Intelligent Metering MCCB-MTMTM5EL RS485 Communication Guide

Document Version

date of issue

Shanghai Matis Electric Co., Ltd.

Catalogue

catalogue........ 2
1 Communication Instructions........ 3
1.1 summary........ 3
1.2 MODBUS-RTU communication protocol........ 3
1.3 MODBUS-RTU transmission mode........ 4
1.4 CRC Check Code Generation Process........ 5
1.5 Example of communication message........ 7
1.6 instrument data backhaul analysis........ 7
2 address table........ 9
3 Test Method of Serial Debug Assistant........ 17
4 Usage Notes........ 23
5 Common Communication Issues and Solutions........ 24
5.1 The instrument did not send back data........ 24
5.2 Instrument sends back data but is inaccurate........ 25
5.3 Instrument data is intermittent and communication is unstable........ 26
6 Version event........ 27

1 Communication INSTRUCTIONS

1.1 Summary

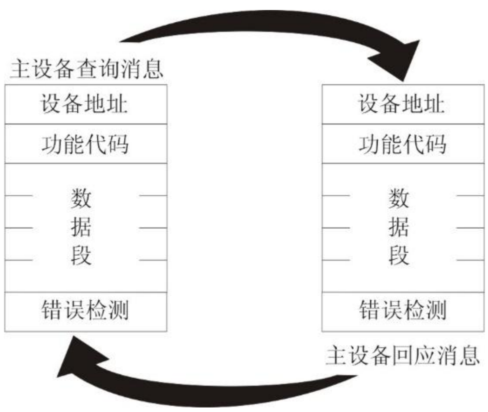
In networking communications, the wiring should be kept away from strong electric cables or other strong electromagnetic environments. It is recommended to use a bus-type (T-type) network topology for wiring, and star or other connection methods are not advised.

Required and optional hardware devices and models

1.2 MODBUS-RTU Communication Protocol

The MODBUS protocol employs a master-slave response communication method over a single communication line. The master computer first addresses a unique terminal device (slave) with a signal. The slave then transmits a response signal in the opposite direction to the master. This enables all communication data streams to travel in opposite directions along the same communication line, operating in a half-duplex mode.

The MODBUS protocol permits communication exclusively between host devices (e.g., PCs or PLCs) and terminal devices, but not between standalone terminals. This design prevents terminal devices from occupying communication channels during initialization, restricting their function to responding to incoming query signals.



Query-Response Cycle Table

Host Query: The query message frame contains four key components: device address code, function code, data information code, and checksum. The address code specifies the target slave device's function code, instructing the selected slave to perform specific operations. For example, function codes 03 or 04 require the slave to read and return register contents. The data segment contains additional information required for the function execution, such as the number of registers to start reading from. The checksum verifies the frame's integrity using CRC16, providing a reliable method for the slave to validate the message content.

Slave response: When a slave device generates a valid response, the reply message contains the slave address code, function code, data information code, and CRC16 checksum. The data information code includes the data collected by the slave device, such as register values or status. If an error occurs, the slave device is not expected to respond.

1.3 MODBUS-RTU Transmission Mode

The transmission mode refers to a series of independent data structures within a data frame and the finite rules for data transmission. Below is the definition of the transmission mode compatible with the MODBUS protocol-RTU method. Each byte consists of: 1 start bit, 8 data bits, (parity bit), 1 stop bit (when parity bit is present) or 2 stop bits (when parity bit is absent).

The structure of data frame is the format of message.

Address code	FC	Data field	Check code
1 byte	1 byte	N bytes	2 byte

Address Code: Located at the frame's beginning, this 8-bit binary code (decimal values 0-255) uses only 1-247 in our system, reserving other addresses. These bits identify the user-specified terminal device that will receive data from its connected host. Each terminal must have a unique address, and only the addressed device responds to queries containing this address. When a response is sent, the slave address data in the response informs the host which terminal is communicating with it.

Function code: It tells the addressed terminal what function to perform. The following table lists the function codes supported by multifunctional power products, along with their meanings and functions.

Data code: Contains either the data required for a terminal to perform specific functions or the data collected during terminal response queries. The data may include numerical values, reference addresses, or configuration parameters. For example, the function field code instructs the terminal to access a register, while the data field specifies the starting register and the number of data points to read. The slave data code returns both the data length and the corresponding data.

Checksum: The error check (CRC) field occupies two bytes and contains a 16-bit binary value. The CRC value is calculated by the transmitting device and appended to the data frame. The receiving device recalculates the CRC value upon data reception and compares it with the value in the received CRC field. If the two values do not match, an error has occurred.

1.4 CRC Check Code Generation Process

The process for generating a CRC is:

1）Set a 16-bit register to FFFFH (16-ary, all ones), known as the CRC register.

2）The result of XOR operation between the first byte's 8 bits and the low byte of CRC register is stored back to CRC register.

