TOHO ELECTRONICS INC.

Operation Manual, Communications
(TOHO protocol and MODBUS)

Model: TTX-700
Designation: Module Controller

Contents

1. Before using the product

1.1 On this operation manual

This is an operation manual regarding communications with a TTX-700 (hereinafter referred to as "this product").

1.2 What can be done with communications

With this product, users can write and read items specified in "10. Table of identifiers," such as "reconfiguring, starting, or stopping items that are operable with the front keys" and "reading information displayable on the display."

However, reading and writing with ordinary commands are performed with regarding to the RAM in this product. Written data can be turned back into the values before the writing (the values stored on the EEPROM) by turning power off and on again. To store the written data on the EEPROM of this product, execute a store request message. (See "3.6. Communications precautions.")

Settings regarding options not added and other unnecessary settings cannot be read or written.

1.3 Positioning communications (priority ranking)

Data and parameters in this product can be changed with keys while in operation in the communications mode.

While this product is in operation in the RO (read-only) mode, no data or parameter setting can be changed by communications. (Provided that communications modes can be changed.)

1.4 Setting before communications

Before performing communications, this product must be set.

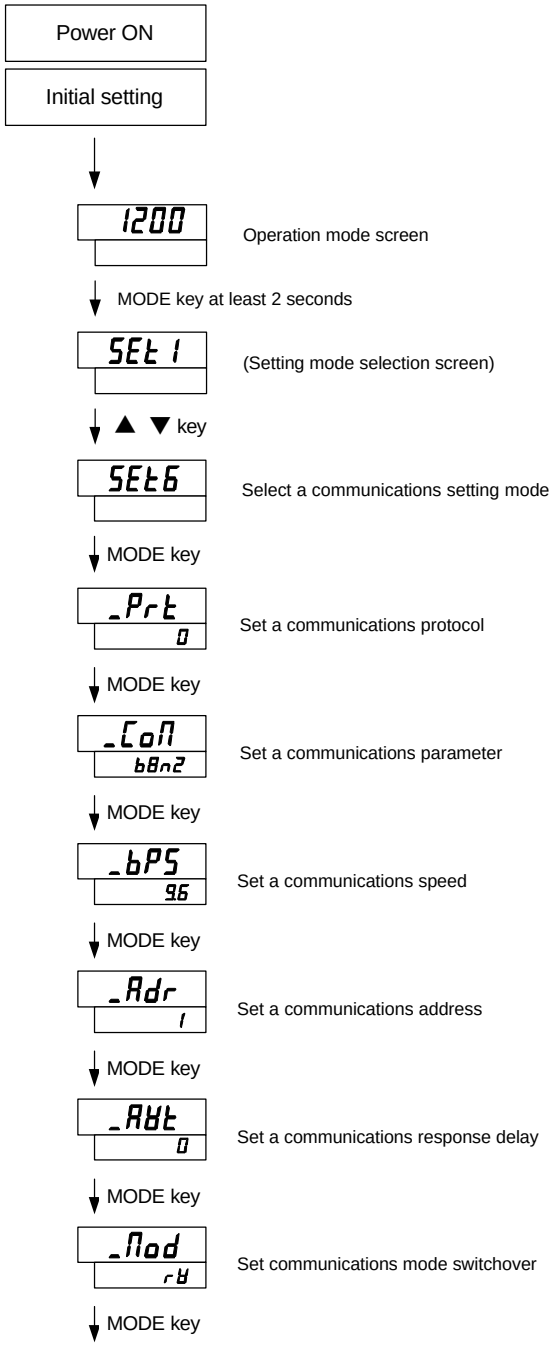
This product is compatible with the TOHO communications protocol and the MODBUS communications (RTU, ASCII).

Select a protocol with the protocol setting (-Prt) on communications 1/2 setting (SEtG). For the TOHO communications protocol, see "2. Settings regarding TOHO communications." For the MODBUS communications protocol, see "5. Settings regarding MODBUS communications."

2. Settings regarding TOHO communications

2.1 Overview

Before communications is performed, initial settings must be made on this product. Enter such settings with the keys on the front panel. To switch to a series of setting screens, take the steps described below. For details, see the operation manual furnished with this product.



When the settings are over, press the MODE key at least 2 seconds to go back to the operation mode. The parameters indicated above are initial values.

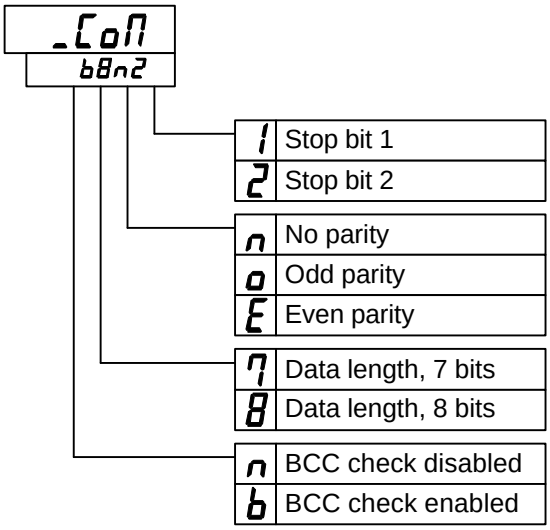
2.2 Setting a data length

2.3 Setting a stop bit length

2.4 Setting a parity

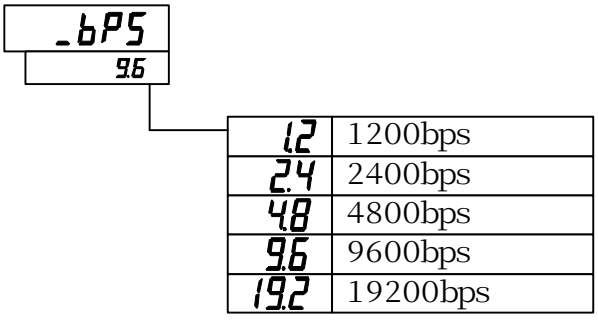
2.5 Setting whether to conduct a BCC check

While in the "Set a communications parameter" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.
The initial value is [8n2].



2.6 Setting a communications speed

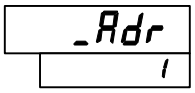
While in the "Set a communications speed" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.
The initial value is [9.6].



2.7 Setting an address

While in the "Set a communications address" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.

The initial value is •.



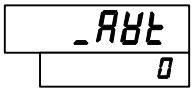
Setting range: 1 to 99 stations (It cannot be set to a 0.)

2.8 Setting a response delay

Set a time from the time when the high-level computer finished sending a "request message" until the time when it delivers the line and enters an input state.

While in the "Set a response delay" on the preceding page, operate the ▲ and ▼ keys to make the settings.

The initial value is 0.

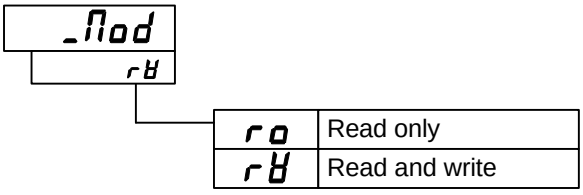


Setting range: 0 to 250ms

* If the response delay is set to a short setting, the communications may not be conducted normally.
* In a real operation, the processing time for this product will be added, in addition to the response delay.

2.9 Switching communications mode

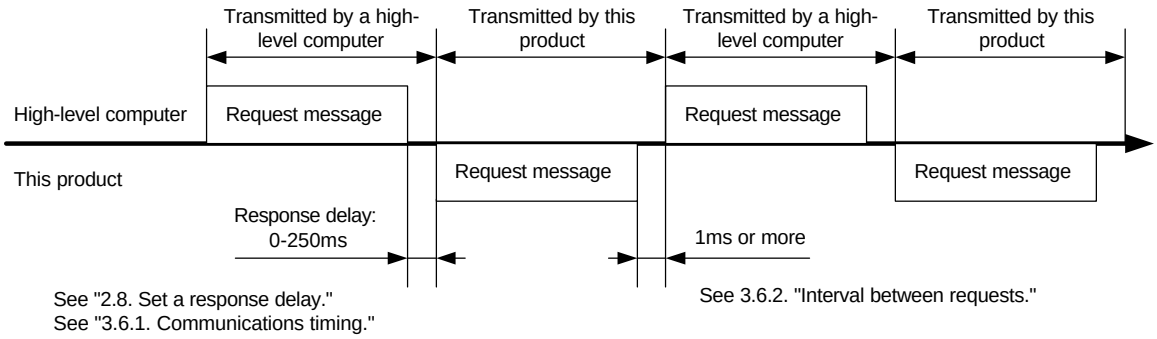
While in the "Set communications mode switchover" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.



