TOHO ELECTRONICS INC.

Operation Manual, Communications
(TOHO protocol and MODBUS)

Model: TRM-006A Series
Designation: Digital Indicator

Thank you very much for purchasing a TRM-006A Series (with communications). Please read this operation manual carefully and use this product correctly.

Contents

1. Before using the product 4
   - 1.1 On this operation manual
   - 1.2 Conditions for communications
   - 1.3 What can be done with communications
   - 1.4 Positioning communications (priority ranking)
   - 1.5 Setting before communications

2. Settings regarding TOHO communications 5
   - 2.1 Overview
   - 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
   - 2.6 Setting a communications speed
   - 2.7 Setting an address
   - 2.8 Setting a response delay
   - 2.9 Switching communications mode

3. TOHO communications control 8
   - 3.1 Communications procedure
   - 3.2 Message types
   - 3.3 Composition of a request message (transmitted from a high-level computer to this product)
   - 3.4 Composition of a response message (transmitted from this product to a high-level computer)
   - 3.5 Description of codes
   - 3.6 Communications precautions

4. Examples of TOHO communications 15
   - 4.1 Examples of communications to be read
   - 4.2 Examples of communications to be written

5. Settings regarding MODBUS communications 17
   - 5.1 Overview
   - 5.2 Setting a data length
   - 5.3 Setting a stop bit length
   - 5.4 Setting a parity
   - 5.5 Setting a BCC check
   - 5.6 Setting a communications speed
   - 5.7 Setting an address
   - 5.8 Setting a response delay
   - 5.9 Switching communications mode

6. MODBUS communications control 20
6.1 Communications procedure
6.2 Message types
6.3 Composition of an RTU request message (transmitted from a high-level computer to this product)
6.4 Composition of an RTU response message (transmitted from this product to a high-level computer)
6.5 Description of RTU codes
6.6 Precautions on RTU communications
6.7 Example of CRC-16 calculations
6.8 Composition of an ASCII request message (transmitted from a high-level computer to this product)
6.9 Composition of ASCII response messages (transmitted from this product to a high-level computer)
6.10 Description of ASCII codes
6.11 Precautions on ASCII communications
6.12 Example of LRC calculations

7. Specifications 34
7.1 Communications standard category
7.2 Communications specifications

8. Connections 35

9. Table of identifiers (codes) 36

10. Table of ASCII codes 39

1. Before using the product

1.1 On this operation manual

This is an operation manual regarding communications with a TRM-006A Series (hereinafter referred to as "this product").

1.2 Conditions for communications

The communications function of this product is optionally specified. For that reason, you should specify a communications option (RS-485) in purchasing this product.

1.3 What can be done with communications

With this product, users can write and read items specified in "9. Table of identifiers (codes)," 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 "Communications precautions." in chapter 3.6, 6.6 and 6.11.)

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

1.4 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.)

1.5 Setting before communications

Before performing communications, this product must be set. See "2. Settings regarding TOHO communications" and "5. Settings regarding MODBUS communications."

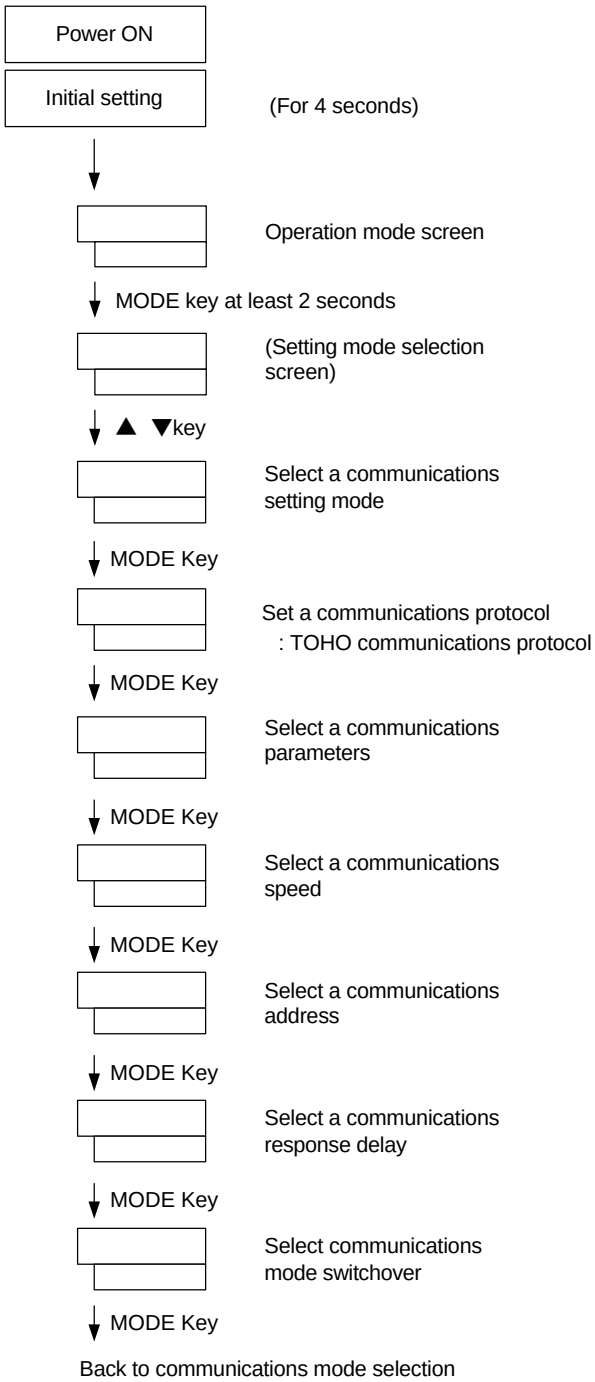
2. Settings regarding TOHO communications

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.

