is 01H.</td>
</tr>
<tr>
<td>30</td>
<td>(NewAddr)</td>
<td>New address ID is 30H (Hexadecimal).</td>
</tr>
<tr>
<td>50</td>
<td>(TypeCode)</td>
<td>Counter input mode.</td>
</tr>
<tr>
<td>06</td>
<td>(BaudRate)</td>
<td>Baud rate is 9600.</td>
</tr>
<tr>
<td>00</td>
<td>(CheckSumFlag)</td>
<td>00 means checksum is disable, and frequency gate is 0.1 second.</td>
</tr>
<tr>
<td>&lt;CR&gt;</td>
<td>Carriage return</td>
<td>0x0D.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Baudrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>1200 bps</td>
</tr>
<tr>
<td>04</td>
<td>2400 bps</td>
</tr>
<tr>
<td>05</td>
<td>4800 bps</td>
</tr>
<tr>
<td>06</td>
<td>9600 bps</td>
</tr>
<tr>
<td>07</td>
<td>19200 bps</td>
</tr>
<tr>
<td>08</td>
<td>38400 bps</td>
</tr>
</tbody>
</table>

Table -1 Baud rate setting code
### Table 2 Check sum flag setting

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Checksum</td>
<td></td>
<td>0: disable, 1: enable</td>
</tr>
<tr>
<td>6</td>
<td>Frequency Gate Time</td>
<td></td>
<td>0: 0.1 second, 1: 1 second</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
<td>Must to be 0</td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td></td>
<td>Must to be 000000</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. 4. Read Configuration

@Description
Read the configuration of module on a specified address ID.

@Syntax
$$(Addr)2<CR>

$ Command leading code
(Addr) Address ID.
2 Command code for reading configuration

@Response
!(Addr)(TypeCode)(BaudRate)(CheckSumFlag)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(TypeCode) Input mode
(BaudRate) Current setting of communication baud rate, refer to Table 3-1 for details.
(CheckSumFlag) Current setting of check-sum flag, refer to Table 2. for details.

@Example
User command: $302<CR>
Response: !30500600<CR>

! Command is valid.
30 Address ID.
50 Counter Input Mode.
06 Baud rate is 9600 bps.
00 checksum is disable, frequency gate is 0.1 second.
3. 5. Read Module Name

@Description
Read OMR module's name.

@Syntax
$(Addr)M<CR>

$ Command leading code.
(Addr) Address ID
M Read module name

@Response
!(Addr)(ModuleName) <CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(ModuleName) OMR module's name.

@Example
User command: $30M<CR>
Response: !306080<CR>

! Command is valid.
30 Address
6080 OMR-6080 (Counter/Frequency module)
3. 6. Read Firmware Version

@Description
Read OMR module's firmware version.

@Syntax
$(Addr)F<CR>

$ Command leading code.
(Addr) Address ID
F Read module firmware version.

@Response
!(Addr)(FirmRev) <CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(FirmRev) OMR module's firmware version.

@Example
User command: $30F<CR>
Response: !30A1.50<CR>

! Command is valid.
30 Address
A1.50 Firmware Version
3. 7. Set Input Mode

@Description
Set the input signal mode of counter/frequency to either TTL or photo isolated mode.

@Syntax
$(Addr)B(InType)<CR>

$ Command leading code.
(Addr) Address ID
B Set input mode Command
(InType) 0: TTL input
          1: photo isolated input

@Response
!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $30B0<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>Set Input mode.</td>
</tr>
<tr>
<td>0</td>
<td>(InType)</td>
<td>TTL input.</td>
</tr>
</tbody>
</table>
3. 8. Read Input Mode

@Description
Read the input signal mode of counter/frequency module.

@Syntax
$(Addr)B<CR>

$ Command leading code.
(Addr) Address ID
B Read input mode Command

@Response
!(Addr)(InType)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(InType) 0: TTL input mode.
1: Photo isolated input mode.

@Example
User command: $30B<CR>
Response: !301<CR>

! Command is valid.
30 Address
1 Photo isolated input.
3. 9. Read Counter/Frequency Value in HEX Format

@Description
Read the Counter/Frequency module of counter 0 or 1 and return the acquired data in hexadecimal format.

