Futaba

Pressure Measuring Unit

MPS01A

RS485 Communication Specifications

MOLD MARSHALLING SYSTEM
OUTPUT 1
OUTPUT 2
ALARM
SET
8.8.8.8.
MPa
Alm Cancel
SET
MPS01A
Futaba

1. Specifications

Item No.	Item	Specificati
1	Interface	EIA standard RS485 compliant
2	Transmission line	2-wire, multi-drop
3	Transmission distance	Up to 500 m
4	Transmission speed (bps)	19200
5	Communication method	Half-duplex
6	Synchronous method	Start-stop synchronization
7	Data format	Start bits: 1 Data bits: 8 Parity bits: none Stop bits: 1
8	Max connectivity	32 devices including master (due to the restriction at the physical layer)
9	Signal logic	V (A) ＞ V (B) 0 (SPACE) V (A) ＜ V (B) 1 (MARK)
10	Protocol	ModBus
11	Transmission mode	Remote Terminal Unit (RTU) mode
12	Function code	0 3 H (Read Holding Registers) 0 8 H (Loopback Diagnostic Test)
13	Error control	Parity check, CRC-16
14	Exception code	0 1 H (Illegal function) 0 2 H (Illegal data address) 0 3 H (Illegal data value) 0 4 H (Slave device failure)

2. Signal name

Signal name	Description
S G	Signal Ground
T ／ R (A)	Transmitted/Received Data
T ／ R (B)	Transmitted/Received Data

3. Communication parameter

Abbreviation	Name	Setting range	Description	Factory default
Adr	Slave address	1 to 247	Address to identify each controller	1
bPS	Transmission speed	19200	Transmission speed setting value	19200
bit	Data format	8n1	8 bits, no parity, 1 stop bit	8n1
int	Interval time [ms]	10 to 1000	Communication interval time setting value	10

Note 1) When any communication parameter is changed, it is necessary to power the system off and on again. Otherwise, the new value will not be used for communication.

Note 2) It is necessary to set the unit to have the same transmission speed as the master.

Note 3) It is necessary to set the unit to have the same data format as the master.

4. Message configuration

In ModBus protocol, the master (host computer) always starts a communication, and the slave (this measuring unit) responds to the master. One master and one or more slave devices (up to 247 devices) are connected to the network.

A communication is always initiated by the master. The slave devices will never transmit data without receiving a command (request) message from the master.

The salve devices will never communicate with each other.

The master issues only one command at once, and cannot transmit multiple commands at the same time.

・The master sends a command (request) message to a slave device.

・The slave device checks whether the address in the received message matches its own address.

・If matched, the slave executes the command and sends back the response message.

・If mismatched, the slave leaves the command message and wait for the next command message.

Message frame (1+1+(maximum address+1)×2+1+2 bytes at maximum)

Slave address	Function code	Data	Error detecting code (CRC)	
1 byte	1 byte	0～ (maximum address +1) ×2+1 byte	2 bytes CRCLow	 CRCHi

1) Slave address

This is a number necessary to identify the slave and addressed from 1 to 247.
The master transmits a signal to only one slave.
Handling of address 0
Address 0 is used as broadcast message from the master.
When receiving this message, the slave executes the command, but sends back no response message.

2) Function code

This is a code used to specify the action to perform.
A function code is followed by data associated with this command and the response to it.
For details, refer to “5. Function code”.

3) Error detecting code

This is a code used to detect errors in messages.
CRC-16 is used.
For details, refer to “CRC-16 calculation”.

5. Function code

〈Function〉

Code	Name	Function
03H	Read Holding Registers	Read the contents of a continuous block of holding register
08H	Diagnostic	Communication diagnostic

〈Message length〉

Code	Command message (byte)		Response message (byte)	
	Minimum	Maximum	Minimum	Maximum
03H	8	8	7	1+1+(maximum address×2+1)+2
06H	8	8	8	8
08H	8	8	8	8

5-1. 03H　Read Holding Registers (Read registers)

This function code is used to read the contents of the holding registers with the continuous numbers (address) for the specified quantity starting at the specified number (address).