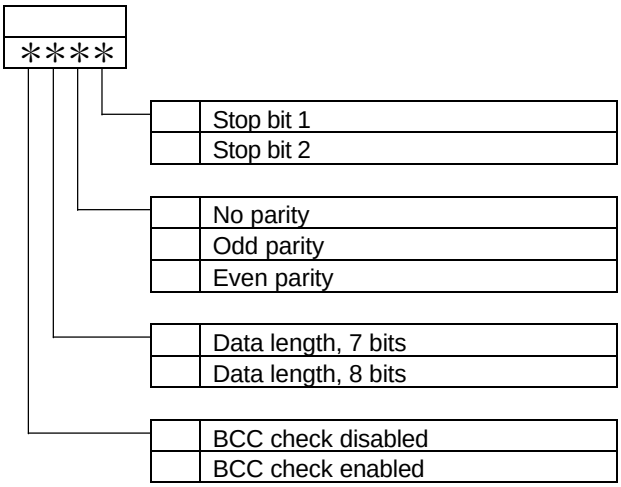
2.2 Setting a data length

2.3 Setting a stop bit length

2.4 Setting a parity

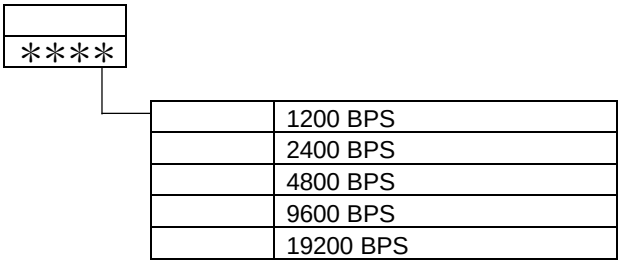
2.5 Setting whether to conduct a BCC check

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



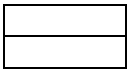
2.6 Setting a communications speed

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



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 .

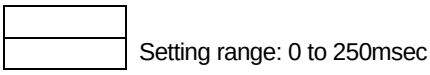


Setting range: 1 to 99 stations (It cannot be set to a 0.)

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.

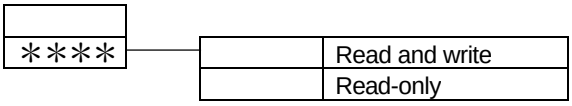
While in the "Set a response delay" on the preceding page, operate the ▲ and ▼ keys to make the settings. The initial value is 0.



* 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

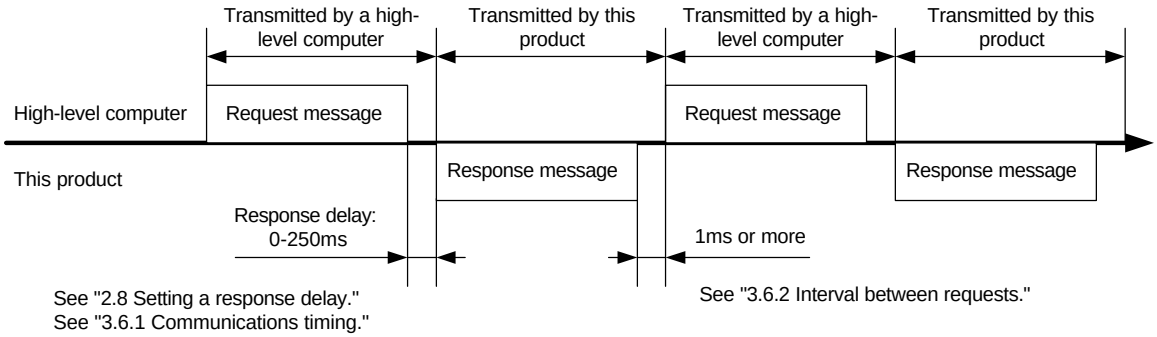
While in the "Set communications mode switchover" screen on the preceding page, operate the ▲ and ▼ keys to make the settings.