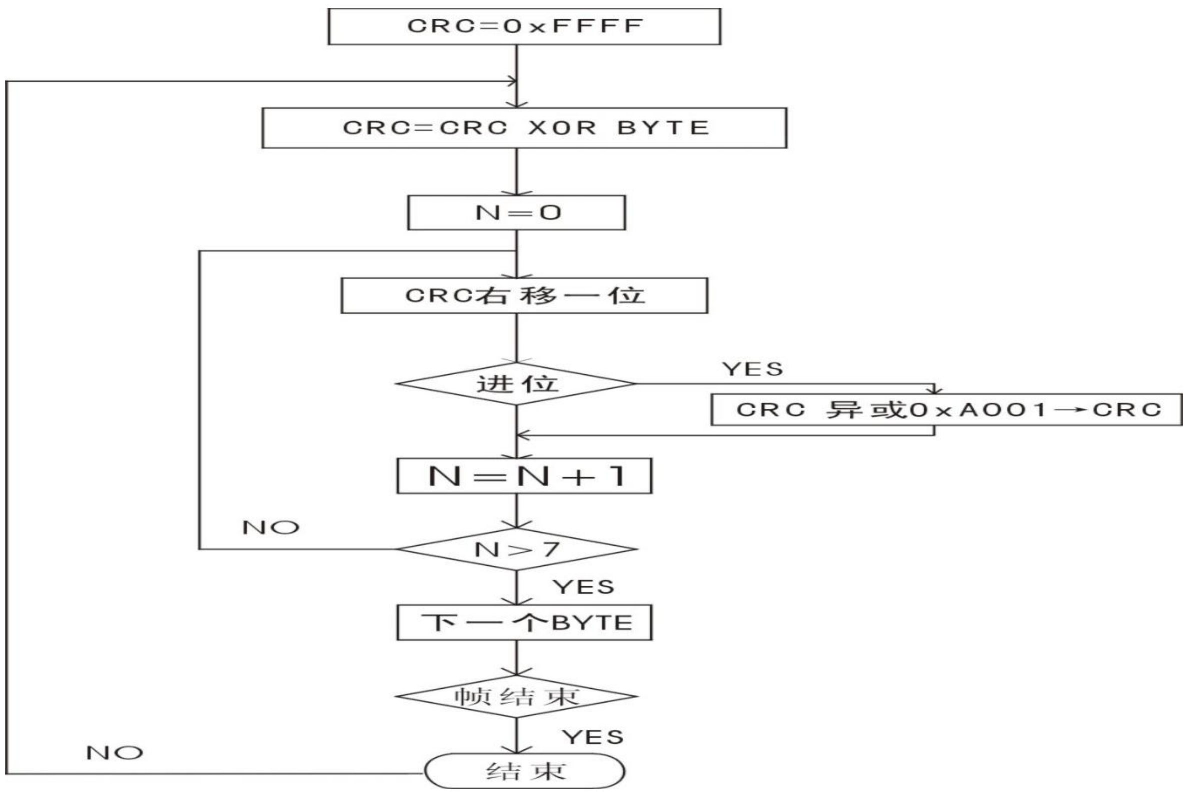
3）Right-shift the CRC register by one bit, set the most significant bit to 0, and clear the least significant bit for detection.

4）If the value removed in the previous step is 0, repeat step 3 (next shift): set to 1, and XOR the CRC register with a preset fixed value (0A001H).

5）Repeat steps 3 and 4 until 8 shifts are completed, processing one full 8-bit.

6）Repeat steps 2 through 5 to process the next eight bits, until all bytes are processed.

7）The value of the CRC register is the CRC value.



Flow chart of CRC16 checksum calculation program

```
unsigned int CRC16(unsigned char *crc,unsigned char n,unsigned char after){
     unsigned char crc_lo=0xff,crc_hi=0xff,savelo,savehi;
     unsigned char crc_reg,crc_reg1;
     unsigned int nCRC;
     for(crc_reg=0;crc_reg<n;crc_reg++)
    {
        crc_lo^=*crc++;
        for(crc_reg1=0;crc_reg1<8;crc_reg1++)
        {
            savehi=crc_hi;
            savelo=crc_lo;
            crc_hi>>=1;
            crc_lo>>=1;
            if((savehi&0x01)==0x01)
                crc_lo|=0x80;
            if((savelo&0x01)==0x01)
            { crc_hi^=0xa0;
               crc_lo^=0x01;
            }
        }
    }
    if(after)
    {
        *crc++=crc_lo; // Low byte first
        *crc=crc_hi; //High byte is at the end
    }
    nCRC=crc_hi;
    nCRC<<=8;
    nCRC|=crc_lo;
    return nCRC;
}
```

1.5 Example of Communication Message

Read data: Function code 03H or 04H

This feature allows users to access data collected and recorded by the terminal device, as well as system parameters. The host can request an unlimited number of data points at once, but the request must not exceed the defined address range.

Set the instrument communication address to 1.

The master transmits the query data frame as follows (hexadecimal representation):

Instrument address	Order	Start address high bit	Start Address Low Bit	High digit of data count	Least significant bit of data count	CRC16 low bit	CRC16 high
01	03	00	00	00	02	C4	0B

If the communication is normal, the slave machine returns the data frame as follows (hexadecimal representation):

Instrument address	Order	DL	Data	CRC16 low bit	CRC16 high
01	03	04	43 55 66 80	D5	A7

The instrument value displayed is 43556680H. According to the IEEE-574 data format definition and calculation method, the correct value is 213.4.