The content of the holding register is divided into high order 8 bits and low order 8 bits, and packed in the response data in numerical order.

〈Command message〉

Slave address	Function code	Starting address	Quantity of registers	Error detecting code
０１H～Ｆ７H	０３H	００００H～ Maximum address H	０００１H～ Maximum address+1 H	
1 byte	1 byte	2 byte	2 byte	2 byte

Slave address	Function code	Number of bytes	Data	Error detecting code
Same as command message	Same as command message	０２～ (Maximum address+1)×2H		
1 byte	1 byte	1 byte	n byte	2 byte

〈Exception response〉

- If the slave receives a command message, but detects an error (other than communication error) with the content, the slave does not execute the command and returns an exception response message.
- Whenever the slave determines any of the content of a message as an error by means of its self-diagnostic function, the slave returns to an exception response message to the command message.

Slave address	Function code	Exception code	Error detecting code
Same as command message 1 byte	83H 1 byte	０１H～０４H 1 byte	2 byte

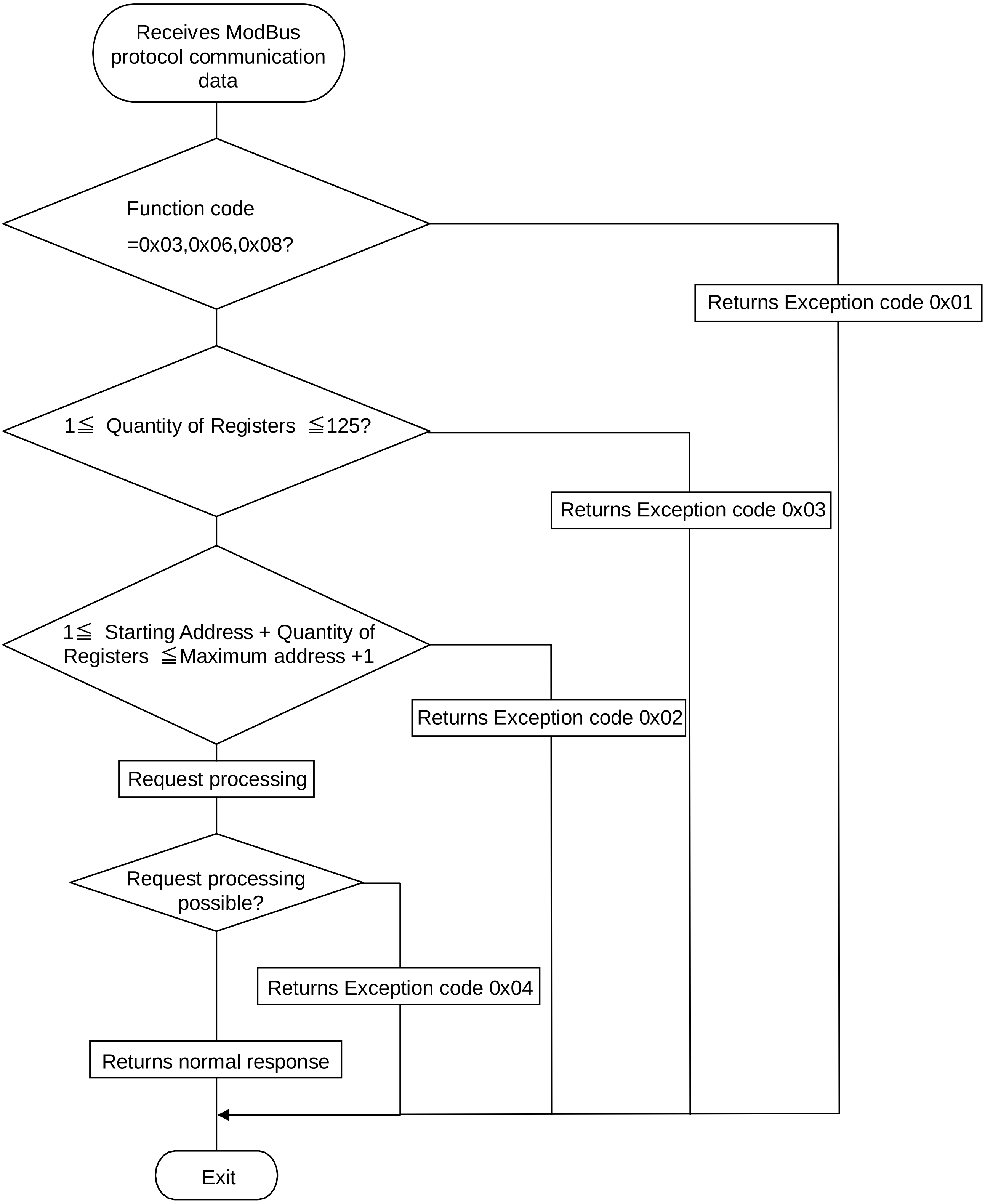
Receives ModBus protocol communication data
Function code =0x03,0x06,0x08?
Returns Exception code 0x01
1≦ Quantity of Registers ≦125?
Returns Exception code 0x03
1≦ Starting Address + Quantity of Registers ≦Maximum address +1
Returns Exception code 0x02
Request processing
Request processing possible?
Returns Exception code 0x04
Returns normal response
Exit

