



INSTRUCTION MANUAL

DOCUMENT NO. 027 - 2178
Rev. 1.01 Sept. 29, 2004

***CV10 MODBUS
Communications
Protocol***

Saftronics, Inc.

5580 Enterprise Pkwy
Ft. Myers, FL 33905
Tel. (239) 693 - 7200
Fax (239) 639 - 2431
www.saftronics.com

Contents

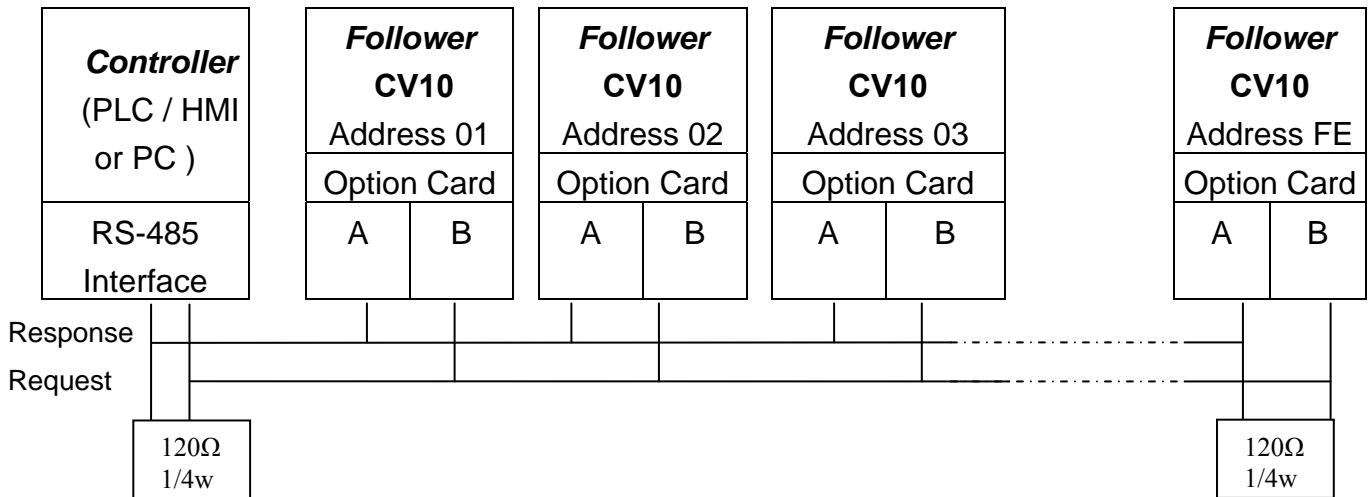
1. Communication Data Frame	2
1.1 Hardware Installation	2
1.2 Data Frame for ASCII Mode	2-3
1.3 Follower Address	3
1.4 Function Code	3
2. CMS (Checksum and time out definition)	4
2.1 LRC	4
2.2 CRC check	4
2.3 Timeout	5
3. Command START ADDRESS	6-8
4. Exception Code	8
5. Command Start Address Description	9
5.1 Ladder Start Address Description	9
5.2 Function block read	10
5.2.1 Timer function block read	10
5.2.2 Counter function block read	11
5.2.3 Encoder function block read	12
5.2.4 Analog function block read	13
5.2.5 Control function block read	14
5.2.6 Coil status read	15
5.3 Ladder page write	16-17
5.4 Function block write	18
5.4.1 Timer function block write	18-19
5.4.2 Counter function block write	19-20
5.4.3 Encoder function block write	20-21
5.4.4 Analog function block write	21-22
5.4.5 Control function block write	23-24
5.4.6 Coil status write	24-25
5.5 Inverter control	26
5.5.1 Command data (Readable and writeable)	26
5.5.2 Supervision data (Only for reading)	27-28
5.5.3 Read data from holding register [03H]	29-30
5.5.4 Loop back test [08H]	31-32
5.5.5 Write to the holding register [06H]	33
5.5.6 Write data to multiple holding registers [10H]	34-35
Appendix: MODBUS Register Addresses	36-39

1. Communication Data Frame

CV10 series inverter can be communication controlled by a PC or other controller with the communication protocol, **MODBUS ASCII Mode & Mode RTU, RS485 or RS232.**

Frame length maximum 80 bytes

1.1 Hardware Installation



****Terminate the communication line with a (120Ω, 1/4W) resistor at both ends.****

1.2 Data Frame for ASCII Mode

STX(3AH)	Start Byte = 3AH
Address Hi	Communication Address: 2-digit ASCII Code
Address Lo	
Function Hi	Function Code (command): 2-digit ASCII Code
Function Lo	
Command Start Address	Command Start byte: 4-digit ASCII Code
Command Start Address	
Command Start Address	
Command Start Address	

Data Frame for ASCII Mode *con't*

Data length	The length of the command: 4-digit ASCII Code
Data length	
Data length	
Data length	
LRC Check Hi	LRC Check Code: 2-digit ASCII Code
LRC Check Lo	
END Hi	End Byte: END Hi = CR(0DH) , END Li = LF(0AH)
END Lo	

Data frame For RTU Mode

MASTER (PLC etc.) sends a request to the FOLLOWER, and the FOLLOWER sends a response to the MASTER. The data received is illustrated here.

