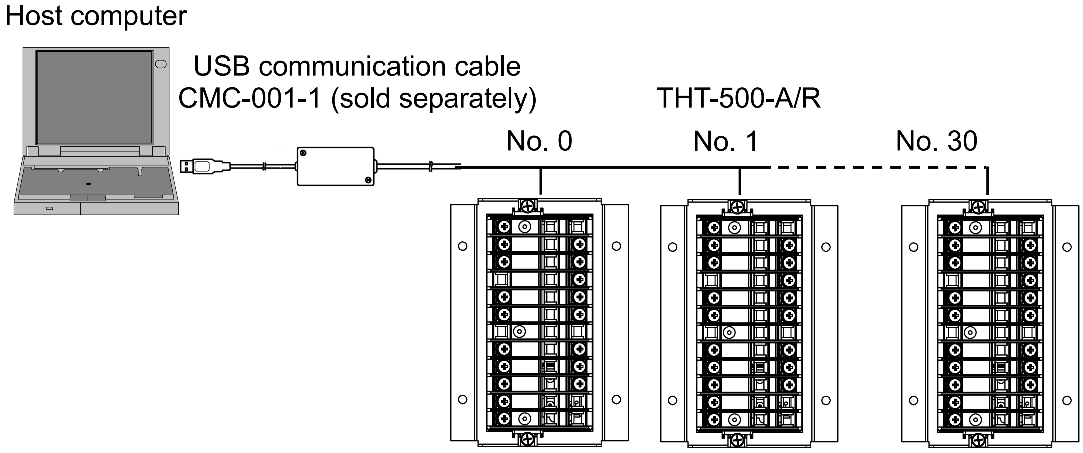
COMMUNICATION INSTRUCTION MANUAL **THT-500-A/R (C5)**

No. THT5CE2 2018.04

This manual contains instructions for communication functions of the THT-500-A/R.

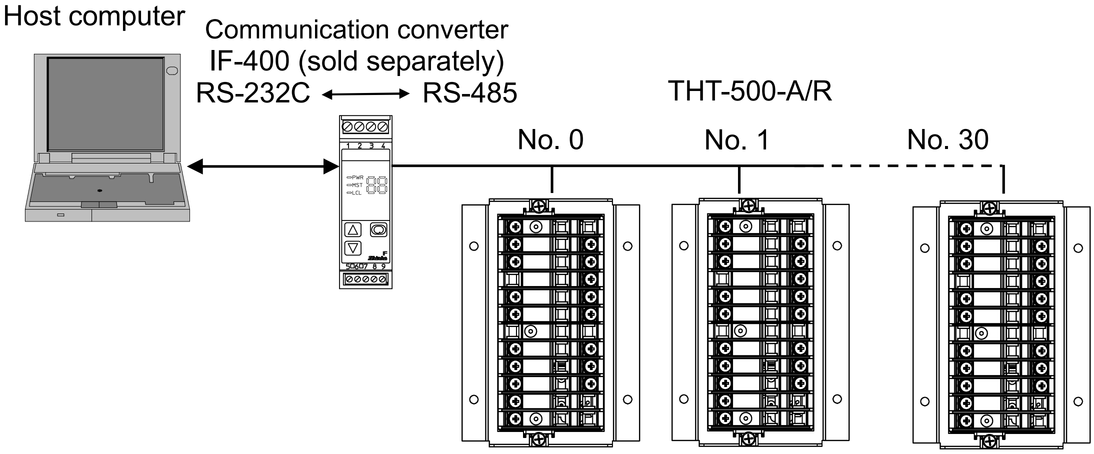
1. System Configuration

1.1 When Using USB Communication Cable CMC-001-1 (sold separately)



(Fig. 1.1-1)

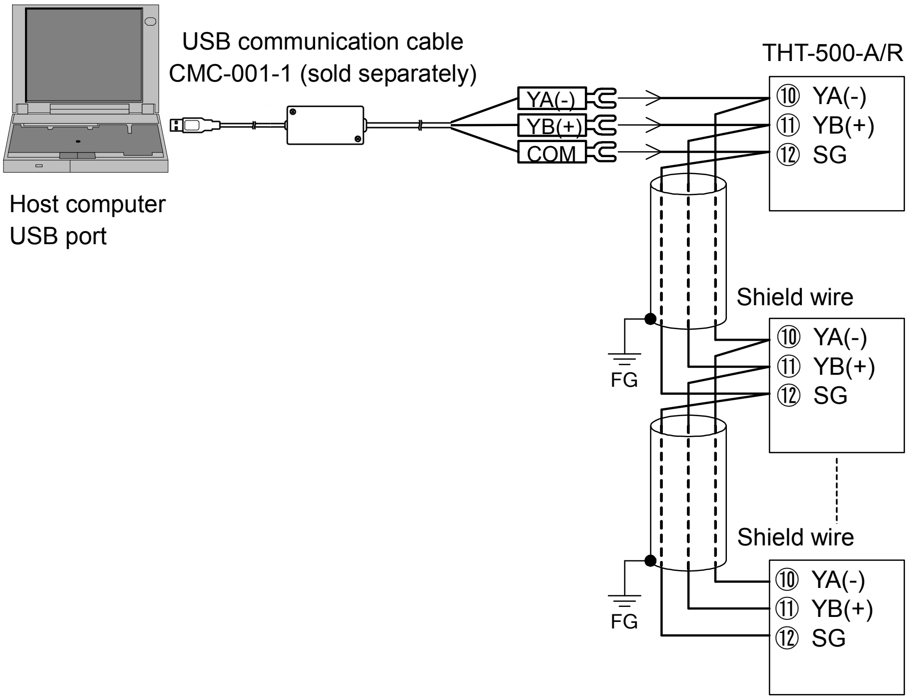
1.2 When Using Communication Converter IF-400 (sold separately)



(Fig. 1.2-1)

