PRODUCT SPECIFICATION

PRODUCT TYPE: **ACM-100**

PRODUCT DESCRIPTION: **MEMS MODBUS PROTOCOL OUTPUT TRI-AXIS ACCELEROMETER**

麦新敏微

Contact Us
HANGZHOU MAIXINMINWEI TECHNOLOGY CORP., LTD
Email: sales@memsmag.com
Whatsapp: +8618151836753
Tel/wechat: +8618621961329
Website: www.mxmwkj.com

1. Product characteristics

The ACM-100 tri-axis accelerometer is a widely used acceleration sensor series product produced by MXMW Hi-Tech Company with Swiss patented technology, which can be used in various fields such as vibration testing and impact testing. The product adopts digital interface output, with RS232, RS485, TTL, Modbus protocol, and RS422 optional. Different address codes can be set, and multiple sensors can be connected in series for long-distance measurement and data analysis at multiple points. ACM-100 is a single crystal silicon capacitive sensor, consisting of a silicon chip that has undergone micro mechanical processing, a low-power ASIC for signal adjustment, a microprocessor for storing compensation values, and a temperature sensor.

The product has a small size, low power consumption, calibrated, fixed structure, and stable output. The new electronic configuration provides solid-state power for reset and comprehensive protection against overvoltage. The long-term stability and typical deviation of the proportional factor within the full range are less than 0.1%. This series of products has the characteristics of fixed structure, low power consumption, and excellent deviation stability, ensuring outstanding output reliability.

2. Product performance

- Tri-axis (X, Y, Z) optional
- Measuring range: ±1g/±2g/±4g/±8g/±16g/±32g/±40g
- Voltage input: DC 9~36V (DC 5V can be customized)
- Excellent deviation stability and vibration resistance
- Full range accuracy: 0.01g;
- High resolution: 0.001g
- Output mode: RS232/RS485/TTL/RS422/Modbus optional
- Widely operating temperature: -55 ~ +100℃
- Size: (60*59*29mm) (can be customized)
- Protection level: IP67 (IP68 customized)

3. Product application

- Crash records, fatigue monitoring and prediction
- Ship navigation attitude measurement
- Roadbed analysis and high-speed railway fault detection
- Satellite solar antenna positioning
- Unmanned aerial vehicles
- Transportation system monitoring

4. Product specification

Parameter	Conditions	ACM-I00-2	ACM-I00-8	ACM-I00-40	Unit
Measuring range		±2	±8	±40	g
Bias calibration		<2	<5	<10	mg
Measuring axis	axis	X, Y, Z	X, Y, Z	X, Y, Z	
Zero bias stability (yearly)		1.5(<5)	7.5(<25)	22(<75)	mg
Resolution threshold	@Hz	<1	<5	<15	mg
Bias temperature coefficient	-55 ~ 100°	0.1	0.5	1.5	mg/℃
Bandwidth		0~⩾400	0~⩾400	0~⩾400	Hz
Resonance frequency		1.6	6.7	6.7	KHz
Output rate	5Hz, 15Hz, 35Hz, 50Hz can be set (RS485 does not have this function)				
Output signal	RS232/RS485/RS422/TTL optional				
Reliability	MIL-HDBK-217, Level 2				
Impact resistance	20000g, 2ms, 1/2sine				
Anti-vibration	10grms、10 ~ 1000Hz				
Waterproof level	IP67				
Cable	Standard 1.5m length, wear-resistant, oil-proof, wide temperature, shielded cable 5*0.2mm2				
Weight	180g (excluding packaging box)				
Connector	6-pin aviation plug				
Capacitive loading	1000				

* This performance parameter only lists ±2g, ±8g, ±40g series for reference, and for other measurement ranges, please refer to the nearest adjacent parameters

