Appendix I. RS485 Communication Protocol

I-1. Use introduce

This chapter introduces something about the install and handle of RS485 communication between inverter and PLC, PC, factory computer.

RS485 standard interface

- Can communicate with all computer
- Using multi-drop link system, can link more to 127 inverters
- Completely isolated, and noise shield
- The user would use all types of RS232-485 inverter, if only the inverter had "automatic RTS control" function inside.

I-2. Specification

Communication function

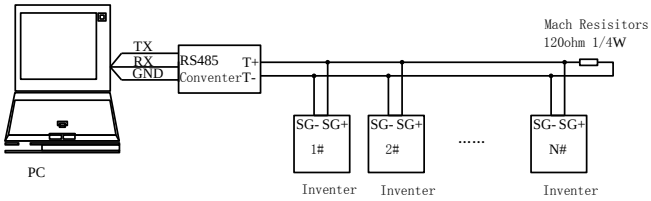
Items	Specification
Communication baud rate	38400/**19200**/9600/4800/2400/1200 bps is selectable.
Communication Protocol	Modbus protocal, RTU fomat
Interface methods	Asynchronism communication methods, semi-duplex, the previous high byte, low byte in the post, and low-effective-bit pre-emptive.
Data fumula	1 start bit, 8 data bits, 1 stop bit, No parity bit.
	1 start bit, 8 data bits, 1 stop bit, even parity bit.
	1 start bit, 8 data bits, 1 stop bit, odd parity bit.
	1 start bit, 8 data bits, 2 stop bit, No parity bit.
	1 start bit, 8 data bits, 2 stop bit, even parity bit.
	1 start bit, 8 data bits, 2 stop bit, odd parity bit.
Slave address	Slave addresses can be set up 1~ 127 0 for broadcast address, host address 128 for the proportion of linkage
Communication connect A	Terminals SG+, SG-, shield SH, Default 19200bps.
Communication connect B	RJ45, 8-core shielded cable, fixed 19200bps, N parity bit

I-3. Communication connection

I-3-1. Definition for Communication port A:

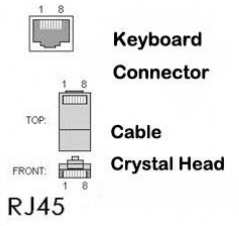
- Link RS485 communication cables to inverter control terminals (SG+), (SG-).
- When using RS232-485 transform, connect Inverter "SG+" to RS485 "T+", Inverter "SG-" to RS485 "T-".
- After Confirming connection again, turn on inverter power.
- If connection is right, set communication parameters as following:

- A26 baud rate 0: 1200, 1: 2400, 2: 4800, 3: 9600, 4: 19200, 5: 38400
- A27 current inverter communication address 1~127 (If there are more than 1 inverters, don't use the same number);
- When using RS485 running control methods, set F05=0/1/2(Keypad + RS485/CAN)



I-3-2. Definition for Communication port B:

Communication Port B pins	1	2	3	4	5	6	7	8
Communication port B signal	GND	+5V	485+	485-	485+	485-	+5V	GND
EIA/TIA T568A	White green	green	White orange	blue	white Blue	orange	White Brown	brown
EIA/TIA T568B	White Orange	Orange	White Green	Blue	White Blue	green	White Brown	brown



I-3-3. Data safety and reliability

- The number of inverter can be connected is no more than 127.
- Though the length of communication cable can add up to 1300m, considering the stability, the length limit within 800m.
- All the control signal cable use the screen cable, and is linked to the signal terminal "SH" of RS485.
- Data packet using CRC (vertical lengthy test) frame detection to ensure data reliability.
- completely isolated RS485 communication module to ensure reliable communications, support hot-swappable, after modular access, you can enter the work.
- the system is tested in 6 kinds of baud rate: 0:1200, 1:2400, 2:4800, 3:9600, 4:19200, 5:38400
- However, if under deteriorating environmental conditions, lowering the baud rate can improve the communication quality.
- Interval time of sending from frame to frame is more than 50ms.

I-4. Communication Protocol

Communication architecture is inverter as a slave, the computer as a host.

I-4-1. The basic format description

1: start of frame

Interval> 2ms,

I-4-2. Slave Address

From the machine's local address, through the A27 parameter settings, can only have that uniquely identifies a network's local address.

Setting Range 1 ~ 127.

00H = 0 ID address is broadcast mailing address, 128 ~ 255 reserved.

I-4-3. Function Code

Host to send commands from the machine to the command response.

