(D)P Modular/C Modular Instrument Builds Communication and Transmission to Computer by standards of RS232 and RS485. The particular explanation is as follows:

Notes:（This product just supports funtions codes: 03, 06）　201900531 Version

Baud Rate：9600
Start Bit：1
Data Bit：8
Check Bit：NO
Stop Bit：1

1．MODBUS_RTU Frame Structure

Message sending starts at a pause interval of at least 3.5 character time; the entire message frame must be a continuous data transmission stream. If there is a pause time of more than 3.5 character time before the frame is completed, the receiving device will refresh the incomplete message and assume that the next byte is the address domain of a new message. Similarly, if a new message starts with the previous message in less than 3.5 characters, the receiving device will consider it as a continuation of the previous message.

An example of standard frame structure:

Start	Functional domain	Functional domain	Data domain	CRC check	End
T1-T2-T3-T4	8Bit	8Bit	n 个 8Bit	16Bit	T1-T2-T3-T4

Address domain: The host chooses the slave device by putting the message address into slave which will be connected. The address range of a single slave is from 1 to 15 (decimal).
Address 0 is consider as a broadcast address so that all slaves can recognize it.

Functional domain: The effective encoding range is 1~255 (decimal); when a message is sent from the host to the slave, the function code will tell the slave what to do. For example, read/write data content of a set of registers, etc.

Data domain: The data domain sent by the host to the slave contains the additional information which is necessary for the slave to complete the action of the functional domain, such as register address.

CRC Check: After CRC generation, low bytes are in front, high bytes are in the back.
Note: The response interval between frames is< 5 ms while the communication rate is >9600 bps.

Serial Number	**Name**	**Explanation**	**Range**	**Decimal point digit capacity**	**Read-write**	**Parameter communication address**
0	ON/OFF	Power output/stop setting	1，0	0	r/w	0001H
1	OP.S	Ptotect status(Note1）	0-0xFFFF	0	r	0002H
2	Specification and type	Specification and type	0-65535	0	r	0003H
3	Tail classificati	Tail classification	0-0xFFFF	0	r	0004H

	on					
4	Decimal point digit capacity	V_A_W digit capacity（Note2）	0-0xFFFF	0	r	0005H
5	U	Voltage display value	0-65535	2	r	0010H
6	I	Current display value	0-65535	3	r	0011H
7	P	Power display value	32 digit capacity	3	r	0012H(high 16 bit) ，0013H(low 16 bit)
9	SetU	Set voltage	0-65535	2	r/w	0030H
10	SetI	Set current	0-65535	3	r/w	0031H
12	OVP	Set over volate protect value	0-65535	2	r/w	0020H
13	OCP	Set over current protect value	0-65535	2	r/w	0021H
14	OPP	Set over power protect value	0-65535	2	r/w	0022H(high 16 bit) ，0023H(low 16 bit)
15	RS-Adder	Set communication address	1～250	0	r/w	9999H

The ret note: The red rerial number part is public.The blue rerial number part is programmable private.The black serial number part is optional.

Note1：The protection status bit is defined as follows:

```
    union   _ST
    {
            struct
                    {
                        uint8_t isOVP:1；
                        uint8_t isOCP:1；
                        uint8_t isOPP:1；
                        uint8_t isOTP:1；
                        uint8_t isSCP:1；
                    }OP;
                    uint8_t   Dat;
    }
```

OVP：Over voltage protection OCP：Over current protection OPP：Over power protection OTP： Over tempreture protection SCP： short-circuit protection

Note 2: Decimal point digit capacity information as follow:

case 0x0005://voltage current power decimal point digit capacity

```
Dat=ShowPN;//((2<<8)|(3<<4)|(3<<0));//0.00V 0.000A 0.000W
break;
```

For example when read:ox0233,mean that voltage 2 decimal,current 3 decimal,power 3 decimal.

MODBUS RTU

Type and Format of the Communication Data：

Information transmission is asynchronous and in bytes. The communication information transmitted between the master station and the slave station is in a 10-bit word format:

Word format（serial data）	10 bit binary
Start bit	1 bit
Data bit	8 bits
Parity bit	None
Stop bit	1 bit

- **Communication data（information frame）format**

Data fomat	address code	Function code	data erea	CRC check
Data length	1 byte	1 byte	N byte	16 bit CRC code（redundancy cycle code）

1.Transmission：

When the communication command is sent from the sending device (host) to the receiving device (slave), the slave that accords with the corresponding address code will receive the communication command and will read the information according to the function code and the related requirements.

If the CRC check is correct, the corresponding task will be executed, and then the execution result (data) will be sent back to the host. The returned information includes the address code, function code, executed data and CRC check code. No information will be returned if the CRC check fails.

1.1 Address Code：

The address code is the first byte (8 bits) of each communication information frame, ranging from 1 to 250. This byte indicates that the slave whose address is set by the user will receive the information sent by the host. Each slave must have a unique address code, and only the slave that accords the address code will be able to respond to the returned information.

During the returning process from the slave, the returning data starts with their own respective address codes. The address code sent by the host indicates the slave address which will be sent, and the address code returned by the slave indicates the slave address to be returned. The corresponding address code indicates where the information comes from.