The data length varies depending on the command (Function).

FOLLOWER Address
Function Code
DATA
CRC CHECK
Signal Interval

** The drive response time is 10ms.

1.3 Follower 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 on....., Max to No. 254 (FE hex.)

1.4 Function Code

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

2. CMS (Checksum and time-out definition)

2.1 LRC

Ex. ADDRESS	01H
FUNCTION	03H
COMMAND	01H
	00H
DATA LENGTH	0AH

	0FH----- two's complement
Checksum =	F1H
CS (H) =	46H (ASCII)
CS (L) =	31H (ASCII)

2.2 CRC CHECK: CRC check code is calculated from the Follower Address to end of the data.

The calculation method is as follow:

- (1) Load a 16-bit register with FFFF hex (all 1's). Call this the CRC register.
- (2) Exclusive OR the first byte of the message with the low-order byte of the 16-bit CRC register, putting the result into 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 is a "1": Exclusive OR the CRC register with the polynomial value of A001 hex (1010 0000 0000 0001).
- (5) Repeat Steps (3) and (4) until 8 shifts been performed. When this is done, a complete byte is processed.
- (6) Repeat Steps (2) through (5) for next byte of the message. Continue doing this until all bytes have been processed. The final content of the CRC register is the CRC value. Now append the 2 Byte CRC Checksum to the end of the message. The low-order byte will be transmitted first, followed by the high-order byte.

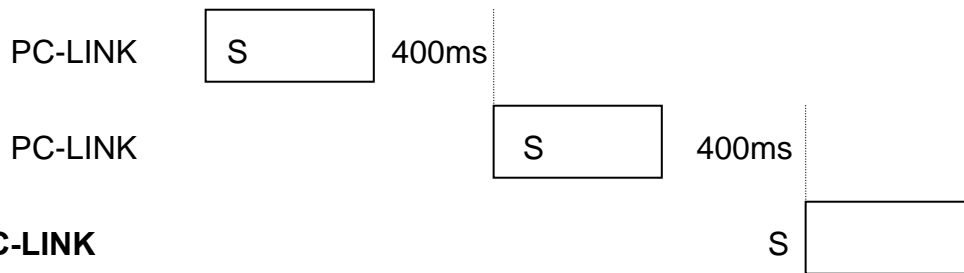
● **CRC calculation application program in C**

```

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

```

2.3 TIMEOUT (400ms) & RETRY (max.: 2 times)



PC-LINK

When a time-out, checksum error is detected, the drive will retry a maximum two times, and if the error is still present the display will show “**ERR6**”.

3. Command START ADDRESS

Function	Function Description	Command Start Address	Data length (WORD)
03	Ladder page1 read	200H	0AH
	Ladder page2 read	20AH	0AH
	Ladder page3 read	214H	0AH
	Ladder page4 read	21EH	0AH
	Ladder page5 read	228H	0AH
	Timer1 Function read	264H	05H
	Timer2 Function read	269H	05H
	Timer3 Function read	26EH	05H
	Timer4 Function read	273H	05H
	Timer5 Function read	278H	05H
	Timer6 Function read	27DH	05H
	Timer7 Function read	282H	05H
	Timer8 Function read	287H	05H
	Counter1 Function read	28CH	04H
	Counter2 Function read	290H	04H
	Counter3 Function read	294H	04H
	Counter4 Function read	298H	04H
	Encoder1 Function read	2ACH	05H
	Encoder2 Function read	2B1H	05H
	Encoder3 Function read	2B6H	05H
	Encoder4 Function read	2BBH	05H
	Analog1 Function read	2C0H	03H
	Analog2 Function read	2C3H	03H
	Analog3 Function read	2C6H	03H
	Analog4 Function read	2C9H	03H
	Control function read	2CCH	06H
	Contro2 function read	2D2H	06H
	Contro3 function read	2D8H	06H
	Contro4 function read	2DEH	06H
	Contro5 function read	2E4H	06H
	Contro6 function read	2EAH	06H
	Contro7 function read	2F0H	06H
	Contro8 function read	2F6H	06H
All Coil status read	2FCH~303H	08H	

(Command START ADDRESS *con't*)