Function Code Categories

0x02＝read slave fault record

Frame start address	Slave adress	Function code	Registers address	CRC checksum	frame end address
Interval> 2ms idle bit	1 bytes	1 bytes	2 bytes	2 bytes	Interval> 2ms idle bit

Slave response

Frame start address	Slave adress	Function code	Registers address	CRC checksum	frame end address
Interval> 2ms idle bit	1 bytes	1 bytes	2 bytes	2 bytes	Interval> 2ms idle bit

0x03＝read single register

Host command

Frame start address	Slave adress	Function code	Registers address	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	2 bytes	2 bytes	Interval> 2ms idle bit

Slave response

Frame start address	Slave adress	Function code	Registers address	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	2 bytes	2 bytes	Interval> 2ms idle bit

0x13＝Read multiple registers

Host command

Frame start address	Slave adress	Function code	Register N.	Register add.1	…	Register add. n.	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	1 bytes	2 bytes	…	2 bytes	2 bytes	Interval> 2ms idle bit

Slave response

Frame start address	Slave adress	Function code	Register N.	Register add.1	…	Register add. n.	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	1 bytes	2 bytes	…	2 bytes	2 bytes	Interval> 2ms idle bit

0x06 = write to slave a single register (power-down does not save)
0x26 = write to slave a single register (power-down save)
Host Command

Frame start address	Slave adress	Function code	Register address	Register data	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	2 bytes	2 bytes	2 bytes	Interval> 2ms idle bit

Slave response

Frame start address	Slave adress	Function code	Register address	Register data	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	1 bytes	2 bytes	2 bytes…	Interval> 2ms idle bit

0x16 = Write multiple registers to the slave (power-down does not save)

0x36 = Write multiple registers to the slave (power-down save)
Host Command

Frame start address	Slave adress	Function code	Register N.	Register add.1	Register data 1	…	Register add. n	Register data n	CRC checksum	frame end adress
Interval > 2ms idle bit	1 bytes	1 bytes	1 bytes	2 bytes	2 bytes	…	2 bytes	2 bytes	2 bytes	Interval > 2ms idle bit

Response command

Frame start address	Slave adress	Function code	Registers N.	CRC checksum	frame end adress
Interval> 2ms idle bit	1 bytes	1 bytes	1 bytes	2 bytes	Interval> 2ms idle bit

If slave response and get back to below function code, it means communications abnormal.

0xA0 = Invalid operation, setting under this state is invalid

0xA1 = function code is invalid

0xA2 = Fault record is empty

0xA3 = register address is invalid

0xA4 = slave is busy, EEPROM delay.

0xA5 = administrator restricted

0xA6 = set value is beyond limit.

0xA7 = CRC checksum error

0xA8 = frame format error

I-4-4. Register Address:

The register address includes two bytes, data setting is constituted by a two-byte.

Function code	Register Address high byte		Register Address low byte	
0x03/0x13 (read slave function code parameter)	Parameter group		Parameter serial number	
	F	0x00	0~63	
	A	0x01	0~63	
	o	0x02	0~71	
	H	0x03	0~55	
	U	0x04	0~15	
	P	0x05	0~15	
	E	0x06	0~23	
	C	0x07	0~47	
	b	0x08	0~23	
	y	0x09	0~23	
	L	0x0A	0~31	
	S	0x0B	0~15	
	R	0x10	0x00	Running status[NOTE 2]
			0x01	Reserved status 1
			0x02	Reserved status 2
			0x03	Reserved status 3
0x06/0x16(set slave function code parameter and write RAM) 0x26/0x36(set slave function code parameter and write EEPROM)	Parameter group	High byte data	Low byte data	
	F	0x00	0~63	
	A	0x01	0~63	
	o	0x02	0~71	
	H	0x03	0~55	
	U	0x04	0~15	
	P	0x05	0~15	
	E	0x06	0~23	
	C	0x07	0~47	
	b	0x08	0~23	
	y	0x09	0~23	
	L	0x0A	0~31	
	R	0x10	0x00	Running command[NOTE 3]
			0x01	Reserved status 1
			0x02	Reserved status 2
			0x03	Reserved status 3

0x02(read the fault record)	**Fault record**	**data**	**Fault inquiry content**	**data**
	Fault history record 1	0x00	Faulty type[NOTE 4]	0x00
	Fault history record 2	0x01	Set frequency	0x01
	Fault history record 3	0x02	Actual frequency	0x02
	Fault history record 4	0x03	Actual current	0x03
	Fault history record 5	0x04	Dc voltage	0x04
			Running status [NOTE 2]	0x05
			Running time	0x06
			IGBT temperature	0x07

NOTE 1:

<table>
<tr><th>Function</th><th colspan="4">0x03/0x13 reading operation</th><th>0x06/0x16 writing operation
0x26/0x36 reading operation</th></tr>
<tr><td>y00 reset the factory setting</td><td colspan="4">Return 0</td><td>Only can write into 5</td></tr>
<tr><td>y01 upload parameter onto keyboad</td><td colspan="4">Return 0</td><td>Invalid operation</td></tr>
<tr><td>y02 latest fault record</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td rowspan="3">y03~y07
fault history record</td><td colspan="3">Empty record</td><td>00H</td><td rowspan="3">Invalid operation</td></tr>
<tr><td colspan="3">New record</td><td>01H</td></tr>
<tr><td colspan="3">Confirmed record</td><td>02H</td></tr>
<tr><td>y08 reset fault record</td><td colspan="4">Return 0</td><td>Valid operation</td></tr>
<tr><td>y09 rated output current</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td>y10 rated output voltage</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td rowspan="3">y11 products series</td><td>80</td><td>0</td><td colspan="2">3</td><td rowspan="3">Invalid operation</td></tr>
<tr><td>Family serial</td><td>Function code</td><td colspan="2">Input voltage level</td></tr>
<tr><td colspan="4">The number should be decimalization.</td></tr>
<tr><td>y12 soft ware version</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td>y13 product date –year</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td>y14 product month–date</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td>y15 user decode</td><td colspan="4">Valid operation</td><td>Invalid operation</td></tr>
<tr><td>y16 user input password</td><td colspan="4">Valid operation</td><td>Valid operation</td></tr>
<tr><td>y17 parameter group protection</td><td colspan="4">Valid operation</td><td>Valid operation</td></tr>
</table>

NOTE 2:

<table>
<tr><th>BIT</th><th>15 BIT</th><th>14 BIT</th><th>13 BIT</th><th>12 BIT</th></tr>
<tr><td>meaning</td><td>0: N fault
1: urgent stopping fault</td><td>0: N fault
1: decelerating fault</td><td>0: N fault
1: alarming fault</td><td>0: confirmed fault
1: unconfirmed fault</td></tr>
<tr><td>bit</td><td>11 BIT</td><td>10 BIT、</td><td>9 BIT</td><td>8 BIT</td></tr>
<tr><td>meanning</td><td>reserved</td><td>reserved</td><td>reserved</td><td>0: N JOG.
1: JOG running</td></tr>
<tr><td>bit</td><td>7 BIT</td><td>6 BIT</td><td colspan="2">5 BIT 、 4 BIT</td></tr>
<tr><td>meaning</td><td>0: lower frequency not arriving
1: arrive lower frequency</td><td>0: upper frequency not arriving
1: arrive upperer frequency</td><td colspan="2">00: stopping 01: accelerating
10: decelerating
11: running in a even speed</td></tr>
<tr><td>`bit</td><td>3 BIT</td><td>2 BIT</td><td>1 BIT</td><td>0 BIT</td></tr>
<tr><td>meaning</td><td>0: running reverse
1: running forward</td><td>reserve</td><td>reserve</td><td>0: V/F control
1: SV control</td></tr>
</table>

NOTE 3:

bit	15 BIT	14 BIT	13 BIT	12 BIT
meaning	reserve	reserve	reserve	reserve
Bit	11 BIT	10 BIT	9 BIT	8 BIT
meaning	reserve	reserve	reserve	reserve
bit	7 BIT	6 BIT	5 BIT	4 BIT
meaning	0: N free-stop 1:free-stop command	reserve	0: JOG stopping 1: JOG running	reserve
bit	3 BIT	2 BIT	1 BIT	0 BIT
meaning	reserve	0:reverse command 1:forward command	reserve	0:stop command 1:run command

NOTE 4: fault style code

Serial nmber nnnnumber	LED display	Fault message
0	E.OCP	IGBT power driver protect and produce hardware interrupt.
1	E.OCC	OC signal from current self-inspected circuit impact.
2	E.OCF	OC signal from drive circuit.
3	E.OU	Over voltage
4	E.LU	Lower voltage
5	E.OL	Over load
6	E.UL	Lower load
7	E.PHI	Input phase lose
8	E.EEP	EEPROM error
9	E.ntC	Over heat
10	E.dAt	Time limit fault
11	E.Set	External fault
12	E.PId	PID regulation fault
13	E.OHt	Motor over heat fault
14	E.OL2	Motor over load falult
15	E.PG	PG error
16	E.PHo	Output phase lose
17	E.COA	Rs485 communication connect A fault
18	E.COb	Rs485 communication connect B fault
19	E.CAL	Parameter Identification Failure

I-4-5. CRC checkup sum

Data meaning: data frame CRC checkup sum, using 2 bytes.

Checkup sum ＝all the CRC checkup sum in one data frame.

Valid setup and communications under normal circumstances, the host command and slave responses are as follows:

Host Command

Frame start address	Slave address	Host command code	Registers address	Setting data	CRC checkup sum
Interval> 2ms idle bit	0x08	0x06	0x0001	0x1388	0xD5C5

Slave response

Frame start address	Slave address	Host response code	Registers address	Setting data	CRC checkup sum
Interval> 2ms idle bit	0x08	0x06	0x0001	0x1388	0xD5C5

I-5. Example of communication protocol:

```
unsigned int cal_crc16    (unsigned char *data, unsigned int length)
{
unsigned int i, crc_result=0xffff;
  while(length--)
  {
    crc_result^=*data++;
    for(i=0;i<8;i++)
    {
    if(crc_result&0x01)
    crc_result=(crc_result>>1)^0xa001;
    else
    crc_result=crc_result>>1;
    }
  }
crc_result=((crc_result&0xff)<<8)|(crc_result>>8);
return(crc_result);
```