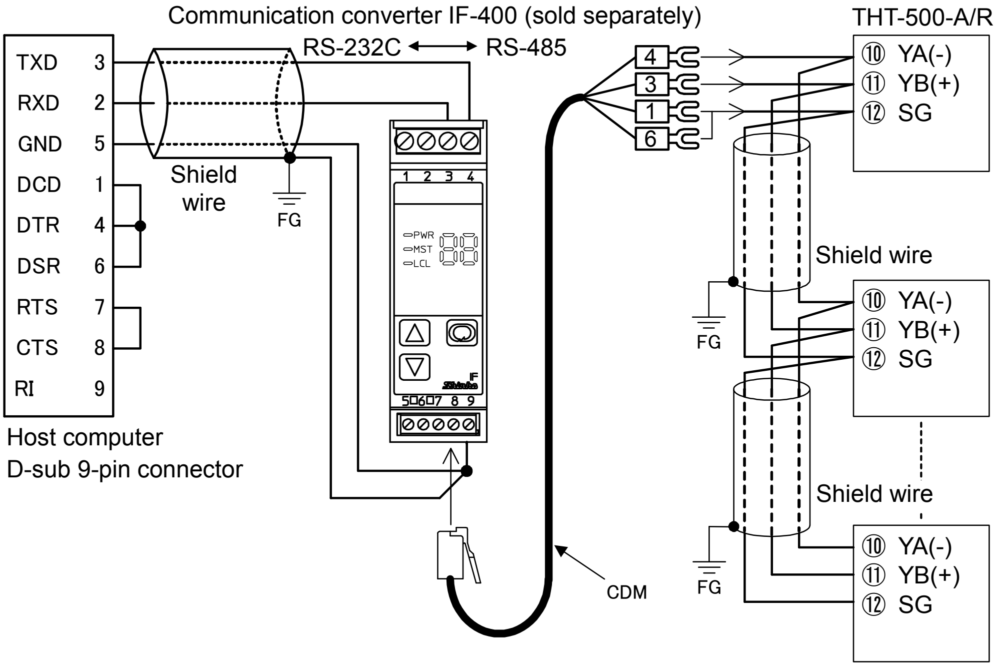
2. Wiring

2.1 When Using USB Communication Cable CMC-001-1 (sold separately)



(Fig. 2.1-1)

2.2 When Using Communication Converter IF-400 (sold separately)



(Fig. 2.2-1)

Shield wire

Connect only one end of the shield to the FG terminal to avoid a ground loop. If both ends of the shield wire are connected to the FG terminal, the circuit will be closed, resulting in a ground loop. This may cause noise. Be sure to ground the FG terminal.

Recommended cable: OTSC-VB 2PX0.5SQ (made by Onamba Co., Ltd.) or equivalent (Use a twisted pair cable.)

Terminator (Terminal resistor)

Communication converter IF-400 (sold separately) has a built-in terminator.

The terminator is mounted at the end of the wire when connecting multiple peripheral devices to a personal computer. The terminator prevents signal reflection and disturbance.

Do not connect a terminator to the communication line because each THT-500-A/R has built-in pull-up and pull-down resistors.

3. Setting Communication Parameters

Set communication parameters via communication.
If communication parameters are changed, the changed parameters will be effective when the power to the THT-500-A/R unit is turned OFF, then ON.

Refer to factory default values below.

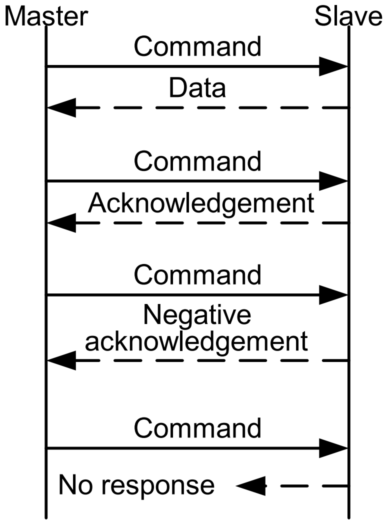
Data Item		Factory Default
0001H	Communication protocol	Shinko protocol
0002H	Instrument number	0
0003H	Communication speed	9600 bps
0004H	Data bit/Parity	7 bits/Even
0005H	Stop bit	1 bit
0006H	Response delay time	10 ms

Data Item of Communication Parameters

Data Item	Explanation of Data Item, Setting Range (Selection Item)	
0001H	**Communication protocol** • Selects communication protocol. • Selection item:	
	0000H	Shinko protocol (Factory default)
	0001H	Modbus ASCII
	0002H	Modbus RTU
0002H	**Instrument number** • Sets the instrument number. The instrument numbers should be set one by one when multiple instruments are connected in Serial communication, otherwise communication is impossible. • Setting range: 0 to 95 (Factory default: 0)	
0003H	**Communication speed** • Selects a communication speed equal to that of the host computer. • Selection item:	
	0000H	9600 bps (Factory default)
	0001H	19200 bps
	0002H	38400 bps
0004H	**Data bit/Parity** • Selects data bit and parity. • Selection item:	
	0000H	8 bits/No parity
	0001H	7 bits/No parity
	0002H	8 bits/Even
	0003H	7 bits/Even (Factory default)
	0004H	8 bits/Odd
	0005H	7 bits/Odd
0005H	**Stop bit** • Selects the stop bit. • Selection item:	
	0000H	1 bit (Factory default)
	0001H	2 bits
0006H	**Response delay time** • Response from the controller can be delayed after receiving command from the host computer. • Setting range: 0 to 1000 ms (Factory default: 10 ms)	

4. Communication Procedure

Communication starts with command transmission from the host computer (hereafter Master) and ends with the response of the THT-500-A/R (hereafter Slave).



(Fig. 4-1)

- **Response with data**
  When the master sends the Read command, the slave responds with the corresponding set value or current status.
- **Acknowledgement**
  When the master sends the Write command, the slave responds by sending the acknowledgement after the processing is terminated.
- **Negative acknowledgement**
  When the master sends a non-existent command or value out of the setting range, the slave returns a negative acknowledgement.
- **No response**
  The slave will not respond to the master in the following cases:
  - Global address (Shinko protocol) is set.
  - Broadcast address (Modbus protocol) is set.
  - Communication error (framing error, parity error)
  - Checksum error (Shinko protocol), LRC discrepancy (Modbus ASCII mode), CRC-16 discrepancy (Modbus RTU mode)

Communication timing of the RS-485

Master Side (Take note while programming)

When the master starts transmission through the RS-485 communication line, the master is arranged so as to provide an idle status (mark status) transmission period of 1 or more characters before sending the command to ensure synchronization on the receiving side.

Set the program so that the master can disconnect the transmitter from the communication line within a 1 character transmission period after sending the command in preparation for reception of the response from the slave.