Function	Function Description	Command Start Address	Data length (WORD)
10	Ladder page1 write	200H	0AH
	Ladder page2 write	20AH	0AH
	Ladder page3 write	214H	0AH
	Ladder page4 write	21EH	0AH
	Ladder page5 write	228H	0AH
	Timer1 Function write	264H	04H
	Timer2 Function write	269H	04H
	Timer3 Function write	26EH	04H
	Timer4 Function write	273H	04H
	Timer5 Function write	278H	04H
	Timer6 Function write	27DH	04H
	Timer7 Function write	282H	04H
	Timer8 Function write	287H	04H
	Counter1 Function write	28CH	03H
	Counter2 Function write	290H	03H
	Counter3 Function write	294H	03H
	Counter4 Function write	298H	03H
	Encoder1 Function write	2ACH	04H
	Encoder2 Function write	2B1H	04H
	Encoder3 Function write	2B6H	04H
	Encoder4 Function write	2BBH	04H
	Analog1 Function write	2C0H	03H
	Analog2 Function write	2C3H	03H
	Analog3 Function write	2C6H	03H
	Analog4 Function write	2C9H	03H
	Control function write	2CCH	06H
	Contro2 function write	2D2H	06H
	Contro3 function write	2D8H	06H
	Contro4 function write	2DEH	06H
	Contro5 function write	2E4H	06H
	Contro6 function write	2EAH	06H
	Contro7 function write	2F0H	06H
Contro8 function write	2F6H	06H	
06	Coil status write	2FCH	01H

Function	Function Description	Command Start Address	Data length (WORD)
06	RUN & Stop(PLC)	330H	1
	All memory clear (Clear PLC Memory)	331H	1
	PASSWORD	332H	1

Note: 'Write Ladder page write' and 'Clear all memory' are not available when the PLC is running.

4. Exception Code

ASCII Mode	
STX	':'
Address	'0'
	'1'
Function	'8'
	'6'
Exception Code	'5'
	'1'
LRC Check	'2'
	'8'
END	'CR'
	'LF'

RTU Mode		
FOLLOWER Address		02H
Function		83H
Exception code		52H
CRC-16	High	C0H
	Low	CDH

During a communication error the drive will response with an Exception Code and send a message back to the main system consisting of a Function Code that is "ANDED (and 80h)" with 80 Hex.

Exception Code	Description
51 Hex.	Function Code Error
52 Hex.	Register Encoding Error
53 Hex.	Data Quantity Error
54 Hex.	DATA Setting Error
55 Hex.	Write Mode Error

5. Command Start Address Description

5.1 Ladder (*) page read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
30H	
30H	
30H	Data Length
30H	
30H	
41H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
31H	Data length (Byte)
34H	
Send out the data from PLC[0]~PLC[19], total 40 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	* Register Number
00H	
00H	Data length
0AH	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
14H	Data length
Sent out the data from PLC[0]~PLC[19], total 20 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2 Function block read

5.2.1 Timer function block read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
36H	
34H	
30H	Data Length
30H	
30H	
30H	
35H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	Data length (Byte)
41H	
*Send out the data from PLC[200]~PLC[209], total 20 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
64H	
00H	Data Length
05H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
0AH	Data Length
*Send out the data from PLC[200]~PLC[209], total 10 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2.2 Counter function block read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
38H	
43H	
30H	Data Length
30H	
30H	
34H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	STX
31H	Address
30H	
33H	
30H	Data Length (Byte)
38H	
*send out the data from PLC[280]~PLC[287], total 16 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
8CH	
00H	Data Length
04H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
08H	Data Length
*Send out the data from PLC[280]~PLC[287], total 8 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2.3 Encoder function block read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
41H	
43H	
30H	Data Length
30H	
30H	
35H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	Data Length (Byte)
41H	
*Send out the data from PLC[344]~PLC[353], total 20 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
ACH	
00H	Data Length
05H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
0AH	Data Length
*Send out the data from PLC[344]~PLC[353], total 10 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2.4 Analog function block read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
43H	
30H	
30H	Data Length
30H	
30H	
33H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	Data Length (Byte)
36H	
*Send out the data from PLC[384]~PLC[389], total 12 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
C0H	
00H	Data Length
03H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
06H	Data Length
**Send out the data from PLC[384]~PLC[389], total 12 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2.5 Control function block read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
43H	
43H	
30H	Data Length
30H	
30H	
36H	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	Data Length (Byte)
43H	
*Send out the data from PLC[408]~PLC[419], total 24 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
CCH	
00H	Data Length
06H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
0CH	Data Length
**Send out the data from PLC[408]~PLC[419], total 12 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.2.6 Coil status read

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
46H	
43H	
30H	Data Length
30H	
30H	
38H	
?	
?	CHECKSUM
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
33H	
31H	Data Length (Byte)
30H	
*Send out the data from PLC[504]~PLC[519], total 32 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
03H	Function Code
02H	*Register Code
FCH	
00H	Data Length
08H	
	CRC High order byte
	CRC Low order byte