Figure 1 Register readout flowchart

Example 1) When reading out data of three registers starting from a holding register “００００H” with a slave address “2”

Command message

Slave address		０２H
Function code		０３H
Start number	High order	００H
	Low order	００H
Quantity of registers	High order	００H
	Low order	０３H
CRC-16	High order	０５H
	Low order	Ｆ８H

Response message (normal)

Slave address		０２H
Function code		０３H
Number of data (＊1)		０６H
Contents of the first holding register	High order	００H
	Low order	００H
Contents of the next holding register	High order	００H
	Low order	０３H
Contents of the next holding register	High order	００H
	Low order	６３H
CRC-16	High order	８５H
	Low order	ＡＣH

(＊1) Number of holding registers ×2

Response message (exception)

Slave address		０２H
Function code ＋ ８０Ｈ		８３H
Exception code		０３H
CRC-16	High order	Ｆ１Ｈ
	Low order	３１Ｈ

5-2. 08H Diagnostic (Loopback diagnostic test)

The loopback test returns the command messages directly as response messages without changing the content. This function is used to check communication between the master and slave.

〈Command message〉

Slave address	Function code	Sub-function code	Data	Error detection code
０１H～Ｆ７H 1 byte	０８H 1 byte	００００H（Fix） 2 byte	００００H～ＦＦＦＦH 2 byte	2 byte

〈Normal response〉

Slave address	Function code	Sub-function code	Data	Error detection code
Same as command message	Same as command message	Same as command message	Same as command message	
1 byte	1 byte	2 byte	2 byte	2 byte

〈Exception responses〉

・If the slave receives a command message, but detects an error (other than communication error) with the content, the slave does not execute the command and returns an exception response message.

・Whenever the slave determines any of the content of a message as an error by means of its self-diagnostic function, the slave returns to an exception response message to the command message.

Slave address	Function code	Exception code	Error detection code
Same as command message	88 H		
1 byte	1 byte	1 byte	2 byte

