BMS communication protocol

48NPFC-XX-2.X

.

1. Summary

1.1 Protocol summary

This doc specify the communication protocol between BMS and the upper device. Based on Modbus protocol, self-defined data are added to complete the data interaction between BMS and upper device. The upper equipment requests or sets BMS data, and the BMS reports or receives relevant data.

1.2 Scope of application

This protocol is applicable to the BMS designated by our company, mainly for the communication between UPS and BMS.

2. Communication way

Adopt master-slave communication mode.

2.1 Reference standard

Modbus

2.2 Physical protocol

RS485

2.3 Data transfer rate

Baud rate 9600

2.4 Data format

Start bit 1, data bit 8, stop bit 1, no parity bit.

.

3. Data definition

There are two types of information:

（1）Command information send by the host to the slave

（2）The response information returned from the slave to the host

3.1 Command for read block

Command message

Number	1	2	3	4	5
Format	NODE	FUN	Register	NR	CRC
Byte number	1	1	2	2	2

Annotations in the basic format

Number	Symbol	Meaning	Description
1	NODE	Slave address for Modbus protocol	
2	FUN	Function code, fixed 0x03	
3	Register	The start register for request data	The register address that has been defined
4	NR	Register number	
5	CRC	Modbus RTU CRC16,	Send low byte first, high byte later

Respond message

Number	1	2	3	4	5
Format	NODE	FUN	NR_BYTE	DATA	CRC
Byte number	1	1	1	n	2

Annotations in the basic format

Number	Symbol	Meaning	Description
1	NODE	Slave address for Modbus protocol	
2	FUN	Function code, fixed 0x03	
3	NR_BYTE	Data bytes that return, It's equal to 2 times NR	
4	DATA	Data, data number is equal to NR_BYTE	
5	CRC	Modbus RTU CRC16	Low byte first, high byte later

.

3.2 Command for block write

Command message

Number	1	2	3	4	5	6	7
Format	NODE	FUN	Register	NR	NR_BYTE	DATA	CRC
Byte number	1	1	2	2	1	n	2

Annotations in the basic format

Number	Symbol	Meaning	Description
1	NODE	Slave address for Modbus protocol	
2	FUN	Function code, fixed 0x10	
3	Register	The start register for write data	The register address that has been defined
4	NR	Register number for write	
5	NR_BYTE	Data length, it must be two times NR.	
6	DATA	Data for write, its length must equal to NR_BYTE	
7	CRC	Modbus RTU CRC16	Low byte first, high byte later

Respond message

Number	1	2	3	4	5
Format	NODE	FUN	Register	NR	CRC
Byte number	1	1	2	2	2

Annotations in the basi

Number	Symbol	Meaning	Description
1	NODE	Slave address for Modbus protocol	
2	FUN	Function code, fixed 0x10	
3	Register	The start register for write data	
4	NR	Register number for write	
5	CRC	Modbus RTU CRC16	Low byte first, high byte later

.

3.3 Error return

Respond message

Number	1	2	3	4
Format	NODE	FUN	ERROR	CRC
Byte number	1	1	1	2

Annotations in the basic format

Number	Symbol	Meaning	Description
1	NODE	Slave address for Modbus protocol	
2	FUN	Function code + 0x80	It means a fault is detected
3	ERROR	Fault code Error code 0x01 means function code is not supported Error code 0x02 means register is not supported Error code 0x03 means register number, that is visited, is over range. Error code 0x04 device error or CRC error	
4	CRC	Modbus RTU CRC16	Low byte first, high byte later

4. Detailed register information

4.1 register message

Item	Address (Hex)	Data length(byte)	Unit	Remark
Current	01	2	10mA	
Max voltage of cell	02	2	mV	
Min voltage of cell	03	2	mV	
Summary voltage of cells	04	2	10mV	
Max temperature of ell	05	2	0.1K(Kelvin temperature)	
Min temperature of cell	06	2	0.1K(Kelvin temperature)	

.

Mosfet status	07	2	/	Bit0: Discharge MOSFET state 1: DSG MOSFET connect 0: DSG MOSFET disconnect Bit1: Charge MOSFET state 1: CHG MOSFET connect 0: CHG MOSFET disconnect Bit2: Charge and discharge enable 1: Enable 0: Disable Bit3~bit7: reserved, keep to 0
Remain capacity	08	2	10mAh	
Full charge capacity	09	2	10mAh	
RSOC	0A	2	%	
SOH	0B	2	%	
Cycle count	0C	2	/	
Pack voltage	0D	2	10mV	
Charge protect state	0E	2	/	Details show in below table 1
Charge protect state 2	0F	2	/	Details show in below table 1
Discharge protect state	10	2	/	Details show in below table 1
Discharge protect state 2	11	2	/	Details show in below table 1
RTC h:m:s	12	4	/	Data0: Not used, keep to 0 Data1: hour (0~23) Data2: minute (0~59) Data3: second (0~59)
RTC y:m:d	14	4	/	Data0: Not used, keep to 0 Data1: year (offset is 2000) Data2: month (1~12) Data3: day (1~31)
Parallel mode	16	2	/	Data0: Parallel mode 0: BMS idle 1: BMS waiting 2: Charge initialize 3: Charge waiting 4: Parallel complete 5: Enter precharge 6-7: Error Data1: Parallel number
Charge alarm	17	2	/	Details show in below table 1

.

