

Communication protocol

SR1568

Revision History

Rev	Date	Author	Description
1.0	10-Nov-2016	Ji GenJun	Initial version
1.1	17-Nov-2016	Ji GenJun	Update some description
1.2	22-Nov-2016	Ji GenJun	Add command to change baud rate Add crc16 codes

Revision History	- 1 -
General	- 3 -
Alarm	- 3 -
CmdList:	- 4 -
Appendix:	- 5 -
Status list:	- 5 -
Example (old version, not updated):	- 7 -
GET_STAT:.....	- 7 -
CRC16.....	- 8 -

General

This document describe protocols for server to communicate with SR1568 via RS485, the default setting is low baud rate (“9600, n, 8, 1”). From version V1.00.0034, SR1568 can support high baud rate “38400, n, 8, 1”, but when controller power up, it is in low baud rate.

The interface is RS485. So after server received data from SR1568, please delay about 15ms, before you send next command

Alarm

Send Alarm, when controller get alarm from normal status, it will send this data to server.

Name	Data sent
Alarm	ADS123:ALARM 10\r\n001;002;...\r\n (maybe more than one error, will be append as “Code;”)

List:

Code	Error information
000	No Error
001	Sensor Error
002	Over Pressure
003	Low Pressure
004	VFS Error
005	No Flow

CmdList:

No	Command	Data sent	Data received(Sample)
1	Get power on time	ADS123:POWER\r\n	ADS123:POWER 2\r\n
2	Get controller name	ADS123:GET_PRO DUCT \r\n	ADS123:SR1568\r\n
3	Get controller software version	ADS123:GET_VER SION \r\n	ADS123:V1.00.0010\r\n
4	Get controller SN	ADS123:GET_SN \r\n	ADS123:19009000\r\n
5	Get status	ADS123:GET_STAT \r\n	ADS123:GET_STAT 61\r\n (all after 61\r\n will be data of status) Check the status format below.
6	Restart controller	ADS123:RESTART \r\n	ADS123:RESTART OK\r\n
7	Get Alarm	ADS123:GET_ALA RM \r\n	ADS123:ALARM 10\r\n001;002;...\r\n (maybe more than one error, will be append as "Code;")
8	SET_TIME	ADS123: SET_TIME xxx\r\n	xxx is the total seconds from 00:00:00, for example, current time is 10:01:30 then xxx is $10*3600+1*60 + 30 = 36090$
9	SET_RATE	ADS123: SET_RATE xxx\r\n	xxx is rate value, it can be 9600 or 38400

Appendix:

Status list:

Name	Type	length	Note	Description
Crc16	ushort	2	CRC16	CRC16 of all data below
Sys	byte	1	System index(1~23)	System index(1~23)
T0	char[5]	5	Temperature of T0 (None, 0-100, Break)	Temperature of T0 (“None, Break, xxx.x”)
T1	char[5]	5	T1	
T2	char[5]	5	T2	
T3	char[5]	5	T3	
T4	char[5]	5	T4	
T5	char[5]	5	T5	
T6	char[5]	5	T6	
R1	byte	1	R1 status (0-100)	Status of pump R1
R2	byte	1	R2 status (0-100)	Status of pump R2
R3	byte	1	R3 status (0-100)	Status of pump R3
R4	byte	1	R4 status (0, 1)	Status of value R4
R5	byte	1	R5 status(0, 1)	Status of value R5
HK	byte	1	HK status(0, 1)	Status of value HK
Solar1Status	byte	1	0=OK, 1=Emer, 2=Cooling, 3=Min, 4=AntiFreeze	0=OK, 1=Emer, 2=Cooling, 3=Min, 4=AntiFreeze
Solar2Status	byte	1	0=OK, 1=Emer, 2=Cooling, 3=Min, 4=AntiFreeze	0=OK, 1=Emer, 2=Cooling, 3=Min, 4=AntiFreeze
Load1Status	byte	1	0=OK, 1=Emerg, 2=Max	0=OK, 1=Emerg, 2=Max