5. Electrical indicators

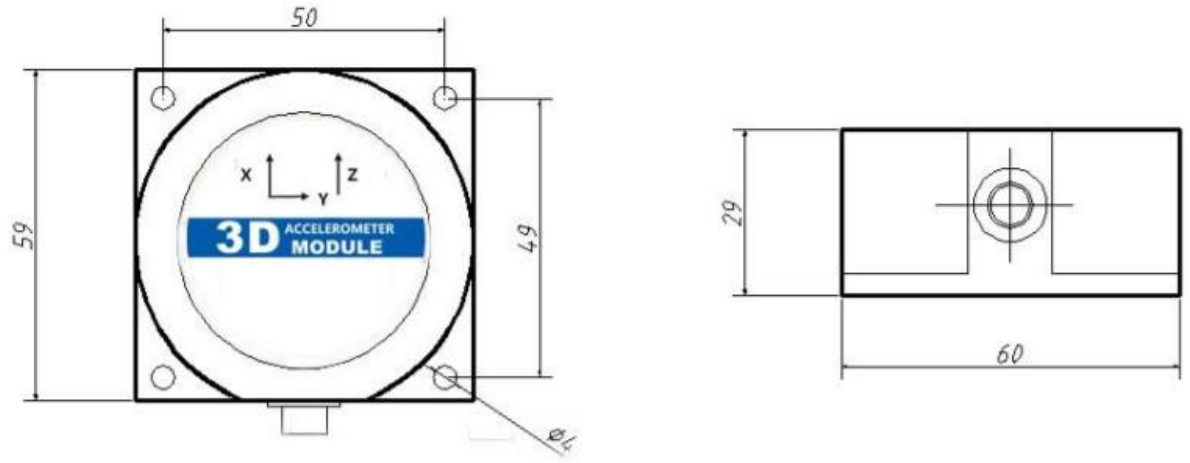
Parameter	Condition	Minimum value	Typical value	Maximum value	Unit
Supply voltage		9	12	36	V
	customizable		5		
Working current	no load		30		mA

Operating temperature		-40		+85	℃
Storage temperature		-55		+125	℃

6. Mechanical properties

Connector	Lead wire (1.5m) or waterproof aviation socket (customized)
Protection level	IP67
Material	Aluminum alloy frosted and oxidized
Installation	4 M4 screws