INV (PLC) → PC	
01H	Address
03H	Function Code
10H	Data Length
*Send out the data from PLC[504]~PLC[519], total 16 Bytes	
?	CRC High order byte
?	CRC Low order byte

5.3 Ladder page write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	
30H	*Register Number
32H	
30H	
30H	
30H	Data Length (Byte)
30H	
30H	
41H	
31H	DATA
34H	
*Send out the data from PLC[0]~PLC[19], total 40 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Number
32H	
30H	
30H	
30H	Data Length
30H	
30H	
41H	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode (Ladder page write con't)

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
00H	
00H	Data Length
0AH	
14H	DATA
*end out the data from PLC[0]~PLC[19], total 20 Bytes	
	CRC High order byte
	CRC Low order byte

INV (PLC) →PC	
01H	Address
03H	Function Code
02H	* Register Code
00H	
00H	Data Length
0AH	
?	CRC High order byte
?	CRC Low order byte

5.4 Function block write

5.4.1 Timer function block write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	
30H	*Register Number
32H	
36H	
34H	
30H	Data Length (Byte)
30H	
30H	
34H	
30H	DATA
38H	
*Send out the data from PLC[200]~PLC[207], total 16 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	
30H	Address
31H	
31H	Function Code
30H	
30H	* Register Number
32H	
36H	
34H	
30H	Data Length
30H	
30H	
34H	
?	CHECKSUM
?	
0DH	END
0AH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
64H	

INV (PLC) → PC	
01H	Address
10H	Function Code
02H	*Register Code
64H	

RTU Mode (5.4.1 Timer function block write *con't*)

00H	Data Length		00H	Data Length
04H			04H	
08H	DATA		?	CRC High order byte
*Send out the data from PLC[200]~PLC[207], total 8 Bytes			?	CRC Low order byte
?	CRC High order byte			
?	CRC Low order byte			

5.4.2 Counter function block write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	
30H	*Register Code
32H	
38H	
43H	
30H	Data Length (Byte)
30H	
30H	
33H	
30H	DATA
36H	
*Send out the data from PLC[280]~PLC[285], total 12 bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Code
32H	
38H	
43H	
30H	Data Length
30H	
30H	
33H	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode (5.4.2 Counter function block write *con't*)

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
8CH	
00H	Data Length
03H	
06H	DATA
*Send out the data from PLC[280]~PLC[285], total 6 Bytes	
?	CRC High order byte
?	CRC Low order byte

INV (PLC) → PC	
01H	Address
10H	Function Code
02H	*Register Code
8CH	
00H	Data Length
03H	
?	CRC High order byte
?	CRC Low order byte

5.4.3 Encoder function block write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	
30H	*Register Code
32H	
41H	
43H	
30H	Data Length (Byte)
30H	
30H	
34H	
30H	DATA
38H	

INV (PLC) → PC	
3AH	
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Code
32H	
41H	
43H	
30H	Data Length
30H	
30H	
34H	
?	CHECKSUM
?	

ASCII Mode (5.4.3 Encoder function block write *con't*)

*Send out the data from PLC[344]~PLC[353], total 16 Bytes		0DH	END
		0AH	END
?	CHECKSUM		
?			
0DH	END		
0AH	END		

RTU Mode (5.4.3 Encoder function block write *con't*)

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
ACH	
00H	Data Length
04H	
08H	DATA
* Send out the data from PLC[344]~PLC[353], total 8 Bytes	
?	CRC High order byte
?	CRC Low order byte

INV (PLC) → PC	
01H	Address
10H	Function Code
02H	*Register Code
ACH	
00H	Data Length
04H	
?	CRC High order byte
?	CRC Low order byte

5.4.4 Analog function block write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	

INV (PLC) → PC	
3AH	
30H	Address
31H	
30H	Function Code
33H	

ASCII Mode (5.4.4 Analog function block write *con't*)

30H	*Register Code		30H	*Register Code
32H			32H	
43H			43H	
30H			30H	
30H	Data Length (Byte)		30H	Data Length
30H			30H	
30H			30H	
33H			33H	
30H	DATA		?	CHECKSUM
36H			?	
* Send out the data from PLC[384]~PLC[389], total 12 Bytes			0DH	END
?	CHECKSUM		0AH	END
?				
0DH				
0AH				

RTU Mode (5.4.4 Analog function block write *con't*)

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
C0H	
00H	Data Length
03H	
06H	DATA
* Send out the data from PLC[384]~PLC[389], total 12 Bytes	
?	CRC High order byte
?	CRC Low order byte

