Programmable High-Speed Counter/Timer

CTH Series Communication Manual · RS232 · Modbus RTU Firmware V3.00 · Rev 1

Thank you for purchasing an INICS product.

This communication manual describes the communication specifications, frame structure, addresses (Mapping Table) and communication examples for connecting the CTH Series programmable high-speed counter/timer to a host device such as a PC or PLC over RS232 (Modbus RTU).

About This Manual

- Read this communication manual thoroughly before using the product.
- For wiring, function settings and **safety precautions**, always refer to the **User Manual** supplied with the product.
- This communication manual is not supplied with the product. Download it from our website (www.iinics.com).
- The contents of this manual may change without notice for product performance and firmware improvements.

■ Notation

Notation	Description
※	Supplementary explanation of the function
e.g.)	Example of the function
0064H	Hexadecimal value (H: Hexadecimal). e.g.) 10H = 16
301004 (03EB)	Modbus register number and the Address actually sent in the frame (hexadecimal)
01 | 04 | 03 EB	Field separator in a frame (address | function code | Address ...), **bold**: CRC

Contents

1. Communication Specifications
2. Wiring and Communication Settings
3. Modbus RTU Frame
4. Frame Structure by Function Code
5. Exception Responses
6. Addresses (Mapping Table)
7. Data Format and Write Rules
8. Communication Examples
9. Troubleshooting

1. Communication Specifications

The CTH Series uses the **Modbus RTU** protocol. Broadcast commands (address 0) are not supported.

Item	Description
Interface	RS232, 1:1 connection (max. 15 m)
Protocol	Modbus RTU (slave)
Comm. address	1~127 (setting mode Adr, default 1)
Baud rate	9600 / 19200 / 38400 / 115200 bps (setting mode bPS, default 9600)
Data format	Start bit 1 bit, data 8 bits, parity None, stop bit 1 bit (fixed)
Error check	CRC-16 (Modbus)
Function codes	01, 02, 03, 04, 05, 06, 16 (10H)
Max. per request	01 · 02: max. 5, 03 · 04: max. 32 registers, 16: max. 27 registers
Response time	Responds immediately after the request is received (3.5-character idle time)
Comm. write protect	Setting mode CPrt = ON: writes by function codes 06 · 16 are blocked (RESET · BATCH RESET by function code 05 are allowed)

※ The product does not respond in these cases: CRC error, a different comm. address, address 0 (broadcast), wrong frame length for function codes 01~06

※ Set the response timeout of the host device to 100 ms or more, and send the next request at least 3.5 character times (2 ms recommended) after receiving a response. The product discards data received during 3.5 character times after it finishes sending a response (for echo wiring where its own response comes back).

2. Wiring and Communication Settings

■ 2.1 Wiring

Product terminal	PC (D-SUB 9 pin)	PLC	Remarks
RXD	RXD (pin 2)	Receive (RD)	Connect same names
TXD	TXD (pin 3)	Send (SD)	Connect same names
GND (0V)	GND (pin 5)	SG	Common

※ The RXD · TXD labels on the product terminals are **named from the host device side**. Always wire **after turning off the power**.

※ For a PC without an RS232 port, use a USB-RS232 converter. To connect RS485 equipment, use an RS232/RS485 converter with automatic direction control. (1:1 communication)

※ The connector pin assignment of PLCs differs by model. Refer to the manual of the PLC.

■ 2.2 Communication Settings

Setting item	Display	Setting
Comm. address	Adr	1~127 (default 1)
Baud rate	bPS	9600 / 19200 / 38400 / 115200 bps (front display 9.6 / 19.2 / 38.4 / 115.2, default 9600)
Comm. write protect	CPrt	OFF: write allowed (default), ON: write protected (only read · RESET · BATCH RESET possible)

- Communication settings can be changed **only in the front panel setting mode**; over communication they can only be read.
- Set the communication conditions of the host device (baud rate, 8 data bits, parity None, 1 stop bit) the same as the product.
- You can set up and operate the product with a free Modbus program on a PC (**QModMaster** etc.) or with any general Modbus RTU master such as a PLC. For connection and usage, see the **CTH Series Modbus Connection Guide**.