3. TOHO communications control

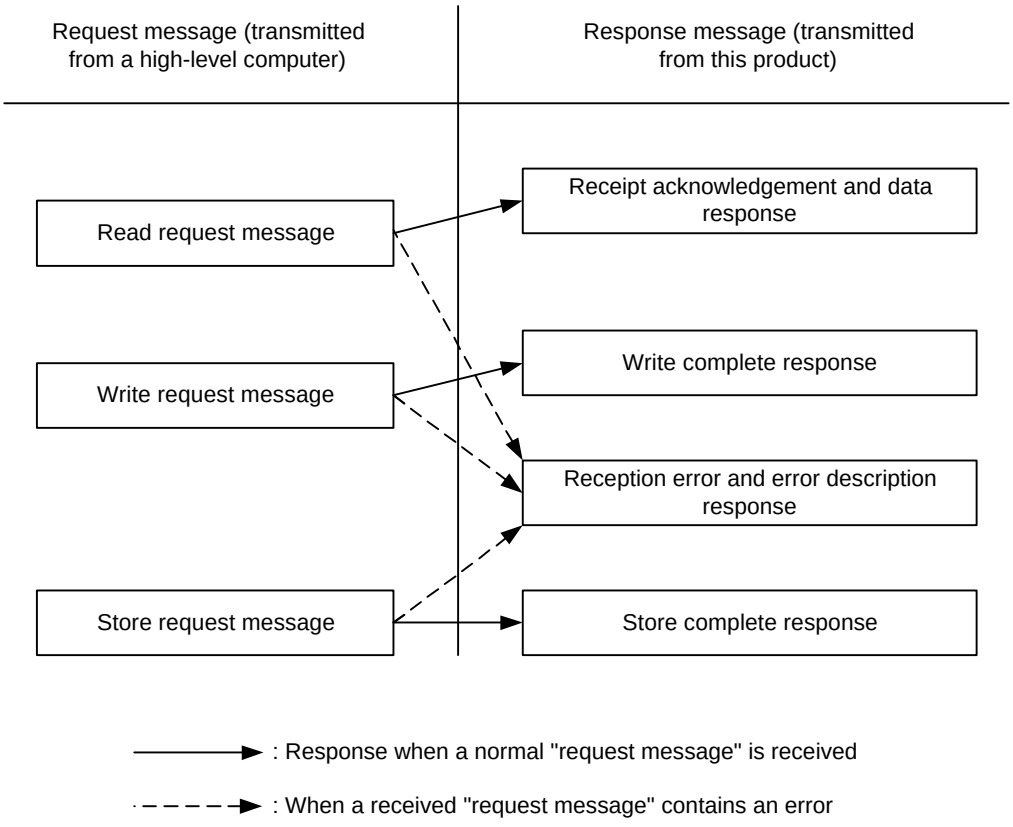
3.1 Communications procedure

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:

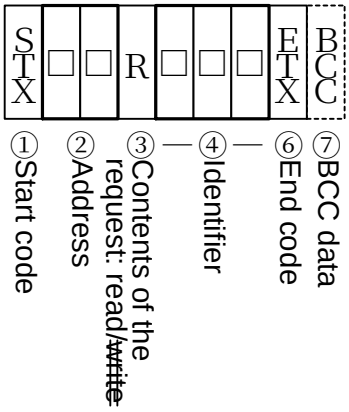


- All codes (except for BCC) from STX and data to ETX are expressed in ASCII codes.
- In assembling a program for a high-level computer, see "9. Table of identifiers (codes)" and "10. Table of ASCII codes" at the end of the book.

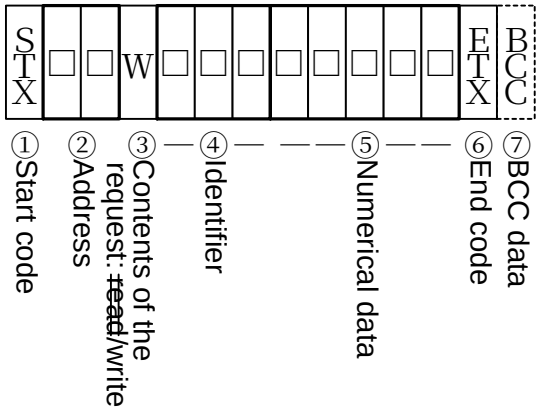
3.3 Composition of a request message (transmitted from a high-level computer to this product)

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

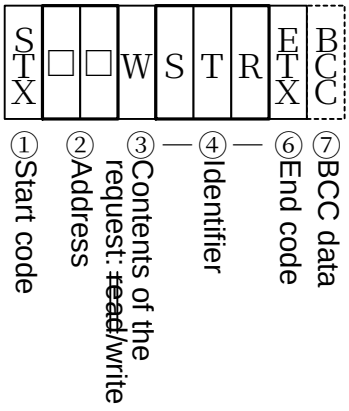
3.3.1 Composition of a read request message



3.3.2 Composition of a write request message



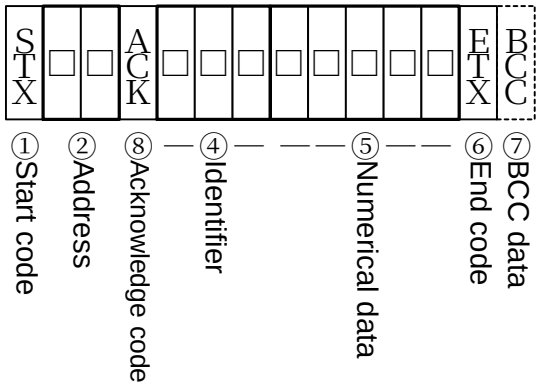
3.3.3 Composition of a store request message



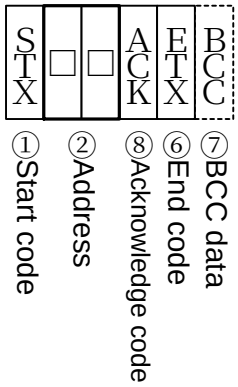
3.4 Composition of a response message (transmitted from this product to a high-level computer)

- For codes ① to 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."

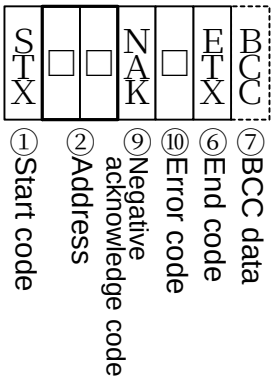
3.4.1 Response message in response to a read request message



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 type as indicated below are expressed in ASCII codes.
- For the ASCII codes, see "10. 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.

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 "9. Table of identifiers (codes)."