INV (PLC) → PC	
01H	Address
10H	Function Code
02H	*Register Code
C0H	
00H	Data Length
03H	
?	CRC High order byte
?	CRC Low order byte

5.4.5 Control function block write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
31H	Function Code
30H	
30H	*Register Code
32H	
43H	
43H	Data Length (Byte)
30H	
30H	
30H	
36H	DATA
30H	
43H	
* Send out the data from PLC[408]~PLC[419], total 24 Bytes	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	
30H	Address
31H	
30H	Function Code
33H	
30H	*Register Code
32H	
43H	
43H	Data Length
30H	
30H	
30H	
36H	CHECKSUM
?	
?	
0DH	END
0AH	END

RTU Mode

PC → INV (PLC)	
01H	Address
10H	Function Code
02H	*Register Code
CCH	

INV (PLC) → PC	
01H	Address
10H	Function Code
02H	*Register Code
CCH	

RTU Mode (5.4.5 Control function block write *con't*)

00H	Data Length		00H	Data Length
06H			03H	
0CH	DATA		?	CRC High order byte
*Send out the data from PLC[408]~PLC[419], total 12 Bytes			?	CRC Low order byte
?	CRC High order byte			
?	CRC Low order byte			

5.4.6 Coil status write

ASCII Mode

PC → INV (PLC)	
3AH	STX
30H	Address
31H	
30H	Function Code
36H	
30H	* Register Code
32H	
46H	
43H	
*Data to be written to	16-Bit data comprising of 4 ASCII codes
*Data to be written to	
*Data to be written to	
*Data to be written to	
?	CHECKSUM
?	
0DH	END
0AH	END

INV (PLC) → PC	
3AH	STX
30H	Address
31H	
30H	Function Code
36H	
30H	* Register Code
32H	
46H	
43H	
*Data to be written to	16-Bit data comprising of 4 ASCII codes
*Data to be written to	
*Data to be written to	
*Data to be written to	
?	CHECKSUM
?	
0DH	END
0AH	END

RTU Mode (5.4.6 Coil status write con't)

PC → INV (PLC)	
01H	Address
06H	Function Code
02H	Register Code
FCH	
*Data to be written to	16-Bit data
*Data to be written to	
?	CRC High order byte
?	CRC Low order byte

INV (PLC) → PC	
01H	Address
06H	Function Code
02H	Register Code
FCH	
*Data to be written to	16-Bit data
*Data to be written to	
?	CRC High order byte
?	CRC Low order byte

note: Write to Coil Start code

INPUT	02FDH
TIMER	02FEH
COUNTER	02FFH
Auxiliary coil	0300H
Control coil	0301H
ANALOG	0302H
ENCODER	0302H
OUTPUT	0303H

5.5 Inverter Control

5.5.1 Command DATA (Readable and Writable)

Register Code	Bit	Content
0100H	Not used	
0101H	Operation Signal	0 Operation Command 1 : Run 0 : Stop
		1 Reverse Command 1 : Reverse 0 : Forward
		2 External Fault 1 : Fault (EFO)
		3 Fault Reset 1 : Reset
		4 Log Command 1 : Log
		5 Multi function Command S1 1 :“ON” (Define 5-00 Function)
		6 Multi function Command S2 1 :“ON” (Define 5-01 Function)
		7 Multi function Command S3 1 :“ON”(Define 5-02 Function)
		8 Multi function Command S4 1 :“ON”(Define 5-03 Function)
		9 Multi function Command S5 1 :“ON”(Define 5-04 Function)
		A Multi function Command S6 1 :“ON”(Define 5-05 Function)
		B Multi function Command AIN 1 :“ON”(Define 5-06 Function)
		C Multi function Command 1 1 : R1A “ON” (Define 8-02 Function)
D Multi function Command 2 1 : R2A “ON” (Define 8-03 Function)		
E-F	(Not used)	
0102H	Frequency Command (1/0.01Hz)	
0103~011FH	Not used	

(Note) Bits that are not used are defined as “0”, registers defined as “Not used”, are Read only.

5.5.2 Supervision Data (Read only)

Register code	Bit	Content
0120H	0	Operation State 1 : Run 0 : Stop
	1	Direction State 1 : Reverse 0 : Forward
	2	Inverter ready 1 : yes 0 : no
	3	No Fault 1 : Fault
	4	DATA setting error 1 : Error
	5-F	Not used

Note: Define the unused Bits as 0. Write 0 to bits that are not used.