3. TOHO communications control

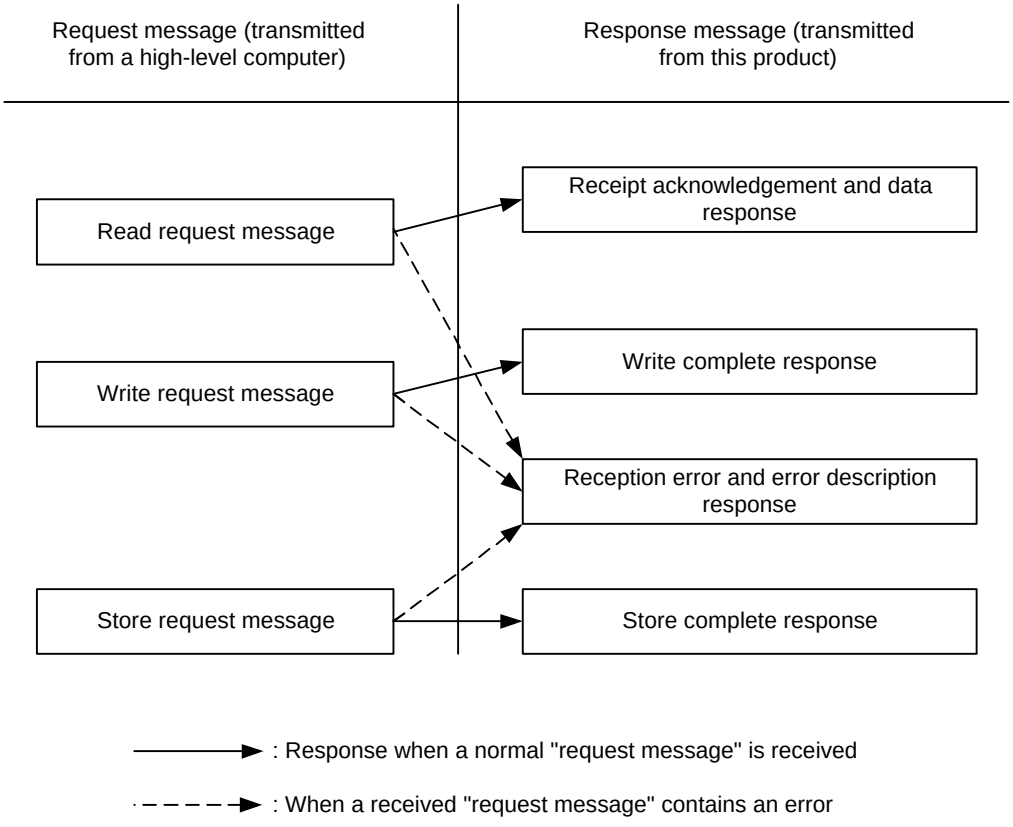
3.1 Communications procedure

This product returns a "response message" in response to a "request message" from a high-level computer. It therefore does not initiate a transmission.



3.2 Message types

- Messages are roughly divided into the following types:

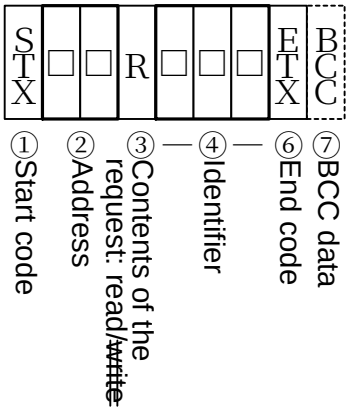


- All codes (except for BCC) from STX data to ETX are expressed in ASCII codes.
- In assembling a program for a high-level computer, see "10. Table of identifiers (codes)" and "9. Table of ASCII codes" at the end of the book.

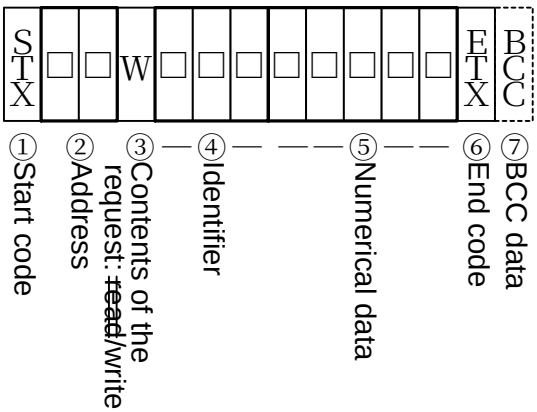
3.3 Composition of a request message (transmitted from a high-level computer to this product)

- For codes ① to ⑩, see "3.5. Description of codes."
- For specific examples of request messages, see "4.1. Examples of reading communications" and "4.2. Examples of writing communications."

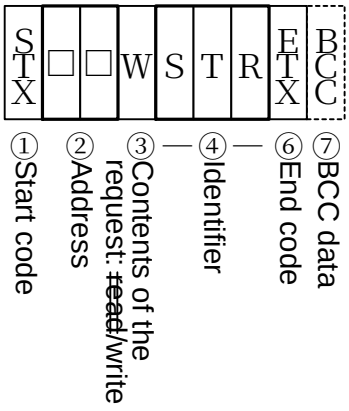
3.3.1 Composition of a read request message



3.3.2 Composition of a write request message



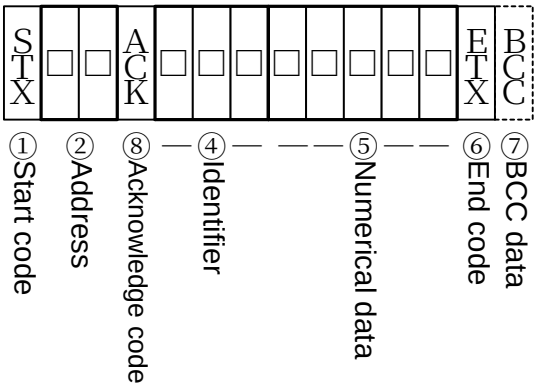
3.3.3 Composition of a store request message



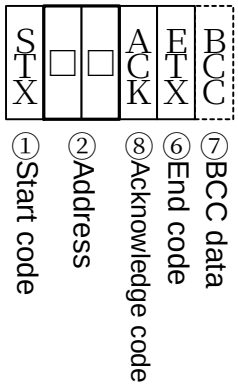
3.4 Composition of a response message (transmitted from this product to a high-level computer)

- For codes ① through ⑩, see "3.5. Description of codes."
- For specific examples of request messages, see "4.1. Examples of communications to be read" and "4.2. Examples of communications to be written."

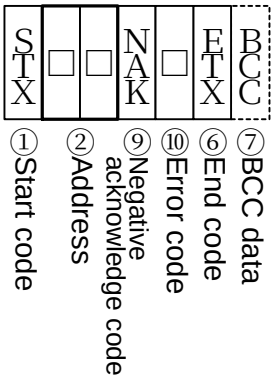
3.4.1 Response message in response to a read request message



3.4.2 Response message in response to a write/store request message



3.4.3 Response message in the case of an error



3.5 Description of codes

- The codes from ① STX, ② address to ⑩ ERR as indicated below are expressed in ASCII codes.
- For the ASCII codes, see "9. Table of ASCII codes."
- For conversion to ASCII codes, see "4. Examples of TOHO communications."

① STX

This code is needed for the receiver to detect the top of the message. It is affixed to the top of a character string to be sent.

② Address

This is the address of the party (this product) with whom a high-level computer communicates. The address in the response message from this product indicates the sender of the response message. Note that, when CH2 is used, two addresses are occupied. (Setting ADR to 1 causes this product to occupy addresses 1 and 2.)

③ Contents requested

Enter a code R or W.

R: to read data from this product

W: to write or store data in this product

④ Identifier

An identifier is a classification code (identifier) for data to be read or written and expressed in a three-digit alphanumerical ASCII code. See "10. Table of identifiers (codes)."

⑤ Numerical data

At the time of writing or reading, 5-digit numerical data can be written.

Example: The table below indicates the significances of 5-digit numerical data 00010.

Example	Significance of the value
Proportional band (P)	→1.0%
Data (PV), etc, whose decimal point can be shifted	
When the decimal point setting (DP) is 0	10
When the decimal point setting (DP) is 0.1	1.0

If DP = 0.1, the numerical data "12000" means 1200.0. In the case of text data, it is "□□INP." (The □ is a space.)

⑥ ETX

This code is needed for the receiver to detect the end of a message. It is affixed to the end of a character string to be sent (except for BCC).

⑦ BCC

This is a check code for error detection and is the exclusive OR (EX-OR) of all characters from STX to ETX.

