Protocol introduction

The communication protocol is MODBUS-RTU protocol, this product only supports function codes 0x03, 0x06, 0x10; the communication interface is TTL serial port;

The information transmission is asynchronous, and the Modbus-RTU mode takes 11-bit bytes as the unit

Word Format (Serial Data)	10 bit binary
Start bit	1 bit
Data bit	8 bit
Parity bit	None
Stop bit	1 bit

Data frame structure:

Data Frame Interval	Address Code	Function Code	Data Area	CRC
3.5 more than bytes	1 byte	1 byte	N byte	2 byte

Before sending data, the data bus static time is required, that is, the time of no data sending is greater than 3.5 (for example: the baud rate is 9600 5ms at the time)

The message transmission must start with a pause interval of at least 3.5 byte times, and the entire message frame must be sent as a continuous

If there is a pause of more than 3.5 byte times before the frame is complete, the receiving device will refresh the incomplete message and assumes the next byte is the address field of a new message. Likewise, if a new message is less than 3.5 character time following the previous message, the receiving device will consider it a continuation of the previous message.

1.2 Function code:

In the ModBus communication protocol, the function code represents the second byte transmitted within each communication information frame. It serves as a crucial identifier to determine the desired action to be executed by the slave device, when sent as a host request. Function codes within the range of 1 to 127 are defined in the ModBus protocol.

When acting as a slave response, the slave device returns the same function code received from the master. This response signifies that the slave has acknowledged the master's request and performed the relevant operations accordingly. It's important to note that this particular device only supports function codes 0x03, 0x06, and 0x10.

Function code	Definition	Operation (binary)
0x03	Read register	Read data from one or more registers

0x06	Write single register	Write a set of binary data to a single register
0x10	Write multiple registers	Write multiple sets of binary data to multiple registers

1.3 Data area

The data area within the ModBus communication protocol contains various types of information that the slave device needs to send back to the host or actions it needs to perform. This information can include data such as open/off values for input/output, analog input/output readings, registers, and more.

Additionally, the data area may include reference addresses and other relevant details. For instance, if the host uses function code 03 to instruct the slave, it can request the value of a specific register by providing the starting address and length of the register to be read.

The response from the slave will include both the length and content of the requested register's data.

0x03 read function host format

Address code	Function code	Register start address	Number of register addresses n(1 ~ 32)	CRC 2
1 byte	1 byte	2 bytes	2 bytes	2 bytes

0X03 Read function slave return format

Address code	Function code	Register start address	Return the number of registers n	Register data	CRC
1 byte	1 byte	2 bytes	1 byte	2*n bytes	2 bytes

0x06 Write Single Register Function Host Format

Address code	Function code	Register start address	Register data	CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

0X06 Write single register function slave return format

Address code	Function code	Register start address	Register data	CRC
1 byte	1 byte	2 bytes	2*n bytes	2 bytes

0x10 Write Multiple Registers Function Host Format

Address code	Function code	Register start address	Number of register addresses n(1~32)	Write bytes 2*n	Register data	CRC
1 byte	1 byte	2 bytes	2 bytes	1 byte	2*n bytes	2 bytes

0X10 Write Multiple Registers Slave Master Format

Address code	Function code	Register start address	Number of register addresses n(1 ~ 32)	CRC
1 byte	1 byte	2 bytes	1 byte	2 bytes

0x10 Write Multiple Registers Slave Master Format

Address code	Function code	Register start address	Number of register addresses n(1 ~ 32)	CRC
1 byte	1 byte	2 bytes	1 byte	2 bytes

Protocol register introduction (the data in a single register address is double-byte data)

Name	Description	Bytes	Dec. point	Unit	R/W	Register address
V-SET	Voltage Setting	2	2	V	R/W	0000H
I-SET	Current setting	2	3	A	R/W	0001H
VOUT	Output voltage display value	2	2	V	R	0002H
IOUT	Output current display value	2	3	A	R	0003H
POWER	Output power display value	2	2	W	R	0004H
UIN	Input voltage display value	2	2	V	R	0005H
AH-LOW	Output AH low 16 bits	2	0	mAh	R	0006H
AH-HIGH	Output AH high 16 bits	2	0	mAh	R	0007H
WH-LOW	Output WH low 16 bits	2	0	mWh	R	0008H
WH-HIGH	Output WH high 16 bits	2	0	mWh	R	0009H
OUT_H	Open time - hours	2	0	H	R	000AH
OUT_M	Start time - minutes	2	0	M	R	000BH
OUT_S	Open time - seconds	2	0	S	R	000CH
T_IN	internal temperature value	2	1	F/C	R	000DH
T_EX	External temperature value	2	1	F/C	R	000EH
LOCK	Key lock	2	0	-	R/W	000FH
PROTECT	Protection status	2	0	-	R/W	0010H
CVCC	Constant voltage and constant current state	2	0	-	R	0011H
ONOFF	Output On/Off	2	0	-	R/W	0012H
F-C	Temperature F/C	2	0	-	R/W	0013H
B-LED	Backlight brightness level	2	0	-	R/W	0014H
SLEEP	Off screen time	2	0	M	R/W	0015H
MODEL	Product number	2	0	-	R	0016H
VERSION	Firmware version	2	0	-	R	0017H
SLAVE-ADD	Slave address	2	0	-	R/W	0018H