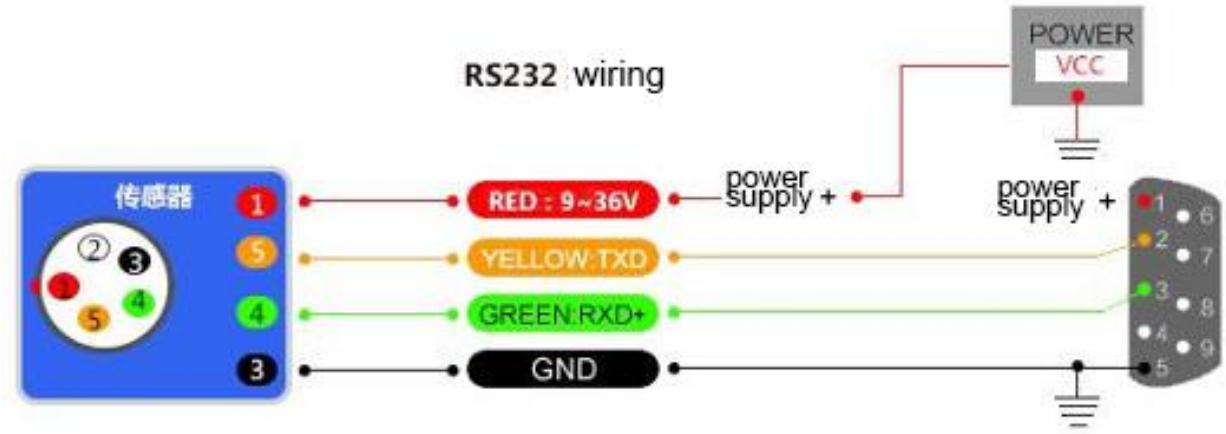
7. Product dimension



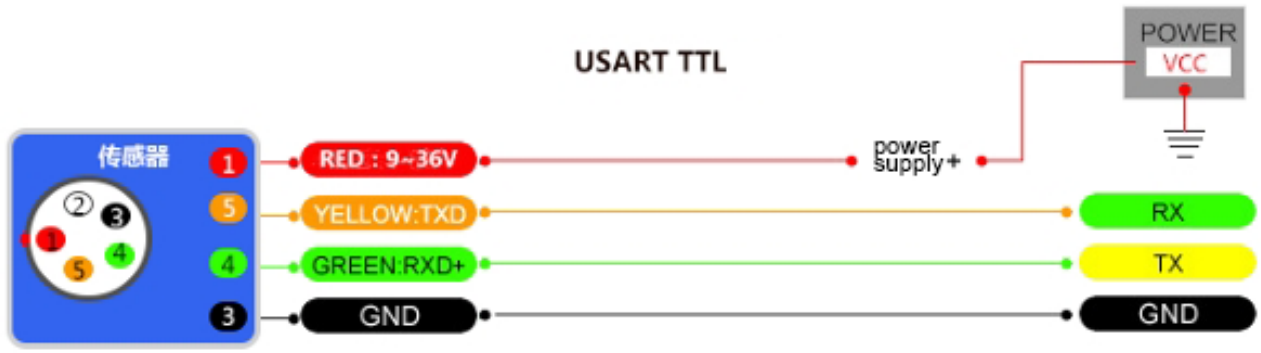
Product size: L60*W59*H29MM

8. Electrical connections

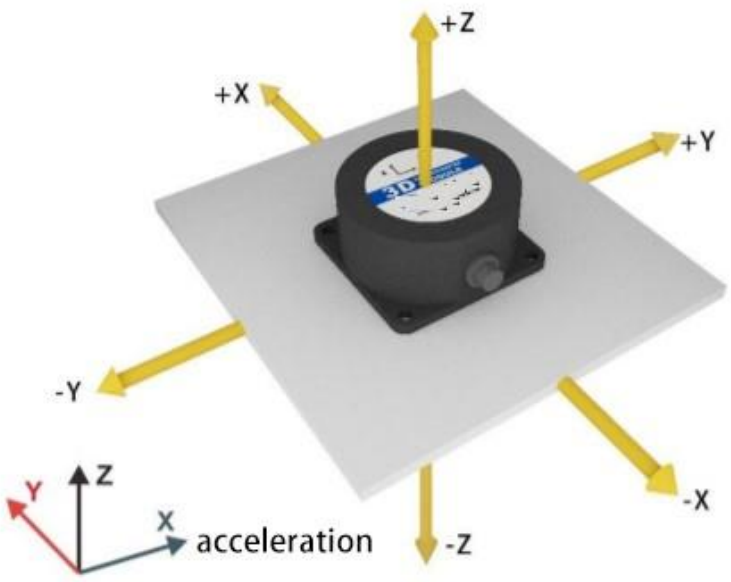
Thread Color Function	RED 1	WHITE 2	BLACK 3	GREEN 4	YELLOW 5
RS232	VCC	NC	GND	RXD	TXD
RS485 Modbus	VCC	NC	GND	(B、D-)	(A、D+)
TTL	VCC	NC	GND	RXD	TXD



Thread Color Function	RED	BLACK	GREEN	YELLOW	WHITE	BROWN
RS422	1	2	3	4	5	6
	VCC	GND	RXD-(B-)	RXD+(A+)	TXD+(A+)	TXD-(B-)



9. Product measurement direction



10. Debugging software

You can download the accelerometer debugging assistant on the official website of MXMW Company (www.mxmwkj.com) to set the analog range. If you want to directly access the accelerometer, you can access it through the communication protocol of the accelerometer and the Volkswagen version of the serial port debugging assistant, so that the sensor can be easily integrated into your system

Accelerometer Debugging Assistant

Device model: Select the corresponding product model
Serial port: Select the COM port corresponding to the device;
Device address: fill in the current address code of the sensor, the factory default is 01
Baud rate: Select the current baud rate of the sensor, the factory default is 9600
Status monitoring: Connect to the serial port and click Start to collect data.
Status setting: Set the sensor function parameters

11. Communication protocol

1) Data frame format: (RTU model 8 data bits, 1 stop bit, no parity, default rate 9600).

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03 read 0x06 write	XX	XX	XX	XX	XX	XX

Data format:	hexadecimal
Address code:	Factory default 0x01 (users can set it according to their needs, the least exceeding 0xFF)
Function code:	0x03 read holding register 0x06 preset single register
Register address:	The starting address of the register that needs to be read and written
Registers number:	The number of registers that need to be read and written
CRC check:	Address code, function code, register starting address, register number, CRC

	check, calculated by the computer through a special CRC16 check tool (**note:** when the address code, function code or register start address changes, The CRC check will change. Please change the CRC check accordingly when your command changes.)
Note:	When using the Modbus serial port assistant software, it is not necessary to add CRC check when accessing the sensor communication; the ordinary serial port debugging assistant needs.