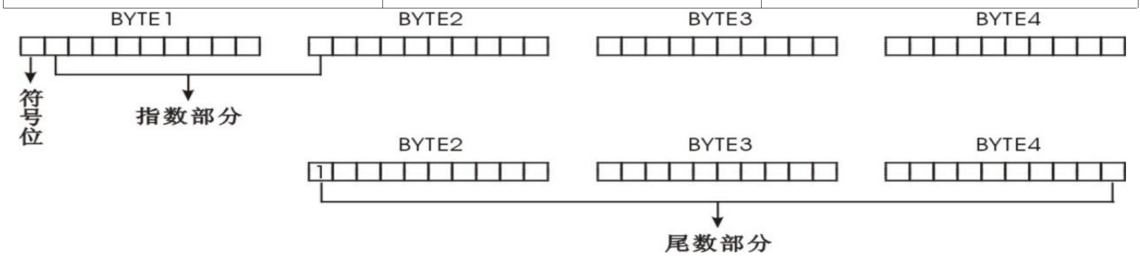
1.6 Instrument Data Backhaul Analysis

The floating point format is used to represent the power parameters.

(1)

Definition: A floating-point data type (float) conforms to the IEEE-754 data format, with the following definition and calculation method:

1 Sign bit	8 E7... E0	23 Positional tail number (p22... p0)



32 IEEE-574 Data Format Definition of Floating Point

Sign bit: SIGN=0 indicates positive, SIGN=1 indicates negative

Exponential part: E=Exponential part-126

Decimal part: M=fill the decimal part with the highest digit 1

Data result = (SIGN) × 2E × M / (256 × 65536)

(2)

Example: After the host sends the query data frame, the ammeter returns the correct data frame, which is 43556680 (H).

Convert to binary data bits: 01000011,01010101,01100110,10000000. According to the IEEE-574 standard, the result is:

The most significant bit is the sign bit: SIGN=0 indicates a positive number.

Exponential part: (10000110)B = 134; Exponential part E= 134-126 =8

The final digits are M = (11010101 01100110 10000000), B = D5 66 80, and H = 13985408.

Results: 28 × 13985408 / (256 × 65536) =213.4

2 Address Book

MODBUS address information table (addresses are in hexadecimal)

<table>
<tr><th>Address</th><th>Command type</th><th>Description</th><th colspan="3">Byte type (empty indicates: one character)</th><th>Explain</th></tr>
<tr><td colspan="7">Output winding</td></tr>
<tr><td>0</td><td>Output winding</td><td>Remote reset</td><td colspan="3"></td><td>Automatically set to 0 upon success</td></tr>
<tr><td>1</td><td>Output winding</td><td>Remote closing</td><td colspan="3"></td><td>Automatically set to 0 upon success</td></tr>
<tr><td>2</td><td>Output winding</td><td>Remote disconnector</td><td colspan="3"></td><td>Automatically set to 0 upon success</td></tr>
<tr><td colspan="7">Holding register</td></tr>
<tr><td>0</td><td>Holding register</td><td>Device address</td><td colspan="3"></td><td>1–255</td></tr>
<tr><td>1</td><td>Holding register</td><td>Baud rate</td><td colspan="3">1200–57600</td><td>Default is 38400</td></tr>
<tr><td>2</td><td>Holding register</td><td>Year (0-99)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>3</td><td>Holding register</td><td>Month (1-12)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>4</td><td>Holding register</td><td>Day (1-31)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>5</td><td>Holding register</td><td>Time (0-23)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>6</td><td>Holding register</td><td>Points (0-59)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>7</td><td>Holding register</td><td>Seconds (0-59)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>8</td><td>Holding register</td><td>Continue to have</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>9</td><td>Holding register</td><td>Overvoltage value (250-300)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>10</td><td>Holding register</td><td>Undervoltage value (150-200)</td><td colspan="3"></td><td>Read-write</td></tr>
<tr><td>0B</td><td>Holding register</td><td>Leakage current rating (see Table 1)</td><td colspan="3">High 8 bits: leakage type</td><td>Read-write</td></tr>
<tr><td></td><td></td><td></td><td colspan="3">Low 8 bits: leakage current rating</td><td>Read-write</td></tr>
<tr><td rowspan="4">0C</td><td rowspan="4">Holding register</td><td rowspan="4">Set value 1</td><td>Price</td><td>250A</td><td>400A</td><td rowspan="4">Read-write</td></tr>
<tr><td>0</td><td>100</td><td>200</td></tr>
<tr><td>1</td><td>125</td><td>225</td></tr>
<tr><td>2</td><td>140</td><td>250</td></tr>
</table>

			3	160	315	

			4	180	350	
			5	200	400	
			6	225	----	
			7	250	----	
0D	Holding register	Ir 1 Delay time	Price	Delay time		Read-write
			0	3		
			1	4		
			2	6		
			3	8		
			4	10		
			5	12		
			6	16		
			7	18		
			8	off		
0E	Holding register	Ir2 setpoint	Price	Factor (Ir2=n*Ir1)		
			0	2		
			1	2.5		
			2	3		
			3	4		
			4	5		
			5	6		
			6	7		
			7	8		
			8	10		
			9	12		
0F	Holding register	Ir2 delay time	Price	Delay time		
			0	0.1		
			1	0.1		
			2	0.2		
			3	0.3		
			4	0.4		
			5	0.6		
			6	0.8		
			7	1.0		