To avoid collision of transmissions between the master and the slave, send the next command after carefully checking that the master has received the response.

If a response to the command is not returned due to communication errors, set the Retry Processing to send the command again. (It is recommended to execute Retry twice or more.)

Slave Side

When the slave starts transmission through the RS-485 communication line, the slave is arranged so as to provide an idle status (mark status) transmission period of 1 ms or more (*) before sending the response to ensure synchronization on the receiving side.

The slave is arranged so as to disconnect the transmitter from the communication line within a 1 character transmission period after sending the response.

(*) Can be set in [Response delay time] within a range of 0 to 1000 ms. (See p.4.)

5. Shinko Protocol

5.1 Transmission Mode

Shinko protocol is composed of ASCII.

Hexadecimal (0 to 9, A to F), which is divided into high order (4-bit) and low order (4-bit) out of 8-bit binary data in command is transmitted as ASCII characters.

Data format Start bit: 1 bit
Data bit: 7 bits
Parity: Even
Stop bit: 1 bit

Error detection: Checksum

5.2 Command Configuration

All commands are composed of ASCII.

The data (set value, decimal number) is represented by a hexadecimal number.

The negative numbers are represented in 2's complement.

Numerals written below the command represent number of characters.

(1) Write command

• Write a single piece of data

Header (02H)	Address	Sub address (20H)	Command type (50H)	Data item	Data	Checksum	Delimiter (03H)
1	1	1	1	4	4	2	1

(2) Read command

• Read a single piece of data

Header (02H)	Address	Sub address (20H)	Command type (20H)	Data item	Checksum	Delimiter (03H)
1	1	1	1	4	2	1

(3) Response with data

• Response to ‘Read a single piece of data’

Header (06H)	Address	Sub address (20H)	Command type (20H)	Data item	Data	Checksum	Delimiter (03H)
1	1	1	1	4	4	2	1

(4) Acknowledgement

Header (06H)	Address	Checksum	Delimiter (03H)
1	1	2	1

(5) Negative acknowledgement

Header (15H)	Address	Error code	Checksum	Delimiter (03H)
1	1	1	2	1

Header: Control code to represent the beginning of the command or the response.
ASCII is used.
Write command, Read command: STX (02H) fixed
Response with data, Acknowledgement: ACK (06H) fixed
Negative acknowledgement: NAK (15H) fixed

Instrument number (Address): Numbers by which the master discerns each slave.
Instrument number 0 to 94 and Global address 95.
ASCII (20H to 7FH) is used by adding 20H to instrument numbers 0 to 95 (00H to 5FH).
95 (7FH) is called Global address, which is used when the same command is sent to all the slaves connected. However, the response is not returned.

Sub address: 20H fixed

Command type: Code to discern Write command and Read command.

Command Type	Contents	Description
20H	Read a single piece of data	Reads a single piece of data.
50H	Write a single piece of data	Writes a single piece of data.

Data item: Classification of the command object.
Composed of 4-digit hexadecimal numbers, using ASCII.
Refer to '7. Communication Command Table'. (p. 19)

Data: The contents of data (values) differ depending on the Write command.
Composed of 4-digit hexadecimal numbers, using ASCII.
Refer to '7. Communication Command Table'. (p. 19)

Checksum: 2-character data to detect communication errors.
Refer to 5.3 Checksum Calculation on p. 8.

Delimiter: Control code to represent the end of command.
ASCII code ETX (03H) fixed

Error code: Represents an error type using ASCII.

Error Code	Contents
1 (31H)	Non-existent command
2 (32H)	Not used
3 (33H)	Value outside the setting range
4 (34H)	Not used
5 (35H)	Not used

5.3 Checksum Calculation

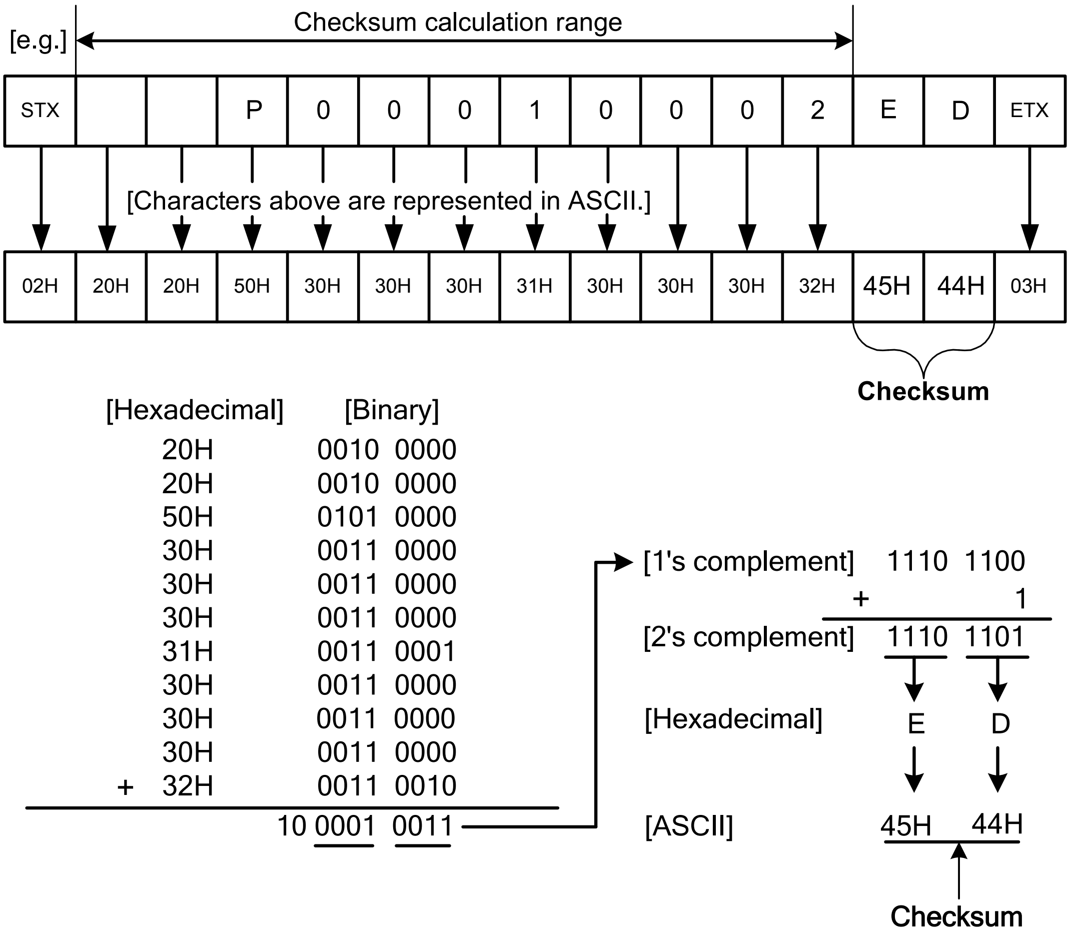
Checksum is used to detect receiving errors in the command or data.
Set the program for the master side as well to calculate the checksum of the response data from the slaves so that communication errors can be checked.