3. Modbus RTU Frame

Modbus RTU sends data in a continuous binary format. When the host device (master) sends a **request**, the product (slave) sends a **response**. Frames are separated by an idle time of **3.5 character times or more** (about 4 ms at 9600 bps), which marks the start and end of each frame.

Address	Function code	Data	CRC
1 byte	1 byte	N bytes	2 bytes
1~127	01~06, 10H	Address · count · value etc.	Low byte first

■ 3.1 Value Representation

- All frame values in this manual are written in **hexadecimal**. 2-byte values (Address, count, data) are sent **high byte first**.
- Only the CRC is sent **low byte first**.
- Registers are sent as the **Address** counted from 0. e.g.) register 301004 → Address 1003 = 03EBH (see 6.1 Reading register numbers)

■ 3.2 CRC-16 Calculation

The CRC (Cyclic Redundancy Check) detects data errors that occur during transmission. The sender calculates the CRC **from the address to the last data byte** and appends it to the frame; the receiver recalculates it the same way and checks that it matches the received CRC. The product does not respond to a request with a wrong CRC.

Item	Description
Type	CRC-16 (Modbus)
Initial value	FFFFH
Polynomial	A001H (bit-reversed 8005H)
Range	Address ~ last data byte (excluding the 2 CRC bytes)
Send order	Low byte → high byte

Calculation steps

Step	Calculation
1	Set the CRC to the initial value **FFFFH**.
2	XOR one byte of the frame with the low byte of the CRC.
3	Shift the CRC right by 1 bit. If the least significant bit (bit0) before the shift was **1**, **XOR** the shifted value **with A001H**; if it was **0**, just shift.
4	Repeat step 3 **8 times**. (1 byte done)
5	Repeat steps 2 ~ 4 for the next byte.
6	After all bytes, append the CRC to the end of the frame in the order **low byte, high byte**.

■ 3.3 CRC Calculation Example (present value read request: 01 | 04 | 03 EB | 00 02)

① **First byte 01H:** start from FFFFH XOR 0001H = **FFFEH** and repeat 8 times.

Repeat	CRC before shift	bit0	Shift right 1 bit	XOR A001H	Result
1	FFFE	0	7FFF	-	7FFF
2	7FFF	1	3FFF	XOR A001	9FFE
3	9FFE	0	4FFF	-	4FFF
4	4FFF	1	27FF	XOR A001	87FE
5	87FE	0	43FF	-	43FF
6	43FF	1	21FF	XOR A001	81FE
7	81FE	0	40FF	-	40FF
8	40FF	1	207F	XOR A001	**807E**

② **Process the remaining bytes the same way.** (CRC after each byte)

Byte	01	04	03	EB	00	02
CRC after byte	807E	E301	C162	A600	00A6	**BB01**

③ **Send:** final CRC = **BB01H** → append low byte **01H**, then high byte **BBH**.

Frame sent	01 | 04 | 03 EB | 00 02 | **01 BB**

※ Receive check: calculating the whole received frame (including the 2 CRC bytes) the same way gives **0000H**.
Any other result means a corrupted frame.

■ 3.4 CRC Calculation Program Example (C language)

```
unsigned short Crc16(const unsigned char *buf, int len)
{
    unsigned short crc = 0xFFFF;              /* 1. initial value */
    int i, k;

    for (i = 0; i < len; i++) {
        crc ^= buf[i];                         /* 2. XOR with low byte */
        for (k = 0; k < 8; k++) {              /* 4. repeat 8 times */
            if (crc & 0x0001) crc = (crc >> 1) ^ 0xA001;   /* 3. */
            else              crc = (crc >> 1);
        }
    }
    return crc;   /* 6. send: crc & 0xFF (low) -> crc >> 8 (high) */
}
```

※ e.g.) buf = 01H, 04H, 03H, EBH, 00H, 02H (len = 6) → returns BB01H → send 01 BB

※ The program example is for reference only. If your PLC or communication program calculates the CRC automatically, you do not need to calculate it.

