F4C Process Controller

RS-485 COMMUNICATION INSTRUCTION MANUAL

MODBUS Protocol Reference Guide

1. COMMUNICATION FUNCTIONS

1.1. General

- ☐ The F4CC process controller provides a communication function by RS-485 interface, by which it can transmit and receive data to and from host computer, PLC, graphic display panel, etc.
- ☐ The communication system consists of master station and slave station. Up to 247 slave stations can be connected per master station.
- ☐ In order that the master station and slave station can communicate, the format of the transmit/receive data must coincide. For the F4C process controller, the format of the communication data is determined by the MODBUS protocol (RTU mode).
- ☐ Please use a RS-232C→RS-485 converter in case of designating a personal computer or other devices which have a RS-232C interface as a master station.

2. SPECIFICATIONS

2.1 Communication Specifications

Item	Specification	
Electrical specification	Based on EIA RS-485	
Transmit system	2-wire, half-duplex	
Synchronizing system	Asynchronous mode	
Number connection unit	Up to 247 units	
Transmission distance	500m max	
Transmission speed	2400 / 4800 / 9600 / 19200 selectable	
Data format	Start bit	1 bit
	Data length bit	8 bits
	Parity bit	None
	Stop bit	2 bits
Transmission code	HEX value (MODBUS RTU mode)	
Error detection	CRC-16 bits	

A typical MODBUS protocol character is shown below:

1	2	3	4	5	6	7	8	9	10	11
1 Start bit	8 Data bits								2 Stop bits	

The format (11 bits) for each byte in RTU mode is:

Coding System: 8-bits binary

Bits per byte: 1 start bit.

8 data bits,least significant bit sent first.

2 stop bit.

2.2 Communication Setting

In order that the master station and F4C process controller can correctly communicate, following settings are required.

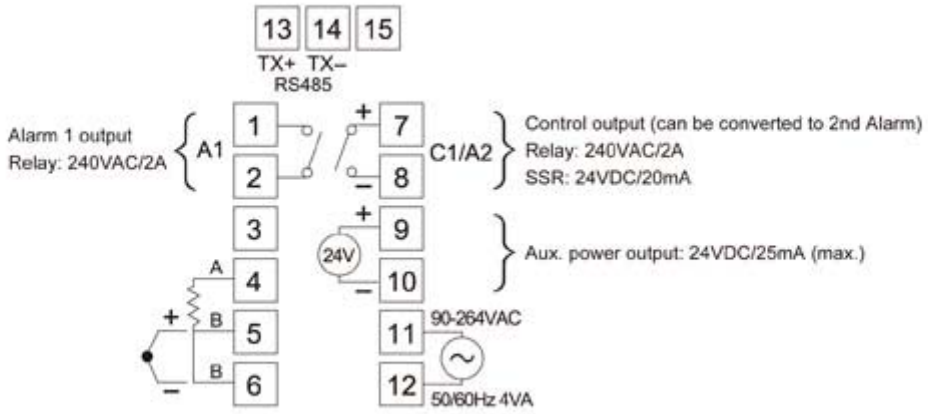
- ✓ All communication condition settings of the master station are the same as those of F4C process controller.

✓ All F4C process controller devices connected on a line are set to an address (ADDR), which is different from each other.

The parameters to be set are shown in the following table.

Parameter	Item	Default	Setting range	Remarks
BaudRate	Transmission speed	19200	2400/4800/9600/19200	Set the same transmission speed to the master station and all F4C controllers.
ID	Slave address	247	1 to 247	Set a different value to each F4C controller.

2.3 Communication Wiring



✓ Use twisted pair cables with shield for communication wiring on the terminals 13 and 14. Recommended cable: UL2464, UL2448, etc.

✓ The total extension length of the cable is up to 500m. A master station and up to 247 units of the F4C process controller can be connected per communication bus.

✓ Both ends of the cable should be connecting with terminate resistors 100Ω 1/2W.

✓ The shield wire of the cable should be grounded at one end on the master station unit side.

2.4 MODBUS Communication Protocol

2.4.1 General

The MODBUS serial line is a Master-Slaves protocol. Only one master (at the same time) is connected to the bus, and one or several ***F4C process controllers*** (247 maximum) are also connected to the same communication bus. A MODBUS communication is always initiated by the master. The ***F4C process controller*** will never transmit data without receiving a request from the master. The ***F4C process controller*** will never communicate with each other. The master initiates only one MODBUS transaction at the same time.