@Syntax

```plaintext
#(Addr)(CounterNo)<CR>
```

# Command leading code. (1-character)
(Addr) Address ID (2-character)
(CounterNo) 0: Counter 0.
1: Counter 1. (1-character)

@Response

```plaintext
> Data<CR>
or
?(Addr)<CR>
```

> Command is valid
?
Command is invalid.
(Addr) Address ID.

@Example

- User command:

  ```plaintext
  #300<CR>
  ```

  Response:

  ```plaintext
  >0000FFFF<CR>
  ```

  30 Address ID
  0 Read counter 0 value
  0000FFFF Return value 0x0000FFFF = 65,535

- User command:

  ```plaintext
  #2F1<CR>
  ```

  Response:

  ```plaintext
  >00001234<CR>
  ```

  2F Address ID
  1 Read Counter 1 Value
  00001234 Return value 0x00001234 = 4,660
3. 10. Read Counter/Frequency Value in DEC Format

@Description
Read the Counter/Frequency module of counter 0 or 1 and return the acquired data in decimal format.

@Syntax

```
#(Addr)(CounterNo)D<CR>
```

#  Command leading code. (1-character)
(Addr)  Address ID (2-character)
(CounterNo)  0: Counter 0.
            1: Counter 1. (1-character)
D  Decimal command code.

@Response

```
Data<CR>
or
?(Addr)<CR>
```

>  Command is valid
?  Command is invalid.
(Addr)  Address ID.

@Example

User command:  #300<CR>
Response:  >0000065535<CR>

30  Address ID
0  Read counter 0 value
0000065535  Return value 65535

User command:  #2F1<CR>
Response:  >0000001234<CR>

2F  Address ID
1  Read Counter 1 Value
0000001234  Return value 1234
3. 11. Set Gate Mode

@Description
Set the counter input module’s gate control to either high, low or disable.

@Syntax
\$(Addr)A(Gmode)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
A Gate command code
(Gmode) 0: the gate is low
1: the gate is high
2: the gate is disable

@Response
!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $30A0<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>Set gate mode.</td>
</tr>
<tr>
<td>0</td>
<td>(Gmode)</td>
<td>The gate is low.</td>
</tr>
</tbody>
</table>
3. 12. Read Gate Mode

@Description
Read the counter input module’s gate status.

@Syntax
$(Addr)A<CR>

$ Command leading code.
(Addr) Address ID (2-character)
A Gate command code

@Response
!(Addr)(Gmode)<CR>
or
?(Addr)<CR>
!
Command is valid.
?
Command is invalid.
(Addr)
Address ID.
(Gmode)
0: the gate is low
1: the gate is high
2: the gate is disable

@Example
User command: $30A<CR>
Response: !301<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>Set gate mode.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
1 The gate is high.
3. 13. Set Maximum Counter Value

@Description
Set the maximum counter value of counter 0 or counter 1.

@Syntax
$(Addr)3(CounterNo)(MaxData)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
3 Maximum counter value command.
(CounterNo) 0: counter 0
1: counter 1
(MaxData) The maximum counter value which consists of 8 hexadecimal digits. When counting value exceeds the maximum counter value, an overflow flag status will set.

@Response
!(Addr)<CR> or 
?(Addr)<CR>
!
Command is valid.

? Command is invalid.
(Addr) Address ID.

@Example
User command: $303000000000<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>3</td>
<td>(CounterNo)</td>
<td>Set maximum counter value.</td>
</tr>
<tr>
<td>0</td>
<td>(MaxData)</td>
<td>Counter 0.</td>
</tr>
<tr>
<td>00010000</td>
<td></td>
<td>65536(0x00010000)</td>
</tr>
</tbody>
</table>

3. 14. Read Maximum Counter Value

@Description
Read the maximum counter value of counter 0 or counter 1.