			8	off	
10	Holding register	Ir3 setpoint	Price	Multiple (Ir3=n * Ir1)	
			0	4	
			1	6	
			2	7	
			3	8	
			4	10	
			5	11	
			6	12	
			7	13	
			8	14	
11	Holding register	Test closing of power supply	0: Disabled, 1: Enabled		
12	Holding register	Standard table calibration value	Read the standard table		Read-write
13	Holding register	Calibration type	11:UA 12:UB 13:UC 14:IA 15:IB 16:IC 17: Leakage current 18:UABC 19:IABC		Read-write
14	Holding register	Calibration level	0-19 writable		Read-write
Incoming register					
0	Incoming register	Device Serial Number 0			Read only
1	Incoming register	Device Serial Number 1			Read only
2	Incoming register	Device Serial Number 2			Read only
3	Incoming register	Device Serial Number 3			Read only
4	Incoming register	Device Serial Number 4			Read only
5	Incoming register	Fault type (see Table 2)	Abandoned		Read only
6	Incoming register	Fault time-year			Read only
7	Incoming register	Fault time-month			Read only
8	Incoming register	Fault time-day			Read only
9	Incoming register	Fault time			Read only
A	Incoming register	Fault time division			Read only

B	Incoming register	Fault time-seconds		Read only
C	Incoming register	Device type	100/250/400/630/800	Read only
D	Incoming register	Device status (see Table 3)		Read only

E	Incoming register	Hitch ID	(Add 1 for new faults)	Read only
F	Incoming register	Continue to have		Read only
10	Incoming register	Continue to have		Read only
11	Incoming register	Continue to have		Read only
12	Incoming register	Continue to have		Read only
13	Incoming register	Continue to have		Read only
14	Incoming register	Continue to have		Read only
15	Incoming register	Continue to have		Read only
16	Incoming register	Continue to have		Read only
17	Incoming register	Continue to have	Single precision floating point type	Read only
19	Incoming register	Current C	Single precision floating point type	Read only
1B	Incoming register	Current B	Single precision floating point type	Read only
1D	Incoming register	Current A	Single precision floating point type	Read only
1F	Incoming register	Voltage A	Single precision floating point type	Read only
21	Incoming register	Voltage B	Single precision floating point type	Read only
23	Incoming register	Voltage C	Single precision floating point type	Read only
25	Incoming register	Leakage current	Single precision floating point type	Read only
27	Incoming register	Current N	Single precision floating point type	Read only

Note: All labeled types are 16-bit integers.

Table 1:

Leakage current rating								
High byte	Description	Explain	Lower byte	Description	Explain			
Tracking type			Protection level					
Price			Price					
0	Do not track			Auto 1	Auto 2	Auto 3	Auto 4	Do not track
1	Auto 1		0	50	100	200	50	50
2	Auto 2		1	100	200	300	100	100

3	Auto 3		2	200	300	400	200	200
4	Auto 4		3	300	400	600	300	300
5	Disable protection		4	400	500	800	400	400

			5	----	----	-----	500	500
			6	----	----	-----	600	600
			7	----	----	-----	800	800

Table 3 (Device Status):

Device status		
Price	Meaning	Description
0	Unknown	
1	Closed-circuit operat-ion	
2	Standby split	
3	Closing	
4	Failed to close the circuit	
5	Re-closing	
6	Unsuccessful reclosing	
7	Splitting	
8	Failed to open the circuit breaker	
9	Circuit breaker interlock	
10	In closing pro-cess	
11	In the process of tripping	
12	Failed to close the circ-uit breaker	
13	Failed to open the circuit breaker	

Table A (Fault Type):

Fault type		
Price	Meaning	Description
0	Trouble-free	
1	Sudden ch-ange	
2	Residual current	
3	Get an ele-ctric shock	Obligate
4	Missing zero	
5	Overload	
6	Short	
7	Phase loss	

8	Undervoltage	
9	Overvoltage	
10	Landing	
11	Interrupt	
12	Fixed time testing	
13	Long-range	
14	Push-button	
15	Earth leakage searchlockout	
16	Fault of current transformer	Obligate
17	Failed to close the circuit	
18	Hand movement	
19	Phase error	Obligate
20	Under load	Obligate
21	Twinkling	Obligate

2018–01

The new register table is as follows:

Register address table					
Address	Description	Unit	Give tacit consent to	03 a slight pause in reading	06 write
01F4	485 check bit		1		0: None 1: EVEN 2:ODD
01F5	485 stop bit		0		0:1 digit 1:2 positions 2:1.5 bits
01F6	Modbus communication address		1		1-255 is configurable
01F7	Modbus communication rate		4800		2400-38400 is configurable
03F5	A mutual power	W			–
03E6	B-Phase Mutual Power	W			–
03E7	C-phase power	W			–
03E8	A-phase real-time voltage	0.1V	–		–
03E9	B-phase real-time voltage	0.1V	–		–
03EA	C phase real-time voltage	0.1V	–		–