The master issues a MODBUS command message to the ***F4C process controller*** in two modes:

1. Unicast Mode: the master addresses an individual ***F4C process controller***. After receiving and processing the command message, the ***F4C process controller*** returns a response message to the

master. Each ***F4C process controller*** must have a unique address (1 ~ 247) set by the ID parameter.

2. Broadcast mode: the master can send a command message to all ***F4C process controllers***. No response is returned to a broadcast command sent by the master. The broadcast commands are necessarily writing commands. ALL ***F4C process controllers*** must accept the broadcast for writing function. The address 0 is reserved to identify a broadcast exchange

2.4.2 Composition of Command Message

Command message and response message consist of 4 fields; Slave Address, Function code, Data and CRC check code. And these are sends in this order. The allowable character transmitted for all fields are hexadecimal 0-9,A-F.

RTU message framing

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low CRC Hi

In the following, each field is explained.

1. Slave Address (ID)

Address is the number specifying a ***F4C process controller***. The individual addresses are set by the ID parameter in the range of 1~247 decimal. A master addresses a ***F4C process controller*** by placing the ***F4C process controller*** address in the address field of the message. When the ***F4C process controller*** returns its response, it places its own address in this address field of the response to let the master know which ***F4C process controller*** is responding.

Address 0 is used for the broadcast address, which all ***F4C process controllers*** recognize.

When the broadcast address (address 0) is applied on the command message, no any response message will be sent from the ***F4C process controller***.

2. Function Code

This is a code to designate the function executed by ***F4C process controller***. When a message is sent from a master to a ***F4C process controller***, the function code field tells the ***F4C process controller*** what kind of action to perform. When the ***F4C process controller*** responds to the master, it uses the function code field to indicate either a normal response or that some kind of error occurred. For normal response, the ***F4C process controller*** simply echoes the original function code. For an exception response, the ***F4C process controller*** returns a code that is equivalent to the original function code with its most-signification bit set to logic 1.

The listing below shows the function codes supported by the ***F4C process controller***.

Function code		
Code	Function	Object Type
03	Read-out	16-bit word Read/Write Register

04	Read-out	16-bit word Read Only Register
06	Write-in	16-bit word Read/Write Register

3. Data

Data are the data required for executing function codes. The composition of data varies with function codes.

A data register is assigned to each parameter in the ***F4C process controller***. For reading/writing parameter by communication, designate the data register. Refer to chapter 3 "**Data Format and Data Registers Map**" for details.

4. CRC check

This is the code to detect message errors (change in bit) in the signal transmission.

On the MODBUS protocol (RTU mode), CRC-16 (Cyclical Redundancy Check) is applied.

CRC-16 is the 2-bytes (16-bits) error check code. From the first byte (address) of the message to the end of the data field are calculated.

The slave station calculates the CRC of the received message, and does not respond if the calculated CRC is different from the contents of the received CRC code.

The Cyclical Redundancy Checking (CRC) field is two bytes, containing a 16–bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The device that receives recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16–bit register to all 1's. Then a process begins of applying successive 8–bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits and the parity bit, do not apply to the CRC.

During generation of the CRC, each 8–bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8–bit character is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final content of the register, after all the characters of the message have been applied, is the CRC value.

A procedure for generating a CRC is:

1. Load a 16-bits register with FFFF hex (all 1's). Call this the CRC register.
2. Exclusive OR the first 8-bit byte of the message with the low-order byte of the 16-bit CRC registers, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the LSB), Zero-filling the MSB. Extract

and examine the LSB.

4. If the LSB was 0: Repeat Step 3.
   If the LSB was 1: Exclusive OR the CRC registers with the polynomial value 0xA001 (1010 0000 0000 0001).
5. Repeat step 3 and 4 until 8 shifts have been performed. When this is done, a complete 8-bit byte will have been processed.
6. Repeat step 2 through 5 for the next 8-bit byte of the message. Continue doing this until all bytes have been processed.
7. The final content of the CRC register is the CRC value.
8. When the CRC is placed into the message, its upper and lower bytes must be swapped as described below.

For example, if the CRC value is x1241H (0001 0010 0100 0001):

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi
							0x41	0x12

2.4.3 Response of The F4C Process Controller

Once the command message has been processed by the F4C process controller, a response message is built depending on the result of processing.

1. Normal Response

To a relevant command message, the F4C process controller creates and sends back a response message, which corresponds to the command message. The composition of response message in this case is the same as command message. Content of the data field depend on the function code. For details, refer to Sec 3.5.