Discharge alarm	18	2	/	Details show in below table 1
BMS control 1	19	2	/	Manufacturer use for debug
BMS control 2	1A	2	/	Manufacturer use for debug
BMS control 3	1B	2	/	Manufacturer use for debug
Temperature 1-6	1C	2*6	0.1K(Kelvin temperature)	
Cell voltage 1-16	22	2*16	mV	
Reserved	32-3B	/	/	
Manufacture name	3C	4	ASCII	
Manufacture barcode	3E	10	ASCII	
Manufacture Date	48	2	ASCII	
Manufacture week	49	2	ASCII	
Manufacture SN	4A	4	ASCII	
Main MCU Firmware version	4C	4	/	Data0: Format is ASCII Data1: Format is hex Data2: Format is ASCII Data3: Format is hex
Hardware version	4E	2	/	Data0: Format is ASCII Data1: Format is hex
Sub MCU Firmware version	4F	2	/	Data0: Format is ASCII Data1: Format is hex

Example:

When Summary voltage of cell is 50V, suppose BMS communication node is 01, to visit BMS's summary voltage of cell, master can send the message 01 03 00 04 00 01 C5 CB(CRC16).And BMS will return the data 01 03 02 13 88 B5 12(CRC16),(13 88)H = 5000 (10mV). Note, BMS will not return data if slave node ID, that send by master, is not match with BMS node ID.

.

Table 1,

Item	Bit	Content	Meaning	Remarks
Charge Protect	15	OCC2	Second level charge over current	1: Enable charge MOSFET disconnection 0: Disable charge MOSFET disconnection
	14	Reserved		
	13	CANID	CAN ID initializing	
	12	2NDOVP	Second level cell over voltage	
	11	OCC	Charge over current	
	10	OCV	Cell over voltage	
	9	CLT	Charge under temperature	
	8	CHT	Charge over temperature	
	7	SUV	Severe under voltage	
	6	FETHT	Mosfet over temperature	
	5	AFESCD	AFE detect discharge short circuit	
	4	AFEOCD	AFE detect discharge over current	
	3	AFEOCC	AFE detect charge over current	
	2	AFEComm	AFE communication fail	
	1	BoostNRDY	Mosfet driver status	
	0	PRES	"In System" signal	
Charge Protect 2	4~15	Reserved		1: Enable charge MOSFET disconnection 0: Disable charge MOSFET disconnection
	3	IDError	CAN ID conflict	
	2	Reserved		
	1	Shutdown	Low voltage shutdown	
	0	ShutdownByCmd	Command Shutdown	
Discharge protect	15	Shutdown	Low voltage shutdown	1: Enable discharge MOSFET disconnection 0: Disable discharge MOSFET disconnection
	14	DHT	Discharge over temperature	
	13	DLT	Discharge low temperature	
	12	OCD	Discharge over current	
	11	CUV	Cell under voltage	
	10	FETHT	Mosfet over temperature	
	9	Reserved		
	8	Reserved		
	7	Reserved		
	6	IDError	CAN ID conflict	
	5	Reserved		
	4	AFESCD	AFE detect discharge short circuit	
	3	AFEOCD	AFE detect discharge over	

.

			current	
	2	AFEOCC	AFE detect charge over current	
	1	BoostNRDY	Mosfet driver status	
	0	PRES	"In System" signal	
Discharge protect 2	15	Reserved		1: Enable discharge MOSFET disconnection 0: Disable discharge MOSFET disconnection
	14	Reserved		
	13	CANID	CAN ID initializing	
	12	Reserved		
	11	Reserved		
	10	Reserved		
	9	Reserved		
	8	Reserved		
	7	Reserved		
	6	ShutdownByCmd	Command shutdown	
	5	DHT2	Second level discharge over temperature	
	4	2NDOVP	Second level cell over voltage	
	3	FgInit	Fuel gauge initialing	
	2	OCD2	Second level discharge over current	
	1	Short	Detect discharge short circuit	
	0	AFEComm	AFE communication fail	
Charge alarm	4-15	Reserved		1: Charge alarm 0: No alarm
	3	OCC	Charge over current	
	2	CLT	Charge low temperature	
	1	OCV	Cell over voltage	
	0	CHT	Charge over temperature	
Discharge alarm	4-15	Reserved		1: Discharge alarm 0: No alarm
	3	CUV	Cell under voltage	
	2	FETHT	Mosfet over temperature	
	1	DLT	Discharge low temperature	
	0	DHT	Discharge over termperature	

5. Modbus CRC16

5.1 The way of calculation

Refer to the instructions in the links below
https://blog.csdn.net/liming0931/article/details/99218915

C language code can refer to the code in the link as shown in the figure below:

```
unsigned int CRC16_2(unsigned char *buf, int len)
{
    unsigned int crc = OxFFFF;
    for (int pos = 0; pos < len; pos++)
    {
        crc A= (unsigned int)buf[pos];          ADR byte irto Least sig. byte of crc
        for (int i = 8; i != 0; i--)            // t:-on          'oc": bit
        {
            if ((crc & Ox0001) != 0)            4 tine L5·E· i5 set
            {
                crc >>= 1;
                crc A= Ox4001;
            }
            else
            {
                crc >>= 1;



    crc = ((crc & Ox0Off) << 8) I ((crc & Oxff00) >> 8);
    return crc;

```