If the BCC check is set to "Disabled" in the communications settings in this product, this code (BCC) will not be incorporated in the response message. See "2. Settings regarding TOHO communications."

⑧ ACK

It is an acknowledge code. If a message received by this product is error-free, this code will be incorporated in the "response message" from this product and returned.

⑨ NAK

It is a negative acknowledge code. If a "request message" received by this product is error-ridden, this code will be incorporated in the "response message" from this product and returned.

If the "request message" received is error-ridden, the error contents (⑩ ERR type) will be incorporated in the "response message" from this product, following NAK.

⑩ ERR type

If a "request message" received from this product is error-ridden, the error contents (either of the numbers in the table below) will be incorporated in the "response message" from this product, following "⑨ NAK."

The error number 0 is an instrument error (memory error or A/D conversion error). It will be incorporated in the "response message" regardless of whether there is an error in the "request message."

The error number 9 is an AT error. It will therefore be incorporated in the "response message" regardless of whether there is an error in the "request message." Remove the cause of the error immediately and start the AT again.

If there are two or more errors occurring at the same time, the largest error number will be incorporated.

The table below indicates the error contents and classifications.

Error No.	Error contents in the "request message" received by this product
0	Instrument error (memory error or A/D conversion error)
1	The numerical data deviated from the "range of settings designated specifically with setting items."
2	The change of requested items is disabled or there are no items to be read.
3	An ASCII code other than the numerical data was specified in the field of numerical data. An ASCII code other than numbers and "-" was specified in the field of codes.
4	Format error
5	BCC error
6	Overrun error
7	Framing error
8	Parity error
9	A PV error occurred during AT. Or AT will not end 3 hours later.

3.6 Communications precautions

3.6.1 Communications timing

Set a sufficient response delay to make sure that this product is switched over from transmission to reception with regard to a high-level computer in using an RS-485.

See the figure in "3.1. Communications procedure" and "2.8. Setting a response delay."

3.6.2 Interval between requests

In transmitting a series of "request messages" from a high-level computer, allow for an interval of 1msec or more from the reception of a "response message" from this product to a next transmission.

3.6.3 Response conditions

This product will not return a "response message" unless it receives a "request message" containing an STX and ETX (BCC).

If, therefore, the "request message" is error-ridden, this product will not return a "response message" (error reply) containing a NAK and ERR unless the conditions mentioned above are met.

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

The moment this product receives an STX, it clears all codes received before that.

3.6.4 Errors in address specification

This product will not respond to any "request message" that specifies an address other than that specified for itself. If, therefore, the address portion of a "request message" is error-ridden, none of the mobile units will return a "response message."

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

The moment this product receives an STX, it clears all codes received before that.

3.6.5 Number of digits in data and the decimal position

See "3.5. Description of codes, ⑤ Numerical data."

3.6.6 Operation after receiving a store request message

This product starts to store data after correctly receiving a store request message from a high-level computer.

This product only stores data different from the contents of the EEPROM (data that is changed). The time (TW) required for storing data is within 6 seconds.

This product transmits a storage-complete reply (ACK) when the data is stored.

This product will not guarantee that the data is stored if this product is turned off during a storage operation. Do not turn off this product for 6 seconds after transmitting a store request message.

3.6.7 Operation after turning on the power

This product will not perform communications (no response) for about 4 seconds after it is turned on. Allow for a delay until communications is started after this product is turned on.

3.6.8 Storing data other than a store request message

Store all parameters in the EEPROM in either of the two cases described below, even if no store request message is received.

1) If a parameter is changed by key operation
2) If auto-tuning is started and ends normally.

3.6.9 Changing the settings (SV or SV2) by communications during auto-tuning

Even if the settings (SV or SV2) used in control for auto-tuning are changed by communications, the settings (SV or SV2) will not be changed until the auto-tuning ends.

4. Examples of TOHO communications

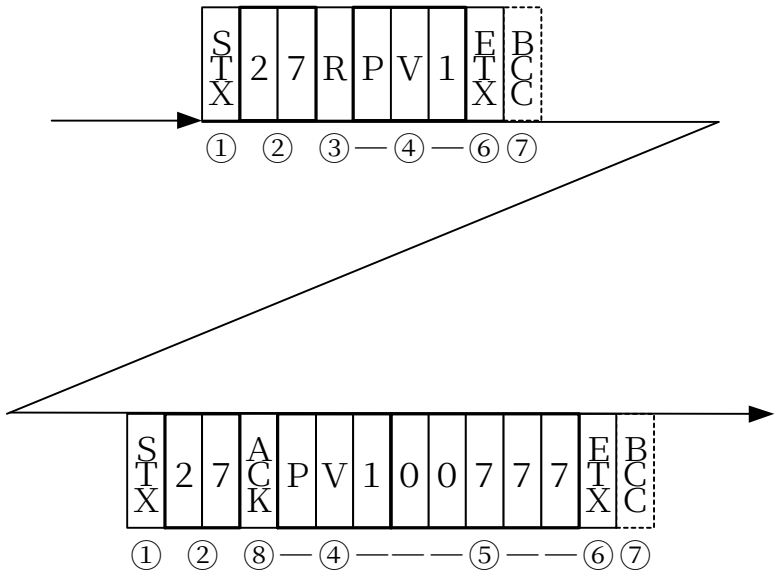
4.1 Examples of communications to be read

Example: Request message: This requests this product set at address 27 to read the PV. (High-level computer)

In response to that,

Response message: This returns PV data (00777).

Read request message (transmitted from the high-level computer)



Code	Code, data	ASCII code, note 2)
① Start code	STX	02H
② Address	27	32H 37H
③ Request contents	R (Read)	52H
④ Identifier, note 1)	PV1	50H 56H 31H
⑤ Numerical data	00777	30H 30H 37H 37H 37H
⑥ End code	ETX	03H
⑦ BCC data request		61H
response		02H
⑧ Acknowledge code	ACK	06H

Note 1): See "10. Table of identifiers (codes)."
Note 2): For the ASCII codes, see "9. Table of ASCII codes."

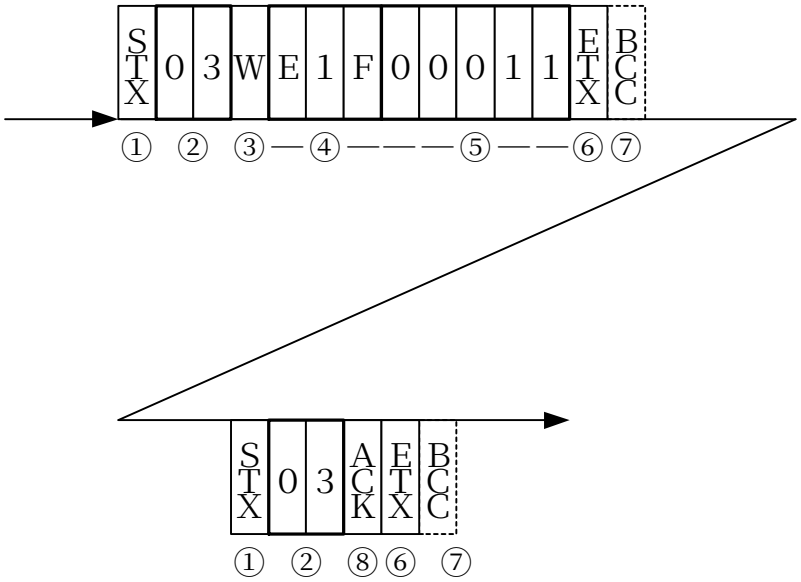
4.2 Examples of communications to be written

Example: Request message: (High-level computer) This requests this product set at address 03 to set "the E1F setting to 011" (write 011).
(This sets the function in event 1 to the deviation upper and lower limits + hold.)

In response to that,

Response message: (This product) This returns a notice that the request message has been received.
*Check that it has been written by reading the data separately.