4. Frame Structure by Function Code

The request and response frames and an example are shown for each function code. All examples use **comm. address 1**.

■ 4.1 Read Coil Status (function code 01)

Reads the outputs (OUT1 · OUT2 · BATCH) and the reset status. (Coil)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	01H
Start	2	0000H~0004H (000001~000005)
Count	2	1~5
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	01H
Byte count	1	01H
Status	1	Status of the start register is bit0 (1 = ON)
CRC	2	

e.g.	Request: 01 | 01 | 00 01 | 00 03 | **2D CB**
	Response: 01 | 01 | 01 | 03 | **11 89**

※ Response status 03H = bit0 OUT2 ON, bit1 OUT1 ON, bit2 BATCH output OFF

■ 4.2 Read Input Status (function code 02)

Reads the terminal input status. (Input Status)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	02H
Start	2	0000H~0004H (100001~100005)
Count	2	1~5
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	02H
Byte count	1	01H
Status	1	Status of the start register is bit0 (1 = input active)
CRC	2	

e.g.	Request: 01 | 02 | 00 00 | 00 05 | **B8 09**
	Response: 01 | 02 | 01 | 01 | **60 48**

※ Response status 01H = INA input active, other inputs OFF

■ 4.3 Read Holding Registers (function code 03)

Reads setting values and function settings. (Holding Register)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	03H
Start	2	Holding Register Address (see 6.6 · 6.7)
Count	2	1~32
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	03H
Byte count	1	Count × 2
Data	Count × 2	High · low byte for each register
CRC	2	

e.g.	Request: 01 | 03 | 00 00 | 00 02 | **C4 0B**
	Response: 01 | 03 | 04 | 13 88 00 00 | **7E 9D**

※ PS2 (400001~400002) = 0000 1388H = 5000

■ 4.4 Read Input Registers (function code 04)

Reads the present value, monitoring data and product information. (Input Register)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	04H
Start	2	Input Register Address (see 6.4 · 6.5)
Count	2	1~32
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	04H
Byte count	1	Count × 2
Data	Count × 2	High · low byte for each register
CRC	2	

e.g.	Request: 01 | 04 | 03 EB | 00 02 | **01 BB**
	Response: 01 | 04 | 04 | E2 40 00 01 | **0D E8**

※ Present value (301004~301005) = 0001 E240H = 123456

■ 4.5 Force Single Coil (function code 05)

Executes RESET or BATCH RESET. (Coil)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	05H
Register	2	0000H: RESET 0004H: BATCH RESET
Data	2	FF00H: execute, 0000H: no action
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address ~ data	6	Same values as the request
CRC	2	

e.g.	Request: 01 | 05 | 00 00 | FF 00 | **8C 3A**
	Response: 01 | 05 | 00 00 | FF 00 | **8C 3A**

※ Executed even while comm. write protect (CPrt = ON) is set.

■ 4.6 Preset Single Register (function code 06)

Writes a value to one Holding Register.

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	06H
Register	2	Holding Register Address
Data	2	Value to write
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address ~ data	6	Same values as the request
CRC	2	

e.g.	Request: 01 | 06 | 00 35 | 00 01 | **58 04**
	Response: 01 | 06 | 00 35 | 00 01 | **58 04**

※ Changes the counter output mode (400054) to 1 (N). Because a function setting changes, the present value and outputs are reset.

■ 4.7 Preset Multiple Registers (function code 16, frame value 10H)

Writes values to several consecutive Holding Registers at once. (Used for 32-bit values)

Request (host → product)		
Field	**Size**	**Description**
Address	1	Comm. address
Function	1	10H
Start	2	Holding Register Address
Count	2	1~27
Byte count	1	Count × 2
Data	Count × 2	High · low byte for each register
CRC	2	

Response (product → host)		
Field	**Size**	**Description**
Address ~ count	6	Address | function code | start | count
CRC	2	

e.g.	Request: 01 | 10 | 00 00 | 00 02 | 04 | 13 88 00 00 | **77 01**
	Response: 01 | 10 | 00 00 | 00 02 | **41 C8**

※ Writes 5000 (0000 1388H) to PS2 (400001~400002). Always write the two registers of a 32-bit value together with function code 16.

