IMO iSmart

Modbus Communication Manual – 1st Edition

Table of Contents

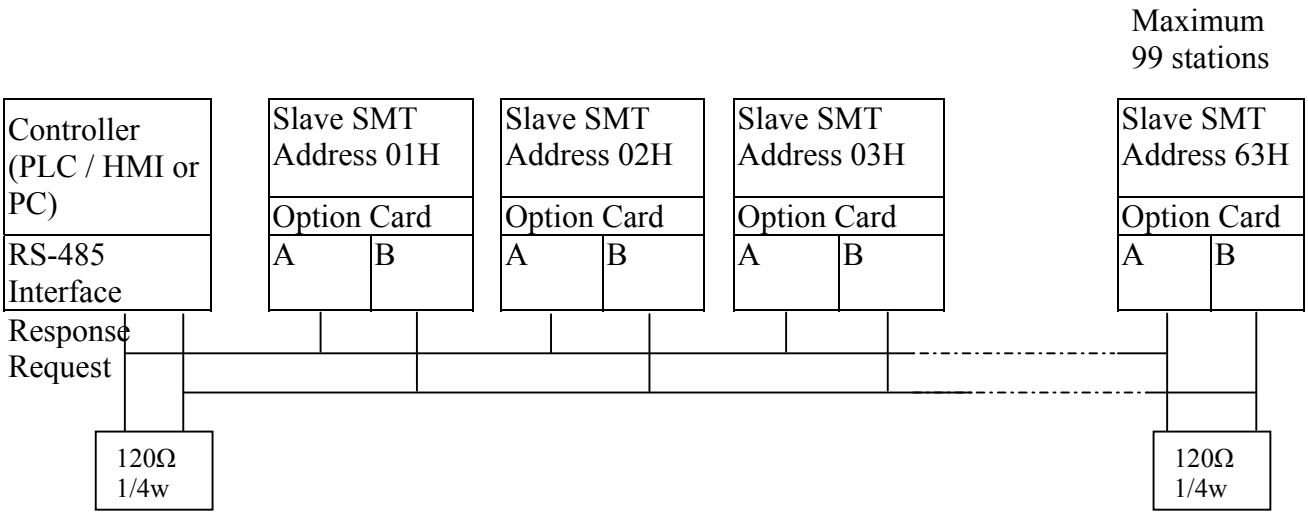
Communication Data Frame

SMT series intelligent relay can be controlled through communication with a PC (Loader Protocol) or by another controller (master) with the communication protocol, RTU Modbus.

Communication Parameters

Transmission method	RS485 twisted pair
Baud rate	38400
Stop bit	2
Parity	None
Maximum frame length	64 bytes

Hardware Installation



Note: It is necessary to connect a terminal resistor with an impedance of 120Ω, 1/4W at both ends of the communication cable.

Data frame for RTU Mode

The MASTER (PLC etc.) sends a request to the SLAVE, and a SLAVE responds to MASTER's command.

The signal receiving is illustrated here.

The data length is varied with the command (Function).

** The interval should be maintained at 10ms between command signal and request.

SLAVE Address	1byte
Function Code	1byte
DATA	nbyte
CRC16 CHECK	2byte
Signal Interval	Signal Interval

SLAVE Address

00H : Broadcast to all the drivers
01H : to the No.01 Driver
0FH : to the No.15 Driver
10H : to the No.16 Driver
and so on...., Max to No.99(63H)

Function Code

03H : Read the register contents
06H : Write a WORD to register
08H : Loop test
10H : Write several WORD's to registers (complex number register write)

CMS (Checksum and time-out definition)

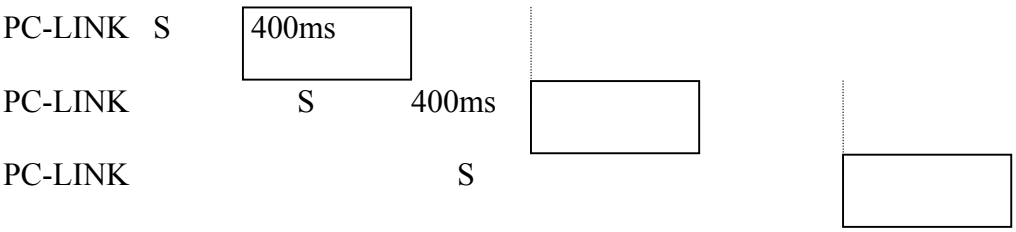
CRC CHECK

The CRC check code is taken from Slave Address to end of the data. The calculation method is illustrated as follow:
(1) Load a 16-bit register with FFFF hex (all1'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 register, 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 examines the LSB.
(4) (If the LSB was 0): Repeat Steps (3) (another shift) (If the LSB was 1): Exclusive OR the CRC register with the polynomial value A001 hex (1010 0000 0000 0001).
(5) Repeat Steps (3) and (4) until 8 shifts been performed. When this is done, a complete 8-bit byte will be processed.
(6) Repeat Steps (2) through (5) for next 8-bit byte of the message, Continue doing this until all bytes have been processed. The final content of the CRC register is the CRC value. Placing the CRC into the message: When the 16-bit CRC (2 8-bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte, For example, if the CRC value is 1241 hex, the CRC-16 (Low) put the 41h, the CRC-16 (Hi) put the 12h.