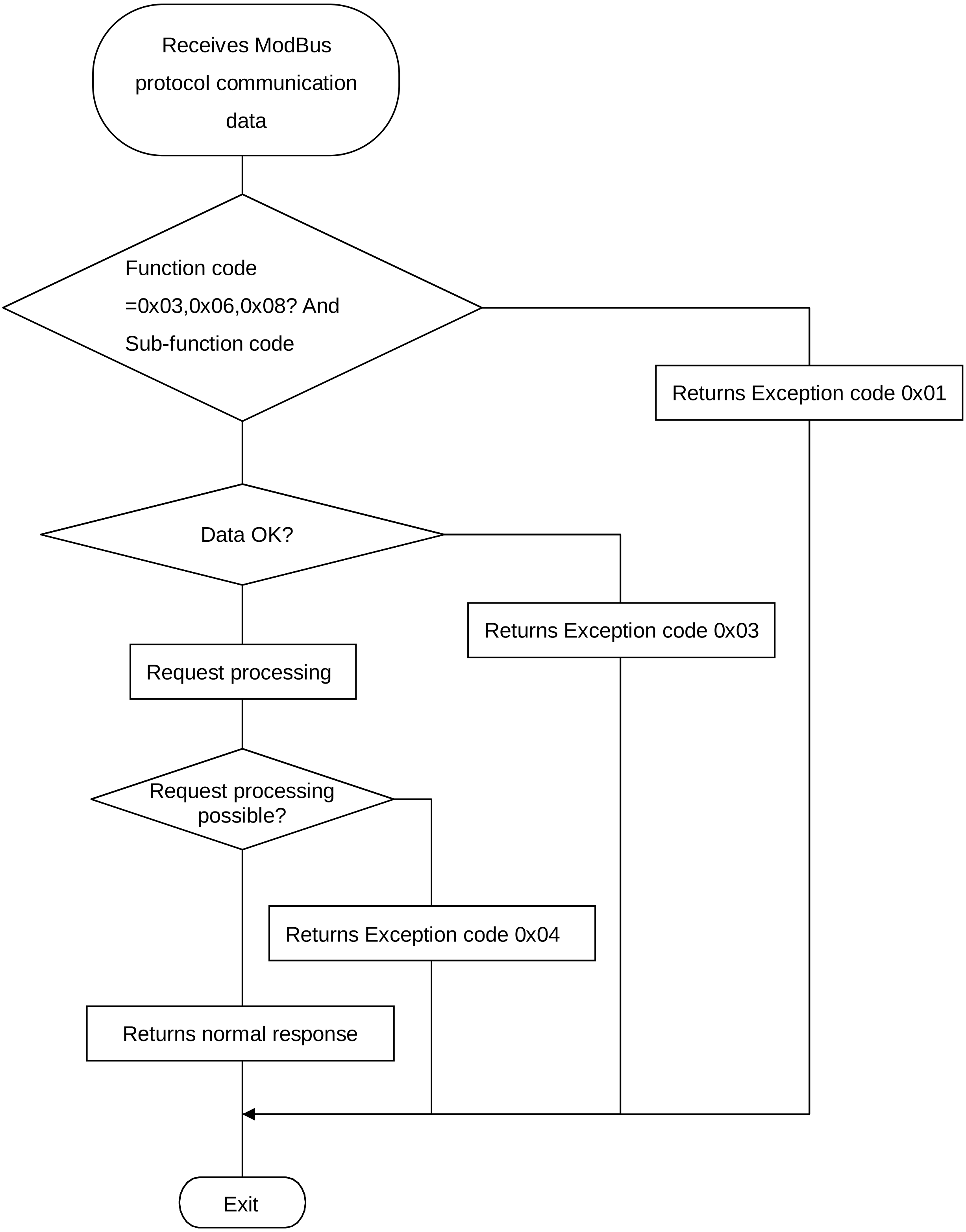
Receives ModBus protocol communication data
Function code =0x03,0x06,0x08? And Sub-function code
Returns Exception code 0x01
Data OK?
Returns Exception code 0x03
Request processing
Request processing possible?
Returns Exception code 0x04
Returns normal response
Exit

Figure 2 Loopback diagnostic test flowchart

Example 1） Loopback test of slave address “1”

Command message

Slave address		０１$_H$
Function code		０８$_H$
Sub-function code (＊1)	High order	００$_H$
	Low order	００$_H$
Data	High order	１Ｆ$_H$
	Low order	３４$_H$
CRC-16	High order	Ｅ９$_H$
	Low order	ＥＣ$_H$

(＊1): Sub-function code is fixed to “００００H”.

Response message (normal)

: Same content as the command message

Slave address		０１$_H$
Function Code		０８$_H$
Sub-function code		００$_H$
		００$_H$
Data	High order	１Ｆ$_H$
	Low order	３４$_H$
CRC-16	High order	Ｅ９$_H$
	Low order	ＥＣ$_H$

Response message (Exception）

Slave address		０１$_H$
Function code＋８０H		８８$_H$
Exception code		０３$_H$
CRC-16	High order	０６$_H$
	Low order	０１$_H$

Exception code table

Exception code	Content
1	Illegal Function (Unsupported function code is specified)
2	Invalid data address is specified. Write request is issued to a read-only address Starting address or quantity of registers that extends beyond the highest configured data address. For communication diagnostic (０８$_H$), a sub-function code other than “００００$_H$” is specified.
3	Data value that falls outside the setting range is specified as write data
4	Self-diagnostic error An unrecoverable error irreparable error occurred while executing the request. The parameter is being changed through the panel operation at that time.