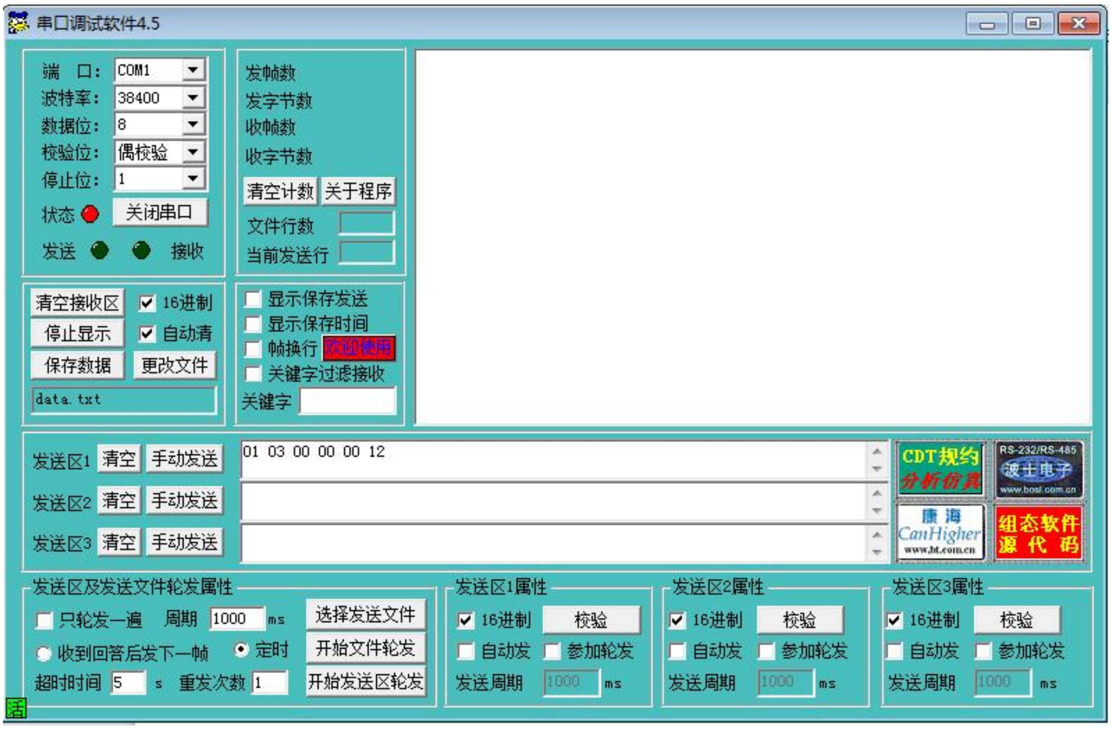
03EB	Real time leakage current	mA	–		–
03EC	A-phase real-time current H	0.1A	–		–
03ED	A-phase real-time current L	0.1A	–		–
03EE	B-phase real-time current H	0.1A	–		–
03EF	B-phase real-time current L	0.1A	–		–
03F0	C-phase real-time current H	0.1A	–		–
03F1	C-phase real-time current L	0.1A	–		–
03F2	Device status (see Table 3)	–	–		–
03F3	Shell current	A			111 Separating brake 222 Switch on 333 Reset
03F4	Set current IR 1	A			Write according to the settings list
03F5	Overload Delay IR 1_T	S			Write according to the settings list
03F6	Short-circuit current IR2	A			Write according to the settings list
03F7	Short Circuit Delay IR2_T	0.1S			Write according to the settings list
03F8	Instantaneous current IR3	A			Write according to the settings list
03F9	Overvoltage action value	0.1V			250.0-350.0 is adjustable
03FA	Undervoltage action value	0.1V			100.0-200.0 is adjustable
03FB	Phase lag action value	0.1V			10.0-100.0 is adjustable
03FC	Leakage current operating value	mA			Write according to the settings list
03FD	Power on function	–	OFF		0-1 Settable
03FE	Power-off alert function	–	OFF		0-1 Settable
03FF	Modbus communication address		1		1-255 is configurable
0400	Modbus communication rate		4800		2400-38400 is configurable
0401	Device time-year				0-99 is configurable
0402	Device Time-Month				1-12 can be set
0403	Device time-Day				1-31 is configurable
0404	Device time-hours				0-23 is adjustable
0405	Device time-minutes				0-59 is adjustable

0406	Device time-seconds				0-59 is adjustable
Fault log (read-only)					
0410	Fault type (Table A)	–	–	–	–
0411	Fault phase	–	–	–	–
0412	Voltage A during fault	0.1v	–	–	–

0413	Fault voltage B	0.1v	–	–	–
0414	Voltage C during fault	0.1v	–	–	–
0415	Fault current A	A	–	–	–
0416	Fault current B	A	–	–	–
0417	Current C at fault	A	–	–	–
0418	Fault leakage current	mA	–	–	–
0419	Failure time-years	–	–	–	–
041A	Failure time-month	–	–	–	–
041B	Failure time-day	–	–	–	–
041C	Fault time (hours)	–	–	–	–
041D	Fault time-minutes	–	–	–	–
041E	Fault time-seconds	–	–	–	–
041F	Record index		Fault When the matter is self Dynamic Update		Write 0 query previous Write 1 query next
Calibration current settings (read-only)					
07D0–07D9	0-9 range setting value	A	–	–	–
Delay time details (read-only)					
07DA–07E3	0-9 calibration time setting value	S	–	–	–
Short-circuit current details (read-only)					
07E4–07ED	0-9 short-circuit current settings	A	–	–	–
Short delay time details (read-only)					
07EE–07F7	0-9 short-circuit delay setting value	0.1S	–	–	–
Instantaneous Current Details (Read-Only)					
07F8–0801	0-9 instantaneous current setting values	A	–	–	–
Leakage current action level details (read-only)					
0802–080B	0-9 leakage current settings	mA	–	–	–

3 Serial @-@ Port Debugging Assistant Test Method

Open the serial port debugger



Set the communication parameters as shown in the figure:

The default device address is 1

The default is 38400-8-parity-1

For example, to retrieve data from addresses 1-5 in the hold register (see table), you can use the debug assistant to send the following data.