2. Exception Response

If contents of a command message have an abnormality (for example, non-actual function code is designated) other than transmission error, the slave station does not execute that command but creates and sends back a response message at error detection.
The composition of response message at error detection is shown on below; the value used for function code field is the function code of command message plus x80H.

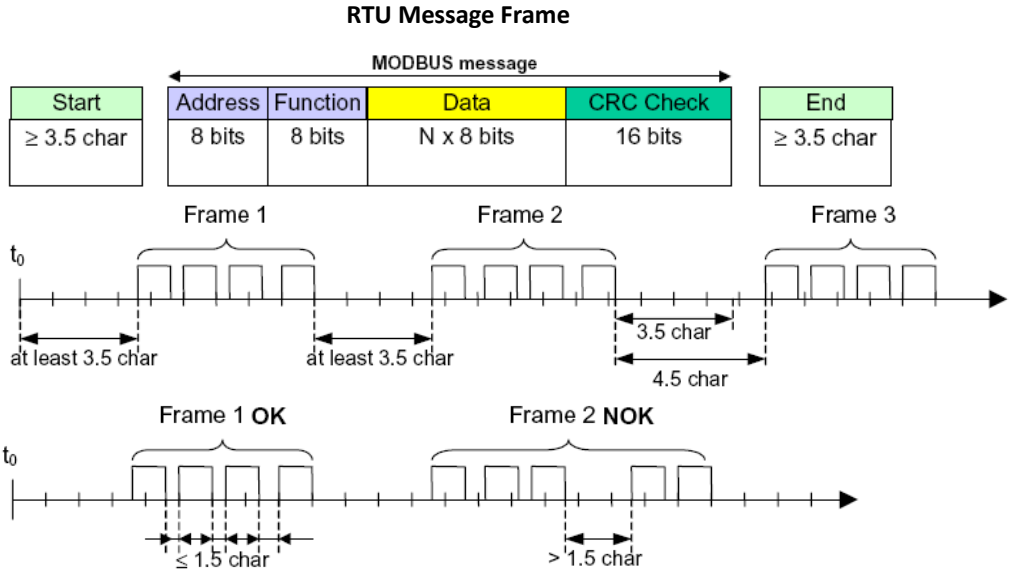
Slave Address	Function code (Function code + x80H)	Error code	CRC check
8 BITS	8 BITS	8 BITS	16 BITS

Error Code	Contents	Description
01	Illegal function	The function code received is not an allowable action for the slave.
02	Illegal data address	The data address received is not an allowable address for the slave.
03	Illegal data value	A value contained in the data field is not an allowable

		value for the slave.

2.4.4 MODBUS Message RTU Framing

In RTU mode, message frame are separated by a silent interval of at least 3.5 character times. The entire message frame must be transmitted as a continuous stream of characters. If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and will be discarded by the F4C process controller.



1. Transmission procedure of master station

Since the communication system uses the 2-write RS-485 interface, there may be 2 statuses on a line below.

(a) Vacant status (no data on line)

(b) Communication status (data is existing)

The master station must proceed to a communication upon conforming to the following items.

1-1. Before sending a command message, at least 3.5 character times silent interval must be provided.

1-2. For sending, the interval between bytes of a command message must be below 1.5 character times.

1-3. Within 1.5 character times after sending a command message, the receiving status is posted.

1-4. Provide 3.5 character times vacant status between the end of response message reception and beginning of next command message sending (same as in 1-1).

1-5. For ensuring the safety, make a confirmation of the response message and make an arrangement so as to provide 3 or more retries in case of no response, error occurrence,

etc.

2. Transaction of the F4C process controller

(1). Detection of the command message frame

The **F4C process controlle** connected on the line are initially at a receiving status and monitoring the line. When 1.5 character times or more vacant status has appeared on the line, the end of preceding frame is assumed and, within following 1.5 character times, a receiving status is posted. When data appears on the line, the **F4C process controlle*r*** receive it. While 1.5 character times more vacant status is detected again, the end of that frame is assumed. Data, which appeared on the line from the first 3.5 character times or more vacant status to the next 3.5 character times or more vacant status, is fetched as one frame.

(2). Response of **F4C process controlle**

After frame detection, The **F4C process controlle*r*** carries out that frame as a command message. If the command message is destined to the own station, a response message is returned. Its processing time is 1 to 10ms (depends on contents of command message). After sending a command message, therefore, the master station must observe the following.

1-1. Receiving status is posted within 1.5 character times after sending a command message.

1-2. 3.5 character times or more vacant status precedes the response message sending.