Write request message (transmitted from a high-level computer)



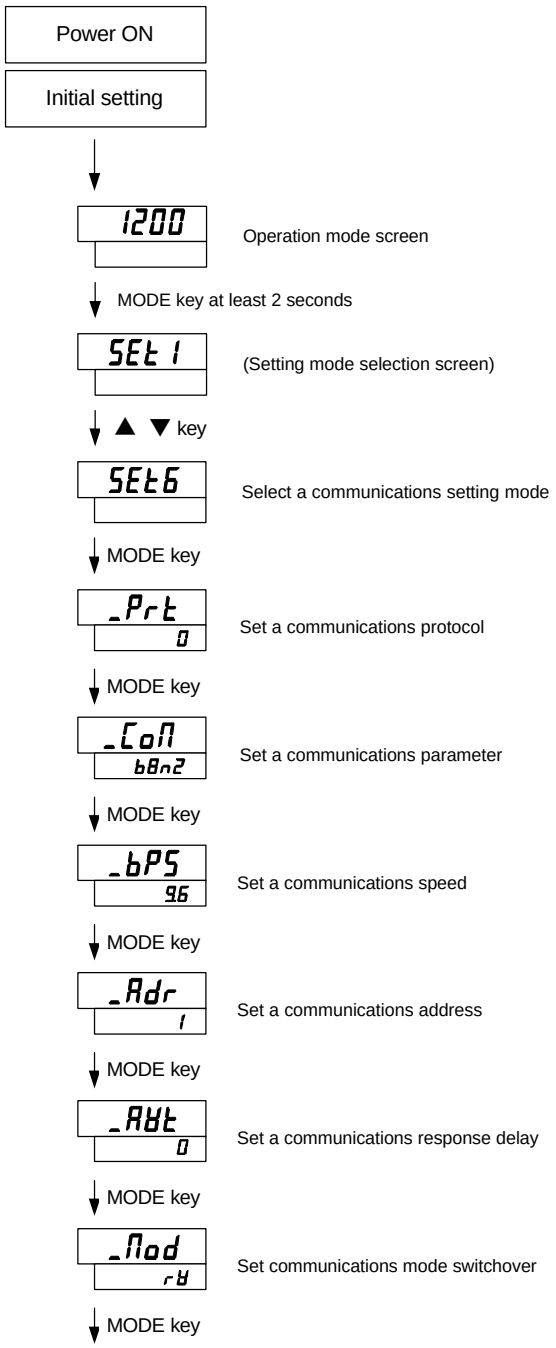
Code	Code, data	ASCII code, note 2)
① Start code	STX	02H
② Address	03	30H 33H
③ Request contents	W (Write)	57H
④ Identifier, note 1)	E1F	41H 34H 46H
⑤ Numerical data	00011	30H 30H 30H 31H 31H
⑥ End code	ETX	03H
⑦ BCC data request		57H
response		04H
⑧ Acknowledge code	ACK	06H

Note 1): See "10. Table of identifiers (codes)."
Note 2): For the ASCII codes, see "9. Table of ASCII codes."

5. Settings regarding MODBUS communications

5.1 Overview

Before communications is performed, initial settings must be made on this product. Enter such settings with the keys on the front panel. To switch to a series of setting screens, take the steps described below. For details, see the operation manual furnished with this product.



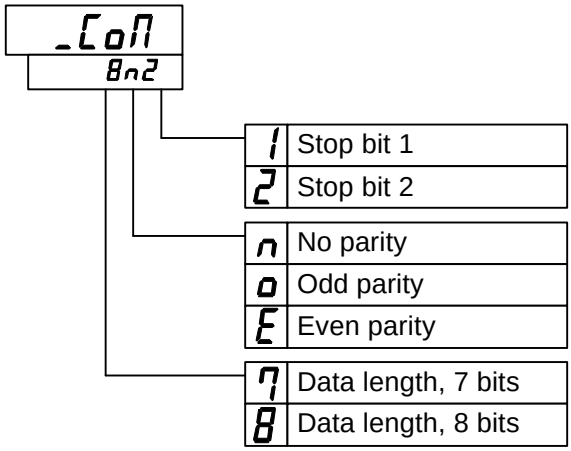
When the settings are over, press the MODE key at least 2 seconds to go back to the operation mode. The parameters indicated above are initial values.

5.2 Setting a data length

5.3 Setting a stop bit length

5.4 Setting a parity

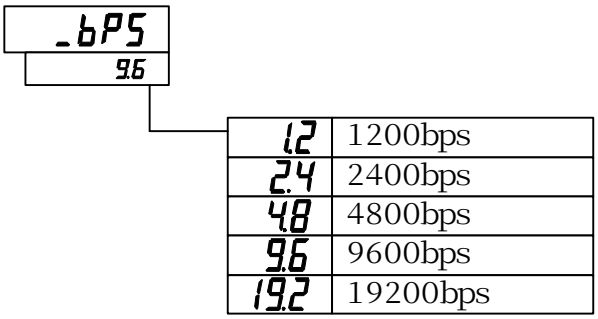
While in the "Set a communications parameter" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.
The initial value is [8n2].



* The ASCII mode settings come only in three types: 7n2, 7o1, and 7E1.
The RTU mode settings come only in three types: 8n2, 8o1, and 8E1.

5.5 Setting a communications speed

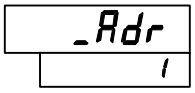
While in the "Set a communications speed" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.
The initial value is [9.6].



5.6 Setting an address

While in the "Set a communications address" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.

The initial value is •.



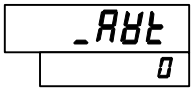
Setting range: 1 to 247 stations (It cannot be set to a 0.)

5.7 Setting a response delay

Set a time from the time when the high-level computer finished sending a "request message" until the time when it delivers the line and enters an input state.

While in the "Set a response delay" on the preceding page, operate the ▲ and ▼ keys to make the settings.

The initial value is 0.

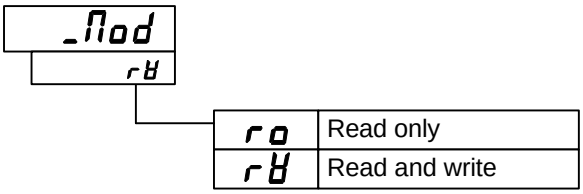


Setting range: 0 to 250ms

* If the response delay is set to a short setting, the communications may not be conducted normally.
* In a real operation, the processing time for this product will be added, in addition to the response delay.

5.8 Switching communications mode

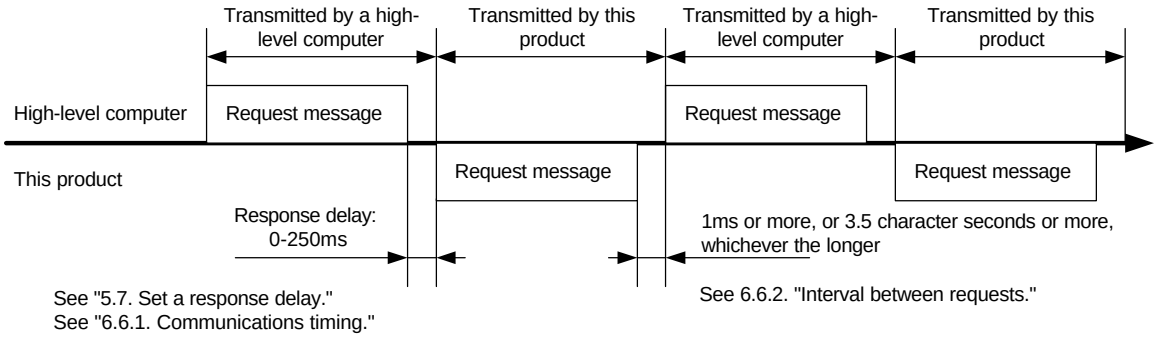
While in the "Set communications mode switchover" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.



6. MODBUS communications control

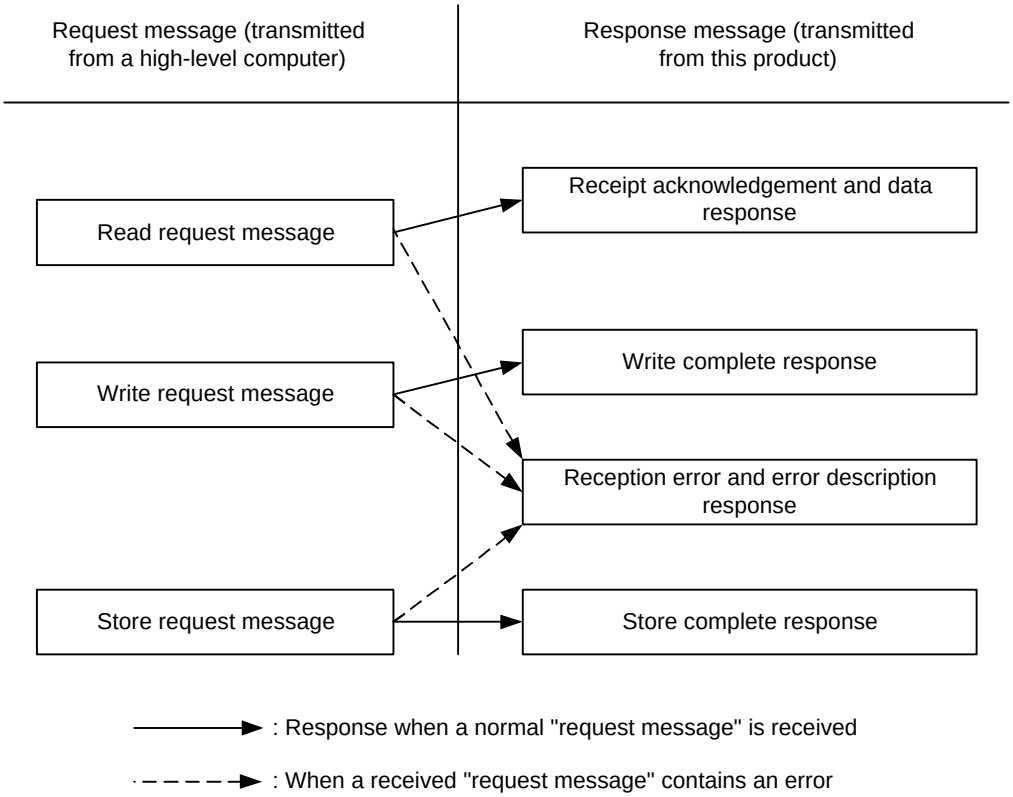
6.1 Communications procedure

This product returns a "response message" in response to a "request message" from a high-level computer. It therefore does not initiate a transmission.