01 03 00 01 00 05

01: Device address

03: Keep the register

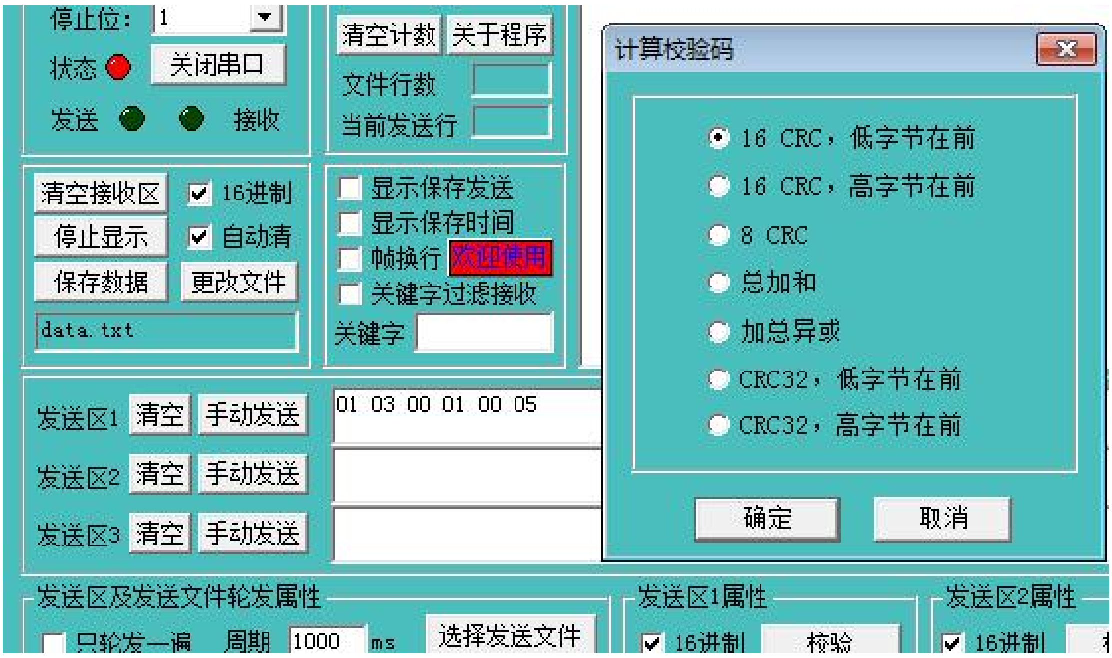
00 01: Read the first address of the address

00 05: Data length read

Generate verification code with Assistant

Click the verification button in the corresponding response area 

Open the dialog box below



Select 16CRC with low byte first, then confirm to generate a 16-bit CRC checksum automatically. This enables data transmission to the device.

The data format for production is as follows:

01 03 00 01 00 05 D4 09

01: Device address

03: Keep the register

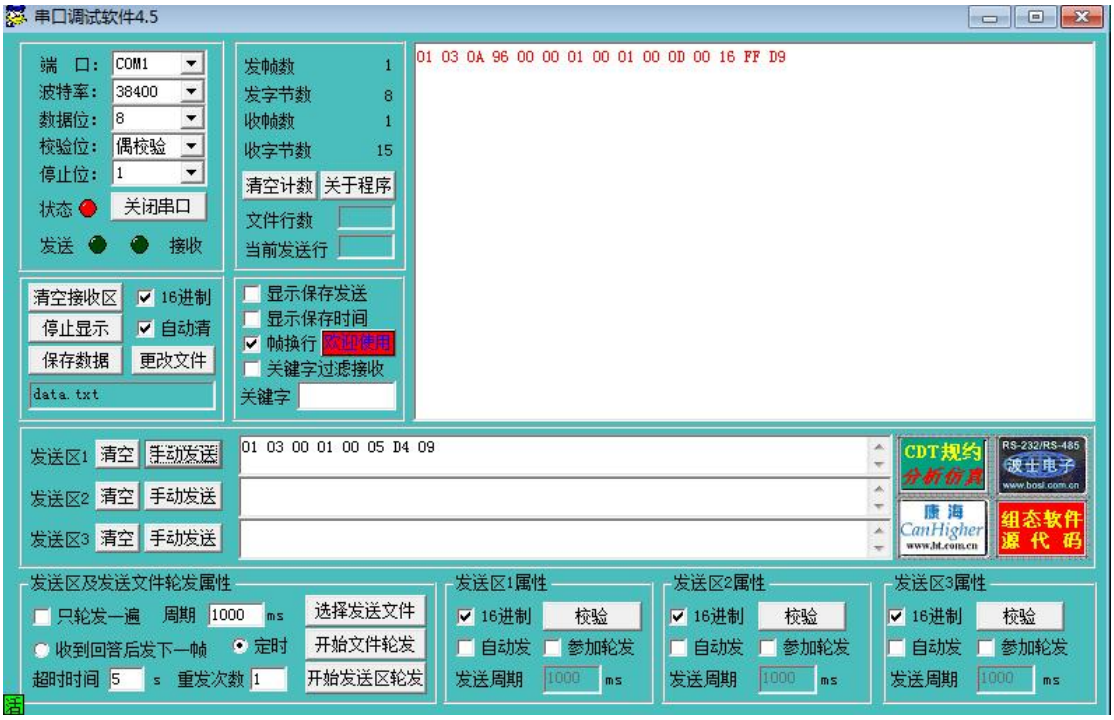
00 01: Read the first address of the address