@Syntax
$(Addr)3(CounterNo)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
3 Maximum counter value command code
(CounterNo) 0: counter 0
1: counter 1

@Response

\![\text{(Addr)}\text{(MaxData)}]<\text{CR}>

or

\?[\text{(Addr)}]<\text{CR}>

! Command is valid.
?
Command is invalid.
\text{(Addr)} Address ID.
\text{(MaxData)} The maximum counter value which consists of 8 hexadecimal digits.

@Example

User command: $3031<\text{CR}>
Response: !3000001234<\text{CR}>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Read maximum counter value.</td>
</tr>
<tr>
<td>1</td>
<td>(CounterNo)</td>
<td>Counter 1.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
00001234 4660(0x00001234).
3. 15. Set Initial Count Value

@Description
Set the initial count value of counter 0 or counter 1.

@Syntax
$(Addr)P(CounterNo)(IniData)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
P Set initial count value command code.
(CounterNo) 0: counter 0
1: counter 1
(IniData) The initial count value which consists of 8 hexadecimal digits.

@Response
!(Addr)<CR> Command is valid.
?(Addr)<CR> Command is invalid.

@Example
User command: $30P000000100<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td>Set initial count value.</td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
<tr>
<td>00000100</td>
<td>(IniData)</td>
<td>256(0x00000100)</td>
</tr>
</tbody>
</table>
3. 16. Read Initial Count Value

@Description
Read the initial count value of counter 0 or counter 1.

@Syntax
\[ $(\text{Addr})G(\text{CounterNo})<\text{CR}> \]

$ Command leading code.
(Addr) Address ID (2-character)
G Read initial counter value command code
(CounterNo) 0: counter 0
1: counter 1

@Response
\[ !(\text{Addr})(\text{IniData})<\text{CR}> \]
or\[ ?(\text{Addr})<\text{CR}> \]
!
Command is valid.
?
Command is invalid.
(Addr) Address ID.
(IniData) The initial count value which consists of 8 hexadecimal digits.

@Example
User command: \$30G1<CR>
Response: !30000000FF<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>G</td>
<td>(CounterNo)</td>
<td>Read initial count value.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Counter 1.</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>Command is valid.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Address of counter/frequency module.</td>
</tr>
<tr>
<td>000000FF</td>
<td></td>
<td>255(0x000000FF).</td>
</tr>
</tbody>
</table>
3. 17. Start/Stop Counter

@Description
Start or stop counting of counter 0 or counter 1.

@Syntax
$(Addr)5(CounterNo)(SStatus)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
5 Start/stop counter command code.
(CounterNo) 0: counter 0
1: counter 1
(SStatus) 0: stop counting
1: start counting

@Response
!(Addr)<CR> Command is valid.
?(Addr)<CR> Command is invalid.

@Example
User command: $30501<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Start/stop counting command.</td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
<tr>
<td>1</td>
<td>(SStatus)</td>
<td>Start counting.</td>
</tr>
</tbody>
</table>
3. 18. Read Start/Stop Counter Status

@Description
Read the status of counter 0 or counter 1 for its active or inactive condition.

@Syntax
$(Addr)5(CounterNo)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
5 Start/stop counter command code.
(CounterNo) 0: counter 0
1: counter 1

@Response
!(Addr)(SStatus)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(Sstatus) 0: stop counting
1: start counting

@Example
User command: $3050<CR>
Response: !301<CR>

! Command is valid.
30 Address of counter/frequency module.
1 Counter 0 is counting.
3. 19. Clear Counter

@Description
Clear the value of counter 0 or counter 1.

@Syntax
$(Addr)6(CounterNo)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
6 Clear counter command code.
(CounterNo) 0: counter 0
1: counter 1

@Response
!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $3060<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Clear counter command code.</td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
</tbody>
</table>
3. 20. Read then Clear Overflow Flag

@Description
Read the status of the overflow flag of counter 0 or counter 1, and then clear the flag afterward.

@Syntax
$(Addr)7(CounterNo)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
7 Read then clear overflow command code.
(CounterNo) 0: counter 0
1: counter 1

@Response
!(Addr)(OFlag)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(OFlag) 0: the overflow flag has not been set
1: the counting value has exceeded the maximum count, the overflow flag has been set.