〈No response〉

The slave address specified in the command message does not match its own address.

The CRC in the command message does not equal the CRC recalculated by the slave.

A transmission error (overrun, framing error or parity error) is detected.

The Interval of the data making up a message is longer than 30 ms.

6. CRC-16 calculation

The CRC is an error check code of two bytes (16 bits). After configuring a message, the master calculates the CRC value and appends the calculation result to the end of the message.

The slave recalculate a CRC based on the message it received and compares the calculated value to the actual CRC code in the CRC field of the message. If the two values are not equal, the slave returns no response. The CRC is calculated by using only the data that does not include start, stop or parity bit.

〈CRC code generation procedure〉

1) Load a 16-bit register with FFFFH.

2) Exclusive OR the first one byte (8 bits) of the message with the low order byte of the CRC register, with the result put into the CRC register.

3) Shift the CRC register one bit to the right.

4) If the carry flag is 1:

Exclusive OR the CRC register with the A001H, with the result put into the CRC register.

If the carry flag is 0: Repeat the Step 3.

5) Repeat Steps 3) and 4) until 8 shifts have been performed.

6) Exclusive OR the next one byte (8 bits) of the message with the CRC register.

7) Repeat Steps 3) through 6) for all (one byte) messages (excluding the CRC).

8) The calculated CRC is appended to the message as a two-byte error check code with the low order byte appended first.

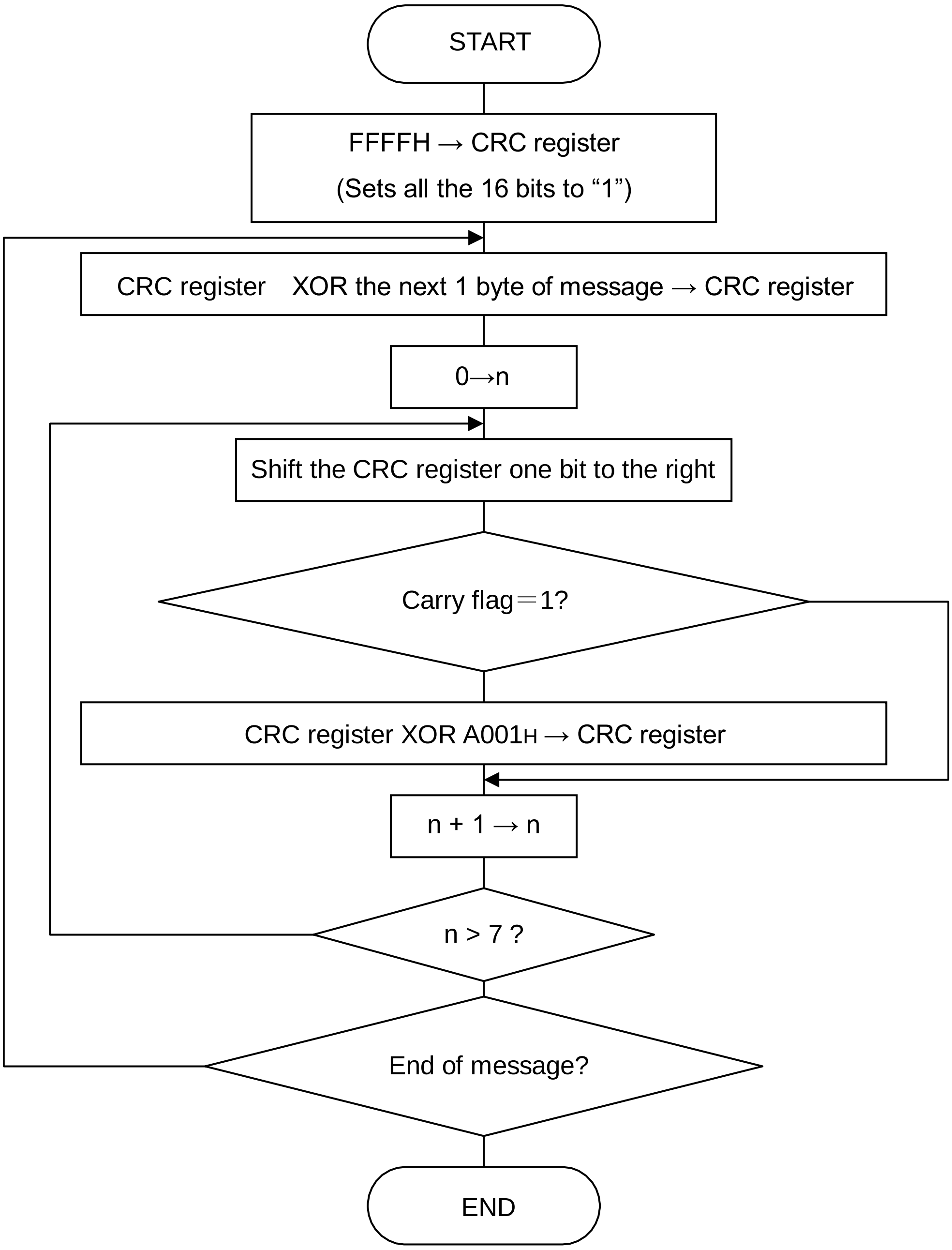
START
FFFFH → CRC register
(Sets all the 16 bits to “1”)
CRC register XOR the next 1 byte of message → CRC register
0→n
Shift the CRC register one bit to the right
Carry flag＝1?
CRC register XOR A001H → CRC register
n + 1 → n
n > 7 ?
End of message?
END

