

---

# BMS Protocol for Lithium Battery Pack RS485-Modbus

REV1.30

**Prepared by: Wang MR**

**Checked by:**

**Approved by:**

---

## Revision history

Index	Description	Version	Date	Author
0	Document created.	V1.00	2016-04-25	Wang
1	Add register Id 0x1009~0x1016	V1.10	2016-06-27	Wang
2	Modify 4.1 Data acquisition	V1.20	2016-07-05	Wang
3	Get rid of Warning Flag's BIT12.	V1.30	2016-08-01	Wang
4				
5				
6				
7				
8				

## 1. CRC parity

CRC parity range is the check of all bytes before CRC field. It uses 16 bit CRC parity.

### 1.1 High significant byte CRC value

```
static unsigned char auchCRCHI[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1,
0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1,
0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40
};
```

---

## 1.2 Low significant byte CRC value

```
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB,
0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE,
0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2,
0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E,
0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B,
0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27,
0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD,
0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8,
0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4,
0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94,
0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59,
0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D,
0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
0x41, 0x81, 0x80, 0x40
};
```

---

### 1.3 Program reference of realization

```
unsigned short CRC16 (puchMsg, usDataLen) /* The function returns the CRC as a unsigned
short type */
unsigned char *puchMsg ; /* message to calculate CRC upon */
unsigned short usDataLen ; /* quantity of bytes in message */
{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
    unsigned ulIndex ; /* will index into CRC lookup table */
    while (usDataLen--) /* pass through message buffer */
    {
        ulIndex = uchCRCLo ^ *puchMsg++ ; /* calculate the CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[ulIndex] ;
        uchCRCHi = auchCRCLo[ulIndex] ;
    }
    return (uchCRCHi<< 8 | uchCRCLo) ;
}
```

Code: 《MODBUS over Serial Line Specification and Implementation Guide V1.02》

## 2. Communication Parameters

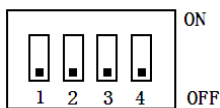
Baud Rate: 9600

Parity bit: No

Data Bits: 8

Stop Bit: 1

BMS code switch:



Communication slave address depend on BMS code switch value. It's range: 0~15.

Such as:

Slave address 0: 1、2、3、4 all are down.

Slave address 1: 1 is up, 2、3、4 are down.

Slave address 2: 2 is up, 1、3、4 are down.

.....

Slave address 15: 1、2、3、4 all are up.

### 3. Frame format of communication data

#### 3.1 List of function code

Function code	Meaning	Notes
0x04	Read command	support to read single or multi register sequentially
0x11	Reading the information of the product	Model of product, software version, hardware version, serial number

Note:

1. Function code 0x04 means that slave node upload battery pack information collected to host node when slave node accept command 0x04 from host node.
2. Function code 0x11 means that slave node upload product information to host node when slave node accept command 0x11 from host node.

#### 3.2 Read the collected information frame format

Note 1: Type of command 0x04

Note 2: MSB means high significant byte, LSB means low significant byte.

Note 3: Each register store two bytes; Register data type is one byte of data, it is required to store in (LSB).

##### 3.2.1 Host node sending frame format

Item	0	1	2	3	4	5	6	7
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Explanation	Controller address	Type of command(0x04)	beginning register address		Resister number n		CRC parity	

##### 3.2.2 Normal response of frame format from slave node

Item	0	1	2	3	4	...	L+1	L+2	L+3	L+4
Field definition	ADDR	CMD	Length	MSB	LSB	...	MSB	LSB	LSB	MSB
Explanation	Controller address	Type of command	Bytes transferred $L=n*2$	First register's value		...	Last register's value		CRC parity	

### 3.2.3 Abnormal response of format from slave node

Item	0	1	2	3	4
Field definition	ADDR	CMD+128	Err Code	LSB	MSB
Explanation	Controller address	Type of command +128	Error Code	CRC parity	

Note: CRC parity range is the check of all bytes before CRC field.

## 3.3 Read product information frame format

Note 1: Type of command 0x11

Note 2: MSB means high significant byte, LSB means low significant byte.

Note 3: Each register store two bytes; Register data type is one byte of data, it is required to store in (LSB).

### 3.3.1 Host node sending frame format

Item	0	1	2	3
Field definition	ADDR	CMD	LSB	MSB
Explanation	Controller address	Type of command(0x11)	CRC parity	

### 3.3.2 Normal response of frame format from slave node

Item	0	1	2	3	4	...	L+1	L+2	L+3	L+4
Field definition	ADDR	CMD	Length	BYTE1 ... BYTE n					LSB	MSB
Explanation	Controller address	Type of command	Bytes transferred L=n	Model of product+ '*' + software version+ '*' + hardware version + '*' + serial number+ '*'					CRC parity	

Note: Each content should be separated by '\*' of model of product, software version, hardware version, serial number

### 3.3.3 Abnormal response of format from slave node

Item	0	1	2	3	4
Field definition	ADDR	CMD+128	Err Code	LSB	MSB
Explanation	Controller address	Type of command +128	Error Code	CRC parity	

Note: CRC parity range is the check of all bytes before CRC field.