00 05: Data length read

D4 09: Generate verification code with assistant

Then click the send button. If the connection is correct, the data will be returned.

The returned data is as follows:



The received data format is as follows:

01 03 0A 96 00 00 01 00 01 00 0D 00 16 FF D9

01: Device address

03: Keep the register

0A: Data length

96 00: The first address's content. The query data table shows this address contains the device's baud rate.

00 01: The first address contains the year of the device's time.

00 01: The first address contains the month of the device's time. 00 0D: The first address contains the day of the device's time.

00 16: The content of the first address is the time of the device FF D9:The check code is generated by the assistant

write hold register

For example, change the date to 13 years, 09 months, and 12 days using the Debug Assistant

The data format for production is as follows:

01 10 00 02 00 03 06 00 0D 00 09 00 0C BA 8C

01: Device address

10: Write multiple hold registers

00 02: Write the first address (year address)

00 03: Number of write registers (only 3 for date)

06: The data length is =number*2

00 0D: Year

00 09: Month

00 0C: Day

BA 8C: Generate verification code with assistant

The received data format is as follows:

01 10 00 02 00 03 21 C8

01: Device address

10: Write multiple hold registers

00 02: Write the first address (year address)

00 03: Number of write registers (only 3 for date)

21 C8:Verification code

Input register operations are similar to hold register operations. Note that the input register is read-only and write operations are invalid.

The data format for production is as follows:

01 04 00 01 00 05 61 C9

01: Device address

03: Input register

00 01: Read the first address of the address

00 05: Data length read

61 C9: Generate verification code with assistant

Then click the corresponding send button. If the connection is correct, the data will be returned in the following format:

01 04 0A FF 57 51 56 71 49 87 23 00 00 C2 19

01: Device address

04: Keep the register

0A: Data length

FF 57: The first address contains the device serial number 0, as shown in the query data table.

51 56: The first address contains the device serial number 1, as shown in the query data table.

71 49: The first address contains the device serial number 2, as shown in the query data table.

87 23: The first address contains the device serial number 3, as shown in the query data table.

00 00: The content of the first address Query the data table to find that the content of this address is the device serial number 4 C2 19:Check code

Operation of output coil

write-on-energized coil

The data format for production is as follows:

01 05 00 01 FF 00 DD FA

01: Device address

05: Write output coil

00 01: Coil address (closing coil address)

FF 00: Set

DD FA: Generate with Assistant

Then click the corresponding send button. If the connection is correct, the data will be returned in the following format:

01 05 00 01 FF 00 DD FA

01: Device address

05: Write output coil

00 01: Coil address (closing coil address)

FF 00: Set

DD FA:Check code

4 Use @-@ NOTE

When instruments communicate with the host, the host can be a standard PC, frequency converter, PLC, or similar devices. As instruments typically provide RS485 interfaces, external signal converters (e.g., RS232-to-RS485 converters) are required. We recommend using active RS 485 converters due to their higher power capacity and stronger load handling capability. Passive converters, which draw power from the device, have weaker load capacity and are prone to issues in long-distance multi-instrument network communications. The host computer must be equipped with corresponding host monitoring software.

Second: Bus Matching. One method involves adding a matching resistor, typically a 120Ω–200Ω terminal resistor, between ports A and B at the bus's end. This reduces signal reflections caused by impedance discontinuities and effectively suppresses noise interference. For multi-meter network communication, only the terminal meter requires a matching resistor, while others need not.

Thirdly, the number of network nodes depends on the RS-485 chip's driving capability and the receiver's input impedance. In practice, the baud rate should be adjusted according to factors like cable length, wire diameter, network layout, and transmission speed. While the RS-485 bus theoretically supports a communication range of up to 1200 meters, the actual transmission distance should not exceed 1 kilometer to ensure reliable communication quality.

Fourth: The RS-485 bus employs a parallel two-wire interface. A single faulty chip may cause the bus to become "locked out," necessitating isolation between its two-wire ports A and B and the bus. Typically, a 4~10 Ω PTC resistor is connected in series between each port and the bus, with a 5V TVS diode bridged to ground to eliminate line surge interference. If PTC resistors and TVS diodes are unavailable, standard resistors and Zener diodes can be used as substitutes.