⑤ Numerical data

These are data to be read or written, and are all expressed in five digits regardless of the type.

Negative data: The "-" (minus) sign is in a single digit at the largest digit.

Position of the decimal point: 5-digit data does not include a decimal point.

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

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 "0" 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, the high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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, the high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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

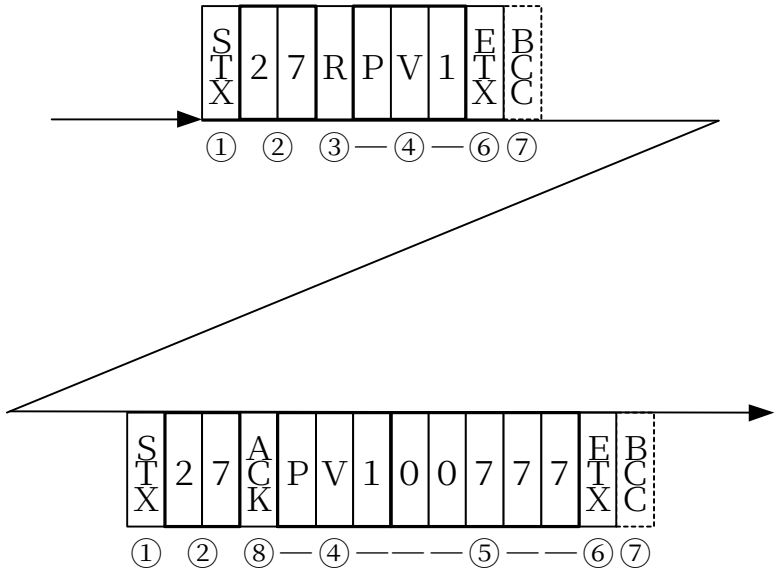
4.1 Examples of communications to be read

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).
(This product)

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

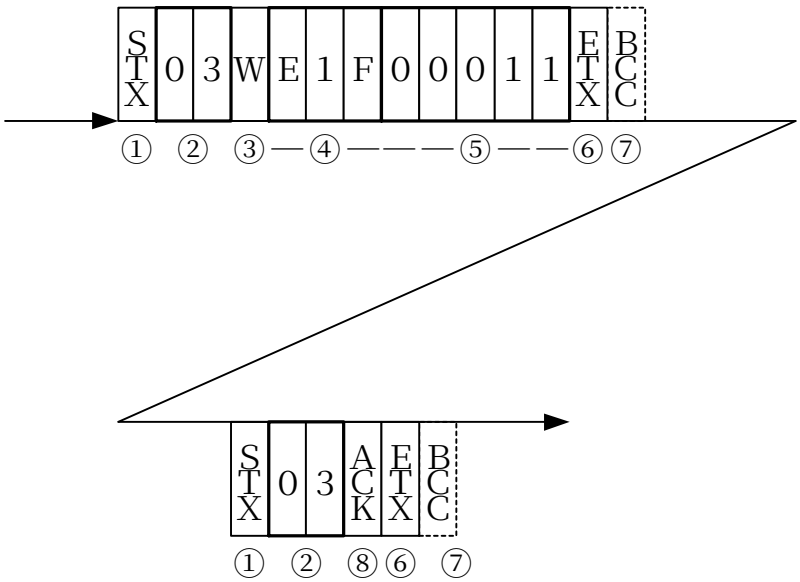
4.2 Examples of communications to be written

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	00135	30H 30H 30H 31H 31H
⑥ End code	ETX	03H
⑦ BCC data request		53H
response		04H
⑧ Acknowledge code	ACK	06H