The ASCII code (hexadecimal) corresponding to the characters which range from the address to that before the checksum is converted to binary notation, and the total value is calculated.
The lower one byte of the total value is converted to 2’s complement, and then to hexadecimal numbers, that is, ASCII code for the checksum.

- 1’s complement: Reverse each binary bit. 0 will become 1 and vice versa.
- 2’s complement: Add 1 to 1’s complement.

[Example of checksum calculation]

When writing Modbus RTU (0002H) to Communication protocol (0001H):
Address (instrument number): 0 (20H)



(Fig. 5.3-1)

5.4 Command Example

Numerals written below the command represent number of characters.

(1) Read [Address 1, Wet bulb input value (0080H)]

- Read command from the master

Header	Address	Sub	Command	Data item	Checksum	Delimiter
		address	type	[0080H]		
(02H)	(21H)	(20H)	(20H)	(30H 30H 38H 30H)	(44H 37H)	(03H)
1	1	1	1	4	2	1

- A response from the slave in normal status [When wet bulb input value is 25℃ (0019H)]

Header	Address	Sub	Command	Data item	Data	Checksum	Delimiter
		address	type	[0080H]	[0019H]		
(06H)	(21H)	(20H)	(20H)	(30H 30H 38H 30H)	(30H 30H 31H 39H)	(30H 44H)	(03H)
1	1	1	1	4	4	2	1

(2) Write [Address 1, Communication protocol (0001H)]

- Write command from the master [When writing Modbus RTU (0002H) to Communication protocol]

Header	Address	Sub	Command	Data item	Data	Checksum	Delimiter
		address	type	[0001H]	[0002H]		
(02H)	(21H)	(20H)	(50H)	(30H 30H 30H 31H)	(30H 30H 30H 32H)	(45H 43H)	(03H)
1	1	1	1	4	4	2	1

- A response from the slave in normal status

Header	Address	Checksum	Delimiter
(06H)	(21H)	(44H 46H)	(03H)
1	1	2	1

(3) Read [Address 1, Communication protocol (0001H)]

- Read command from the master

Header	Address	Sub	Command	Data item	Checksum	Delimiter
		address	type	[0001H]		
(02H)	(21H)	(20H)	(20H)	(30H 30H 30H 31H)	(44H 45H)	(03H)
1	1	1	1	4	2	1

- A response from the slave in normal status [When Modbus RTU (0002H) is selected for the Communication protocol]

Header	Address	Sub	Command	Data item	Data	Checksum	Delimiter
		address	type	[0001H]	[0002H]		
(06H)	(21H)	(20H)	(20H)	(30H 30H 30H 31H)	(30H 30H 30H 32H)	(31H 43H)	(03H)
1	1	1	1	4	4	2	1

6. Modbus Protocol

6.1 Transmission Mode

There are 2 transmission modes (ASCII and RTU) in Modbus protocol.

6.1.1 ASCII Mode

Hexadecimal (0 to 9, A to F), which is divided into high order (4-bit) and low order (4-bit) out of 8-bit binary data in command is transmitted as ASCII characters.

Data format	Start bit:	1 bit
	Data bit:	7 bits (8 bits) (Selectable)
	Parity:	Even (No parity, Odd) (Selectable)
	Stop bit:	1 bit (2 bits) (Selectable)

Error detection : LRC (Longitudinal Redundancy Check)

6.1.2 RTU Mode

8-bit binary data in command is transmitted as it is.

Data format	Start bit:	1 bit
	Data bit:	8 bits
	Parity:	No parity (Even, Odd) (Selectable)
	Stop bit:	1 bit (2 bits) (Selectable)

Error detection: CRC-16 (Cyclic Redundancy Check)

6.2 Data Communication Interval

6.2.1 ASCII Mode

No communication interval limit between characters

6.2.2 RTU Mode

1.5 character transmission times or less
(Communication speed 9600 bps, 19200 bps: 1.5 character transmission times,
Communication speed 38400 bps: 750 μs)
To transmit continuously, an interval between characters which consist of one message, must be within 1.5 character transmission times.
If an interval lasts longer than 1.5 character transmission times, the THT-500-A/R assumes that transmission from the master is finished, which results in a communication error, and will not return a response.

6.3 Message Configuration

6.3.1 ASCII Mode

ASCII mode message is configured to start by Header [: (colon) (3AH)] and end by Delimiter [CR (carriage return) (0DH) + LF (Line feed) (0AH)].
Data section: Max. 2 x 252 characters

Header (:)	Slave address	Function code	Data	Error check LRC	Delimiter (CR)	Delimiter (LF)

6.3.2 RTU Mode

RTU mode is configured to start after idle time is processed for more than 3.5 character transmissions, and end after idle time is processed for more than 3.5 character transmissions.
(Communication speed 9600 bps, 19200 bps: 3.5 character transmission times,
Communication speed 38400 bps: 1.75 ms)
Data section: Max. 252 bytes

3.5 idle characters	Slave address	Function code	Data	Error check CRC-16	3.5 idle characters

(1) Slave Address

Slave address is an individual instrument number on the slave side, and is set within the range 0 to 95 (00H to 5FH). The master identifies slaves by the slave address of the requested message. The slave informs the master which slave is responding to the master by placing its own address in the response message.

Slave address 0 (00H, Broadcast address) can identify all the slaves connected. However, slaves do not respond.

(2) Function Code

The function code is the command code for the slave to undertake one of the following actions.

Type	Function Code	Sub-Function Code	Contents
Data access	03 (03H)		Reads a single piece of data from slave(s).
	06 (06H)		Writes a single piece of data to slave(s).
Diagnostics	08 (08H)	00	Echoes back the request message.
	43 (2BH)	14	Reads device identification information.