CRC calculation application program

```
UWORD ch_sum ( UBYTE long , UBYTE *rxdbuff ) {
                    BYTE i = 0;
                    UWORD wkg = 0xFFFF;
                    while ( long-- ) {
                    wkg ^= rxdbuff++;
          for ( i = 0 ; i < 8; i++ ) {
        if ( wkg & 0x0001 ) {          wkg = ( wkg >> 1 ) ^ 0xa001;   }
           else { wkg = wkg >> 1;    }       }
  }  return( wkg );
}
```

TIME-OUT (400ms) & RETRY (max : 2 times)



(When SMT time-out or detect checksum error, or SMT response error code = checksum error, PC-LINK retry maximum two times, and if two times after still error, then display "Communication error")

Command

03H Read Register

PC → PLC

Address		01H
Function Code		03H
*Register Address	(High)	00H
	(Low)	00H
Data Length（Hi）		00H
Data Length（Lo）		13H
CRC-16 (Lo)		04H
CRC-16 (Hi)		07H

PLC→PC(OK)

Address	01H
Function Code	03H
Data (byte)	26H
*Send out the data	
CRC-16 (Lo)	
CRC-16 (Hi)	

PLC→PC(ERROR)

Address	01H
Function Code	83H
Exception Code	52H
CRC-16 (Lo)	C0H
CRC-16 (Hi)	CDH

06H Write single Register

PC → PLC

Address		01 H
Function Code		06H
*Register Address	(High)	01H
	(Low)	02H
Write Data	High	17H
	Low	70H
CRC-16 (Lo)		27H
CRC-16 (Hi)		E2H

PLC→PC(OK)

Address		01H
Function Code		06H
*Register Address	High	01H
	Low	02H
Write Data	High	17H
	Low	70H
CRC-16 (Lo)		27H
CRC-16 (Hi)		E2H

PLC→PC(ERROR)

SLAVE Address	01H
Function Code	86H
Exception Code	52H
CRC-16 (Lo)	C3H
CRC-16 (Hi)	9DH

08H LOOP BACK CHECK

The check code checking the transmission of the signal between MASTER and SLAVE could be discretionary.

PC → PLC

SLAVE Address		01 H
Function Code		08H
Check Code	High	00H
	Low	00H
DATA	High	A5H
	Low	37H
CRC-16	High	DAH
	Low	8DH

PLC→PC(OK)

SLAVE Address		01H
Function Code		08H
Check Code	High	00H
	Low	00H
DATA	High	A5H
	Low	37H
CRC-16	High	DAH
	Low	8DH

PLC→PC(ERROR)

SLAVE Address		01H
Function Code		88H
Exception Code		20H
CRC-16	High	47H
	Low	D8H

10H Write multipile Registers

PC → PLC

Address		01H
Function Code		10H
*Register Address	(High)	00H
	(Low)	00H
Data Length（Hi）		00H
Data Length（Lo）		13H
Byte counters		26H
Send out the data		
CRC-16 (Lo)		
CRC-16 (Hi)		

PLC→PC(OK)

Address		01H	01H
Function Code		10H	10H
*Register Address	(High)	00H	
	(Low)	00H	
Data Length（Hi）		00H	
Data Length（Lo）		13H	
CRC-16 (Lo)		81H	
CRC-16 (Hi)		C4H	

PLC→PC(ERROR)

SLAVE Address	01H
Function Code	90H
Exception Code	52H
CRC-16 (Lo)	ACH
CRC-16 (Hi)	3DH