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

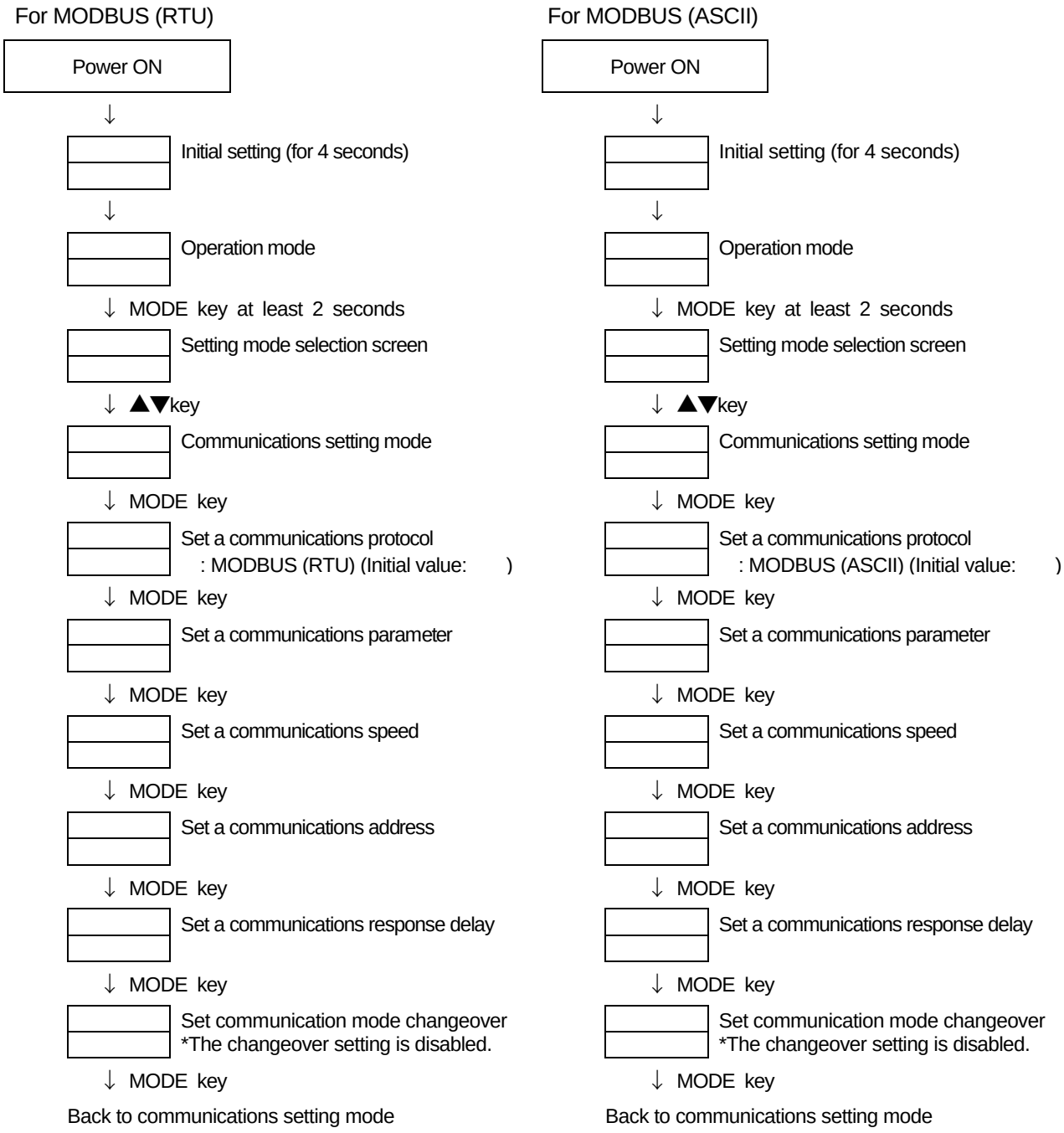
5. Settings regarding MODBUS communications

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.

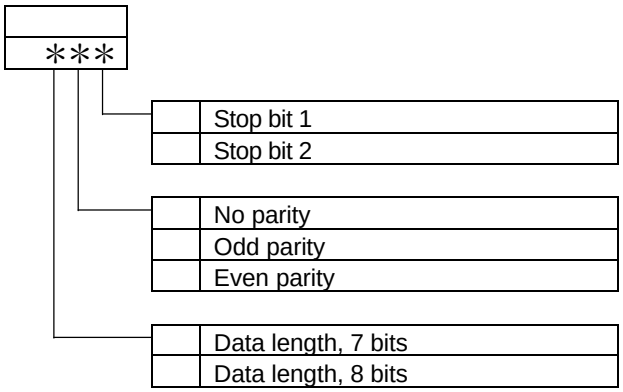
5.2 Setting a data length

5.3 Setting a stop bit length

5.4 Setting a parity

5.5 Setting a BCC check

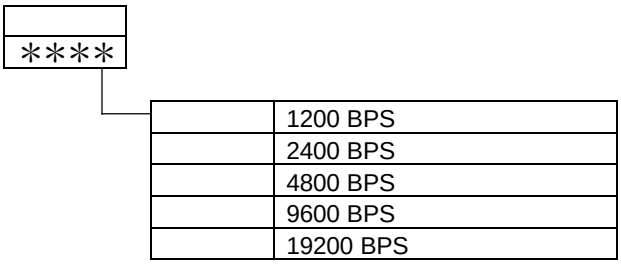
The BCC check is disabled.
Initial value of MODBUS (RTU): Initial value of MODBUS (ASCII):



* The RTU mode settings come only in three types: .
The ASCII mode settings come only in three types: .

5.6 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 .



5.7 Setting an address

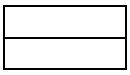
While in the "Set a communications address" screen on the preceding page, operate the ▲ and ▼ keys to make the settings. The initial value is .

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

5.8 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.

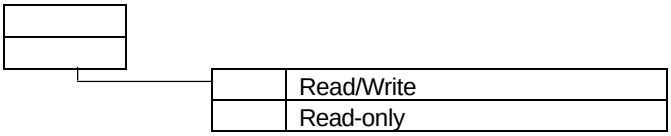


Setting range: 0 to 250msec

* 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.

5.9 Switching communications mode

While in the "Set communication mode changeover" screen on page 17, operate the ▲ and ▼ keys and make a setting.

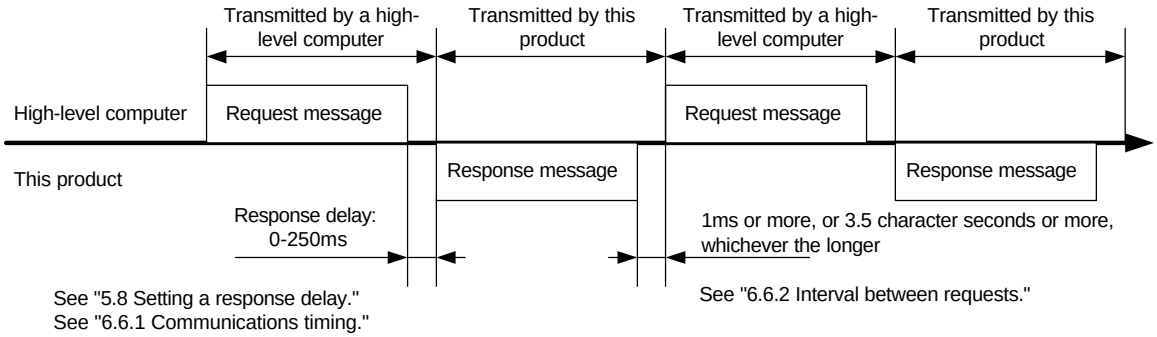


* The changeover setting is disabled.