The function code is used to discern whether the response is normal (acknowledgement) or if any error (negative acknowledgement) has occurred when the slave returns the response message to the master.

When acknowledgement is returned, the slave simply returns the original function code.

When negative acknowledgement is returned, the MSB of the original function code is set as 1 for the response.

For example, if the master sends request message setting 13H to the function code by mistake, slave returns 93H by setting the MSB to 1, because the former is an illegal function.

For negative acknowledgement, the exception codes below are set to the data of the response message, and returned to the master in order to inform it of what kind of error has occurred.

Exception Code	Contents
1 (01H)	Illegal function (Non-existent function)
2 (02H)	Illegal data address (Non-existent data address)
3 (03H)	Illegal data value (Value out of the setting range)
17 (11H)	Not used
18 (12H)	Not used

(3) Data

Data differs depending on the function code.

A request message from the master is composed of a data item, amount of data and setting data.

A response message from the slave is composed of the number of bytes, data and exception codes in negative acknowledgements, corresponding to the request message.

The effective range of data is -32768 to 32767 (8000H to 7FFFH).

Refer to Section “7. Communication Command Table” (p. 19).

(4) Error Check

ASCII Mode

After calculating LRC (Longitudinal Redundancy Check) from the slave address to the end of data, the calculated 8-bit data is converted to two ASCII characters, and are appended to the end of message.

How to Calculate LRC

① Create a message in RTU mode.
② Add all the values from the slave address to the end of data. This is assumed as X.
③ Make a complement for X (bit reverse). This is assumed as X.
④ Add a value of 1 to X. This is assumed as X.
⑤ Set X as an LRC to the end of the message.
⑥ Convert the whole message to ASCII characters.

RTU Mode

After calculating CRC-16 (Cyclic Redundancy Check) from the slave address to the end of the data, the calculated 16-bit data is appended to the end of message in sequence from low order to high order.

How to calculate CRC-16

In the CRC-16 system, the information is divided by the polynomial series. The remainder is added to the end of the information and transmitted. The generation of a polynomial series is as follows.
(Generation of polynomial series: $X^{16} + X^{15} + X^{2} + 1$)

① Initialize the CRC-16 data (assumed as X) (FFFFH).
② Calculate exclusive OR (XOR) with the 1st data and X. This is assumed as X.
③ Shift X one bit to the right. This is assumed as X.
④ When a carry is generated as a result of the shift, XOR is calculated by X of ③ and the fixed value (A001H). This is assumed as X. If a carry is not generated, go to step ⑤.
⑤ Repeat steps ③ and ④ until shifting 8 times.
⑥ XOR is calculated with the next data and X. This is assumed as X.
⑦ Repeat steps ③ to ⑤.
⑧ Repeat steps ③ to ⑤ up to the final data.
⑨ Set X as CRC-16 to the end of message in sequence from low order to high order.

6.4 Message Example

6.4.1 ASCII Mode

Numerals written below the command represent the number of characters.

(1) Read [Slave address 1, Wet bulb input value (0080H)]

- A request message from the master

Header	Slave address	Function code	Data item	Amount of data	Error check LRC	Delimiter CR+LF
			[0080H]	[0001H]		
(3AH)	(30H 31H)	(30H 33H)	(30H 30H 38H 30H)	(30H 30H 30H 31H)	(37H 42H)	(0DH 0AH)
1	2	2	4	4	2	2

- Response message from the slave in normal status [When wet bulb input value is 25℃ (0019H)]

Header	Slave address	Function code	Number of response bytes	Data	Error check LRC	Delimiter CR+LF
			[02H]	[0019H]		
(3AH)	(30H 31H)	(30H 33H)	(30H 32H)	(30H 30H 31H 39H)	(45H 31H)	(0DH 0AH)
1	2	2	2	4	2	2

(2) Write [Slave address 1, Communication protocol (0001H)]

- A request message from the master [When writing Modbus RTU (0002H) to Communication protocol]

Header	Slave address	Function code	Data item	Data	Error check LRC	Delimiter CR+LF
			[0001H]	[0002H]		
(3AH)	(30H 31H)	(30H 36H)	(30H 30H 30H 31H)	(30H 30H 30H 32H)	(46H 36H)	(0DH 0AH)
1	2	2	4	4	2	2

- Response message from the slave in normal status

Header	Slave address	Function code	Data item	Data	Error check LRC	Delimiter CR+LF
			[0001H]	[0002H]		
(3AH)	(30H 31H)	(30H 36H)	(30H 30H 30H 31H)	(30H 30H 30H 32H)	(46H 36H)	(0DH 0AH)
1	2	2	4	4	2	2

- Response message from the slave in exception (error) status (When a value out of the setting range is set)

The function code MSB is set to 1 for the response message in exception (error) status [86H (38H 36H)].

The exception code 03H (30H 33H: Value out of the setting range) is returned (error).

Header	Slave address	Function code	Exception code	Error check LRC	Delimiter CR+LF
			[03H]		
(3AH)	(30H 31H)	(38H 36H)	(30H 33H)	(37H 36H)	(0DH 0AH)
1	2	2	2	2	2

(3) Read [Slave address 1, Communication protocol (0001H)]

- A request message from the master

Header	Slave address	Function code	Data item	Amount of data	Error check LRC	Delimiter CR+LF
			[0001H]	[0001H]		
(3AH)	(30H 31H)	(30H 33H)	(30H 30H 30H 31H)	(30H 30H 30H 31H)	(46H 41H)	(0DH 0AH)
1	2	2	4	4	2	2

- Response message from the slave in normal status [When Modbus RTU (0002H) is selected for the Communication protocol]

Header	Slave address	Function code	Number of response bytes	Data	Error check LRC	Delimiter CR+LF
			[02H]	[0002H]		
(3AH)	(30H 31H)	(30H 33H)	(30H 32H)	(30H 30H 30H 32H)	(46H 38H)	(0DH 0AH)
1	2	2	2	4	2	2

- Response message from the slave in exception (error) status (When data item is incorrect)

The function code MSB is set to 1 for the response message in exception (error) status [83H (38H 33H)].

The exception code 02H (30H 32H: Non-existent data address) is returned (error).