Register code	Content	
0121H	00	The inverter is normal
	01	Program abnormal(CPF) 24 Under voltage during running (LV-C)
	02	EEPROM abnormal (EPR) 25 ~ 28 (not used)
	03	Over voltage (OV) 29 (Err8)
	04	Under voltage(LV) 30 Stop at 0 Hz(STP0)
	05	Inverter over heat (OH) 31 Direct start disable (STP1)
	06 ~ 09 (not used)	32 Control panel emergency stop (STP2)
	10	Over current during decelerating (OC-D) 33 Emergency stop (E.S)
	11	Over current during accelerating (OC-A) 34 External BB(bb)
	12	Over current at constant speed (OC-C) 35 Auto Tuning error(ATER)
	13	Over voltage at constant speed / decelerating (OV-C) 36 PID feedback signal loss(PDER)
	14	Inverter over heat at constant speed (OH-C) 37 Communication error(EFO)
	15	Inverter over speed (OVSP) 38 Encoder signal loss (ECER) *1
	16	CPU interrupted (CTER) 39 Analog converting error(Err4)
	17	Over current at Stop (OC-S) 40 Parameter locked(LOC)
	18~19 (Not used)	41 Keypad operation error (Err1)
	20	Over current at stop(OC) 42 Parameter setting error (Err2)
	21	Motor over load (OL1) 43 Modifying the parameter in communication(Err5)
	22	Inverter over load (OL2) 44 Communication failure (Err6)
	23	Over torque detected (OL3) 45 Parameter setting error (Err7)

Register code (con't)	Content (con't)		
0122H	Sequent input value	0	Terminal S1 1 : OFF
		1	Terminal S2 1 : OFF
		2	Terminal S3 1 : OFF
		3	Terminal S4 1 : OFF
		4	Terminal S5 1 : OFF
		5	Terminal S6 1 : OFF
		6	Terminal AIN 1 : OFF
	7-9	(not used)	
	Terminal output	A	Multifunction output 1(RELAY1) (1 : R1A ON 0 : R1A OFF)
		B	Multifunction output 2(RELAY2) (1 : R2A ON 0 : R2A OFF)
C~F		(Unused)	
0123H	Frequency command (1/0.01Hz)		
0124H	Output frequency (1/0.01Hz)		
0125H	Output voltage command (1/1V)		
0126H	DC Bus voltage command (1/1V)		
0127H	Output current (1/0.1A)		
0128H	Reserved		
0129H	Output torque (1/0.1%)		
012AH	PID Feedback value (1/0.1%)		
012BH	PID input value (100% / Max output frequency, 10/1% , sign attached)		
012CH	TM2 AIN input value (1/0.1%) (Data only available when parameter 1-06=2)		
012DH	TM2 AIN2/S6 input value (1/0.1%)		
012FH	Analog Input AIN (8192=100%)		
0130H	Pulse Frequency		

(Note :) The ready-to-use register is not available for the data write.

5.5.3 Read data from holding register [03H]

Read the data from a register at a specified address.

Example: Read the frequency reference command from address 1.

ASCII Mode

Command Signal	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function
33H	Code
30H	Start to encode
31H	
32H	
33H	
30H	Data
30H	
30H	
31H	
?	
?	LRC CHECK
0DH	END
0AH	

Respond signal (Normally)	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function Code
33H	
30H	DATA
32H	Quantity
31H	Initial holding register
37H	
37H	
30H	LRC CHECK
?	
?	END
0DH	
0AH	

Respond signal (Error)	
3AH	STX
30H	FOLLOWER
32H	Address
38H	Function
33H	Code
35H	Exception Code
32H	
0DH	END
0AH	

RTU Mode (5.5.3 Read data from holding register [03H] *con't*)

Command Signal		
FOLLOWER Address		01 H
Function Code		03H
Start to encode	High	01H
	Low	23H
Data quantity	High	00H
	Low	01H
CRC-16	High	74H
	Low	3CH

Respond signal (Normally)		
FOLLOWER Address		01H
Function Code		03H
DATA		02H
Initial holding register	High	17H
	Low	70H
CRC-16	High	AFH
	Low	82H

Respond signal (Error)		
FOLLOWER Address		02H
Function Code		83H
Exception Code		52H
CRC-16	High	C0H
	Low	CDH

5.5.4 LOOP BACK TEST [08H]

The test function to check the transmission of the signal between MASTER and FOLLOWER.

ASCII Mode

Command Signal	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function
38H	Code
30H	Check Code
30H	
30H	
30H	
41H	DATA
35H	
33H	
37H	
?	LRC CHECK
?	
0DH	END
0AH	

Respond signal (Normally)	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function Code
38H	
30H	Check Code
30H	
30H	
30H	
41H	DATA
35H	
33H	
37H	
?	LRC CHECK
?	
0DH	END
0AH	

Respond signal (Error)	
3AH	STX
30H	FOLLOWER
31H	Address
38H	Function
38H	Code
32H	Exception
30H	Code
?	LRC CHECK
?	
0DH	END
0AH	

RTU Mode (5.5.4 LOOP BACK TEST [08H] con't)

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

Respond signal (Normally)		
FOLLOWER Address		01H
Function Code		08H
Check Code	High	00H
	Low	00H
DATA	High	A5H
	Low	37H
CRC-16	High	DAH
	Low	8DH

Respond signal (Error)		
FOLLOWER Address		01H
Function Code		88H
Exception Code		01H
CRC-16	High	47H
	Low	D8H

5.5.5 Write to holding register [06H]

Send data to a specific holding register.