1.2 Function Code：

The function code is the second byte of each communication information frame transmission. The functional codes defined by ModBus communication protocol is from 1 to 127. As a host request, it tells the slave what action to perform through the function code. As a slave response, the function code returned from the slave machine is the same as the function code sent from the host machine, and indicates that the slave machine has responded to the host machine and has carried out relevant operations.

Table 8.1 MODBUS Partial Function Code

Function code	definition	operation（binary）
02	Read switching input DI	Reead one chanel or multi-chanel switching status output number(remote communication)

01	Read status output OUT	Read one chanel or multi-chanel switching status output status data
03	Read register data	Read one or more register data
05	Write switching output OUT	Control one chanel relay “connect/break”, remote communication
06	Write single channel register	Write a set of binary data into a single register
10	Write multi-channel register	Write multiple sets of binary data into multiple registers

1.3 Data Area:

The data area includes what information needs to be sent back from the slave or what actions to perform. This information can be data (such as switching input/output, analog input/output, registers, etc.), reference addresses, etc. For example, the host tells the slave to return the value of the register (including the starting address of the register to be read and the length of the read register) through the function code 03, so that the data returned will include the length and content of the register. For different slaves, the address and data information are different (communication information table should be shown).

The power supply adopts Modbus communication protocol, and the host computer (PLC, RTU, PC, DCS, etc.) can read its data register arbitrarily by using communication command (function code 03) (see appendix for details of its data information table).

The response command format is from machine address, function code, data area and CRC code. The data in the data area is two bytes, and the high position is in the front.

1.4 Silent Period Requirements:

Before the data is transmitted, the quiescent time (no data transmission time) of data bus must be longer than (when the baud rate of 5 ms is 9600).

2. MODBUS Function Code Description

2.1 Function Code "03"：Read Multiport Register

For example: The host needs to read address 01, three slave register data with starting address 0001.

The address and data of the slave data register are:

Register address	Register data (hexadecimal)	Corresponding parameters
0010	0BB8(30.00V)	U
0011	01F4(5.00A)	I
0012	3A98(150.00W)	P

Message format sent by host:

Host sending	Bytes	Messages sent	Remarks
Slave address	1	01	Slaves sent to address 01
Function code	1	03	Read register
Initial address	2	0010	Initial address 0001
Read data length	2	0003	Read 3 registers (6 bytes in total)
CRC code	2	040E	CRC Code Calculated by Host

Message format returned from slave machine response:

Slave response	bytes	Response information	Remarks
Slave address	1	01	来自从机01 From slave 01
Function code	1	03	Read register
data length（bytes）	1	06	A total of 6 bytes
Data of register 1	2	0BB8	Content of register 0001
Data of register 2	2	01F4	Content of register 0002
Data of register 3	2	3A98	Content of register 0003
CRC code	2	D311	CRC Code Obtained from Slave Computing

2.2 Function Code "06"：Write Single Register

The host uses this function code to store the data in the internal data storage of the power supply. Registers in the Modbus protocol refer to 16 bits (2 bytes), and the high bits are in the front. So the memory is two bytes.

For example, the host should save 0E10 to the slave register with address of 0004 (slave address code 01).

Message format sent by host：

Host sending	bytes	Messages sent	Remarks

Slave address	1	01	Slaves sent to address 01
Function code	1	06	Write a single register
Initial address	2	0004	The address of the register to be written to
save data	2	0E10	Data to be written to 0004 address
CRC code	2	CDA7	CRC Code Calculated by Host

Message format returned from slave machine response：

Slave response	bytes	Messages sent	Remarks
Slave address	1	01	Slaves sent to address 01
Function code	1	06	Write a single register
Initial address	2	0004	The address of the register to be written to
Save data	2	0E10	0004 address data
CRC code	2	CDA7	CRC Code Obtained from Slave Computing

3. Error Check Code（CRC Check）：

The host or slave computer can use the check code to identify whether the received information is correct or not. Due to electronic noise or some other interference, information may sometimes make errors in the transmission process. Error Check Code (CRC) can check whether the information of host or slave is wrong in the process of communication data transmission. Error data will be abandoned (whether sent or received), which increases the security and efficiency of the system.

The CRC (Redundant Cyclic Code) of MODBUS protocol contains two bytes, which is 16-bit binary number. The CRC code is calculated by the transmitting device (host) and placed at the end of the transmitting information frame. The device receiving the information (slave) recalculates the CRC of the received information and compares the calculated CRC with the received CRC. If the two do not match, it indicates an error.

Steps of CRC Code Algorithm：

1、Preset a 16-bit register as hexadecimal FFFFFF (that is, all 1), the register is called CRC register;

2、The first 8-bit binary data (the first byte of the communication information frame) is XOR with the lower 8-bit of the 16-bit CRC register, then the result is placed in the CRC register.

3、Move the contents of the CRC register one bit to the right (toward the lower position) and fill the highest position with 0, and check the move-out position after the right move.

4、If the removal bit is 0: repeat step 3 (move one bit to the right again); if the removal bit is 1: CRC register will be XOR with the polynomial A001 (1010 0000 0000 0001);

5、Repeat steps 3 and 4 until moving right eight times, so that all eight bits of data are processed.

6、Repeat steps 2 to 5 to process the next byte of the communication information frame

7、After calculating all bytes of the communication information frame according to the above steps, the high and low bytes of the 16-bit CRC register are exchanged.

8、The final CRC register content is the CRC code.