* After executing the command, the overflow flag will clear to zero if it has been set.
@Example

User command: $3070<CR>
Response: !301<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Read counter overflow command code.</td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
1 Counter 0 is overflowed.
3. 21. Enable/Disable Digital Filter

@Description
Enable or disable the digital filter function.

@Syntax
$(Addr)4(FStatus)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
4 Enable/Disable filter command code.
(FStatus) 0: disable filter
1: enable filter

@Response
!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $3040<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Enable/Disable filter command.</td>
</tr>
<tr>
<td>0</td>
<td>(FStatus)</td>
<td>Disable filter.</td>
</tr>
</tbody>
</table>
3. 22. Read Filter Status

@Description
Read the digital filter enable/disable status.

@Syntax
$(Addr)4<CR>

$ Command leading code.
(Addr) Address ID (2-character)
4 Enable/Disable filter command code.

@Response
!(Addr)(FStatus)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(FStatus) 0: disable filter
1: enable filter

@Example
User command: $304<CR>
Response: !301<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Enable/Disable filter command.</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>Command is valid.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Address of counter/frequency module.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Digital filter is enable.</td>
</tr>
</tbody>
</table>
3. 23. Set Minimum Input Signal Width at High Level

@Description
Set the minimum input signal width at high level, for signal level high less then this value will be filtered out as noise.

@Syntax
$(Addr)0H(MinFData)<CR>

$ Command leading code.
Addr Address ID (2-character)
0H Set minimum input signal width at high level command code.
(MinFData) The minimum width data at high level. The unit is µs and its resolution is 1 µs. This value range from 4 µs to 1020 µs, which is a 4-digit integer. (4-character)

@Response
!(Addr)<CR> or ?(Addr)<CR>

! Command is valid.
? Command is invalid.
Addr Address ID.

@Example
User command: $30H0100<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>0H</td>
<td></td>
<td>Set minimum input signal width.</td>
</tr>
<tr>
<td>0100</td>
<td>(MinFData)</td>
<td>100 µs</td>
</tr>
</tbody>
</table>
3. 24. Read Minimum Input Signal Width at High Level

@Description
Read the minimum input signal width at high level.

@Syntax
$(Addr)0H<CR>

$ Command leading code.
(Addr) Address ID (2-character)
0H Set minimum input signal width at high level command code.

@Response
!(Addr)(MinFData)<CR> or ?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(MinFData) The minimum width data at high level. The unit is µs and its resolution is 1 µs. This value range from 4 µs to 1020 µs, which is a 4-digit integer.

@Example
User command: $300H<CR>
Response: !300100<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>0H</td>
<td></td>
<td>Set minimum input signal width.</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>Command is valid.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Address of counter/frequency module.</td>
</tr>
<tr>
<td>0100</td>
<td></td>
<td>Digital filter value of minimum signal width at high level is 100 µs.</td>
</tr>
</tbody>
</table>
3. 25. Set Minimum Input Signal Width at Low Level

@Description
Set the minimum input signal width at low level, for signal level low less then this value will be filtered out as noise.

@Syntax
$(Addr)0L(MinFData)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
0L Set minimum input signal width at low level command code.
(MinFData) The minimum width data at low level. The unit is µs and its resolution is 1 µs. This value range from 4 µs to 1020 µs, which is a 4-digit integer. (4-character)

@Response
!(Addr)<CR> or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $300L0010<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>0L</td>
<td></td>
<td>Set minimum input signal width.</td>
</tr>
<tr>
<td>0010</td>
<td>(MinFData)</td>
<td>10 µs</td>
</tr>
</tbody>
</table>
3. 26. Read Minimum Input Signal Width at Low Level

@Description
Read the minimum input signal width at low level.

@Syntax
$(Addr)0L<CR>

$ Command leading code.
(Addr) Address ID (2-character)
0L Set minimum input signal width at low level command code.

@Response
!(Addr)(MinFData)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(MinFData) The minimum width data at low level. The unit is µs and its resolution is 1 µs. This value range from 4 µs to 1020 µs, which is a 4-digit integer.