5. Exception Responses

When the product receives a request it cannot process, it responds with the function code with its most significant bit set to 1 (**function code + 80H**) and an exception code.

Address	Function code + 80H	Exception code
1 byte	1 byte (e.g. 03H → 83H)	1 byte

Code	Name	Condition	Action
01	ILLEGAL FUNCTION	Unsupported function code	Use 01~06, 16
02	ILLEGAL DATA ADDRESS	Nonexistent register, write to a read-only register (400151~400156), range includes a register outside the read · write range, start + count out of range for 01 · 02, function 05 to a register other than RESET · BATCH RESET	Check the register
03	ILLEGAL DATA VALUE	Count exceeded, value out of the setting range, byte count mismatch, function 05 data other than FF00H · 0000H, write rule violation (see 7.2)	Check value · count
04	SLAVE DEVICE FAILURE	Write by 06 · 16 while comm. write protect (CPrt = ON) is set	Set CPrt to OFF
06	SLAVE DEVICE BUSY	Write by 06 · 16 while the setting mode or a setting value change is in progress on the front panel (returns automatically after 60 s without key input)	Send again after the front panel setting is finished

e.g.	Request: 01 | 10 | 00 00 | 00 02 | 04 | 13 88 00 00 | **77 01** (PS2 write while comm. write protected)
	Response: 01 | 90 | 04 | **4D C3** (90H = 10H + 80H, exception code 04)

※ If an exception occurs for a write request, none of the values in that request are applied.

6. Addresses (Mapping Table)

■ 6.1 Reading Register Numbers

This manual writes registers as **6-digit register numbers**, with the **Address** actually sent in the frame (hexadecimal) in brackets.

6-digit register number	**First digit** = register type (determines the function code to use) **Remaining 5 digits** = number (counted from 1)
Address	Value sent in the frame = **number – 1** (counted from 0)

Example	First digit → register type	Function code	Address (number – 1)	Registers of this product
000001	**0** → Coil (output · reset)	01 read 05 write	1 – 1 = 0000H	000001~000005
100001	**1** → Input Status (Discrete Inputs)	02 read	1 – 1 = 0000H	100001~100005
301004	**3** → Input Register (value read)	**04** read	**1004 – 1 = 1003 (03EBH)**	300101~300125 301001~301013
400001	**4** → Holding Register (settings)	03 read 06 · 16 write	1 – 1 = 0000H	400001~400006, 400051~400066, 400101~400110, 400151~400156, 400201~400206, 400211

※ **The first digit and the function code number are not the same.** (3 → function code **04**, 4 → function code **03** · 06 · 16) This is because the Modbus register notation convention and the function code numbers were defined separately.

※ The Address alone does not identify the type. (000001 · 100001 · 400001 are all Address 0000H) Always use it together with the function code.

Value to enter by register entry method (e.g. reading the present value 301004)

Register entry method of the host device	Value to enter	Remarks
6-digit register number (type + number from 1)	301004 (5-digit notation: 31004)	The first digit determines the function code
Function code + number from 1	04 + 1004	The host device subtracts 1 when sending
Function code + Address from 0	04 + 1003 (03EBH)	Same as the Address in the frame

※ The entry method differs **by model and program**, not by device type (HMI, PLC, PC program). Check the register entry method in the manual of your HMI, PLC or communication program.

※ Whichever method is used, the frame actually sent is the same: 01 | 04 | 03 EB | 00 02 | **01 BB**

※ Registers not in the tables respond with exception 02. When reading several registers at once, exception 02 is returned if the range includes a register that does not exist.