6. MODBUS communications control

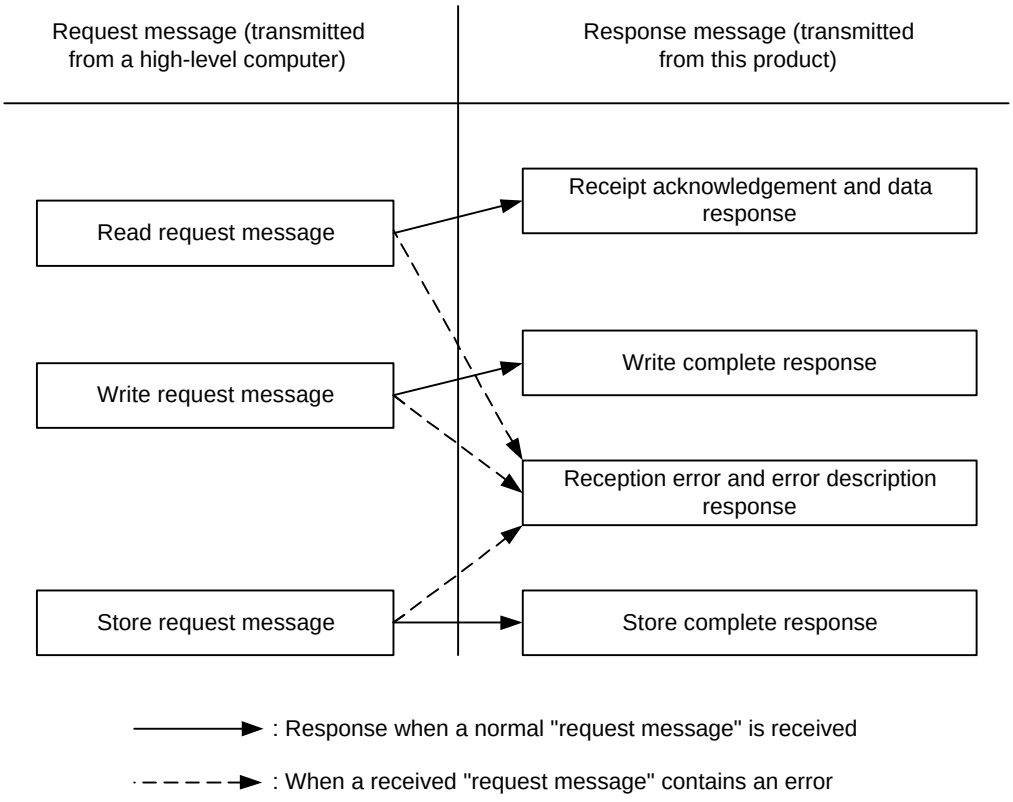
6.1 Communications procedure

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 "9. Table of identifiers (codes)" and "10. 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	Fixed at 2
		Low level	02H	
e)	CRC-16	High level	C6H	
		Low level	31H	

6.3.2 Composition of a write request message

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	

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	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	09H	④
	Data for the first register + 1 (a high-level word)	High level	00H	①
		Low level	00H	②
e)	CRC-16	High level	91H	
		Low level	B4H	

6.4.2 Response message for a write/store request message

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	

6.4.3 Response message in the case of an error

a)	Slave address		1BH
b)	Function code		83H
h)	Error code		02H
e)	CRC-16	High level	E1H
		Low level	36H

← In the case of an error, the function code for the request message + 80H is entered.

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 or 10H.

03H: To read data from this product

10H: To write or store data in this product

c) 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 "9. Table of identifiers (codes)."

The data is written in the holding register.

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 registers 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): 20494E50H.

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 "6.1 Communications procedure" and "5.8 Setting a response delay."

6.6.2 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, the high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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, the high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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 Storing data other than a store request message

This product will 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.

6.6.9 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.

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"	Fixed at 2
		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 "10. 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 "9. 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 registers 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): 20494E50H.

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 "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."

Error number "9" is an AT error. It is therefore incorporated into the "response message" regardless of whether the "request message" is error-ridden. 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 failure (memory error or A/D conversion error)
1	The numerical data was out of a "specific setting range specified with a setting item."
2	The required modification in an item is prohibited, or such an item to be read does not exist.
3	Reservation number
4	Format error
5	LRC error
6	Overrun error
7	Framing error
8	Parity error
9	A PV error occurred during AT. Or AT does not end 3 hours later.

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 "6.1 Communications procedure" and "5.8 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, high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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, the high-level computer transmits the necessary "request message" again if a "request message" is sent to this product 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 Storing data other than a store request message

This product will 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.

6.11.9 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.