Note, please read the following items carefully before use:

1) **Since the MODBUS protocol stipulates that the time between two data frames should be at least greater than 3.5 bytes (for example, at 9600 baud rate, the time is 3.5×(1/9600)×11=0.004s). But in order to leave enough margin, this sensor increases this time to more than 10ms, so please leave at least 10ms time interval between each data frame.**
2) **The master sends a command - 10ms idle - the slave replies the command - 10ms idle - the master sends a command...**

If the user needs to implement CRC16 MODBUS calculation by himself, the C language program is implemented as follows for reference:

```
unsigned short ModBusCRC (unsigned char *ptr, unsigned char size)
{
    unsigned short a,b,tmp,CRC16,V;
    CRC16=0xffff;//CRC Register initial value
    for (a=0;a<size;a++) //N bytes
{
    CRC16=*ptr^CRC16;
    for (b=0;b<8;b++) //8-bit data
{
    tmp=CRC16 & 0x0001;
    CRC16 =CRC16 >>1; //shift one bit to the right
    if(tmp)
    CRC16=CRC16 ^ 0xa001; //XOR polynomial
}
    *ptr++;
}
    V = ((CRC16 & 0x00FF) << 8) | ((CRC16 & 0xFF00) >> 8) ;//High and low byte conversion
    return V;
}
```

For example: the check code of "01 06 00 0B 00 02" is "79 C9"

2) Command format

2.1 Read X-axis (acceleration)

Send command: 01 03 00 01 00 02 95 CB

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)

0x01	0x03	0x00	0x01	0x00	0x02	0X95	0xCB

Answer command:

Address code	Function code	Bytes Count	Register number		CRC check	
Address (1byte)	Function (1byte)	Bytes Count (1 byte)	DataH (2byte)	DataL (2byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x04	XXXX	XXXX	XX	XX

Note: For example, response reply frame: 01 03 04 BD B0 20 C5 06 2B, the X axis is the 1-4 bytes of the register data (the register stores 32-bit floating point numbers, standard IEEE754 standard), and the 1-2 bytes are the high bits of the data (high byte), 3-4 bytes are the data status (low byte), the high byte comes first, and the low byte follows;

MODBUS RTU standard protocol, according to the IEEE754 standard, the angle representation method is as follows:

X-axis acceleration (0XBDB020C5) = -0.086000g

2.2 Read Y-axis (acceleration)

Send command: 01 03 00 03 00 02 34 0B

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x02	0x00	0x02	0x34	0x0B

Answer command:

Address code	Function code	Bytes Count	Register number		CRC check	
Address (1byte)	Function (1byte)	Bytes Count (1 byte)	DataH (2byte)	DataL (2byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x04	XXXX	XXXX	XX	XX

Note: For example, response reply frame: 01 03 04 3F BD 91 68 0A 7D, Y axis is 1-4 bytes of register data (register stores 32-bit floating point number, standard IEEE754 standard), among which 1-2 bytes are high data bits (high byte), 3-4 bytes are the data status (low byte), the high byte comes first, and the low byte follows;

MODBUS RTU standard protocol, according to the IEEE754 standard, the angle representation method is as follows:

Y-axis angle (0x3FBD9168) = 1.48100g

2.3 Read Z-axis (acceleration)

Send command: 01 03 00 05 00 02 D4 0A

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x05	0x00	0x02	0xD4	0x0A

Answer command:

Address code	Function code	Bytes Count	Register number		CRC check	
Address (1byte)	Function (1byte)	Bytes Count (1 byte)	DataH (2byte)	DataL (2byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x04	XXXX	XXXX	XX	XX

Note: For example, response reply frame: 01 03 04 41 6C 2D 0E B2 86, Y axis is 1-4 bytes of register data (register stores 32-bit floating point number, standard IEEE754 standard), among which 1-2 bytes are high data bits (high byte), 3-4 bytes are the data status (low byte), the high byte comes first, and the low byte follows;

MODBUS RTU standard protocol, according to the IEEE754 standard, the angle representation method is as follows:

Z-axis angle (0x416C2D0E) = 14.76100g

2.4 Read X, Y, Z axis acceleration

Send command: 01 03 00 01 00 06 94 08

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x01	0x00	0x06	0x94	0x08

Answer command:

Address code	Function code	Bytes Count	Register number			CRC check	
Address (1byte)	Function (1byte)	Bytes Count (1 byte)	X angle (4byte)	Y angle (4byte)	Z angle (4byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x0C	XXXX XXXX	XXXX XXXX	XXXX XXXX	XX	XX