1-3. Interval between bytes of response message must be smaller than 1.5 character times.

2.5 Function Code Description

2.5.1 Read Data Registers [Function Code: 03]

Read the contents of a contiguous block of registers in the slave.

Broadcast is not possible.

1. Message composition

Command message composition

Address	Function Code	Starting Register	Quantity of Registers*	CRC-16	
x01~xF7	x03	X0000~xFFFF	x0001~x007E	Low-order byte	High-order byte
1 byte	1 byte	2 byte	2 bytes	2 bytes	

Response message composition

Address	Function Code	Byte Count*	Register Value	CRC-16	
x01~ xF7	x03	x02~xFC		Low-order byte	High-order byte
1 byte	1 byte	1 bytes	N x 2 bytes	2 bytes	

* N= Quantity of Registers; Byte Count = N × 2

2. Message transmission (example)

The following show an example of reading the set point (x0000) from the F4C process controller with ID=1.

Command message composition

Address	Function Code	Starting Register	Quantity of Registers	CRC-16
x01	x03	x0000	x0001	x840A

Response message composition

Address	Function Code	Byte Count	Register Value	CRC-16
x01	x03	x02	x03E8	xB8FA

The response data show that the set-point of the F4C is x03E8 (1000)

2.5.2 Read Input Register [Function Code:04]

Read the contents of input registers (1000~1002) in the slave.

Broadcast is not possible.

1. Message composition

Command message composition

Address	Function Code	Starting Register	Quantity of Registers*	CRC-16	
x01~xF7	x04	X000~xFFFF	x0001~x007E	Low-order byte	High-order byte
1 byte	1 byte	2 bytes	2 bytes	2 bytes	

Response message composition

Address	Function Code	Byte Count*	Register Value	CRC-16	
x01~xF7	x04	x02~xFC		Low-order byte	High-order byte
1 byte	1 byte	1 byte	N x2 bytes	2 bytes	

* N= Quantity of Registers; Byte Count = N × 2

2. Message transmission (example)

The following show an example of reading the Measuring Value from the F4C process controller with ID=1

Command message composition

Address	Function Code	Starting Register	Quantity of Registers	CRC-16
x01	x04	x1000	x0001	x350A

Response message composition

Address	Function Code	Byte Count	Register Value	CRC-16

x01	x04	x02	x001B	xF93B

2.5.3 Write Single Register [Function Code:06]

1. Message composition

Command message composition

Address	Function	Register Address	Register Value	CRC-16	
x01~xF7	x06	x0000~xFFFF		Low-order byte	High-order byte
1 byte	1 byte	2 bytes	2 bytes	2 bytes	

Response message composition

Address	Function	Register Address	Register Value	CRC-16	
x01~xF7	x06	x0000~xFFFF		Low-order byte	High-order byte
1 byte	1 byte	2 bytes	2 bytes	2 bytes	

2. Message transmission (example)

The following show an example ofsetting the Input signal type [data register x001C] of the F4C process controller with ID=1 to K type thermocouple

Command message composition

Address	Function Code	Register Address	Register Value	CRC-16
x01	x06	x001C	x0001	x89CC

Response message composition

Address	Function Code	Register Address	Register Value	CRC-16
x01	x06	x001C	x0001	X89CC

3 DATA FORMAT AND DATA REGISTERS MAP

3.1 Data Format

The MODBUS protocol used in F4C process controller is RTU (Remote Terminal Unit) mode.

Transmitted data is “numeric value” and not “ASCII code”

3.2 Data Registers Map (參數位址皆為16進位格式)

Read & Write Word data map: Function code [x03 , x06]

Register	Parameter	Range	Default	Unit
USER LEVEL				
x0000	SV	Low limit ~ High limit	500	
x0001	PuoF	-1000 ~ 1000 (dP=0000) -100.0 ~ 100.0 (dP=000.0) -10.00 ~10.00 (dP=00.00) -1.000 ~ 1.000 (dP=0.000)	0	℃/℉/ENG
x0002	oUtL	0.0 ~ 100.0	N/A	%