Header	Slave address	Function code	Exception code	Error check LRC	Delimiter CR+LF
			[02H]		
(3AH)	(30H 31H)	(38H 33H)	(30H 32H)	(37H 41H)	(0DH 0AH)
1	2	2	2	2	2

6.4.2 RTU Mode

Numerals written below the command represent number of characters.

(1) Read [Slave address 1, Wet bulb input value (0080H)]

- A request message from the master

3.5 idle characters	Slave address (01H)	Function code (03H)	Data item (0080H)	Amount of data (0001H)	Error check CRC-16 (85E2H)	3.5 idle characters
	1	1	2	2	2	

- Response message from the slave in normal status [When wet bulb input value is 25℃ (0019H)]

3.5 idle characters	Slave address (01H)	Function code (03H)	Number of response bytes (02H)	Data (0019H)	Error check CRC-16 (798EH)	3.5 idle characters
	1	1	1	2	2	

(2) Write [Slave address 1, Communication protocol (0001H)]

- A request message from the master [When writing Modbus RTU (0002H) to Communication protocol]

3.5 idle characters	Slave address (01H)	Function code (06H)	Data item (0001H)	Data (0002H)	Error check CRC-16 (59CBH)	3.5 idle characters
	1	1	2	2	2	

- Response message from the slave in normal status

3.5 idle characters	Slave address (01H)	Function code (06H)	Data item (0001H)	Data (0002H)	Error check CRC-16 (59CBH)	3.5 idle characters
	1	1	2	2	2	

- Response message from the slave in exception (error) status (When a value out of the setting range is set)

  The function code MSB is set to 1 for the response message in exception (error) status, and 86H is returned.

  The Exception code 03H (Value out of the setting range) is returned (error).

3.5 idle characters	Slave address (01H)	Function code (86H)	Exception code (03H)	Error check CRC-16 (0261H)	3.5 idle characters
	1	1	1	2	

(3) Read [Slave address 1, Communication protocol (0001H)]

- A request message from the master

3.5 idle characters	Slave address (01H)	Function code (03H)	Data item (0001H)	Data (0001H)	Error check CRC-16 (D5CAH)	3.5 idle characters
	1	1	2	2	2	

- Response message from the slave in normal status [When writing Modbus RTU (0002H) to Communication protocol]

3.5 idle characters	Slave address (01H)	Function code (03H)	Number of response bytes (02H)	Data (0002H)	Error check CRC-16 (3985H)	3.5 idle characters
	1	1	1	2	2	

- Response message from the slave in exception (error) status (When data item is incorrect)

  The function code MSB is set to 1 for the response message in exception (error) status, and 83H is returned. The Exception code 02H (Non-existent data address) is returned (error).

3.5 idle characters	Slave address (01H)	Function code (83H)	Exception code (02H)	Error check CRC-16 (C0F1H)	3.5 idle characters
	1	1	1	2	

6.5 Diagnostics Function

Modbus protocol has the following diagnostics functions.

- Echoes back the request message.
- Reads device identification information.

6.5.1 Message Configuration

ASCII mode

Header (:)	Slave address	Function code	Data	Error check LRC	Delimiter (CR)	Delimiter (LF)

RTU mode

3.5 idle characters	Slave address	Function code	Data	Error check CRC-16	3.5 idle characters

(1) Slave address:

Slave address is an individual instrument number on the slave side, and is set within the range 1 to 95 (01H to 5FH).

The master identifies slaves by the slave address of the requested message.

The slave informs the master which slave is responding to the master by placing its own address in the response message.

Diagnostics function is disabled for the Slave address 0 (00H, broadcast address).

(2) Function code:

The function code is the command code for the slave to undertake the following action types.

Type	Function Code	Sub-function Code	Contents
Diagnostics	08 (08H)	00 (0000H)	Echoes back the request message.
	43 (2BH)	14 (0EH)	Reads device identification information.

Function code is used to discern whether the response is normal (acknowledgement) or if any error (negative acknowledgement) has occurred when the slave returns the response message to the master.

When acknowledgement is returned, the slave simply returns the original function code.

When negative acknowledgement is returned, the MSB of the original function code is set as 1 for the response.

For example, when the master sends request message setting 0FH to the sub-function code by mistake, slave returns ABH by setting the MSB to 1, because the former is a non-existent sub-function code.

For negative acknowledgement, the exception codes below are set to the data of the response message, and returned to the master in order to inform it of what kind of error has occurred.

Exception Code	Contents
1 (01H)	Illegal function (Non-existent function) Sub-function code is not correct.
2 (02H)	Illegal data address (Non-existent data address) For function code 43: Object ID is any value other than 00, 01, 02.
3 (03H)	Illegal data value (Value out of the setting range) For function code 08: Data is less than 1, or has exceeded 100. For function code 43: Read Device ID code is any value other than 01, 04.

(3) Data

Data differs depending on the function code.

For the Function code 08 (08H), a request message from the master side is composed of 'Sub-function code 2 Bytes (0000H)' and 'Data n x 2 Bytes'.

[n: Amount of data (Max. 100)]

In normal status, a response message from the slave side is the same as the request message.

Function code	1 Byte	08H
Sub-function code	1 Byte	0000H fixed
Data	n x 2 Bytes	Random value (Max. 100)

For Function code 43 (2BH), the request message from the master side is composed of Sub-function code 14 (0EH), Read Device ID code and Object ID.

Function code	1 Byte	2BH	
Sub-function code (MEI type)	1 Byte	0EH	
Read Device ID code (Corresponds to Basic category)	1 Byte	01H/04H	
Object ID	1 Byte	00	Vendor name SHINKO TECHNOS CO., LTD.
		01	Product code (model) (e.g.) THT-500-A/R
		02	Version number (D, T, MP) (e.g.) Dxx-xxxx-xx, MPxxxx-xx

Response message from the slave is composed of Sub-function code 14 (0EH) (for request), Read Device ID code and Object ID.

Function code		1 Byte	2BH
Sub-function code (MEI type)		1 Byte	0EH
Data	Read Device ID code	1 Byte	01H/04H
	Conformity level	1 Byte	01H/81H
	More Follows	1 Byte	00H/FFH
	Next Object ID	1 Byte	Object ID number
	Number of Objects	1 Byte	
	List of Object ID	1 Byte	
	List of Object length	1 Byte	
	List of Object value	Object length	

For the response message (negative acknowledgement), an exception code is set and returned.