BAUDRATE_L	Baud rate	2	0	-	R/W	0019H
T-IN-OFFSET	Internal temperature correction	2	1	F/C	R/W	001AH
T-EX-OFFSET	External temperature correction	2	1	F/C	R/W	001BH
BUZZER	Buzzer switch	2	0	-	R/W	001CH
EXTRACT-M	Quickly call out data sets	2	0	-	R/W	001DH
DEVICE	Device status	2	0	-	R/W	001EH
MASTER	Host type	2	0	0	R/W	0030H
WIFI-CONFIG	WIFI-CONFIG	2	0	0	R/W	0031H
WIFI-STATUS	WIFI State	2	0	0	R/W	0032H
IPV4-H	First 2 bytes of the IP-address	2	0	0	R/W	0033H
IPV4-L	Last 2 bytes of the IP-address	2	0	0	R/W	0034H
V-SET	Set voltage	2	2	V	R/W	0050H
I-SET	Set current	2	3	A	R/W	0051H
S-LVP	Low voltage protection value	2	2	V	R/W	0052H
S-OVP	Overvoltage protection value	2	2	V	R/W	0053H
S-OCP	Overcurrent protection value	2	3	A	R/W	0054H
S-OPP	Over power protection value	2	1	W	R/W	0055H
S-OHP_H	Maximum output time--hour	2	0	H	R/W	0056H
S-OHP_M	Maximum output duration—minutes	2	0	M	R/W	0057H
S-OAH_L	Maximum output AH low 16 bits	2	0	maH	R/W	0058H
S-OAH_H	Maximum output AH high 16 bits	2	0	maH	R/W	0059H
S-OWH_L	Maximum output WH low 16 bits	2	0	10m Wh	R/W	005AH
S-OWH_H	Maximum output WH high 16 bits	2	0	10m Wh	R/W	005BH
S-OTP	Over temperature protection value	2	1	F/C	R/W	005CH
S-INI	Power-on output switch	2	0	-	R/W	005DH

Note 1: This product incorporates 10 sets of storage data sets, labeled as M0 to M9. Each set consists of a total of 14 data points, with serial numbers ranging from 20 to 2D. By default, the M0 data set is

automatically loaded upon powering on the product. The M1 and M2 data sets can be quickly accessed through the product panel. The M3 to M9 data sets are considered common storage data groups. The starting address calculation for each data group follows this formula: 0020H + (data group number * 0010H). For example, the starting address of the M3 data group would be: 0050H + (3 * 0010H) = 0080H.

Note 2: The key lock function has two possible values for reading and writing: 0 and 1. A value of 0 indicates an unlocked state, while a value of 1 represents a locked state.

Note 3: The protection status can be read with values ranging from 0 to 3: 0: Normal operation 1: Overvoltage protection (OVP) 2: Overcurrent protection (OCP) 3: Overpower protection (OPP) 4: Low voltage protection (LVP) 5: Over-arc current protection (OAH) 6: Over-heat protection (OHP) 7: Over-temperature protection (OTP) 8: Output enable/disable protection (OEP) 9: Over-wattage protection (OWH) 10: Input current protection (ICP)

Note 4: The constant voltage and constant current states have reading values of 0 and 1. A value of 0 indicates the constant voltage (CV) state, while a value of 1 indicates the constant current (CC) state.

Note 5: The switch output function can be read and written with values of 0 and 1. A value of 0 represents a closed state, while a value of 1 represents an open state.

Note 6: The backlight brightness level ranges from 0 to 5, allowing a total of six levels. Level 0 is the darkest, while level 5 is the brightest.

Note 7: The quick recall data group function can be written with values ranging from 0 to 9. After writing a specific value, the corresponding data group will be automatically called out.