(e.g.) send the frequency command 60.0Hz from the PLC to FOLLOWER with address 1.

ASCII Mode

Command Signal	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function
36H	Code
30H	Start to encode
31H	
30H	
32H	
31H	DATA
37H	
37H	
30H	
?	LRC CHECK
?	
0DH	END
0AH	

Respond signal (Normally)	
3AH	STX
30H	FOLLOWER
31H	Address
30H	Function Code
36H	
30H	Start to encode
31H	
30H	
32H	
31H	DATA
37H	
37H	
30H	
?	LRC CHECK
?	
0DH	END
0AH	

Respond signal (Error)	
3AH	STX
30H	FOLLOWER
31H	Address
38H	Function
36H	Code
35H	Exception
32H	Code
?	LRC CHECK
?	
0DH	END
0AH	

RTU Mode

Command Signal		
FOLLOWER	Address	01 H
Address		
Function Code	06H	
Start to encode	High	01H
	Low	02H
Data	High	17H
	Low	70H
Quantity	High	27H
	Low	E2H
CRC-16	High	27H
	Low	E2H

Respond signal (Normally)		
FOLLOWER	Address	01H
Address		
Function Code	06H	
Start to encode	High	01H
	Low	02H
Data	High	17H
	Low	70H
Quantity	High	27H
	Low	70H
CRC-16	High	27H
	Low	E2H

Respond signal (Error)		
FOLLOWER	Address	01H
Address		
Function Code	86H	
Exception Code	52H	
CRC-16	High	C3H
	Low	9DH

5.5.6 Write data to multiple holding registers [10H]

Send data to multiple registers starting at a specific address.

e.g. send a frequency command 'Running forward 60 Hz.' to Follower with address 1.

ASCII Mode

Command Signal	
3AH	STX
30H	FOLLOWER
31H	
31H	Function Code
30H	
30H	Start coding
31H	
30H	
31H	
30H	Data quantity
30H	
30H	
32H	
30H	DATA *
34H	
30H	Primary DATA
30H	
30H	
31H	
31H	Secondary DATA
37H	
37H	
30H	LRC CHECK
?	
?	

Respond signal (Normally)	
3AH	STX
30H	FOLLOWER
31H	
31H	Function Code
30H	
30H	Start coding
31H	
30H	
31H	
30H	Data quantity
30H	
30H	
32H	
?	LRC CHECK
?	
0DH	END
0AH	

Respond signal (Error)	
3AH	STX
30H	FOLLOWER
31H	
39H	Function Code
30H	
35H	Exception Code
32H	
?	LRC CHECK
?	
0DH	END
0AH	

ASCII Mode (5.5.6 Write data to multiple holding registers [10H] con't)

0DH	END
0AH	

* DATA Quantity is equal to the Number multiplied by 2.

RTU Mode (5.5.6 Write data to multiple holding registers [10H] con't)

Command Signal		
FOLLOWER Address		01 H
Function Code		10H
Start to encode	High	01H
	Low	01H
Data quantity	High	00H
	Low	02H
DATA number *		04H
Primary DATA	High	00H
	Low	01H
Secondary DATA	High	17H
	Low	70H
CRC-16	High	60H
	Low	27H

Respond signal (Normally)		
FOLLOWER Address		01H
Function Code		10H
Start to encode	High	01H
	Low	01H
Data quantity	High	00H
	Low	02H
CRC-16	High	11H
	Low	F4H

Respond signal (Error)		
FOLLOWER Address		01H
Function Code		90H
Exception Code		52H
CRC-16	High	CDH
	Low	FDH

* 'DATA number' is equal to the 'data quantity' multiplied by 2.