6.2 Message types

- Messages are roughly divided into the following types:



- In RTU codes, the data is binary.
- In ASCII codes, all codes are expressed in ASCII codes.
- In assembling a program for a high-level computer, see "10. Table of identifiers (codes)" and "9. Table of ASCII codes" at the end of the book.

6.3 Composition of an RTU request message (transmitted from a high-level computer to this product)

■ For codes a) through i), see "6.5. Description of RTU codes."

6.3.1 Composition of a read request message

a)	Slave address		1BH	
b)	Function code		03H	
c)	Register address	High level	00H	First register address
		Low level	00H	
d)	Number of registers	High level	00H	2 to 100 (even number only)
		Low level	02H	
e)	CRC-16	High level	C6H	
		Low level	31H	

6.3.2 Composition of a write request message

Multiple

a)	Slave address		03H	
b)	Function code		10H	
c)	Register address	High level	00H	First register address
		Low level	C0H	
d)	Number of registers	High level	00H	Fixed at 2
		Low level	02H	
f)	Number of data items		04H	Number of registers × 2
g)	Data for the first register (a low-level word)	High level	00H	③ When writing ①, ②, ③, and ④ H in the data, write them in the order described on the left-hand side. (① represents 1 byte.)
		Low level	6FH	④
	Data for the first register + 1 (a high-level word)	High level	00H	①
		Low level	00H	②
e)	CRC-16	High level	C4H	
		Low level	5AH	

Single

a)	Slave address		03H	
b)	Function code		06H	
c)	Register address	High level	00H	First register address
		Low level	C0H	
g)	Data for the first register (a low-level word)	High level	00H	① When writing ① and ② H in the data, write them in the order described on the left-hand side. (① represents 1 byte.)
		Low level	6FH	②
e)	CRC-16	High level	C4H	
		Low level	5AH	

6.3.3 Composition of a store request message

a)	Slave address		03H	
b)	Function code		10H	
c)	Register address	High level	02H	First register address
		Low level	0EH	
d)	Number of registers	High level	00H	Fixed at 2
		Low level	02H	
f)	Number of data items		04H	Number of registers × 2
g)	Data for the first register (a low-level word)	High level	00H	The data about the storage of settings is arbitrary.
		Low level	00H	
	Data for the first register + 1 (a high-level word)	High level	00H	
		Low level	00H	
e)	CRC-16	High level	60H	
		Low level	FBH	

6.4 Composition of an RTU response message (transmitted from this product to a high-level computer)

■ For codes a) through h), see "6.5. Description of RTU codes."

6.4.1 Response message for a read request message

a)	Slave address		1BH	
b)	Function code		03H	
d)	Number of data items		04H	Number of registers × 2
g)	Data for the first register (a low-level word)	High level	03H	
		Low level	09H	
	Data for the first register + 1 (a high-level word)	High level	00H	
		Low level	00H	
e)	CRC-16	High level	91H	
		Low level	B4H	

6.4.2 Response message for a write/store request message

Multiple

a)	Slave address		03H	
b)	Function code		10H	
c)	Register address	High level	00H	First register address
		Low level	00H	
d)	Number of registers	High level	00H	Fixed at 2
		Low level	02H	
e)	CRC-16	High level	40H	
		Low level	2AH	

Single

a)	Slave address		03H	
b)	Function code		06H	
c)	Register address	High level	00H	First register address
		Low level	C0H	
g)	Data for the first register (a low-level word)	High level	00H	
		Low level	6FH	
e)	CRC-16	High level	C4H	
		Low level	5AH	

6.4.3 Response message in the case of an error

a)	Slave address		1BH	
b)	Function code		83H	← In the case of an error, the function code for the request message + 80H is entered.
h)	Error code		02H	
e)	CRC-16	High level	E1H	
		Low level	36H	

6.5 Description of RTU codes

- The codes from a) slave address to b) function code to h) error code shown below are expressed in 8-bit binary numbers.

a) Slave address

This is the address of the party (this product) with which the high-level computer communicates. The address in the response message from this product represents the source of the response message. Note that, when CH2 is used, 2 addresses are occupied. (When the ADR is set to 1, addresses 1 and 2 are occupied.)

b) Function code

Enter a code 03H, 06H, or 10H.

03H: To read data from this product

06H: To write or store data in this product (multiple)

10H: To write or store data in this product (single)

c) Register address

The locations of the data to be read or that to be written are specified in 2 bytes.

The data is written in the holding register.

For the addresses of the commands, see "10. Table of identifiers (codes)."

d) Number of registers

This specifies the number of registers to be written in. Since this product has a fixed number of registers (which is 2), specify 0002H.

e) CRC-16

This error check code is for detecting message errors. This transmits a CRC-16 (tour redundancy code).

The multinomial for generating a CRC-16 used in this product is $X^{16}+X^{15}+X^{2}+1$.

To learn how to calculate the CRC-16, see "6.7. Example of CRC-16 calculations."

To affix an error code at the end of the message, affix the low-level byte first, then the high-level byte of the CRC.

f) Number of data

This specifies the number of registers to be read and written x 2. Since the number of registered in this product is fixed at 2, specify 04H here.

g) Data portion

This specifies data to be written in the register. The data is fixed at 4 bytes. This product will write data without the decimal point.

Example: In the case of numerical data

Example	Significance of the value
Proportional band (P) = 1.0 %	0000000AH
PV = 1200.0°C	00002EE0H
SV = -10.00°C	FFFFFC18H

In the case of text data, write the ASCII code "□INP" (□ is a space): 2049E50H.

h) Error code

If a message from a high-level computer is error-ridden, it will be incorporated in the "response message" from this product and returned.

The error number "04" is an instrument error (memory error or A/D conversion error, AT error). It will be incorporated in the "response message" regardless of whether there is an error in the "request message."

If there are two or more errors occurring at the same time, the largest error number will be incorporated.

The table below indicates the error contents and classifications.

Error No.	Error contents in the "request message" received by this product
01	Received an unsupported function code.
02	Received an address other than the specified one.
03	The numerical data deviated from the "range of settings designated specifically with setting items."
04	Instrument error (memory error or A/D conversion error, AT error)

6.6 Precautions on RTU communications

6.6.1 Communications timing

Set a sufficient response delay to make sure that this product is switched over from transmission to reception with regard to a high-level computer in using an RS-485.

See the figure in "5.1. Overview" and "5.7. Setting a response delay."

3.6.2 Interval between requests

In transmitting a series of "request messages" from a high-level computer, allow for an interval of 1msec or more or 3.5 character minutes, whichever the longer, from the reception of a "response message" from this product to a next transmission.

6.6.3 Response conditions

If there is a time interval of 3.5 characters or more between data items constituting a "request message," this product cannot recognize it as a "request message." It will therefore not return a "response message." If, therefore, the "request message" contains an error, this product will not return a "response message" (error reply) containing an ERR unless the above conditions are met.

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

The moment a period of 3.5 characters or more has elapsed, it clears all codes received before that.

6.6.4 Errors in address specification

This product will not respond to any "request message" that specifies an address other than that specified for itself. If, therefore, the address portion of a "request message" is error-ridden, none of the mobile units will return a "response message."

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

6.6.5 Number of digits in data and the decimal position

See "6.5. Description of RTU codes, g) Data portion."

6.6.6 Operation after receiving a store request message

This product starts to store data after correctly receiving a store request message from a high-level computer.