Figure 4 CRC calculation flowchart

7. Data range

The numeric range of data used in this communication is “００００H to ＦＦＦＦH”.
“ＦＦＦＦH” represents “-1”.

The data with decimal points are treated as the data without decimal points during communication.
Handling of negative data

Example) －１０．０℃
－１０．０→－１００＝００００H－００６４H＝ＦＦ９ＣH

8. Communication data list

Attribute
RO: Read only

・Read-only register list

Address	Name	Data	Attribute
0000H	Reserved		RO
0001H	Software version	0 to 9999(V 0 . 00to V 99. 99)	RO
0002H	Software checksum	0 x 0000~ 0 x FFFF	RO
0003H	Communication format version	0 to 9999(V 0 . 00toV 99. 99)	RO
0004H	Status flag	1 ／ 0 (1: Shot status 0: Stopped)	RO
0005H	Current pressure value	0 to 9999 (0 to 999.9 MPa)	RO
0006H ～ 000FH	Spare	Unused	RO
		Continued to the next page	

Address	Name	Data	Attribut
0010H	Shot count (first 4 digits)	0 to ９９９９ (First 4 digits of the shot count)	RO
0011H	Shot count (last 4 digits)	(Last 4 digits of the shot count)	RO
0012H	Peak pressure value	0 to ９９９９ (0 to ９９９．９ MPa）	RO
0013H	Time to peak	0 to ６００００ (０ ～ ６００．００ sec)	RO
0014H	Peak pressure detection	１／０ (1: OK 0: NG)	RO
0015H	Point pressure value (pressure at t-second point)	0 to ９９９９ (0 to ９９９．９ MPa)	RO
0016H	T-second point in monitoring (preset time）	0 to ６００００ (0 to ６００．００ sec ）	RO
0017H	Point pressure detection	１／０ (1 : OK 0 : NG)	RO
0018H	Eject pressure value	0 to ９９９９ (０ ～ ９９９．９ MPa）	RO
0019H	Eject time	０ ～ ６００００ (０ ～ ６００．００ 秒 ）	RO
001AH	Eject pressure detection	１／０ (1 : OK 0 : NG)	RO
001BH ～ 001FH	Spare	Unused	RO