Fifth, when setting up the instrument 485 communication network, to ensure the stability of the instrument communication, it is recommended to use 485 repeaters when the actual communication distance exceeds 500 meters, and 485 hubs when the number of slave devices connected to the bus exceeds 32, and the distance to the bus should be controlled within 5 meters.

Sixth: If real-time performance is not critical, a lower communication baud rate (e.g., 2400 bits/s or 4800 bits/s) is recommended. Higher baud rates may degrade communication quality, potentially causing data corruption or loss during backhaul transmission.

7. In multi-instrument network communication, it is recommended to connect all instrument communication terminals in series to ground (GND). This ensures all instruments share the same common ground and reference point, while also reducing interference and damage caused by common-mode voltage.

5 Communication Common Question Resolution

5.1 The Instrument Did Not Send Back Data

Answer: The following steps can be followed sequentially for troubleshooting

1）Check that the instrument has RS485 communication capability.

2）Check whether the A and B terminals of the RS485 bus match the terminal sequence numbers on the instruments and ensure no reverse connections. Verify the proper functioning of the RS485 converter module, and we strongly recommend using an active RS485 converter from a certified manufacturer. If multiple instruments on-site fail to transmit data, confirm the accuracy and reliability of the field communication bus connections. If only one or a few instruments exhibit communication issues, inspect the corresponding communication lines. You can test by modifying the addresses of the faulty and functioning slave instruments to troubleshoot or confirm the host computer software, or by testing the installation positions of the faulty and functioning instruments to identify or resolve instrument malfunctions.

3）Check the communication parameters of the instrument: communication address, baud rate, parity check mode, data transmission mode (word mode, byte mode) and the upper computer communication parameter settings are consistent.

4）Verify the correctness of commands transmitted by the host. The instrument employs the MODBUS-RTU communication protocol, and command frames must be issued in the specified format. Notably, the two-byte CRC checksum must be accurate; otherwise, the instrument will not transmit any data. We recommend first using standard serial debugging software to test the instrument's communication functionality and confirm proper communication. Here is an example.

Set the instrument address to 1 and the word transfer mode to WORD.

Host command (16-ary): 01 03 00 00 00 02 C4 0B

01: Instrument communication address 03: Function code (reads register value)

00 00-Register starting address

00 02-Number of registers C4 0B-CRC checksum value

If communication is normal, the 1# instrument responds and returns data: 01 03

04 ** ** ** ** CRC

01: Instrument communication address 03: Function code (reads register value)

04-Number of bytes returned

** ** ** **-4 bytes data, ** indicates 1 byte, and its value is related to the specific table.

If communication fails, the 1# instrument will not return any data, except for the 2-byte CRC checksum value returned by the CRC instrument.

5) Use a multimeter or similar device to check for short circuits or open circuits in the RS 485 bus.

6) For long-distance communication lines, shielded twisted pair cables with a minimum cross-sectional area of 0.5mm² should be used. When the distance exceeds 500 meters, repeaters are recommended to minimize signal attenuation. In networking setups, if instruments are equipped with RS485 GND ports, these should be connected in series to ensure uniform ground voltage across all devices.

7) Use genuine 9-pin or 25-pin physical serial ports. Some PCs or laptops without RS232 serial ports may experience communication failures when using USB-to-RS232 adapters, as the adapter port is not authentic.

8) Check whether the serial port of the host computer is normal. For example, try a different computer.

9) If none of the above steps resolve the issue, please contact our technical department.

5.2 Instrument Sends Back Data but Is Inaccurate

Answer: The following steps can be followed sequentially for troubleshooting

1) Please carefully read the communication section in the accompanying instrument manual, pay attention to the data storage addresses and formats specified in the address table, and ensure conversion according to the corresponding data format.

2) We recommend customers to request the MODBUS-RTU communication protocol testing software MODSCAN from distributors. This software adheres to the standard MODBUS-RTU protocol and supports data display in formats such as integer, floating-point, and hexadecimal, enabling direct comparison with instrument data.

3) Check the transmission method for the data settings in the instrument communication parameters. If the user sets it to Word (word transfer mode), the data is read using the word address specified in the manual. If set to byte (byte transfer mode), the data is read using the byte address specified in the manual.

4) Request the latest version of the manual and the corresponding RS485 communication backend test software from our company.

5) If none of the above steps resolve the issue, please contact our technical department.

5.3 Instrument Data Is Intermittent and Communication Is Unstable

Answer: The following steps can be followed sequentially for troubleshooting

1）Check the terminal resistance of 120～200 ohms at the end of the 485 bus.

2）Check if the RS485 converter is functioning properly. It is recommended to use an active RS485 converter. If possible, use an oscilloscope to verify the signal transmission on the line.

3）Check whether the background device driver is the latest version. Some configuration software may have different versions for serial port driver upgrades. It is recommended to use the latest version of the driver.

4）Use a lower baud rate, such as 4800 bits per second.

5）Change the interval between each frame of the command.

6）If none of the above steps resolve the issue, please contact our technical department.

6 Version CONTROL

Time	Edition	Event	Handlers

matismart AIoT Electric SHANGHAI MATIS ELECTRIC CO.,LTD.
上海麦豆电气有限公司

Website:www.matismart.com

Mail:info@matismart.com

Telephone fax: 0086 2168682728

Mobile: 0086 186 2187 9631

Address: No.83, West Third Road, Pudong, Shanghai