@Example
User command: $30L<CR>
Response: !300010<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>0L</td>
<td></td>
<td>Set minimum input signal width.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
0010 Digital filter value of minimum signal width at low level is 10 µs.
3. 27. Set TTL Input High Trigger Level

@Description
Set the TTL input high trigger level, for voltage level higher than this value is recognized as logic high.

@Syntax
$(Addr)1H(ThData)<CR>

$ Command leading code.
(Addr) Address ID (2-character)
1H TTL input high trigger level command code.
(ThData) The high trigger level for TTL input. The unit is 0.1 V and its resolution is 0.1 V too. This value range from 0.1 to 5V, which is a 2-digit integer.

@Response
!(Addr)<CR> or ?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: $301H30<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>1H</td>
<td></td>
<td>Set TTL input high trigger level.</td>
</tr>
<tr>
<td>30</td>
<td>(ThData)</td>
<td>3 V</td>
</tr>
</tbody>
</table>
3. 28. Read TTL Input High Trigger Level

@Description
Read the TTL input high trigger level.

@Syntax
$(Addr)1H<CR>

$ Command leading code.
(Addr) Address ID (2-character)
1H TTL input high trigger level command code.

@Response
!(Addr)(ThData)<CR>
or
?(Addr)<CR>
!
Command is valid.
?
Command is invalid.
(Addr) Address ID.
(ThData) The high trigger level for TTL input. The unit is 0.1 V and its resolution is 0.1 V too. This value range from 0.1 to 5V, which is a 2-digit integer.

@Example
User command:  $301H<CR>
Response:    !3024<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>1H</td>
<td></td>
<td>Read TTL input high trigger level.</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>Command is valid.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Address of counter/frequency module.</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>The high trigger level is 2.4 V.</td>
</tr>
</tbody>
</table>
3. 29. Set TTL Input Low Trigger Level

@Description
Set the TTL input low trigger level, for voltage level lower than this value is recognized as logic low.

@Syntax
```
$(Addr)1L(ThData)<CR>
```

$ Command leading code.
(Addr) Address ID (2-character)
1L TTL input low trigger level command code.
(ThData) The low trigger level for TTL input. The unit is 0.1 V and its resolution is 0.1 V too. This value range from 0.1 to 5V, which is a 2-digit integer.

@Response
```
!(Addr)<CR>
```

! Command is valid.
?
Command is invalid.
(Addr) Address ID.

@Example
User command: $301L10<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>1L</td>
<td>Set TTL input low trigger level.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(ThData)</td>
<td>1 V</td>
</tr>
</tbody>
</table>
3. 30. Read TTL Input Low Trigger Level

@Description
Read the TTL input low trigger level.

@Syntax
$(Addr)1L<CR>

$ Command leading code.
(Addr) Address ID (2-character)
1L TTL input low trigger level command code.

@Response
!(Addr)(ThData)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(ThData) The high trigger level for TTL input. The unit is 0.1 V and its resolution is 0.1 V too. This value range from 0.1 to 5V, which is a 2-digit integer.

@Example
User command: $301L<CR>
Response: !3008<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>1L</td>
<td></td>
<td>Read TTL input low trigger level.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
08 The low trigger level is 0.8 V.
### 3. 31. Enable Alarm

@Description
Enables alarm function of counter 0 or counter 1. The digital output will assert if the counter value reaches the alarm limit while the alarm is enable.

@Syntax
```
@(<Addr>)EA(<CounterNo>)<CR>
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>EA</td>
<td>Enable alarm command code.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
</tbody>
</table>

@Response
```
!(<Addr>)<CR>

or

?(<Addr>)<CR>
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Command is valid.</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>Command is invalid.</td>
<td></td>
</tr>
<tr>
<td>(&lt;Addr&gt;)</td>
<td>Address ID.</td>
<td></td>
</tr>
</tbody>
</table>

@Example
User command: `@30EA0<CR>`
Response: `!30<CR>`
3. 32. Disable Alarm

@Description
Disables alarm function of counter 0 or counter 1.