■ 6.2 RESET · Outputs

Register (Address)	Func	Name	Range · description
000001 (0000)	01/05	RESET	Read: 1 during reset, write: FF00H = present value · output reset
000002 (0001)	01	OUT2 output	0: OFF, 1: ON
000003 (0002)	01	OUT1 output	0: OFF, 1: ON
000004 (0003)	01	BATCH output	0: OFF, 1: ON
000005 (0004)	01/05	BATCH RESET	Read: always 0, write: FF00H = BATCH value · output reset

■ 6.3 Terminal Input Status

Register (Address)	Func	Name	Range · description
100001 (0000)	02	INA input	0: OFF, 1: active
100002 (0001)	02	INB input	0: OFF, 1: active
100003 (0002)	02	INHIBIT input	Shared INB/INHIBIT terminal (same as 100002)
100004 (0003)	02	RESET terminal	0: OFF, 1: active
100005 (0004)	02	BATCH RESET terminal	0: OFF, 1: active

※ Regardless of the NPN · PNP setting, the input status is **1 when the sensor is active (ON)**.

■ 6.4 Product Information

Register (Address)	Func	Name	Range · description
300101 (0064)	04	Serial number H	0000H
300102 (0065)	04	Serial number L	0001H
300103 (0066)	04	Hardware version	0100H (V1.00)
300104 (0067)	04	Firmware version	0300H (V3.00)
300105~108 (0068~006B)	04	Model name	4354H 4820H 2020H 2020H (ASCII "CTH")
300109~117 (006C~0074)	04	Reserved	0000H
300118 / 119 (0075 / 0076)	04	Coil start / count	0000H / 5
300120 / 121 (0077 / 0078)	04	Input start / count	0000H / 5
300122 / 123 (0079 / 007A)	04	Holding start / count	0000H / 00D3H
300124 / 125 (007B / 007C)	04	Input Register start / count	0064H / 0391H

■ 6.5 Monitoring Data

Register (Address)	Func	Name	Range · description
301001 (03E8)	04	LED status	See the LED status bit table below
301002~003 (03E9~03EA)	04	BATCH value	0~999999 (32 bit)
301004~005 (03EB~03EC)	04	Present value	Counter: -99999~999999, timer: elapsed time (32 bit)
301006 (03ED)	04	Display unit	Counter: decimal places (0~3), timer: time range code (same as 400102)
301007~008 (03EE~03EF)	04	PS2 setting value	Value of the mode in operation (same as 400001~400002)
301009~010 (03F0~03F1)	04	PS1 setting value	Value of the mode in operation (same as 400003~400004)
301011~012 (03F2~03F3)	04	BATCH setting value	Same as 400005~400006
301013 (03F4)	04	Input logic	0: NPN, 1: PNP (N/P switch status)

301001	LED status bits (1 = on)
bit	5: BA.O 6: OUT2 7: OUT1 10: BA.S 11: LOCK 12: PS2 13: PS1 14: TMR 15: CNT (other bits: 0)

※ The TMR bit follows the indicator. (While the timer is running it changes between 0 and 1 as the indicator flashes)

■ 6.6 Setting Values (Saved / Not Saved)

Register (Address)	Func	Name	Range · description
400001~002 (0000~0001)	03/06/16	PS2 setting value (saved)	0~999999 (32 bit, saved 3 s after the first write)
400003~004 (0002~0003)	03/06/16	PS1 setting value (saved)	0~999999 (32 bit, saved 3 s after the first write)
400005~006 (0004~0005)	03/06/16	BATCH setting value (saved)	0~999999 (32 bit, saved 3 s after the first write)
400201~202 (00C8~00C9)	03/06/16	PS2 setting value (not saved)	0~999999 (32 bit, for frequent changes)
400203~204 (00CA~00CB)	03/06/16	PS1 setting value (not saved)	0~999999 (32 bit, for frequent changes)
400205~206 (00CC~00CD)	03/06/16	BATCH setting value (not saved)	0~999999 (32 bit, for frequent changes)
400211 (00D2)	03/06/16	Time calibration (factory)	–1000~1000 ppm, factory calibration value (do not change)

※ PS1 · PS2 · BATCH (400001~, 400201~, 301007~) are the values of **the mode in operation (counter or timer)**.