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: ..... Multi-drop system (up to 1 pair, 31 stations)
Direction of information: ..... Half duplex
Synchronization system: ..... Asynchronous
Transmission code: ..... ASCII, 7 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: ..... 1,200, 2,400, 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 Character

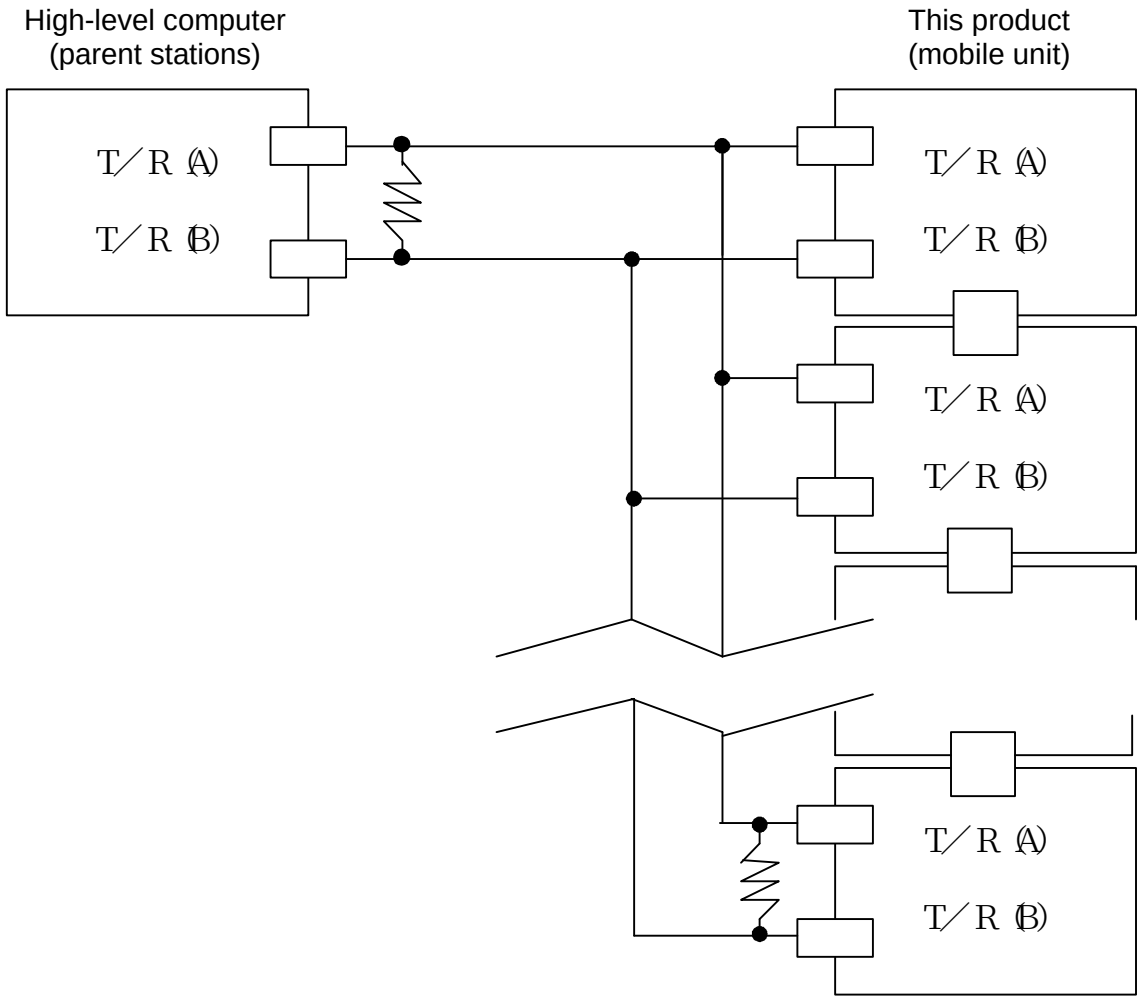
1) TOHO communications protocols
   Start bit length: ..... Fixed at 1 bit
   Stop bit length: ..... Either 1 or 2 bit is selected and this product is set to it.
   Data length: ..... Either 7 or 8 bit is selected and this product is set to it.
   Parity: ..... No. Either odd or even is selected and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

2) MODBUS communications (RTU) protocols
   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

3) MODBUS communications (ASCII) protocols
   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

4) MODBUS communications (RTU/ASCII) function codes
   03H (reading the contents of the holding register)
   10H (writing the contents of two or more holding registers)

8. Connections



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 identifiers (codes)

■ For the setting range, options, initial values, and similar parameters, see the operation manual for this system.

a) Identifier: This code represents an item. Enter this code in the identifier field in the message. The □ in the frame represents an SP (ASCII code: 20H).

b) Character: The character to be displayed on the system screen.

c) Name: Item name

d) R/W: This specifies which is possible: reading, writing, or both.

e) Description:

Note: The R/W to characters that do not meet the display conditions responds with "NAK2."

Example: If no EV2 option is selected, the R/W to the EV2 character becomes "NAK2."