@Syntax

@{(Addr)DA(CounterNo)}<CR>

@ Command leading code.
(Addr) Address ID (2-character)
DA Enable alarm command code.
(CounterNo) 0: counter 0
1: counter 1

@Response

!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example

User command: @30DA0<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>DA</td>
<td></td>
<td>Disable alarm command code.</td>
</tr>
<tr>
<td>0</td>
<td>(CounterNo)</td>
<td>Counter 0.</td>
</tr>
</tbody>
</table>
3. 33. Set Alarm Limit Value of Counter 0

@Description
Set the alarm limit value of counter 0.

@Syntax
@{(Addr)}PA{(ArmData)}<CR>

@ (Leading Code)
(Addr) Address ID (2-character)
PA Set alarm limit value command code.
(ArmData) The alarm limit value which consists of 8 hexadecimal digits.

@Response
!{(Addr)}<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.

@Example
User command: @30PA00020000<CR>
Response: !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>PA</td>
<td></td>
<td>Set alarm limit value of counter 0.</td>
</tr>
<tr>
<td>00020000</td>
<td>(ArmData)</td>
<td>131072(0x00020000)</td>
</tr>
</tbody>
</table>
**3. 34. Set Alarm Limit Value of Counter 1**

@Description
Set the alarm limit value of counter 1.

@Syntax
```
@((Addr)SA(ArmData)<CR>
```
- @ Command leading code.
- (Addr) Address ID (2-character)
- SA Set alarm limit value command code.
- (ArmData) The alarm limit value which consists of 8 hexadecimal digits.

@Response
```
!(Addr)<CR>
or
?(Addr)<CR>
```
- ! Command is valid.
- ? Command is invalid.
- (Addr) Address ID.

@Example
- User command: `@30SA0002FFFF<CR>`
- Response: `!30<CR>`

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>SA</td>
<td></td>
<td>Set alarm limit value of counter 1.</td>
</tr>
<tr>
<td>0002FFFF</td>
<td>(ArmData)</td>
<td>196607(0x0002FFFF)</td>
</tr>
</tbody>
</table>
3. 35. Read Alarm Limit Value of Counter 0

@Description
Read the alarm limit value of counter 0.

@Syntax
@ (Addr) RP <CR>

@ Command leading code.
(Addr) Address ID (2-character)
RP Read alarm limit value command code

@Response
!(Addr)(ArmData)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(ArmData) The alarm limit value which consists of 8 hexadecimal digits.

@Example
User command: @30RP<CR>
Response: !30000FFFF<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>RP</td>
<td></td>
<td>Read alarm limit value of counter 0.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
0000FFFF 65535(0x0000FFFF).
3. 36. Read Alarm Limit Value of Counter 1

@Description
Read the alarm limit value of counter 1.

@Syntax
@\( (\text{Addr})\)RA<CR>

- @ Command leading code.
- (Addr) Address ID (2-character)
- RA Read alarm limit value command code

@Response
!(\(\text{Addr}\))(\text{ArmData})<CR>
or
?(\(\text{Addr}\)<CR>

- ! Command is valid.
- ? Command is invalid.
- (Addr) Address ID.
- (ArmData) The alarm limit value which consists of 8 hexadecimal digits.

@Example
User command: @30RA<CR>
Response: !30001FFFF<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>RA</td>
<td></td>
<td>Read alarm limit value of counter 0.</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>Command is valid.</td>
</tr>
<tr>
<td>30</td>
<td>Address of counter/frequency module.</td>
<td></td>
</tr>
<tr>
<td>0001FFFF</td>
<td></td>
<td>131071(0x0001FFFF).</td>
</tr>
</tbody>
</table>
3. 37. Set Digital Output Values

@Description
Set the value (ON or OFF) of the 2 channel digital outputs.

@Syntax
@/(Addr)DO(DoData)<CR>

@ (Leading Code) Command leading code.
(Addr) Address ID
DO Set digital data output command code.
(DoData) 00: DO0 is OFF, DO1 is OFF
          01: DO0 is ON, DO1 is OFF
          02: DO0 is OFF, DO1 is ON
          03: DO0 is ON, DO1 is ON

@Response
!/(Addr)<CR>  Command is valid.
?/(Addr)<CR>  Command is invalid.