※ Values written to the not-saved registers return to the last saved values at the next power-on.

■ 6.7 Function Settings

Register (Address)	Func	Name	Range · description
Counter function settings			
400051 (0032)	03/06/16	Counter/Timer	0: counter, 1: timer
400052 (0033)	03/06/16	Input mode	1: UP-1, 0: UP-3, 6: UP/DOWN-1, 7: UP/DOWN-3, 8: ENC-1 (x1), 9: ENC-2 (x4)
400053 (0034)	03/06/16	Display mode	Not supported (reads 0, only 0 can be written)
400054 (0035)	03/06/16	Output mode	0: F, 1: N, 2: C, 3: R
400055 (0036)	03/06/16	Counting speed	0: 1 cps, 1: 30 cps, 2: 1 kcps, 4: 10 kcps, 5: 300 kcps (ENC only, front display 300K), 3 is not used
400056 (0037)	03/06/16	OUT2 output time	0: HOLD, 1~9999 (×10 ms = 0.01~99.99 s)
400057 (0038)	03/06/16	OUT1 output time	0: HOLD, 1~9999 (×10 ms = 0.01~99.99 s)
400058 (0039)	03/06/16	Decimal point	0~3 (digits after the decimal point)
400059 (003A)	03/06/16	RESET time	0: 1 ms, 1: 20 ms
400060 (003B)	03/06/16	Prescale decimal point	Fixed at 3 (only 3 can be written)
400061~062 (003C~003D)	03/06/16	Prescale	1~999999 (= 0.001~999.999, 32 bit)
400063~066 (003E~0041)	03/06/16	Start point, memory, key LOCK	Not supported (reads 0, only 0 can be written)

Register (Address)	Func	Name	Range · description
Timer function settings			
400101 (0064)	03/06/16	Counter/Timer	0: counter, 1: timer (same as 400051)
400102 (0065)	03/06/16	Time range	1: 0.01 s, 2: 0.1 s, 3: 1 s, 8: 1 m
400103 (0066)	03/06/16	UP/DOWN	Fixed at 0 (UP) (only 0 can be written)
400104 (0067)	03/06/16	Output mode	0: OND, 3: FLK, 6: INT, 9: OFD
400105 (0068)	03/06/16	OUT2 output time	0: HOLD, 1~9999 (×10 ms = 0.01~99.99 s)
400106 (0069)	03/06/16	OUT1 output time	0: HOLD, 1~9999 (×10 ms = 0.01~99.99 s)
400107 (006A)	03/06/16	Input signal time	0: 1 ms, 1: 20 ms
400108~110 (006B~006D)	03/06/16	Memory, key LOCK, display mode	Not supported (reads 0, only 0 can be written)

Register (Address)	Func	Name	Range · description
Communication settings (read only - changed only in the front panel setting mode)			
400151 (0096)	03	Comm. address	1~127
400152 (0097)	03	Baud rate	2: 9600, 3: 19200, 4: 38400, 5: 115200 bps
400153 (0098)	03	Parity	0: None (fixed)
400154 (0099)	03	Stop bits	0: 1 bit (fixed)
400155 (009A)	03	Response delay	5 (fixed, for compatibility · the product responds right after receiving)
400156 (009B)	03	Comm. write protect	0: OFF (write allowed), 1: ON (write protected)

7. Data Format and Write Rules

■ 7.1 Data Format

- **32-bit values** (present value, PS1, PS2, BATCH, prescale) consist of 2 registers: **the first register is the low 16 bits** and the next register is the high 16 bits.
  e.g.) 123456 = 0001 E240H → 301004 = E240H, 301005 = 0001H
- **Negative values** are in two's complement.
  e.g.) -100 = FFFF FF9CH (low FF9CH, high FFFFH)
- Values with a **decimal point** are read and written as integers without the decimal point.
  e.g.) display with 1 decimal place 123.4 = 1234, 12.34 s in the timer 0.01 s range = 1234
- **Output time** is in units of 10 ms.
  e.g.) 0.50 s = 50, 0 = HOLD