This product only stores data different from the contents of the EEPROM (data that is changed). The time (TW) required for storing data is within 6 seconds.

This product transmits a storage-complete reply after the data is stored.

This product will not guarantee that the data is stored if this product is turned off during a storage operation. Do not turn off this product for 6 seconds after transmitting a store request message.

6.6.7 Operation after turning on the power

This product will not perform communications (no response) for about 4 seconds after it is turned on. Allow for a delay until communications is started after this product is turned on.

6.6.8 Changing the settings (SV or SV2) by communications during auto-tuning

Even if the settings (SV or SV2) used in control for auto-tuning are changed by communications, the changes (SV or SV2) will not be changed until the auto-tuning ends.

6.7 Example of CRC-16 calculations

Following is an example of calculating CRC-16 with VisualBasic6.0.

Variables are declared as shown below.
VisualBasic6.0 cannot use code-free variables. It therefore uses code-equipped 16-bit integer variables as data. Similarly, the CRC calculation results are entered into code-equipped 32-bit integer variables.

```
Dim CRC As Long
Dim i, j, arry_count As Integer

Dim c_next, c_carry As LongDim crc_arry(64) As Integer
```

Then enter calculable data into the crc_arry(), and enter the number of data items into the arry_count. After that, run the following program to cause the calculation results to enter the CRC.

```
i = 0
CRC = 65535
For i = 0 To arry_count
    c_next = crc_arry(i)
    CRC = (CRC Xor c_next) And 65535
    For j = 0 To 7
        c_carry = CRC And 1
        CRC = CRC ¥ 2
        If c_carry Then
            CRC = (CRC Xor &HA001) And 65535
        End If
    Next
Next
```

To affix an error code to the end of the message, affix first the low-level byte and then the high-level byte of the CRC.

6.8 Composition of an ASCII request message (transmitted from a high-level computer to this product)

■ For the codes a) through g), see "6.10. Description of ASCII codes."

6.8.1 Composition of a read request message

a)	Start code		":"	
b)	Slave address		"1" , "B"	
c)	Function code		"0" , "3"	
d)	Register address	High level	"0" , "0"	First register address
		Low level	"0" , "0"	
e)	Number of registers	High level	"0" , "0"	2 to 100 (even number only)
		Low level	"0" , "2"	
f)	LRC		"E" , "0"	
g)	End code		CR, LF	

6.8.2 Composition of a write request message

a)	Start code		":"	
b)	Slave address		"0" , "3"	
c)	Function code		"1" , "0"	
d)	Register address	High level	"0" , "0"	First register address
		Low level	"C" , "0"	
e)	Number of registers	High level	"0" , "0"	Fixed at 2
		Low level	"0" , "2"	
h)	Number of data items		"0" , "4"	Number of registers × 2
i)	Data for the first register (a low-level word)	High level	"0" , "0"	③ When writing ①, ②, ③, and ④ H in the data, write them in the order described on the left-hand side. (① represents 1 byte.)
		Low level	"6" , "F"	④
	Data for the first register + 1 (a high-level word)	High level	"0" , "0"	①
		Low level	"0" , "0"	②
f)	LRC		"E" , "0"	
g)	End code		CR, LF	

6.8.3 Composition of a store request message

a)	Start code		":"	
b)	Slave address		"0" , "3"	
c)	Function code		"1" , "0"	
d)	Register address	High level	"0" , "2"	First register address
		Low level	"0" , "E"	
e)	Number of registers	High level	"0" , "0"	Fixed at 2
		Low level	"0" , "2"	
h)	Number of data items		"0" , "4"	Number of registers × 2
i)	Data for the first register (a low-level word)	High level	"0" , "0"	The data about the storage of settings is arbitrary.
		Low level	"0" , "0"	
	Data for the first register + 1 (a high-level word)	High level	"0" , "0"	
		Low level	"0" , "0"	
f)	LRC		"D" , "7"	
g)	End code		CR, LF	

6.9 Composition of ASCII response messages (transmitted from this product to a high-level computer)

■ For the codes a) through g), see "6.10. Description of ASCII codes."

6.9.1 Response message for a read request message

a)	Start code		":"	
b)	Slave address		"1" , "B"	
c)	Function code		"0" , "3"	
h)	Number of data items		"0" , "4"	Number of registers × 2
i)	Data for the first register (a low-level word)	High level	"0" , "3"	③ When writing ①, ②, ③, and ④ H in the data, write them in the order described on the left-hand side. (① represents 1 byte.)
		Low level	"0" , "9"	④
	Data for the first register + 1 (a high-level word)	High level	"0" , "0"	①
		Low level	"0" , "0"	②
f)	LRC		"D" , "2"	
g)	End code		CR, LF	

6.9.2 Response message for a write/store request message

a)	Start code		":"	
b)	Slave address		"0" , "3"	
c)	Function code		"1" , "0"	
d)	Register address	High level	"0" , "0"	First register address
		Low level	"0" , "0"	
e)	Number of registers	High level	"0" , "0"	Fixed at 2
		Low level	"0" , "2"	
f)	LRC		"E" , "B"	
g)	End code		CR, LF	

6.9.3 Response message in the case of an error

a)	Start code	":"	
b)	Slave address	"1" , "B"	
h)	Function code	"8" , "3"	← In the case of an error, the function code for the request message + 80H is entered.
j)	Error code	"0" , "2"	
f)	LRC	"6" , "0"	
g)	End code	CR, LF	

6.10 Description of ASCII codes

- The codes from a) start code to b) slave address to j) error type described below are expressed in ASCII codes.
- For ASCII codes, see "9. Table of ASCII codes."
- For converting to ASCII codes, see 6.8 and 6.9 "Message composition."

a) Start code

The receiver side is the code required for detecting the top of the message. It is affixed to the top of a character string to be transmitted.

b) Slave address

This is the address of the party (this product) with which the high-level computer communicates. The address in the response message from this product represents the source of the response message. Note that, when CH2 is used, 2 addresses are occupied. (When the ADR is set to 1, addresses 1 and 2 are occupied.)

c) Function code

Enter a code 03H or 10H.

03H: To read data from this product

10H: To write or store data in this product

d) Number of registers

This specifies the number of registers to be written in. Since this product has a fixed number of registers (which is 2), specify 0002H.

e) Register address

The locations of the data to be read or that to be written are specified in 2 bytes.

For the addresses of the commands, see "10. Table of identifiers (codes)."

f) LRC

LRC is an error check code for detecting message errors. An LRC is transmitted. The LRC used in this product is the 2-complement of the sum of the data portions without a carry, except for the start code and end code of the message.

The parts of the data portions expressed as a "1" and "B" are considered as "1BH."

To learn how to calculate the LRC, see "6.12. Example of LRC calculations."

If 12H is calculated as an error code, affix a "1" or "2" at the end of the message.

g) End code

This code is required for the receiver to detect the end of a message. Affix CR (0DH) and LF (0AH) at the end of a character string to be transmitted.

h) Number of data

This specifies the number of registers to be read and written x 2. Since the number of registered in this product is fixed at 2, specify 04H here.

i) Data portion

This specifies data to be written in the register. The data is fixed at 4 bytes. This product will write data without the decimal point.

Example: In the case of numerical data

Example	Significance of the value
Proportional band (P) = 1.0 %	0000000AH
PV = 1200.0°C	00002EE0H
SV = -10.00°C	FFFFFC18H

In the case of text data, write the ASCII code "□INP" (□ is a space): 2049E50H.

j) Error code

If a message from a high-level computer is error-ridden, it will be incorporated in the "response message" from this product and returned.

The error number "04" is an instrument error (memory error or A/D conversion error, AT error). It will be incorporated in the "response message" regardless of whether there is an error in the "request message."

If there are two or more errors occurring at the same time, the largest error number will be incorporated.

The table below indicates the error contents and classifications.

Error No.	Error contents in the "request message" received by this product
01	Received an unsupported function code.
02	Received an address other than the specified one.
03	The numerical data deviated from the "range of settings designated specifically with setting items."
04	Instrument error (memory error or A/D conversion error, AT error)

6.11 Precautions on ASCII communications

6.11.1 Communications timing

Set a sufficient response delay to make sure that this product is switched over from transmission to reception with regard to a high-level computer in using an RS-485.

See the figure in "5.1. Overview" and "5.7. Setting a response delay."

6.11.2 Interval between requests

In transmitting a series of "request messages" from a high-level computer, allow for an interval of 1msec or more or 3.5 character minutes, whichever the longer, from the reception of a "response message" from this product to a next transmission.

6.11.3 Response conditions

This product will not return a "response message" unless the "request message" contains a start code and end code.

If, therefore, the "request message" contains an error, this product will not return a "response message" (error reply) containing an error code unless the above conditions are met.