Note 8: WiFi-related register descriptions are not provided.

name	Detailed description	Reg. address
MASTER	Host type (0x3B3A:WIFI,Other to be determined)	0030H
WIFI-CONFIG	WIFI pairing status(0:invalid 1:Touch pair 2:AP pair)	0031H
WIFI-STATUS	WIFI state 0:invalid network 1:connect router 2: Successfully connected to the server 3:touch pair 4:AP pair	0032H
IPV4-H	IP Address The first two bytes of the address 0xC0A8	0033H
IPV4-L	IP Address The last two bytes of the address 0x0108	0034H

IPV4-H: 0xC0A8 IPV4-L: 0x0108
IPV4 = 192.168.1.8

1.4 Error check code (CRC check):

The check code, also known as the CRC (Cyclic Redundancy Code), can be used by the host or slave to verify the integrity of received information.

During the transmission of data, errors can occur due to electronic noise or other interferences. The CRC code acts as an error check mechanism to determine whether the transmitted information is accurate. If errors are detected, the erroneous data can be discarded, ensuring system security and efficiency.

In the MODBUS communication protocol, the CRC code consists of 16 bits, or 2 bytes, represented as a binary number. The sending device (host) calculates the CRC and appends it to the end of the information frame being sent.

Upon receiving the information, the receiving device (slave) recalculates the CRC based on the received data and compares it with the received CRC. If the calculated CRC does not match the received CRC, it indicates an error. It is important to note that the CRC check code is transmitted with the low bit first and the high bit following.

Calculation method of CRC code:

Begin with a 16-bit register set to the hexadecimal value FFFF (all bits set to 1). This register is referred to as the CRC register.

1. Perform an XOR operation between the first 8 bits of binary data (the first byte of the communication information frame) and the lower 8 bits of the CRC register. Store the result back into the CRC register.
2. Shift the content of the CRC register one bit to the right (towards the least significant bit). Fill the most significant bit with 0. Take note of the bit shifted out during the right shift.
3. If the shifted out bit is 0, repeat step 3 by shifting right one more bit. If the shifted out bit is 1, perform an XOR operation between the CRC register and the predefined formula A001 (1010 0000 0000 0001).
4. Repeat steps 3 and 4 until the right shift has been performed 8 times, processing the entire 8-bit data.
5. Repeat steps 2 to 5 for each subsequent byte of the communication information frame.
6. Once all bytes of the communication information frame have been processed, swap the high and low bytes of the 16-bit CRC register.
7. The content of the CRC register after the swap is the final CRC code.

3. Communication examples

Example 1: The host reads the output voltage and output current display values

The message format sent by the host:

Host sends	Bytes	Message sent	Remark
Slave address	1	01	Send to slave with address 01
Function code	1	03	Read register
Register start address	2	0002H	Register start address
Number of register addresses	2	0002H	2 bytes in total
CRC	2	65CBH	The CRC code is calculated by the host

For example, if the current display value is 5.00V and 1.5A, then the message format returned by the slave machine response:

Slave response	Bytes	Returned information	Remark
Slave address	1	01	From slave 01

Function code	1	03	Read register
Read bytes	1	04	1 byte in total
The content of the register at address 0002H	2	01F4H	Output voltage display value
The contents of the register at address 0003H	2	05DCH	Output current display value
CRC	2	B8F4H	The CRC code is calculated by the slave

Example 2: The host needs to set the voltage to 24.00V The message format sent by the host:

host sends	Bytes	Message sent	Remark
slave address	1	01H	From slave 01
function code	1	06H	Write a single register
Register address	2	0000H	Register address
The contents of the register at address 0000H	2	0960H	Set the output voltage value
CRC code	2	8FB2H	The CRC code is calculated by the host

The format of the message returned by the slave after receiving the response

Slave response	Bytes	returned information	Remark
Slave address	1	01H	Send to slave with address 01
Function code	1	06Hwrite a single register	Write a single register
register address	2	0000H	Register start address
The contents of the register at address 0000H	2	0960H	Set the output voltage value
CRC code	2	8FB2H	The CRC code is calculated by the slave

Example 3: The host needs to set the voltage to 24.00V and the current to 15.00A. The message format sent by the host:

Host sends	Bytes	Message sent	Remark
Slave address	1	01H	From slave 01
Function code	1	10H	Write register
Register start address	2	0000H	Register start address
Number of register addresses	2	0002H	2 bytes in total
Write bytes	1	04H	1 byte in total
The contents of the register at address 0000H	2	0960H	Set the output voltage value
The contents of the register at address 0001H	2	05DCH	Set the output current value
CRC code	2	F2E4H	The CRC code is calculated by the host

The format of the message returned by the slave after receiving the response

Slave response	Bytes	Returned information	Remark
Slave address	1	01H	Send to slave with address 01
Function code	1	10H	Write register
Register start address Number of register addresses	2	0000H	Register start address
Number of register addresses	2	0002H	2 bytes in total
CRC code	2	41C8H	The CRC code is calculated by the slave