Appendix: MODBUS Register Addresses

Register Address	Drive Parameter	Register Address	Drive Parameter
0000H	0-0	0023H	3-11
0001H	0-1	0024H	3-12
0002H	0-2	0025H	3-13
0003H	0-3	0026H	3-14
0004H	0-4	0027H	3-15
0005H	0-5	0028H	3-16
0006H	0-6	0029H	3-17
0007H	0-7	002AH	3-18
0008H	0-8	002BH	3-19
0009H	1-0	002CH	3-20
000AH	1-1	002DH	3-21
000BH	1-2	002EH	3-22
000CH	1-3	002FH	4-0
000DH	1-4	0030H	4-1
000EH	1-5	0031H	4-2
000FH	1-6	0032H	4-3
0010H	1-7	0033H	4-4
0011H	2-0	0034H	4-5
0012H	2-1	0035H	5-0
0013H	2-2	0036H	5-1
0014H	2-3	0037H	5-2
0015H	2-5	0038H	5-3
0016H	2-6	0039H	5-4
0017H	2-7	003AH	5-5
0018H	3-0	003BH	5-6
0019H	3-1	003CH	5-8
001AH	3-2	003DH	5-7
001BH	3-3	003EH	6-0
001CH	3-4	003FH	6-1
001DH	3-5	0040H	6-2
001EH	3-6	0041H	6-3
001FH	3-7	0042H	6-4
0020H	3-8	0043H	6-5
0021H	3-9	0044H	6-6
0022H	3-10	0045H	6-7

(MODBUS Register Addresses *con't*)

Register Address	Drive Parameter	Register Address	Drive Parameter
0046H	6-8	0069H	10-6
0047H	7-0	006AH	10-7
0048H	7-1	006BH	10-8
0049H	7-2	006CH	10-9
004AH	7-3	006DH	11-0
004BH	7-4	006EH	11-1
004CH	7-5	006FH	11-2
004DH	8-0	0070H	11-3
004EH	8-1	0071H	11-4
004FH	8-2	0072H	11-5
0050H	8-3	0073H	11-6
0051H	8-4	0074H	11-7
0052H	8-5	0075H	12-0
0053H	9-0	0076H	12-1
0054H	9-1	0077H	12-2
0055H	9-2	0078H	12-3
0056H	9-3	0079H	12-4
0057H	9-4	007AH	12-5
0058H	9-5	007BH	12-6
0059H	9-6	007CH	13-0
005AH	9-7	007DH	13-1
005BH	9-8	007EH	13-2
005CH	9-9	007FH	13-3
005DH	9-10	0080H	13-4
005EH	9-11	0081H	14-0
005FH	9-12	0082H	14-1
0060H	9-13	0083H	14-2
0061H	9-14	0084H	14-3
0062H	9-15	0085H	14-4
0063H	10-0	0086H	15-0
0064H	10-1	0087H	15-1
0065H	10-2	0088H	15-2 Fault Log 1
0066H	10-3	0089H	15-2 Fault Log 2
0067H	10-4	008AH	15-2 Fault Log 3
0068H	10-5	008BH	15-3

(MODBUS Register Addresses *con't*)

Register Address	Drive Parameter	Register Address	Drive Parameter
008CH	15-4	00AFH	~
008DH	15-5	00B0H	~
008EH	15-6	00B1H	~
008FH		00B2H	~
0090H	3-23	00B3H	~
0091H	3-24	00B4H	~
0092H	3-25	00B5H	~
0093H	3-26	00B6H	~
0094H	3-27	00B7H	~
0095H	3-28	00B8H	~
0096H	3-29	00B9H	~
0097H	5-7	00BAH	~
0098H	~	00BBH	~
0099H	2-4	00BCH	~
009AH	2-8	00BDH	~
009BH	12-7	00BEH	~
009CH	12-8	00BFH	~
009DH	~	00C0H	~
009EH	5-10	00C1H	~
009FH	5-11	00C2H	~
00A0H	~	00C3H	~
00A1H	~	00C4H	~
00A2H	~	00C5H	~
00A3H	~	00C6H	~
00A4H	~	00C7H	~
00A5H	~	00C8H	~
00A6H	~	00C9H	~
00A7H	~	00CAH	~
00A8H	~	00CBH	~
00A9H	~	00CCH	~
00AAH	~	00CDH	~
00ABH	~	00CEH	~
00ACH	~	00CFH	~
00ADH	~	00D0H	~
00AEH	~	00D1H	~

(MODBUS Register Addresses *con't*)

Register Address	Drive Parameter	Register Address	Drive Parameter
00D2H	~	00F3H	~
00D3H	~	00F4H	~
00D4H	~	00F5H	~
00D5H	~	00F6H	~
00D6H	~	00F7H	~
00D7H	~	00F8H	~
00D8H	~	00F9H	~
00D9H	~	00FAH	~
00DAH	~	00FBH	~
00DBH	~	00FCH	~
00DCH	~	00FDH	~
00DDH	~	00FEH	~
00DEH	~	00FFH	~
00DFH	~	~	~
00E0H	~		
00E1H	~		
00E2H	~		
00E3H	~		
00E4H	~		
00E5H	~		
00E6H	~		
00E7H	~		
00E8H	~		
00E9H	~		
00EAH	~		
00EBH	~		
00ECH	~		
00EDH	~		
00EEH	~		
00EFH	~		
00F0H	~		
00F1H	~		
00F2H	~		

NOTES