x0003	rUn	x0000 : Off x0001 : On x0002 : AT1 x0003 : AT2 x0004 : Manual	x0001	N/A
ALRM LEVEL				
x0004	A1SP	-1999 ~ 9999 (dP=0000) -199.9 ~ 999.9 (dP=000.0) -19.99 ~ 99.99 (dP=00.00) -1.999 ~ 9.999 (dP=0.000)	10	℃/℉/ENG
x0005	A1HY	0 ~ 1000 (dP=0000) 0 ~ 100.0 (dP=0000) 0 ~ 10.00 (dP=0000) 0 ~ 1.000 (dP=0000)	0	℃/℉/ENG
x0006	A1FU	x0000 : A.oFF x0001 : A.Hi x0002 : A.Lo x0003 : A.diH x0004 : A.diL x0005 : A.bdH x0006 : A.bdL x0007 : B.oFF x0008 : B.Hi x0009 : B.Lo x000A : B.diH x000B : B.diL x000C : B.dbH x000D : B.dbL	x0003	N/A
x0007	A1nd	x0000 : None x0001 : Stdy x0002 : LAtH x0003 : StLA	x0000	N/A
x0008	A1dt	0 ~ 5999	0	Second/Minute
x0009	A2SP	-1999 ~ 9999 (dP=0000) -199.9 ~ 999.9 (dP=000.0) -19.99 ~ 99.99 (dP=00.00) -1.999 ~ 9.999 (dP=0.000)	10	℃/℉/ENG
x000A	A2HY	0 ~ 1000 (dP=0000) 0 ~ 100.0 (dP=0000) 0 ~ 10.00 (dP=0000) 0 ~ 1.000 (dP=0000)	0	℃/℉/ENG
x000B	A2FU	x0000 : A.oFF x0001 : A.Hi x0002 : A.Lo x0003 : A.diH x0004 : A.diL x0005 : A.bdH x0006 : A.bdL x0007 : B.oFF x0008 : B.Hi x0009 : B.Lo x000A : B.diH x000B : B.diL x000C : B.dbH x000D : B.dbL	x0004	N/A

x000C	A2nd	x0000 : None x0001 : Stdy x0002 : LAtH x0003 : StLA	x0000	N/A
x000D	A2dt	0 ~ 5999	0	Second/Minute
SOFT LEVEL				
x000E	rAnP	0 ~ 9999 (dP=0000) 0.0 ~ 999.9 (dP=000.0) 0.00 ~ 99.99 (dP=00.00) 0.000 ~ 9.999 (dP=0.000)	0	℃/℉/ENG
x000F	SoFt	HiLt ~ LoLt	oFF	℃/℉/ENG
x0010	Reserved	N/A		
PID LEVEL				
x0011	Pb	0.0 ~ 300.0	5.0	%
x0012	t i	0 ~3000	240	Sec
x0013	td	0 ~ 1000	60	Sec
x0014	ar	0.0 ~ 51.0	0.0	%
x0015	Ar	0.0 ~ 100.0	50.0	%
x0016	HYS	0 ~ 1000 (dP=0000) 0.0 ~ 100.0 (dP=000.0) 0.00 ~ 10.00 (dP=00.00) 0.000 ~ 1.000 (dP=0.000)	0	℃/℉/ENG
x0017	Ct	0 ~ 60	15	Sec
x0018	CPb	0.0 ~ 300.0	5.0	%
x0019	CHYS	0 ~ 1000 (dP=0000) 0.0 ~ 100.0 (dP=000.0) 0.00 ~ 10.00 (dP=00.00) 0.000 ~ 1.000 (dP=0.000)	0	℃/℉/ENG
x001A	db	-1000 ~ 1000 (dP=0000) -100.0 ~ 100.0 (dP=000.0) -10.00 ~ 10.00 (dP=00.00) -1.000 ~ 1.000 (dP=0.000)	0	℃/℉/ENG
x001B	CCt	1 ~ 60	15	Sec
OPTI LEVEL				
x001C	tYPE	x0000 : J x0001 : K x0002 : T x0003 : E x0004 : B x0005 : R x0006 : S x0007 : N x0008 : C x0009 : DPT x000A : JPT x000B : mA x000C : mV x000D : V	x0001	N/A