■ 7.2 Write Rules

- Multiple values of function code 16 are **all checked first and then applied at once**. If any value is invalid, none of them are applied.
- Write the two registers of a 32-bit value **together with function code 16**. If only one register is written, the other 16 bits keep their previous value.
- **Saved registers** (400001~, function settings) are stored in non-volatile memory **3 s after the first write**; writes within those 3 s are stored together.
- For setting values that change periodically, use the **400201~ (not saved)** registers. They return to the last saved value at the next power-on. Writing repeatedly to the saved registers shortens the memory life.
- **Changing a function setting** (when the value actually changes) **resets the present value and outputs**, as with the front panel setting.
- Changing PS1 · PS2 during operation works as with the front panel setting. (If already reached or the output is ON, it resets; if not yet reached, operation continues with the new value. In timer FLK mode it does not reset and continues with the new value)

The following values cannot be written; the product responds with **exception 03**.

Condition	Values that cannot be written
Counter output mode C · R	PS2 = 0
Timer output mode FLK · INT · OFD	PS1 = 0 or PS2 = 0
Input mode other than ENC	Counting speed 5 (300 kcps)
ENC input mode	Counting speed 0 · 1 · 2 · 4 (only 5 is possible in ENC input mode)
Setting values · prescale	Values outside the setting value range (0~999999) or prescale range (1~999999)

The following values are written normally but are **changed automatically when stored**. Reading them back after writing may give a different value. (Same rules as the front panel setting)

Condition	Value changed automatically
Counter output mode F · N	OUT2 output time (400056) → HOLD
Counter output mode C · R	If the OUT2 output time is HOLD (0) → 50 (0.50 s) If PS2 = 0 when changing to C · R → PS2 = 5000
Timer output mode FLK · INT · OFD	OUT2 · OUT1 output time (400105 · 400106) → HOLD When changing to these modes: PS1 = 0 → 1000, PS2 = 0 → 5000
Timer time range (400102) changed	PS1 → 1000, PS2 → 5000
Input mode (400052) changed to ENC	Counting speed (400055) → 5 (300 kcps)
Input mode changed from ENC to another mode	If the counting speed is 5 → 1 (30 cps)

※ To change the time range and PS1 · PS2 together, write the time range first, then write PS1 · PS2 in the next request.

8. Communication Examples

Comm. address 1, hexadecimal. | : field separator (address | function code | register | count | byte count | data | CRC, depends on the function code), **bold**: CRC

■ 8.1 Read

Function	Request (host → product)	Response (product → host)	Remarks
Read present value	01 | 04 | 03 EB | 00 02 | **01 BB**	01 | 04 | 04 | E2 40 00 01 | **0D E8**	123456 (0001 E240H)
Read BATCH value	01 | 04 | 03 E9 | 00 02 | **A0 7B**	01 | 04 | 04 | 00 64 00 00 | **BA 5B**	100 (0064H)
Read LED status	01 | 04 | 03 E8 | 00 01 | **B1 BA**	01 | 04 | 02 | A0 C0 | **C1 60**	A0C0H = CNT, PS1, OUT1, OUT2 on
Read output status	01 | 01 | 00 01 | 00 03 | **2D CB**	01 | 01 | 01 | 03 | **11 89**	OUT2 ON, OUT1 ON, BATCH OFF
Read input status	01 | 02 | 00 00 | 00 05 | **B8 09**	01 | 02 | 01 | 01 | **60 48**	INA active

Function	Request (host → product)	Response (product → host)	Remarks
Read setting values (PS2 · PS1 · BATCH)	01 | 03 | 00 00 | 00 06 | **C5 C8**	01 | 03 | 0C | 13 88 00 00 03 E8 00 00 00 0A 00 00 | **DD 30**	PS2 5000, PS1 1000, BATCH 10
Read product info (300101~300108)	01 | 04 | 00 64 | 00 08 | **B0 13**	01 | 04 | 10 | 00 00 00 01 01 00 03 00 43 54 48 20 20 20 20 20 | **97 DC**	Firmware 0300H, model name "CTH"
Read counter function settings (400051~400062)	01 | 03 | 00 32 | 00 0C | **E4 00**	01 | 03 | 18 | 00 00 00 01 00 00 00 00 00 01 00 00 00 0A 00 00 00 01 00 03 03 E8 00 00 | **48 4D**	Counter, UP-1, F, 30 cps, OUT2 HOLD, OUT1 0.10 s, decimal point 0, 20 ms, prescale 1.000 (defaults)