Register Address	Data Length	Usable Cmd	Content															
			F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0000H	1	03H 06H 10H	-	RF	RE	RD	RC	RB	RA	R9	R8	R7	R6	R5	R4	R3	R2	R1
0001H	1		-	GF	GE	GD	GC	GB	GA	G9	G8	G7	G6	G5	G4	G3	G2	G1
0002H	1		-	TF	TE	TD	TC	TB	TA	T9	T8	T7	T6	T5	T4	T3	T2	T1
0003H	1		-	CF	CE	CD	CC	CB	CA	C9	C8	C7	C6	C5	C4	C3	C2	C1
0004H	1		-	MF	ME	MD	MC	MB	MA	M9	M8	M7	M6	M5	M4	M3	M2	M1
0005H	1		XC	XB	XA	X9	IC	IB	IA	I9	I8	I7	I6	I5	I4	I3	I2	I1
0006H	1		Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	X8	X7	X6	X5	X4	X3	X2	X1
0007H	1		-	-	1	-	YC	YB	YA	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
0008H	1		-	NF	NE	ND	NC	NB	NA	N9	N8	N7	N6	N5	N4	N3	N2	N1
0009H	1		-	HF	HE	HD	HC	HB	HA	H9	H8	H7	H6	H5	H4	H3	H2	H1
000AH	1		-	-	-	-	-	-		P1	L8	L7	L6	L5	L4	L3	L2	L1
000BH	1		W16	W15	W14	W13	W12	W11	W10	W9	W8	W7	W6	W5	W4	W3	W2	W1
000CH	1	03H 06H 10H	W32															W17
000DH	1		W48															W33
000EH	1		W64															W49
0010H	1	03H 06H 10H (Only FBD)	B16	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
0011H	1		B32															B17
0012H	1		B48															B33
0013H	1		B64															B49
0014H	1		B80															B65
0015H	1		B96															B81
0016H	1		-	-	-	-	-	-	-	-	-	-	-	-	-	B99	B98	B97

Exception Codes

If an error has occurred during communication, the controller responds with an Exception Code and send the Function Code together with 80H to the main system.

Error Codes	Description
51	Frame error (Function Code error, Register Encoding error, Data Quantity Error)
52	Run mode and command disable
53	Secret mode and command disable
54	Data value over rang
55	SMT system ROM error
56	SMT RTC not exist, can’t operate RTC
57	SMT the other error
58	Commands do not match SMT edit mode
59	Brand ID error

Register Address

(00xxH) Coil Status Address

(01xxH) Control register Address

Register Address	Data Length	Usable Comm.	Content																
				F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0100H	1	03H 06H 10H	RUN / STOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S1
				S=0 STOP S= 1 RUN															
0101H	1	03H	MODE	BRAND ID								SMT MODE							
				BRADE ID： 0～4															
				SMT MODE (Hex) Only C-Type model															
0102H	1	03H	STATUS	A	-	I1	I0	L3	L2	L1	L0	-	S1	S2	B	-	-	D1	D0

<table>
<tr><td></td><td></td><td>06H
10H</td><td>1</td><td colspan="16">A： alarm at expand I/O No. unmatched
＝0： alarm
＝1： don’t alarm
I1I0：expand I/O No.(0~3)
L3 L2 L1 L0
=1 : English
=2 : French
=3 : Spanish
=4 : Italian
=5 : German
=6 : Portugal
=7 : Chinese
B: Backlight on/off
=0 ： auto on/off
=1 ： always on
s1:Power down retain(M coil)
=1 : unretain
=0 : retain
S2:run/stop retain(Counter current value)
=1 : retain
=0 : unretain
D1D0：Data communication mode
=0： data link
=1： remote I/O master
=2： remote I/O slave</td></tr>
<tr><td rowspan="2">0103H</td><td rowspan="2">1</td><td rowspan="2">03H</td><td rowspan="2">STATUS 2 (PA,Error)</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>PA</td><td>-</td><td>-</td><td>-</td><td>-</td><td>S3</td><td>S2</td><td>S1</td><td>S0</td></tr>
<tr><td colspan="16">S3 S2 S1 S0:
0 = OK
1 = ROM error
2 = RAM error
3 = EEPROM error
4 = Program error
5 = Watchdog error
6 = Expand error
7 = Communication error
PA：
0＝PASSWORD OFF
1 = PASSWORD ON</td></tr>
<tr><td>0104H</td><td>1</td><td rowspan="3">03H
10H</td><td rowspan="3">Analog config</td><td colspan="8">A1_GAIN_H</td><td colspan="8">A1_GAIN_L</td></tr>
<tr><td>0105H</td><td>1</td><td colspan="16">A1_OFFSET</td></tr>
<tr><td>0106H</td><td>1</td><td colspan="8">A2_GAIN_H</td><td colspan="8">A2_GAN_L</td></tr>
</table>