Note: For example, the response reply frame: 01 03 0C BD B0 20 C5 3F BD 91 68 41 6C 2D 0E 1F C2, the register stores 32-bit floating point numbers, standard IEEEE754 standard, the X-axis acceleration is 1-4 bytes of the register data, Y The axis acceleration is the 5-8 bytes of the register data, and the Z-axis acceleration is the 9-12 bytes of the register data; the high byte of the data is in front and the low byte is in the back;

MODBUS RTU standard protocol, according to the IEEEE754 standard, the data representation method is as follows:

X-axis acceleration (0x BDB020C5) = -0.086000g

Y-axis acceleration (0x3FBD9168) = 1.481000g

Z-axis acceleration (0x416C2D0E) = 14.76100g

2.5 Accelerometer calibration

Send command: 01 06 00 0A 00 00 A9 C8

Address code	Function code	Register address	Register number	CRC check

Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0A	0x00	0x00	0xA9	0xC8

Answer command:

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0A	0x00	0x00	0XA9	0xC8

Note: Accelerometer calibration is used to remove the zero bias of accelerometers X, Y, and Z. The calibration method for accelerometer is as follows: keep the accelerometer module horizontally stationary and perform calibration.

2.6 Set communication rate

Send command: 01 06 00 0C 00 04 48 0A

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0C	0x00	0x04	0x48	0x0A

Answer command:

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0C	0x00	0x04	0x48	0x0A

Note: Register data field 0x0000 means 2400, 0x0001 means 4800, 0x0002 means 9600, 0x0003 means 19200, 0x0004 means 115200, 0x0005 means 14400, 0x0006 means 38400, 0x0007 means 576 00, **the default value is 0X02:9600.** After each successful communication baud rate change, send a save command, a response command will be sent back at the original baud rate, power on again, and then immediately change the device communication baud rate.

More note: If high frequency output is required, please set the baud rate to 115200.

2.7 Set module address

Send command: 01 06 00 0D 00 02 99 C8

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0D	0x00	0x02	0x99	0xC8

Note: The default address of the sensor is 01.

Answer command:

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x02	0x06	0x00	0x0D	0x00	0x02	0x99	0xFB

1) If multiple sensors are connected to a group of buses at the same time, such as MODBUS, each sensor needs to be set to a different address to achieve separate control and response speed.
2) If the new address is successfully changed, the address codes in all subsequent command and response packets must be replaced by the changed new address code to take effect, otherwise the sensor will not respond to the command.
3) XX module address ranges from 00 to FE.

2.8 Query module address

Send command: FF 03 00 0D 00 01 00 17

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0xFF	0x03	0x00	0x0D	0x00	0x01	0x00	0x17

Note: As the MODBUS protocol stipulates that the product must know the module address to communicate, so when the MODBUS protocol communicates, the address is known in advance, so the address cannot be queried. The query address of this product is to use the custom protocol to query the address FF of the MODBUS protocol, or it can be queried according to the known address code.

Answer command:

Address code	Function code	Bytes Count	Register number		CRC check	
Address (1byte)	Function (1byte)	Bytes Count (1 byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x02	XX	XX	XX	XX

2.09 Factory default

Send command: 01 06 00 0E 00 00 E8 09

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0E	0x00	0x00	0xE8	0x09

Answer command:

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0E	0x00	0x00	0xE8	0x09

Note: Restore factory settings to take effect after power on again

2.10 Update flash (save settings)

Send command: 01 06 00 0F 00 00 B9 C9

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0F	0x00	0x00	0xB9	0xC9

Answer command:

Address code	Function code	Register address		Register number		CRC check	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0F	0x00	0x00	0xB9	0xC9

*For various parameter settings, if the save setting command is not sent after the settings are completed, these settings will disappear after power failure.

Appendix. IEEE754 conversion

1). Convert single-precision floating-point numbers to standard 4-byte numbers

```
//float converted to IEEE754 4 bytes big_endian
//If the compiler adopts little endian mode, please reverse the bdat array first
void float2byte(float fdat, unsigned char * bdat)
{
unsigned char i;
//Get the 4 byte address of the float data
unsigned char *tmp=(unsigned char *)&fdat;
//Indirect addressing, obtain the value in the 4-byte address where float is located
for(i=0;i<(sizeof(float));i++)
*(bdat+i)=*(tmp+i);
}
```

2) .4 bytes converted to standard single precision floating point function

```
//IEEE754 4 bytes converted to float big_endian
//If the compiler adopts little endian mode, please reverse the bdat array first
float byte2float(unsigned char *bdat)
{
return *((float *)bdat);
}
```