### 3.4 Error Code

Err Code	NAME	Remark
1	Illegal function code	Function that does not exist
2	Illegal function address	Register address that does not exist
3	Illegal data operation	Its operation is not allowed

## 4. Data information

### 4.1 Data acquisition

Note 1: if the reading data is invalid value, reported 0xFFFF

Note 2: Each register stores two bytes

Absolute Address	Relative Address	Name	Bytes	Data type	Unit	Range	Remark
34097	0x1000	Voltage of battery pack	2	UINT16	10mV	0..655.35V	
34098	0x1001	Current of battery pack	2	INT16	10mA	-327.68..327.67A	
34099	0x1002	Remaining capacity	2	UINT16	10mAH	0~655.35AH	
34100	0x1003	Average of cell temperature	2	INT16	0.1℃	-40.0..120.0℃	
34101	0x1004	Environment temperature	2	INT16	0.1℃	-40.0..120.0℃	
34102	0x1005	Warning Flag	2	HEX	bit	0000..FFFF	See description
34103	0x1006	Protection Flag	2	HEX	bit	0000..FFFF	See description
34104	0x1007	Fault/Status	2	HEX	bit	0000..FFFF	See description
34105	0x1008	SOC	2	UINT16	0.1%	0..100.0%	
34106	0x1009	SOH	2	UINT16	0.1%	0..100.0%	
34107	0x100A	Full charged capacity	2	UINT16	10mAH	0~655.35AH	
34108	0x100B	Cycle count	2	UINT16	Cyc	0..65535	
34109	0x100C	Max charging current	2	UINT16	10mA	-327.68..327.67A	Charging current of enable limit function
34110	0x100D	Max cell voltage	2	UINT16	mV	0..65535 mV	
34111	0x100E	Min cell voltage	2	UINT16	mV	0..65535 mV	
34112	0x100F	Reserved	2	-	-	-	
34113	0x1010	Max cell temperature	2	INT16	0.1℃	-40.0..120.0℃	
34114	0x1011	Min cell temperature	2	INT16	0.1℃	-40.0..120.0℃	
34115	0x1012	FET temperature	2	INT16	0.1℃	-40.0..120.0℃	
34116	0x1013	Reserved	2	-	-	-	



---

34117	0x1014	Nominal Float Voltage	2	UINT16	10mV	0..655.35V	
34118	0x1015	Design capacity	2	UINT16	10mAH	0~655.35AH	
34119	0x1016	Reserved	2	-	-	-	

## 4.2 Information of product

	Number of bytes	Example	Note
Model of product	15 (max number of bytes)	P16S50A-6232	ASCII transmission
software version	2 (fixed)	0x0100, version: V1.00	16 hexadecimal code
hardware version	2 (fixed)	0x0120, version: V1.20	16 hexadecimal code
serial number	20 (fixed)	20161111011800400026	ASCII transmission

## 5. Instruction

### 5.1 Warning Flag

Byte0	Bit0	1: battery cell overvoltage alarm 0: not occurring	
	Bit1	1: battery cell low voltage alarm 0: not occurring	
	Bit2	1: battery pack overvoltage alarm 0: not occurring	
	Bit3	1: battery pack low voltage alarm 0: not occurring	
	Bit4	1: charging over current alarm 0: not occurring	
	Bit5	1: discharging over current alarm 0: not occurring	
	Bit6	1: battery high temperature alarm 0: not occurring	
	Bit7	1: battery low temperature alarm 0: not occurring	
Byte1	Bit0	1: environment high temperature alarm 0: not occurring	
	Bit1	1: environment low temperature alarm 0: not occurring	
	Bit2	1: MOSFET high temperature alarm	

---

		0: not occurring	
	Bit3	1: low capacity alarm 0: not occurring	
	Bit4	reserve	
	Bit5	reserve	
	Bit6	reserve	
	Bit7	reserve	

## 5.2 Protection Flag

Byte0	Bit0	1: battery cell over voltage protection 0: not occurring	
	Bit1	1: battery cell low voltage protection 0: not occurring	
	Bit2	1: battery pack over voltage protection 0: not occurring	
	Bit3	1: battery pack low voltage protection 0: not occurring	
	Bit4	1: short circuit protection 0: not occurring	
	Bit5	1: over current protection 0: not occurring	
	Bit6	1: charging high temperature protection 0: not occurring	
	Bit7	1: charging low temperature protection 0: not occurring	
Byte1	Bit0	1: discharging high temperature protection 0: not occurring	
	Bit1	1: discharging low temperature protection 0: not occurring	
	Bit2	reserve	
	Bit3	reserve	
	Bit4	reserve	
	Bit5	reserve	
	Bit6	reserve	
	Bit7	reserve	

## 5.3 Fault/Status Flag

Byte0 (Fault)	Bit0	1: front end sampling communication fault 0: not occurring	
	Bit1	1: temperature sensor break 0: not occurring	
	Bit2	reserve	

---

	Bit3	reserve	
	Bit4	reserve	
	Bit5	reserve	
	Bit6	reserve	
	Bit7	reserve	
Byte1 (Status)	Bit0	1: state of charge 0: not occurring	
	Bit1	1: state of discharge 0: not occurring	
	Bit2	1: charge MOSFET is ON 0: charge MOSFET is OFF	
	Bit3	1: discharge MOSFET is ON 0: discharge MOSFET is OFF	
	Bit4	1: charge limit current function is ON 0: charge limit current function is OFF	
	Bit5	reserve	
	Bit6	reserve	
	Bit7	reserve	

**For example:**

Query from host to BMS module.

0x00	0x04	0x10	0x00	0x00	0x17	0xb5	0x15
0x01	0x04	0x10	0x00	0x00	0x17	0xb4	0xc4
...	...	...	...	...	...	...	...
0x0E	0x04	0x10	0x00	0x00	0x17	0xb4	0x3b
0x0F	0x04	0x10	0x00	0x00	0x17	0xb5	0xea