x001D	SCAL	-1999 ~ 9999 (dP=0000) -199.9 ~ 999.9 (dP=000.0) -19.99 ~ 99.99 (dP=00.00) -1.999 ~ 9.999 (dP=0.000)	0	ENG
x001E	SCAH	-1999 ~ 9999 (dP=0000) -199.9 ~ 999.9 (dP=000.0) -19.99 ~ 99.99 (dP=00.00) -1.999 ~ 9.999 (dP=0.000)	1000	ENG
x001F	CUt	x0000 : nonE x0001 : Lo x0002 : Hi x0003 : HiLo	x0000	N/A
x0020	Unit	x0000 : ℃ x0001 : ℉ x0002 : EnG	x0000	N/A
x0021	dP	x0000 : 0000 x0001 : 000.0 x0002 : 00.00 x0003 : 0.000	x0000	N/A
x0022	HLt	x0000 : REV x0001 : DIR	x0000	N/A
x0023	LoLt	Depend on the input type	0	℃/℉/ENG
x0024	HiLt	Depend on the input type	1000	℃/℉/ENG
x0025	FiLt	0.0 ~ 99.9	0.0	Sec.
x0026	PtnE	x0000 : MM.SS x0001 : HH.MM	x0001	N/A
x0027	EroP	x0000 : 0000 x0001 : 0001 x0002 : 0010 x0003 : 0011	x0000	N/A
x0028	LoCY	x0000 : 0000 x0001 : 0001 x0002 : 0010 x0003 : 0011 x0004 : 0100 x0005 : 0101 x0006 : 0110	x0006	N/A
x0029	SYoF	-1999 ~ 9999 (dP=0000) -199.9 ~ 999.9 (dP=000.0) -19.99 ~ 99.99 (dP=00.00) -1.999 ~ 9.999 (dP=0.000)	0	℃/℉/ENG
x002A	id	1 ~ 247	247	N/A
x002B	bAUd	x0000 : 2.4K x0001 : 4.8K x0002 : 9.6K x0003 : 19.2K	x0003	bps
PROG LEVEL				
x0030	StAt	x0000 : Yes x0001 : No	x0000	N/A

x0031	StAr	x0000 : Zero x0001 : PV	x0001	N/A
x0032	bAnd	0~ 2000 (dP=0000) 0.0 ~ 200.0 (dP=000.0) 0.00 ~ 20.00 (dP=00.00) 0.000 ~ 2.00 0(dP=0.000)	20	℃/℉/ENG
x0033	rt 1	0~5999	60	Second/Minute
x0034	SP 1	Low limit ~ High limit	20	℃/℉/ENG
x0035	St 1	0~5999	60	Second/Minute
x0036	SF 1	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x0037	Ln 1	1~9999	1	Count
x0038	rt2	0~5999	60	Second/Minute
x0037	SP2	Low limit ~ High limit	20	℃/℉/ENG
x003A	St2	0~5999	60	Second/Minute
x003B	SF2	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x003C	Ln2	1~9999	1	Count
x003D	rt3	0~5999	60	Second/Minute
x003E	SP3	Low limit ~ High limit	20	℃/℉/ENG
x003F	St3	0~5999	60	Second/Minute
x0040	SF3	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x0041	Ln3	1~9999	1	Count
x0042	rt4	0~5999	60	Second/Minute

x0043	SP4	Low limit ~ High limit	20	℃/℉/ENG
x0044	St4	0~5999	60	Second/Minute
x0045	SF4	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x0046	Ln4	1~9999	1	Count
x0047	rt5	0~5999	60	Second/Minute
x0048	SP5	Low limit ~ High limit	20	℃/℉/ENG
x0049	St5	0~5999	60	Second/Minute
x004A	SF5	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x004B	Ln5	1~9999	1	Count
x004C	rt6	0~5999	60	Second/Minute
x004D	SP6	Low limit ~ High limit	20	℃/℉/ENG
x004E	St6	0~5999	60	Second/Minute
x004F	SF6	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x0050	Ln6	1~9999	1	Count
x0051	rt7	0~5999	60	Second/Minute
x0052	SP7	Low limit ~ High limit	20	℃/℉/ENG
x0053	St7	0~5999	60	Second/Minute
x0054	SF7	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3	x0008	N/A

		x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next		
x0055	Ln7	1~9999	1	Count
x0056	rtB	0~5999	60	Second/Minute
x0057	SPB	Low limit ~ High limit	20	℃/℉/ENG
x0058	StB	0~5999	60	Second/Minute
x0059	SFB	x0000 : RT8 x0001 : RT7 x0002 : RT6 x0003 : RT5 x0004 : RT4 x0005 : RT3 x0006 : RT2 x0007 : RT1 x0008 : END x0009 : HOLD x000A : Next	x0008	N/A
x005A	LnB	1~9999	1	Count

Read Only Word data map: Function code [x03 ,x04] (參數位址皆為16進位格式)

Register	Parameter	Contents	Parameter Unit
x1000	PVPVOF	PV + PVOF	℃/℉/ENG
x1001	SVSVOF	SV + SVOF	℃/℉/ENG
x1002	OUTL	Output persentage	%