Identifier	MODBUS address		Character	Name	R/W	Description
	Low-level W	High-level W				
PV1	0	1		Setting value (PV)	R	Use it as monitor for measurements (PV). When overscale: HHHH When underscale: LLLL
MI1	198	199		PV bottom hold value	R/W	When overscale: HHHH When underscale: LLLL Reset: 00001
MA1	200	201		PV peak hold value	R/W	When overscale: HHHH When underscale: LLLL Reset: 00001
PR1	4	5		Priority screen function setting 1	R/W	RW the priority screen function setting 1 Example: □□ INP (identifier)
PR2	6	7		Priority screen function setting 2	R/W	RW the priority screen function setting 2 Example: □□INP (identifier)
PR3	8	9		Priority screen function setting 3	R/W	RW the priority screen function setting 3 Example: □□INP (identifier)
PR4	10	11		Priority screen function setting 4	R/W	RW the priority screen function setting 4 Example: □□INP (identifier)
PR5	12	13		Priority screen function setting 5	R/W	RW the priority screen function setting 5 Example: □□INP (identifier)
PR6	14	15		Priority screen function setting 6	R/W	RW the priority screen function setting 6 Example: □□INP (identifier)
PR7	16	17		Priority screen function setting 7	R/W	RW the priority screen function setting 7 Example: □□INP (identifier)
PR8	18	19		Priority screen function setting 8	R/W	RW the priority screen function setting 8 Example: □□INP (identifier)
PR9	20	21		Priority screen function setting 9	R/W	RW the priority screen function setting 9 Example: □□INP (identifier)
INP	22	23		Set a input type	R/W	R/W the input type setting
PVG	24	25		Set a PV corrected gain	R/W	R/W the PV corrected gain setting
PVS	26	27		Set a PV corrected zero point	R/W	R/W the PV corrected zero point setting
PDF	28	29		Set an input filter	R/W	R/W the input filter setting
PH1	202	203		PV hold function setting	R/W	Function OFF: 00000 Peak hold: 00001 Bottom hold: 00002 Peak & bottom: 00003

Identifier	MODBUS address		Character	Name	R/W	Description
	Low-level W	High-level W				
□DP	30	31		Set a decimal position	R/W	R/W the decimal position setting No decimal point: 00000 Tenth: 00001 For analog input, following available: Hundredth: 00002 Thousandth: 00003
LOC	34	35		Set a key lock	R/W	R/W the key lock setting
SLH	36	37		Set a scaling upper limit	R/W	R/W the PV range upper limit setting
SLL	38	39		Set a scaling lower limit	R/W	R/W the PV range lower limit setting
E1F	94	95		Set a PV event output 1 function	R/W	R/W the PV event output 1 function setting
E1H	96	97		Set an event output 1 upper limit	R/W	R/W the event output 1 upper limit setting
E1L	98	99		Set an event output 1 lower limit	R/W	R/W the event output 1 lower limit setting
E1C	100	101		Set an event output 1 sensitivity	R/W	R/W the event output 1 sensitivity setting
E1T	102	103		Set an event output 1 delay timer	R/W	R/W the event output 1 delay timer setting
E1B	104	105		Set a special event output 1 function	R/W	R/W the special event output 1 function setting
E1P	106	107		Set an event output 1 polarity	R/W	R/W the event output 1 polarity setting
E2F	112	113		Set a PV event output 2 function	R/W	R/W the PV event output 2 function setting
E2H	114	115		Set an event output 2 upper limit	R/W	R/W the event output 2 upper limit setting
E2L	116	117		Set an event output 2 lower limit	R/W	R/W the event output 2 lower limit setting
E2C	118	119		Set an event output 2 sensitivity	R/W	R/W the event output 2 sensitivity setting
E2T	120	121		Set an event output 2 delay timer	R/W	R/W the event output 2 delay timer setting
E2B	122	123		Set a special event output 2 function	R/W	R/W the special event output 2 function setting
E2P	124	125		Set an event output 2 polarity	R/W	R/W the event output 2 polarity setting
PRT	136	137		Set a communications protocol	R/W	R/W the communications protocol setting Special-purpose protocol: 00000 MODBUS (RTU): 00001 MODBUS (ASCII): 00002
COM	138	139		Set a communications parameter	R/W	R/W the communications parameter setting Example: □B8N2
BPS	140	141		Set a communications speed	R/W	R/W the communications speed setting Example: 00096 (if 9600 bps)
ADR	142	143		Set a communications address	R/W	R/W the communications address setting
AWT	144	145		Set a response delay	R/W	R/W the response delay setting
MOD	146	147		Set communications mode switchover	R/W	R/W the communications mode switchover setting RO: 00000 RW: 00001
TRF	160	161		Set a transmission output function	R/W	R/W the transmission output function setting
TRP	162	163		Set a forward/reverse operation switchover for transmission output	R/W	R/W the forward/reverse operation switchover setting for transmission output

Identifier	MODBUS address		Character	Name	R/W	Description
	Low-level W	High-level W				
TRH	164	165		Set an upper limit for transmission output scaling	R/W	R/W the upper limit setting for transmission output scaling
TRL	166	167		Set an lower limit for transmission output scaling	R/W	R/W the lower limit setting for transmission output scaling
OM1	170	171		Output status monitor	R	Read the output monitor ①②③④⑤ ⑤: EV1 (1:ON 0:OFF) ④: EV2 (1:ON 0:OFF)
STR	176	177		Store data	W	Store data

Identifiers used only in blind setting

Identifier	MODBUS address		Character	Name	L/B	Description
	Low-level W	High-level W				
000	178	179		SET0	L/B	Blinding enabled: 00000 Blinding disabled: 00001
001	180	181		SET1	L/B	Blinding enabled: 00000 Blinding disabled: 00001
002	182	183		SET2	L/B	Blinding enabled: 00000 Blinding disabled: 00001
003	184	185		SET3	L/B	Blinding enabled: 00000 Blinding disabled: 00001
004	186	187		SET4	L/B	Blinding enabled: 00000 Blinding disabled: 00001
005	188	189		SET5	L/B	Blinding enabled: 00000 Blinding disabled: 00001
006	190	191		SET6	L/B	Blinding enabled: 00000 Blinding disabled: 00001

10. Table of ASCII codes

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

47-9928