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

The moment a start code is received, this product clears all codes received before that.

6.11.4 Errors in address specification

This product will not respond to any "request message" that specifies an address other than that specified for itself. If, therefore, the address portion of a "request message" is error-ridden, none of the mobile units will return a "response message."

Therefore transmit the necessary "request message" again if a "request message" is sent to the high-level computer but the latter does not return a "response message" at the end of an appropriate period.

The moment a start is received, this product clears all codes received before that.

6.11.5 Number of digits in data and the decimal position

See "6.10. Description of ASCII codes, i) Data portion."

6.11.6 Operation after receiving a store request message

This product starts to store data after correctly receiving a store request message from a high-level computer.

This product only stores data different from the contents of the EEPROM (data that is changed). The time (TW) required for storing data is within 6 seconds.

This product transmits a storage-complete reply after the data is stored.

This product will not guarantee that the data is stored if this product is turned off during a storage operation. Do not turn off this product for 6 seconds after transmitting a store request message.

6.11.7 Operation after turning on the power

This product will not perform communications (no response) for about 4 seconds after it is turned on. Allow for a delay until communications is started after this product is turned on.

6.11.8 Changing the settings (SV or SV2) by communications during auto-tuning

Even if the settings (SV or SV2) used in control for auto-tuning are changed by communications, the changes (SV or SV2) will not be changed until the auto-tuning ends.

6.12 Example of LRC calculations

Following is an example of calculating LRC with VisualBasic6.0.

Variables are declared as shown below.
VisualBasic6.0 cannot use code-free variables. It therefore uses code-equipped 16-bit integer variables as data. Similarly, the LRC calculation results are entered into code-equipped 16-bit integer variables.

```
Dim LRC As Integer
Dim i, arry_count As Integer

Dim lrc_arry(128) As Integer
```

Then enter calculable data into the 1rc_arry(), and enter the number of data items into the arry_count. After that, run the following program to cause the calculation results to enter the LRC.

```
For i = 0 To arry_count
    LRC = (LRC + lrc_arry(i)) And &HFF
Next

LRC = ((Not LRC) + 1) And &HFF
```

If the error code is calculated as 12H as an example, affix a "1" or "2" at the end of the message.

7. Specifications

7.1 Communications standard category: Compliant with EIA standard RS-485

7.2 Communications specifications

7.2.1 Communications system

Network: ..... For RS-485, multi-drop system (up to 1 pair, 31 stations)
Direction of information: ..... Half duplex
Synchronization system: ..... Asynchronous
Transmission code: ..... ASCII, 7/8 bit code, except for BBC data
(highest-level bit = 0 in 8-bit code)

7.2.2 Interface system

Signal line: ..... 2 lines for transmission and reception
Communications speed: ..... One speed is selected from 1,200, 4,800, 9,600, and 19,200 bps and this product is set to it.
Communications distance: ..... 500m maximum
Provided that it varies somewhat depending on the cable and other ambient conditions.

7.2.3 TOHO communications characters

Start bit length: ..... Fixed at 1 bit
Stop bit length: ..... Either 1 or 2 bit is selected and this product is set to it.
Data length: ..... Either 7 or 8 bit is selected and this product is set to it.
Parity: ..... No. Either odd or even is selected and this product is set to it.
BCC check: ..... Yes or no is selected and this product is set to it.
Communications address: 1-99

7.2.4 MODBUS communications (RTU) characters

Start bit length: ..... Fixed at 1 bit
Stop bit length: ..... Either 1 or 2 bit is selected and this product is set to it. (If parity-equipped, fixed at 1 bit.)
Data length: ..... Fixed at 8 bit.
Parity: ..... No. Either odd or even is selected and this product is set to it.
CRC-16 check: ..... Fixed at yes.
Communications address: 1-247

7.2.5 MODBUS communications (ASCII) characters

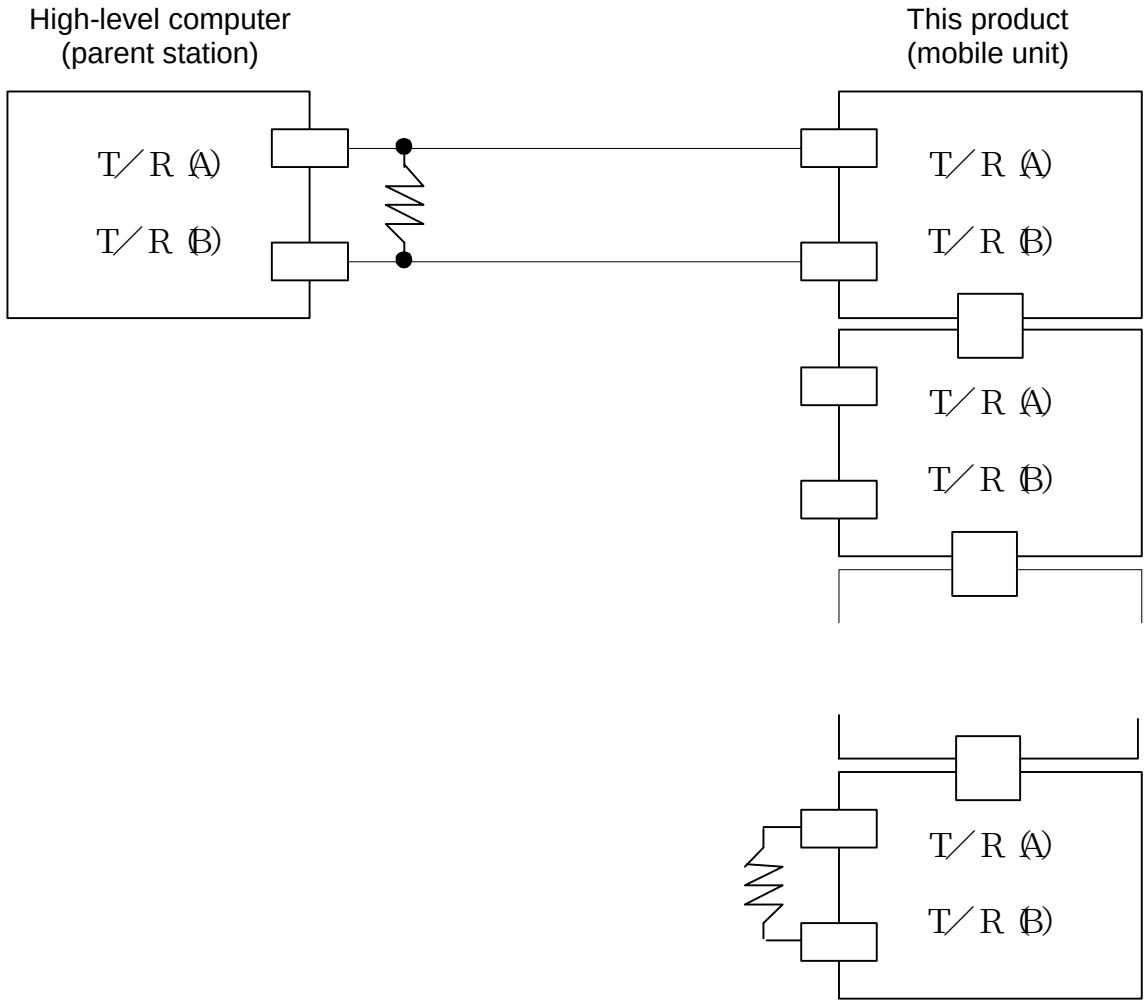
Start bit length: ..... Fixed at 1 bit
Stop bit length: ..... Either 1 or 2 bit is selected and this product is set to it. (If parity-equipped, fixed at 1 bit.)
Data length: ..... Fixed at 7 bit.
Parity: ..... No. Either odd or even is selected and this product is set to it.
LRC check: ..... Fixed at yes.
Communications address: 1-247

7.2.6 MODBUS communications (ASCII/RTU) function codes

03H (reading the contents of the holding register)
10H (writing the contents of two or more holding registers)

8. Connections

8.1 Connections for the RS-485



Communications is such that the connections can be made with the connectors on the sides.

Install an end of line resistor at both of the farthest devices in the parent station and the mobile unit. For a resistance value, use one that matches the characteristic impedance of the cable. Provided that the synthesis is set to at least 75Ω.