Function code	1 Byte	ABH
Exception code	1 Byte	01H/02H/03H

(4) Error check:

16-bit data to detect communication errors.

Refer to Sections ‘6.3 Message Configuration (4) Error Check (pp.11, 12).

6.5.2 Message Example

Message example in RTU mode are shown below.

Numerals written below the command represent the number of characters.

(1) Echo back Slave address 1, Request message

- A request message from the master [Test data 200 (00C8H), 60 (003CH), 10 (000AH)]

3.5 idle characters	Slave address (01H)	Function code (08H)	Sub-function code (0000H)	Data (00C8003C000AH)	Error check CRC-16 (E7D9H)	3.5 idle characters
	1	1	2	n x 2	2	

- Response message from the slave in normal status (Echoes back the same message.)

3.5 idle characters	Slave address (01H)	Function code (08H)	Sub-function code (0000H)	Data (00C8003C000AH)	Error check CRC-16 (E7D9H)	3.5 idle characters
	1	1	2	n x 2	2	

(2) Read Slave address 1, Device identification information (Vendor name)

- A request message from the master

3.5 idle characters	Slave address (01H)	Function code (2BH)	Sub-function code (0EH)	Data (0400H)	Error check CRC-16 (7327H)	3.5 idle characters
	1	1	1	2	2	

- Response message from the slave in normal status (SHINKO TECHNOS CO., LTD.)

3.5 idle characters	Slave address (01H)	Function code (2BH)	Sub-function code (0EH)
	1	1	1

Data (048100000100185348494E4B4F20544543484E4F5320434F2E2C204C54442EH)
31

Error check CRC-16 (1C54H)	3.5 idle characters
2	

Data in the response message becomes as follows.

Data	Read Device ID code	1 Byte	04H
	Conformity level	1 Byte	81H
	More Follows	1 Byte	00H
	Next Object ID	1 Byte	00H
	Number of Objects	1 Byte	01H
	List of Object ID	1 Byte	00H
	List of Object length	1 Byte	24(18H)
	List of Object value	Object length	S(53H)
			H(48H)
			I(49H)
			N(4EH)
			K(4BH)
			O(4FH)
			(20H)
			T(54H)
			E(45H)
			C(43H)
			H(48H)
			N(4EH)
			O(4FH)
			S(53H)
			(20H)
			C(43H)
			O(4FH)
			.(2EH)
			,(2CH)
			(20H)
			L(4CH)
			T(54H)
			D(44H)
			.(2EH)

(3) Read Slave address 1, Device identification information (Product code)

- A request message from the master

3.5 idle characters	Slave address (01H)	Function code (2BH)	Sub-function code (0EH)	Data (0401H)	Error check CRC-16 (B2E7H)	3.5 idle characters
	1	1	1	2	2	

- Response message from the slave in normal status (THT-500-A/R)

3.5 idle characters	Slave address (01H)	Function code (2BH)	Sub-function code (0EH)
	1	1	1

Data (0481000001010B5448542D3530302D412F52H)	Error check CRC-16 (A55FH)	3.5 idle characters
18	2	

Data in the response message becomes as follows.

Data	Read Device ID code	1 Byte	04H
	Conformity level	1 Byte	81H
	More Follows	1 Byte	00H
	Next Object ID	1 Byte	00H
	Number of Objects	1 Byte	01H
	List of Object ID	1 Byte	01H
	List of Object length	1 Byte	11(0BH)
	List of Object value	Object length	T(54H)
			H(48H)
			T(54H)
			-(2DH)
			5(35H)
			0(30H)
			0(30H)
			-(2DH)
			A(41H)
			/(2FH)
			R(52H)

- Response message from the slave in exception (error) status [when Sub-function code (MEI type) is incorrect]
  The function code MSB is set to 1 for the response message in exception (error) status, and ABH is returned.
  Exception code 01H (Non-existent function) is returned (error).

3.5 idle characters	Slave address (01H)	Function code (ABH)	Exception code (01H)	Error check CRC-16 (9EF0H)	3.5 idle characters
	1	1	1	2	

7. Communication Command Table

7.1 A Single Piece of Data Read/Write Command

Shinko Command Type	Modbus Function Code	Data Item		Data
20H/50H	03H/06H	0001H	Communication protocol (*)	0000H: Shinko protocol 0001H: Modbus ASCII 0002H: Modbus RTU
20H/50H	03H/06H	0002H	Instrument number (*)	Set value (0 to 95)
20H/50H	03H/06H	0003H	Communication speed (*)	0000H: 9600 bps 0001H: 19200 bps 0002H: 38400 bps
20H/50H	03H/06H	0004H	Data bit/Parity (*)	0000H: 8 bits/No parity 0001H: 7 bits/No parity 0002H: 8 bits/Even 0003H: 7 bits/Even 0004H: 8 bits/Odd 0005H: 7 bits/Odd
20H/50H	03H/06H	0005H	Stop bit (*)	0000H: 1 bit 0001H: 2 bits
20H/50H	03H/06H	0006H	Response delay time (*)	Set value (0 to 1000)

(*) If communication parameters are changed, the changed parameters will be effective when the power to the THT-500-A/R unit is turned OFF, then ON.

7.2 Read Command

Shinko Command Type	Modbus Function Code	Data Item		Data
20H	03H	0080H	Wet bulb input value	Read value
20H	03H	0081H	Humidity converted value	Read value
20H	03H	0082H	Humidity output value	Read value
20H	03H	0083H	Status flag	2^0 digit: Wet bulb input temperature sensor burnout 0: No 1: Yes 2^1 digit: Wet bulb input temperature sensor short circuit 0: No 1: Yes 2^2 digit: Wet bulb input high limit error(Exceeding 100℃) 0: No 1: Yes 2^3 digit: Wet bulb input low limit error (Less than -25℃) 0: No 1: Yes 2^4 digit: Dry bulb input temperature sensor burnout 0: No 1: Yes 2^5 digit: Dry bulb input temperature sensor short circuit 0: No 1: Yes 2^6 digit: Dry bulb input high limit error(Exceeding 225℃) 0: No 1: Yes 2^7 digit: Dry bulb input low limit error (Less than -25℃) 0: No 1: Yes 2^8 digit: Output selection 0: 4 to 20 mA 1: 0 to 20 mA 2^9 to 2^{15} digit: Undefined
20H	03H	0090H	Dry bulb input value	Read value
20H	03H	0091H	Temperature output value	Read value
20H	03H	00A0H	Software version	Upper byte: Version
20H	03H	00A1H	Instrument model information	2^0 digit: Dry bulb input range 0 to 200℃ 0: No 1: Yes 2^1 digit: Serial communication 0: No 1: Yes 2^2 to 2^{15} digit: Undefined