Load2Status	byte	1	0=OK, 1=Emerg, 2=Max	0=OK, 1=Emerg, 2=Max
WarningStatus	byte	1	0=OK, 1=Warning	0=OK, 1=Warning
DumpStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
BypassStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
TimerStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
ParRelayStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
HolidayStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
SDCardStatus	byte	1	0=NoCard, 1=CardExist	0=NoCard, 1=CardExist
AHStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
Disinfection Status	byte	1	0=OFF, 1=disinfecting, 2=Waiting, 3=Finished	0=OFF, 1=disinfecting, 2=Waiting, 3=Finished
BlockStatus	byte	1	0=OFF, 1=ON	0=OFF, 1=ON
ManualStatus	char[10]	10	First 6 bytes R1~R5, HK status; (0=OFF, 1=ON)	Status of R1~R5, HK
TVFS	char[5]	5	Temperature of VFS (None, 0.0-100.0, Break)	Temperature (None, 0.0-100.0, Break)
Flow	char[5]	5	Flow (None, 0.0-24.0, Break)	Flow (None, 0.0-24.0, Break)
Pressure	char[5]	5	Pressure (None, 0.0-20.0, Break)	Pressure (None, 0.0-20.0, Break)
Today's Measurement	char[5]	5	Today's heat (kWh)	Heat measurement
Total's Measurement (kWh)	char[5]	5	Total heat (kWh)	Total heat measurement (kWh)
Total's Measurement (char[5]	5	Total heat (mWh)	Total heat measurement (mWh)

mWh)				
PowerOn Time	int	4	Power on days	Total time of power on
Current controller time (s)	int	4	Current controller' s time	Total seconds Hour=total/3600, Minutes=(total%3600)/60
Controller Type	Char [8]	8	Controller type	“ SR1568”

Block 数据结构

域	长度(bytes)	
Block length	4	
End Flat	2	是否文件的最后一个 Block(00=最后一包, 01=不是最后一包)
Block Index	4	发送过来的 Block 索引
CRC	2	后面数据的 crc16 值
Data	Xxx	除了最后一包变长, 其他固定为 512 字节

Example (old version, not updated):

GET_STAT:

```

ADS123:GET_STAT 68
o# 58.8888.8888.8 25.1 65.6888.8888.8

```

In binary shown:

```

41 44 53 31 32 33 3A 47 45 54 5F 53 54 41 54 20 36 38 0D 0A 6F 23 14 20 35 38 2E 38 38 38 38 2E
38 38 38 38 2E 38 20 32 35 2E 31 20 36 35 2E 36 38 38 38 2E 38 38 38 38 2E 38 00 00 00 00 00
00 00 00 00 01 00 00 00 00 01 01 00 02 00 00 00 00 00 00 00 00 00

```

Detail:

Name	Lenght	Example data	Example data (Hex)
包头		ADS123:GET_STAT 68 \r\n	41 44 53 31 32 33 3A 47 45 54 5F 53 54 41 54 20 36 38 0D 0A
Crc	0-1		6F 23
Sys	2	14	14

T0	3~7	_58.7	20 35 38 2E 37
T1	8~12	888.8	38 38 38 2E 38
T2	13~17	888.8	38 38 38 2E 38
T3	18~22	_25.0	20 32 35 2E 30
T4	23~27	_65.6	20 36 35 2E 36
T5	28~32	888.8	38 38 38 2E 38
T6	33~37	888.8	38 38 38 2E 38
R1	38	0	00
R2	39	0	00
R3	40	0	00
R4	41	0	00
R5	42	0	00
HK	43	0	00
Solar1Status	44	0	00
Solar2Status	45	0	00
Load1Status	46	0	00
Load2Status	47	0	00
WarningStatus	48	1	01
DumpStatus	49	0	00
BypassStatus	50	0	00
TimerStatus	51	0	00
ParRelayStatus	52	0	00
HolidayStatus	53	1	01
SDCardStatus	54	1	01
AHStatus	55	0	00
DisinfectionStatus	56	2	02
BlockStatus	57	0	00
ManualStatus	58~67	0 0 0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00 00 00

CRC16

```

#define CRC_SEED  0xFFFF
#define POLY16 0x1021

U16 crc16Seed(unsigned char *buf, U32 length, U16 seed)
{
    U16 shift,data,val;
    int i;

    shift = seed;

    for(i=0;i<length;i++) {
        if((i % 8) == 0)
            data = (*buf++)<<8;
        val = shift ^ data;
        shift = shift<<1;
        data = data <<1;
    }
}

```



```
    if(val&0x8000)
        shift = shift ^ POLY16;
    }
    return shift;
}

U16 crc16(unsigned char *buf,U32 length)
{
    return crc16Seed(buf, length, CRC_SEED);
}
```