0107H	1			A2_OFFSET																	
0108H	1			A3_GAIN_H									A3_GAIN_L								
0109H	1			A3_OFFSET																	
010AH	1			A4_GAIN_H									A4_GAN_L								
010BH	1			A4_OFFSET																	
				Note: A_GAIN = (0~999) A_OFFSET = (-5-0~50) (complement)																	
0110H	1	06H 10H	CLEAR CODE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(02XXH) Current value Address

Register Address	Data Length	Usable Comm.	Content		Remark
0200H	1	03H	Timer1		
0201H	1		Timer2		
0202H	1		Timer3		
…….	……		……..		
020EH	1		TimerF		
0210H	2	03H	CNT1		*1
0211H	2		CNT2		
….	….		….		
021EH	2		CNTF		
		03H 10H	RTC current value		
0220H	1		CURRENT_YEAR	CURRENT_MOON	
0021H	1		CURRENT_DAY	CURRENT_WEEK	
0022H	1		CURRENT_HOUR	CURRENT_MINUTE	
0023H	1		CURRENT_SECOND	00	
		03H	ANALOG 1		
0230	1		A1_VALUE_H	A1_VALUE_L	
0231	1		A2_VALUE_H	A2_VALUE_L	
0232	1		A3_VALUE_H	A3_VALUE_L	
0233	1		A4_VALUE_H	A4_VALUE_L	
0234	1		A5_VALUE_H	A5_VALUE_L	
0235	1		A6_VALUE_H	A6_VALUE_L	
0236	1		A7_VALUE_H	A7_VALUE_L	
0237	1		A8_VALUE_H	A8_VALUE_L	
PWM					
0260H	3H	03H	00	PWM_RUN_NUM	
			PW_H	PW_L	
			PT_H	PT_L	

(03XXH) User character Address (Chinese lattice)

Register Address	Data Length	Usable Comm.	Content	Remark
0300H	10H	03H 10H	Character_5	
0301H	10H		Character_6	
….	….			
027H	10H		Character_44	

(04XXH) Preset value Address

Register Address	Data Length	Usable Comm.	Content	Remark
TMR				
0400H	1H	03H 10H	Timer1	
0401H	1H		Timer2	
0402H	1H		Timer3	
…….	……		……..	
040EH	1H		TimerF	
COUNTER				
0410H	5H	03H 10H	CNT1	*2
0411H	5H		CNT2	
….	….		….	
041EH	5H		CNTF	
RTC				
0420H	3H	03H 10H	RTC1	*3
0421H	3H		RTC2	
…	….		….	
042EH	3H		RTCF	
ANALOG				
0430H	1H	03H 10H	ANALOG 1	
0431H	1H		ANALOG 2	
……	…….		…….	
043EH	1H		ANALOG F	
PWM				
0460H	10H	03H 10H	PWM	*4

Limitation Notes

Note 1: Counter current value

High bytes	Low bytes
C_current_V_M	C_current_V_L
00	C_current_V_H

Counter value	0~999999	0~0F423FH (HEX)

Note 2: Counter Preset Value

	High bytes	Low bytes
COUNTER MOD 1~7	C_PRESET_V_M	C_PRESET_V_L
	00	C_PRESET_V_H
	00	00
	00	00
	00	00
COUNTER MOD8	FIX_TIM_H	FIX_TIM_L
	C_ON_PRESET_V_M	C_ON_PRESET_V_L
	00	C_ON_PRESET_V_H
	C_OFF_PRESET_V_M	C_OFF_PRESET_V_L
	00	C_OFF_PRESET_V_H

Counter value	0~999999	0~0F423FH (HEX)

Note 3: RTC Preset Value

	High bytes	Low bytes
RTC MOD1 RTC MOD2	Turn on week	Turn off week
	Turn on time(hour)	Turn on time(min)
	Turn off time(hour)	Turn off time(min)
RTC MOD3	Turn on year	Turn off year
	Turn on month	Turn on day
	Turn off month	Turn off day

Year	00~99
Months	01~12
Days	01~31
Weeks	00~06
Hours	00~23
Minutes	00~59
Seconds	00~59

Note 4: PWM Preset Value

	High bytes	Low bytes
1	PW1_H	PW1_L
2	PT1_H	PT1_L
3	PW2_H	PW2_L
4	PT2_H	PT2_L
5	PW3_H	PW3_L
6	PT3_H	PT3_L
7	PW4_H	PW4_L
8	PT4_H	PT4_L
9	PW5_H	PW5_L
10	PT5_H	PT5_L
11	PW6_H	PW6_L
12	PT6_H	PT6_L
13	PW7_H	PW7_L
14	PT7_H	PT7_L
15	PW8_H	PW8_L
16	PT8_H	PT8_L

PW	Pulse Width Value	00000~32767
PT	Period Value	00001~32767)