7.3 Data

7.3.1 Notes about Write/Read Command

- The data (set value, decimal) is converted to a hexadecimal number.
  Negative numbers are represented in 2's complement.
- When connecting multiple slaves, the address (instrument number) must not be duplicated.
- Do not use undefined Data items. If they are used, negative acknowledgement will be returned or a random value will be written or read, resulting in malfunction.
- Modbus protocol uses Holding Register addresses. The Holding Register addresses are created as follows. A Shinko command data item is converted to decimal number, and the offset of 40001 is added. The result is the Holding Register address.
  Using Data item 0001H (Communication protocol) as an example: Data item in the sending message is 0001H, however, Modbus protocol Holding Register address is 40002 (1 + 40001).

7.3.2 Write Command

- Up to 1,000,000 (one million) entries can be stored in non-volatile IC memory.
  If the number of settings exceeds the limit, the data will not be saved. So, do not change the set values frequently via software communication. (If a value set via software communication is the same as the value before the setting, the value will not be written in non-volatile IC memory.)
- When Write is executed using the Global address [95 (7FH), Shinko protocol] command or Broadcast address [0 (00H) Modbus protocol] command, the command is sent to all the connected slaves. However, a response is not returned.

7.4 Negative Acknowledgement

7.4.1 Error code 1 (31H, Shinko protocol) or Exception code 2 (02H, Modbus protocol)

The slave will return Error code 1 (31H, Shinko protocol) or Exception code 2 (02H, Modbus protocol) in the following case.

- When non-existent data item is read or written.
- When Serial communication (C5 option) is not ordered, and Write is executed for communication parameters.

7.4.2 Error code 3 (33H, Shinko protocol) or Exception code 3 (03H, Modbus protocol)

The slave will return Error code 3 (33H, Shinko protocol) or Exception code 3 (03H, Modbus protocol) in the following case.

- When a value out of the setting range is written.

8. Specifications

<table>
<tr><td>Cable length</td><td colspan="4">1.2 km (Max.), Cable resistance: Within 50 Ω (Terminators are not necessary, but if used, use 120 Ω or more on both sides.)</td></tr>
<tr><td>Communication line</td><td colspan="4">EIA RS-485</td></tr>
<tr><td>Communication method</td><td colspan="4">Half-duplex communication</td></tr>
<tr><td>Communication speed</td><td colspan="4">9600, 19200, 38400 bps (Selectable via communication)</td></tr>
<tr><td>Synchronization method</td><td colspan="4">Start-stop synchronization</td></tr>
<tr><td>Code form</td><td colspan="4">ASCII, binary</td></tr>
<tr><td>Data bit/Parity</td><td colspan="4">7, 8 / Even, Odd, No parity (Selectable via communication)</td></tr>
<tr><td>Stop bit</td><td colspan="4">1, 2 (Selectable via communication)</td></tr>
<tr><td>Communication protocol</td><td colspan="4">Shinko protocol, Modbus ASCII, Modbus RTU (Selectable via communication)</td></tr>
<tr><td rowspan="6">Data format</td><td colspan="4"></td></tr>
<tr><td>Communication Protocol</td><td>Shinko Protocol</td><td>Modbus ASCII</td><td>Modbus RTU</td></tr>
<tr><td>Start bit</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Data bit</td><td>7</td><td>7 (8)
Selectable</td><td>8</td></tr>
<tr><td>Parity</td><td>Even</td><td>Even (No parity, Odd)
Selectable</td><td>No parity (Even, Odd)
Selectable</td></tr>
<tr><td>Stop bit</td><td>1</td><td>1 (2)
Selectable</td><td>1 (2)
Selectable</td></tr>
<tr><td>Number of connectable units</td><td colspan="4">Max 31 units to 1 host computer</td></tr>
<tr><td>Error correction</td><td colspan="4">Command request repeat system</td></tr>
<tr><td>Error detection</td><td colspan="4">Parity, checksum (Shinko protocol), LRC (Modbus ASCII), CRC-16 (Modbus RTU)</td></tr>
</table>

9. Troubleshooting

Check that power is being supplied to the master and slave that customers use. If communication failure still occurs, check the following.

Problem	Possible Cause	Solution
Communication failure	Communication cable is not securely connected, or is disconnected/defective.	Check the communication cable and connector.
	Incorrect wiring of the communication cable and/or connector	Check the communication cable and connector. Refer to Section ‘2. Wiring’ (pp. 2, 3).
	Imperfect contact between the communication cable and the connector, or between the communication connector and instrument port	Check the communication cable and connector.
	Communication speed of the slave does not match that of the master.	Set the same communication speed on the master and the slave. Refer to Section ‘3. Setting Communication Parameters’ (p. 4).
	The data bit, parity and stop bit of the master do not correspond to those of the slave.	Set the same data bit, parity and stop bit on the master and the slave. Refer to Section ‘3. Setting Communication Parameters’ (p. 4).
	The instrument number (address) of the slave does not correspond to that of the command.	Check the instrument number (address) of the slave and the command. Refer to Section ‘3. Setting Communication Parameters’ (p. 4).
	The instrument numbers (addresses) are duplicated in multiple slaves.	Check that each slave has a different instrument number (address). Refer to Section ‘3. Setting Communication Parameters’ (p. 4).
	Make sure that the program is appropriate for the transmission timing.	Check the program. Refer to Section ‘4. Communication Procedure’ (p. 5).
Although communication is occurring, the response is negative acknowledgement.	A non-existent command code has been sent.	Check the command code.
	The Write command data exceeds the setting range of the slave.	Check the setting range of the slave.

For all other malfunctions, please contact our main office or dealers.

SHINKO TECHNOS CO., LTD.
OVERSEAS DIVISION

Head Office : 2-5-1, Senbahigashi, Minoo, Osaka, Japan
URL: http://www.shinko-technos.co.jp/e/ Tel : +81-72-727-6100
E-mail: overseas@shinko-technos.co.jp Fax: +81-72-727-7006