■ 8.2 Write · Reset

Function	Request (host → product)	Response (product → host)	Remarks
Change PS2 (saved) = 5000	01 | 10 | 00 00 | 00 02 | 04 | 13 88 00 00 | **77 01**	01 | 10 | 00 00 | 00 02 | **41 C8**	Saved 3 s after the first write
Change PS1 (saved) = 1000	01 | 10 | 00 02 | 00 02 | 04 | 03 E8 00 00 | **F2 06**	01 | 10 | 00 02 | 00 02 | **E0 08**	Saved 3 s after the first write
Change PS2 (not saved) = 5000	01 | 10 | 00 C8 | 00 02 | 04 | 13 88 00 00 | **7A F7**	01 | 10 | 00 C8 | 00 02 | **C0 36**	Returns to the saved value at the next power-on
Change PS1 · PS2 together (not saved)	01 | 10 | 00 C8 | 00 04 | 08 | 13 88 00 00 03 E8 00 00 | **9D 85**	01 | 10 | 00 C8 | 00 04 | **40 34**	PS2 5000, PS1 1000
Present value reset	01 | 05 | 00 00 | FF 00 | **8C 3A**	01 | 05 | 00 00 | FF 00 | **8C 3A**	Present value · output reset
BATCH reset	01 | 05 | 00 04 | FF 00 | **CD FB**	01 | 05 | 00 04 | FF 00 | **CD FB**	BATCH value · output reset
Change output mode = N	01 | 06 | 00 35 | 00 01 | **58 04**	01 | 06 | 00 35 | 00 01 | **58 04**	Present value · output reset
Change prescale = 0.500	01 | 10 | 00 3C | 00 02 | 04 | 01 F4 00 00 | **B0 E0**	01 | 10 | 00 3C | 00 02 | **81 C4**	500 = 01F4H Present value · output reset

■ 8.3 Exception Responses

Function	Request (host → product)	Response (product → host)	Remarks
Write comm. setting	01 | 06 | 00 96 | 00 02 | **E8 27**	01 | 86 | 02 | **C3 A1**	Read-only register → exception 02 (exception 04 if CPrt = ON)
PS2 = 0 in C mode	01 | 10 | 00 00 | 00 02 | 04 | 00 00 00 00 | **F3 AF**	01 | 90 | 03 | **0C 01**	Write rule violation → exception 03
Read 33 registers	01 | 03 | 00 00 | 00 21 | **85 D2**	01 | 83 | 03 | **01 31**	Count exceeded → exception 03
PS2 write while write protected	01 | 10 | 00 00 | 00 02 | 04 | 13 88 00 00 | **77 01**	01 | 90 | 04 | **4D C3**	CPrt = ON → exception 04

9. Troubleshooting

Symptom	Check
No response	RXD · TXD · GND wiring (connect same names) Baud rate · address of the product and host device (bPS, Adr), 8 data bits · parity None · 1 stop bit CRC calculation and byte order (CRC low byte first), use of address 0 (broadcast)
Occasional no response	Response timeout of the host device (100 ms or more), request interval (send after receiving the response) Cable length (15 m or less) and separation from power lines
Wrong values	Register numbering (from 0 / from 1), word order of 32-bit values (first register = low), decimal point · unit
Exception 02	Check the register, whether the range includes registers outside the read · write range (start + count for 01 · 02), writes to communication setting registers (read only)
Exception 03	Count (01 · 02: 5, 03 · 04: 32, 16: 27 or less), setting range, write rules (see 7.2)
Exception 04 / 06	Check the CPrt setting in setting mode / send again after the front panel setting is finished