@Example
User command:  @30DO01<CR>
Response:  !30<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>DO</td>
<td></td>
<td>Set digital data output.</td>
</tr>
<tr>
<td>01</td>
<td>(DoData)</td>
<td>DO0 is ON, DO1 is OFF</td>
</tr>
</tbody>
</table>
3. 38. Read Digital Output and Alarm Status

@Description
Read the current digital output channel values and the status of alarm function.

@Syntax

@ (Addr)DI<CR>

@ Command leading code.
(Addr) Address ID
DI Read digital data output and alarm status command code.

@Response

!(Addr)(AStatus)(DoData)00<CR>
or
=?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
(AStatus) 0: counter 0 alarm is disabled, counter 1 alarm is disabled.
1: counter 0 alarm is enabled, counter 1 alarm is disabled.
2: counter 0 alarm is disabled, counter 1 alarm is enabled.
3: counter 0 alarm is enabled, counter 1 alarm is enabled.
(DoData) 00: DO0 is OFF, DO1 is OFF
01: DO0 is ON, DO1 is OFF
02: DO0 is OFF, DO1 is ON
03: DO0 is ON, DO1 is ON
@Example

User command:  @30DI<CR>
Response: !30030200<CR>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>(Leading Code)</td>
<td>Command leading code.</td>
</tr>
<tr>
<td>30</td>
<td>(Addr)</td>
<td>Address ID is 30H.</td>
</tr>
<tr>
<td>DI</td>
<td></td>
<td>Set digital data output.</td>
</tr>
</tbody>
</table>

! Command is valid.
30 Address of counter/frequency module.
3 Counter 0 alarm is enabled, counter 1 alarm is enabled.
02 DO0 is OFF, DO1 is ON.
3. 39. Read Command Leading Code Setting

@Description
Read command leading code setting and host watchdog status.

@Syntax

~(Addr)0<CR>

~ Command leading code.
(Addr) Address ID
0 Read command leading code setting.

@Response

!(Addr)(Status)(C1)(C2)(C3)(C4)(C5)(C6)<CR>

or

?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID
(Status) (2-character)
 Bit 0 : Reserved
 Bit 1 : Power failure or watchdog failure
 Bit 2 : Host watchdog is enable
 Bit 3 : Host failure
(C1) Leading code 1, for read configuration status,
 firmware version, etc. default is $. (1-character)
(C2) Leading code 2, for read synchronize sampling,
 digital output, default is #. (1-character)
(C3) Leading code 3, for change configuration.
 default is %. (1-character)
(C4) Leading code 4, for read alarm status, enable
 alarm, etc. default is @. (1-character)
(C5) Leading code 5, for read command leading
 code, change command leading code, etc.
 default is ~. (1-character)
(C6) Leading code 6, this leading code is reserved.
 default is *. (1-character)

@Example

User command: ~060<CR>
Response: !0600$#%@~<CR>

Command leading code setting is $#%@~ for module address ID is 06, current status is factory default setting.
3. 40. Change Command Leading Code Setting

@Description
User can use this command to change command leading code setting as he desired.

@Syntax
~(Addr)10(C1)(C2)(C3)(C4)(C5)(C6)<CR>

~ Command leading code.
(Addr) Address ID, range (00 - FF).
10 Change command leading code setting.
(C1) Leading code 1, for read configuration status, firmware version, etc. default is $.
   (1-character)
(C2) Leading code 2, for read synchronize sampling, digital output, default is #. (1-character)
(C3) Leading code 3, for change configuration. default is %. (1-character)
(C4) Leading code 4, for read alarm status, enable alarm, etc. default is @. (1-character)
(C5) Leading code 5, for read command leading code, change leading code, etc. default is ~.
   (1-character)
(C6) Leading code 6, this leading code is reserved. default is *. (1-character)

@Response
!(Addr)< CR> or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID.
@Examples

<table>
<thead>
<tr>
<th>User command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>~060&lt;CR&gt;</td>
<td>!0600$#%@~*&lt;CR&gt;</td>
</tr>
<tr>
<td><del>0610A#%@</del>*&lt;CR&gt;</td>
<td>!06&lt;CR&gt;</td>
</tr>
<tr>
<td>A06F</td>
<td>!06A1.8&lt;CR&gt;</td>
</tr>
</tbody>
</table>

Read leading code setting is $#%@~* for module address 06 and change leading code $ to A, then use A06F to read firmware version of module on address 06.