9. Table of ASCII codes

ASCII code	00H	01H	02H	03H	04H	05H	06H	07H
	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL

ASCII code	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
	BS	HT	LF	VT	FF	CR	SO	SI

ASCII code	10H	11H	12H	13H	14H	15H	16H	17H
	DLE	DC1	DC2	DC3	DC4	NAK	SYM	ETB

ASCII code	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH
	CAN	EM	SUB	ESC	FS	GS	RS	US

ASCII code	20H	21H	22H	23H	24H	25H	26H	27H
	ｽﾍﾟｰｽ	!	”	#	$	%	&	’

ASCII code	28H	29H	2AH	2BH	2CH	2DH	2EH	2FH
	(	)	*	+	,	−	.	／

ASCII code	30H	31H	32H	33H	34H	35H	36H	37H
	0	1	2	3	4	5	6	7

ASCII code	38H	39H	3AH	3BH	3CH	3DH	3EH	3FH
	8	9	:	;	<	=	>	?

ASCII code	40H	41H	42H	43H	44H	45H	46H	47H
	@	A	B	C	D	E	F	G

ASCII code	48H	49H	4AH	4BH	4CH	4DH	4EH	4FH
	H	I	J	K	L	M	N	O

ASCII code	50H	51H	52H	53H	54H	55H	56H	57H
	P	Q	R	S	T	U	V	W

ASCII code	58H	59H	5AH	5BH	5CH	5DH	5EH	5FH
	X	Y	Z	[	¥	]	^	_

ASCII code	60H	61H	62H	63H	64H	65H	66H	67H
	‘	a	b	c	d	e	f	g

ASCII code	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH
	h	i	j	k	l	m	n	o

ASCII code	70H	71H	72H	73H	74H	75H	76H	77H
	p	q	r	s	t	u	v	w

ASCII code	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
	x	y	z	{	|	}	~	DEL

10. Table of identifiers (codes)

MODBUS ADR Operation mode

Low-level W	High-level W	Iden-tifier	Character	Name	R/W	Description
40000	40001	PV1		Measurement	R	Use it as a monitor for measurements (PV). When overscale: HHHH When underscale: LLLL
40002	40003	SV1		Control setting	R/W	R/W of the setting (SV)

MODBUS ADR Initial setting mode

Low-level W	High-level W	Iden-tifier	Character	Name	R/W	Description
40004	40005	INP	_InP	Set an input type	R/W	000??
40006	40007	PVG	_PuG	Set a PV corrected gain	R/W	
40008	40009	PVS	_PuS	Set a PV corrected zero point	R/W	
40010	40011	PDF	_PdF	Set a PV filter	R/W	
40012	40013	□DP	_ dP	Set a decimal position	R/W	No decimal point : 00000 1 decimal place : 00001 2 decimal place : 00002 3 decimal place : 00003
40014	40015	□FU	_ FU	Set a function key function	R/W	
40016	40017	LOC	_LoC	Set a key lock	R/W	

MODBUS ADR Control setting mode

Low-level W	High-level W	Iden-tifier	Character	Name	R/W	Description
40018	40019	SLH	_SLH	Set an SV limiter upper limit	R/W	
40020	40021	SLL	_SLL	Set an SV limiter lower limit	R/W	
40022	40023	□MD	_ Md	Set control mode	R/W	Control execution : 00000 Manual control : 00001 Control stop : 00002 Auto-tuning in progress : 00003
40024	40025	CNT	_Cnt	Set a control type	R/W	
40026	40027	DIR	_dir	Set positive/reverse operation switchover	R/W	
40028	40029	MV1	_Mv1	Amount of output operation	R/W	W possible during manual control
40030	40031	TUN	_tUn	Set a tuning type	R/W	
40032	40033	ATG	_AtG	Set an AT factor	R/W	
40034	40035	ATC	_AtC	Set an AT sensitivity	R/W	
40036	40037	□P1	_ P1	Set a proportional band	R/W	
40038	40039	□I1	_ I	Set an integral time	R/W	
40040	40041	□D1	_ d	Set a derivative time	R/W	
40042	40043	□T1	_ t1	Set a proportional frequency	R/W	
40044	40045	ARW	_ArW	Anti-reset windup	R/W	
40046	40047	MH1	_MH1	Set an amount-of-operation limiter upper limit	R/W	
40048	40049	ML1	_ML1	Set an amount-of-operation limiter lower limit	R/W	
40050	40051	PBB	_Pbb	Manual reset	R/W	
40052	40053	MV2	_Mv2	Amount of cooling output operation	R/W	W possible during manual control
40054	40055	□P2	_ P2	Set a cooling power proportional band	R/W	
40056	40057	□T2	_ t2	Set a cooling output proportional cycle	R/W	
40058	40059	MH2	_MH2	Set a cooling output amount-of-operation limiter upper limit	R/W	
40060	40061	ML2	_ML2	Set a cooling output amount-of-operation limiter lower limit	R/W	
40062	40063	□C1	_ C1	Set a (heating output) control sensitivity	R/W	
40064	40065	□C2	_ C2	Set a cooling output control sensitivity	R/W	
40066	40067	CP1	_CP1	Set an off-point position	R/W	
40068	40069	CP2	_CP2	Set a cooling output off-point position	R/W	
40070	40071	□DB	_ db	Set a dead band	R/W	

MODBUS ADR Event setting mode

Low-level W	High-level W	Identifier	Character	Name	R/W	Description
40072	40073	E1F	_E 1F	Set an event output function	R/W	
40074	40075	E1H	_E 1H	Set an event output upper limit	R/W	
40076	40077	E1L	_E 1L	Set an event output lower limit	R/W	
40078	40079	E1C	_E 1C	Set an event output sensitivity	R/W	
40080	40081	E1T	_E 1t	Set an event output delay timer	R/W	
40082	40083	E1B	_E 1b	Set an event output special function	R/W	
40084	40085	E1P	_E 1P	Set an event output polarity	R/W	
40086	40087	CM1	_CM 1	CT monitor	R	
40088	40089	CT1	_Ct 1	Set a CT abnormal current	R/W	

MODBUS ADR Timer setting mode

Low-level W	High-level W	Identifier	Character	Name	R/W	Description
40090	40091	TMO	_tMo	Set a timer output destination	R/W	
40092	40093	TMF	_tMF	Set a timer function	R/W	
40094	40095	H/M	_HrM	Set a timer unit	R/W	
40096	40097	TSV	_tSu	Set a timer SV start tolerance	R/W	
40098	40099	TIM	_ti M	Set a timer time	R/W	
40100	40101	TIA	_ti A	Set a timer remaining time	R/W	

MODBUS ADR DI setting mode

Low-level W	High-level W	Identifier	Character	Name	R/W	Description
40102	40103	DIF	_di F	Set a DI function	R/W	
40104	40105	DIP	_di P	Set a DI polarity	R/W	
40106	40107	SV2	1Su2	Set a CH1 SV2	R/W	
40108	40109	SV2	2Su2	Set a CH2 SV2	R/W	

MODBUS ADR Communications setting mode

Low-level W	High-level W	Identifier	Character	Name	R/W	Description
40110	40111	PRT	_Prt	Set a communications protocol	R/W	
40112	40113	COM	_CoM	Set a communications parameter	R/W	Example: □B8N2
40114	40115	BPS	_bPS	Set a communications speed	R/W	Example: 00096 (if 9600 bps)
40116	40117	ADR	_Adr	Set a communications address	R/W	
40118	40119	AWT	_AWt	Set a communications response delay	R/W	
40120	40121	MOD	_Mod	Set communications mode switchover	R/W	RO:00000 RW:00001

MODBUS ADR　Screen-less commands

Low-level W	High-level W	Iden-tifier	Character	Name	R/W	Description
40122	40123	TST		Timer start stop	R/W	
40124	40125	OM1		Output monitor	R	R of the output monito ①②③④⑤ ⑤：Control output (1:ON　0:OFF) ④：Cooling output (1:ON　0:OFF) ③：Event output (1:ON　0:OFF)
40126	40127	EM1		DI monitor	R	R of DI monitor ①②③④⑤ ⑤：DI (1: ON　0: OFF)
40128	40129	□AT		Start/release AT	R/W	Read/write AT start/release Start : 00001 Release : 00000 Reading during startup causes this product to replay with 00001.
40130	40131	STR		Store data	W	Store data

MODBUS ADR　Identifiers used only in blind setting

Low-level W	High-level W	Iden-tifier	Character	Name	R/W	Description
40132	40133	001		SET1	L/B	Blinding enabled : 00000 Blinding disabled 00001
40134	40135	002		SET2	L/B	
40136	40137	003		SET3	L/B	
40138	40139	004		SET4	L/B	
40140	40141	005		SET5	L/B	
40142	40143	006		SET6	L/B	

TOHO ELECTRONICS INC.
Head office: 1-13-21, Tanashioda, Sagamihara Kanagawa 229-1125 Japan.
Phone: +81-42-777-3311 Fax: +81-42-777-3751