*** WARNING ***

- We do not recommend users to change the default setting of leading code, because it will make you confuse....
- The leading code change only use the command conflicts other devices on the network.
3. 41. Set Host Watchdog Timer & Safety Value

@Description
Set host watchdog timer, module will change to safety state when host is failure. Define the output value in this command.

@Syntax
~(Addr)2(Flag)(TimeOut)(SafeValue)<CR>

~ Command leading code.
(Addr) Address ID, range (00 - FF).
2 Set host watchdog timer and safe state value.
(Flag) 0 : Disable host watchdog timer
1 : Enable host watchdog timer (1-character)
(TimeOut) Host timeout value, between this time period host must send (Host is OK) command to module, otherwise module will change to safety state.
Range 01 - FF. (2-character)
One unit is 53.3 ms (Firmware version 1.x)
01 = 1 * 53.3 = 53.3 ms
FF = 255 * 53.3 = 13.6 sec
One unit is 100 ms (Firmware version 2.x)
01 = 1 * 100 = 100 ms
FF = 255 * 100 = 25.5 sec
(SafeValue) 8 channels safety value of digital output channels when host is failure. (2-character)

@Response
!(Addr)<CR>
or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID

@Example
User command: ~0621121C<CR>
Response: !06<CR>
06 Address ID
2 Set host watchdog timer and safe state value.
1 Enable host watchdog timer.
12 Timeout value. 0x12 = 18
18 * 53.3 = 959 ms (Firmware Version 1.x)
18 * 100 = 1800 ms (Firmware Version 2.x)
1C 1C (00011100) Digital output channel DO3, DO4 and DO5 are high, the others are low.
3.42. Read Host Watchdog Timer & Safety Value

@Description
Read host watchdog timer setting and the safety value.

@Syntax
~(Addr)3<CR>

~ Command leading code.
(Addr) Address ID
3 Read host watchdog setting and module safety state value.

@Response
!(Addr)(Flag)(TimeOut)(SafeValue)<CR>

or
?(Addr)<CR>

! Command is valid.
? Command is invalid.
(Addr) Address ID, range (00 - FF).
(Flag) 0 : Host watchdog timer is disable
       1 : Host watchdog timer is enable(1-character)
(TimeOut) Host timeout value.
          Range 01 - FF. (2-character)
          One unit is 53.3 ms (Firmware version 1.x)
          01 = 1 * 53.3 = 53.3 ms
          FF = 255 * 53.3 = 13.6 sec
          One unit is 100 ms (Firmware version 2.x)
          01 = 1 * 100 = 100 ms
          FF = 255 * 100 = 25.5 sec
(SafeValue) 8 channels safety state digital output value when host is failure. (2-character)

@Example
User command: ~063<CR>
Response: 1061121C<CR>

06 Address ID
1 Host watchdog timer is enable.
12 Timeout value. 0x12 = 18
   18 * 53.3 = 959 ms (Firmware Version 1.x)
   18 * 100 = 1800 ms (Firmware Version 2.x)
1C 1C (00011100) Digital output channel DO3, DO4
    and DO5 are high, the others are low.

Between 959 ms (Firmware Version 1.x) or 1800 ms (Firmware Version 2.x) time period, if host does not send
(Host is OK) then digital output will change to safety state 1C (00011100) means digital output DO3 , DO4
and DO5 is high, others are low.
3. 43. Host is OK

@Description
When host watchdog timer is enable, host computer must send this command to every module before timeout otherwise “host watchdog timer enable” module’s output value will go to safety state output value.

Timeout value and safety state output value is defined in 3.14. “Set Host Watchdog Timer & Safety Value”

@Syntax
~**<CR>
~ Command leading code.
** Host is OK.

@Response
Note : Host is OK command has NO response.

@Example
User command:  ~**<CR>