

Chapter I Use the PC computer software to control DPM8600

1. Connect the DPM8600 to the computer with a "USB to TTL data cable", charge the power module (10-75V DC), and switch the seesaw switch to ON to power on; DPM8600-485 series are wired according to A-A, B-B.



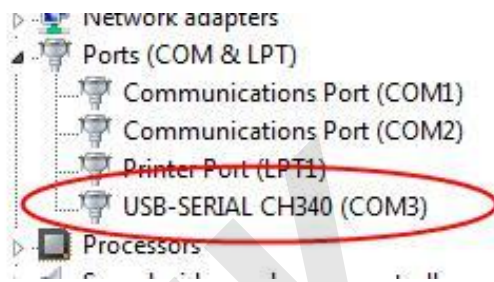
2. Open the data package and double-click to run "install"
3. Select English



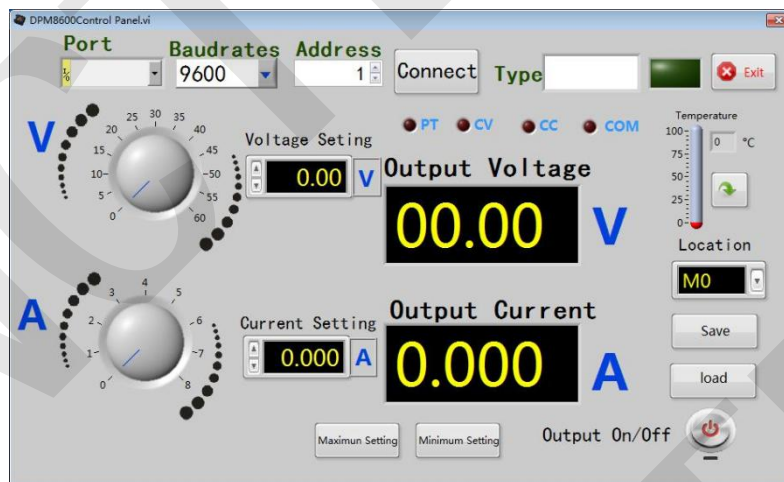
4. Install (USB driver)-(Ni-Visa)-(application) in order, and note that each step cannot be omitted



5. Select "My Computer"-Device Manager-view the assigned port number;



6. Open the computer software as shown in the figure below, and pay attention to selecting the corresponding "port". The "communication baud rate" and "slave address" should be consistent with the internal settings of the DPM8600 power supply. The default communication baud rate of the machine is "9600" and the slave address is "1"; then click the "connect", the model and serial number displayed, indicating that the connection is finished.



Chapter II Use Serial Port Assistant to Debug DPM8600

The DPM8600 can be controlled by the secondary development of the "simple communication protocol" and "MODBUS communication protocol" provided by us. The two communication protocols can be freely selected in the DPM8600 function "5-CS". 0 means "custom communication protocol", 1 means "MODBUS communication protocol".

Before using the communication protocol to program, you can use the serial port assistant to debug the commands. For example: 01w10=1234,\r\n, which means that the voltage setting value is: 12.34V

Chapter III Simple Communication Protocol

Overview

The control instructions generally adopt the command line method, and the communication rate can be selected among seven baud rates (2400, 4800,

9600, 19200, 38400, 57600, 115200), and the machine address code can be set in the range of 01-99 in total 99, please refer to the manual for specific setting operation.

The instructions are issued by the PC, and the machine parses and executes them. When the address codes are the same, the result is returned to the PC. When the address codes are different, no information is returned. This is very suitable for multi-machine centralized control.

The current in the agreement is uniformly expressed as an accuracy of 0.001A, and the display accuracy of DPM-8616 and DPM-8624 is 0.01A. The third decimal place is automatically ignored.

The format of sending data is as follows:

Start bit	:
Address code	01-99
Function code	w,r
Function number	00-99
Connector	=
Operand	0-65535
Spacer	,
End code	\r\n

1. The start character is ":"
2. The address code means the address of the machine, and the setting range is 01~99.
3. The function code is a lowercase letter 'w' or 'r', which means writing or reading.
4. The function number is the number for different functions, and different values represent different parameter settings.
5. The operand is the operand of the command.
6. Spacer: Use "," to space between different data.
7. End code, the end of a command is represented by \r\n, which is carriage return and newline in the ASCII code, expressed in hexadecimal as 0x0d,0x0a

w command

1. 10 command, write voltage setting value

The format is: 01w10=****, where "****" is the value, which represents the set voltage value. For example: 01w10=1234,\r\n, which represents the

voltage set value: 12.34V

2. 11 command, write current setting value

The format is: 01w11=****, where "****" is the value, which represents the set current value. For example: 01w11=12345, which represents the voltage set value: 12.345V

3. 12 command, output status setting

The format is: 01w12=*, where "*" is the value, which represents the output status setting. For example: 01w12=0, means the output is turned off; 01w12=1, means the output is turned on

4. 20 command, set the voltage and current setting values at the same time

The format is: 01w20=****,####,\r\n, where "****" means voltage setting value, "####" means current setting value. For example: 01w20=1234, 2345, which means that the voltage and current setting values are: 12.34V, 2.345A.

r command

1. 00 command, read the maximum output voltage

The format is: 01r00=0. For example: send: 01r00=0, return: 01r00=6000, which means the maximum output voltage is 60V

2. 01 command, read the maximum adjustment current

The format is: 01r01=0. For example: send: 01r01=0, return: 01r01=24000, which means the maximum output current is 24A, the corresponding model is DPM-8624

return: 01r01=16000, which means the maximum output current is 16A, the corresponding model is DPM-8616

return: 01r01=8000, which means the maximum output current is 8A, the corresponding model is DPM-8608

return: 01r01=5000, which means the maximum output current is 5A, the corresponding model is DPM-8605

3. 10 command, read voltage setting value

The format is: 01r10=0. For example: send: 01r10=0, return: 01r10=1234, means that the preset voltage value is 12.34V

4. 11 command, read the current setting value

The format is: 01r11=0. For example: send: 01r11=0, return: 01r11=12345, means that the current setting value is 12.345A

5. 12 command, read the output status

The format is: 01r12=0. For example: send: 01r12=0, return: 01r12=0,

which means the output is off ; send: 01r12=0, return: 01r12=1, which means the output is on

6. 30 command, read the measured output voltage

The format is: 01r30=0. For example: send: 01r30=0, return: 01r30=2345, which means that the current output voltage measured value is 23.45V

7. 31 command, read the measured current voltage

The format is: 01r31=0. For example: send: 01r31=0, return: 01r31=12345, which means that the current output current measured value is 12.345V

8. 32 command, read CC CV status

The format is: 01r32=0. For example: send: 01r32=0, return: 01r32=0, which means constant voltage output (CV); send: 01r32=1, return: 01r32=1, which means constant current output (CC);

9. 33 command, read temperature

The format is: 01r33=0. For example: send: 01r33=0, return: 01r33=0, which means that the internal temperature of the machine is 30°C

ChapterIV MODBUS Communication Protocol

Protocol brief introduction

Using RS485 or transmission interface, the communication protocol is MODBUS-RTU protocol. This product only supports function codes 0x03, 0x06, 0x10. The current in the agreement is uniformly expressed as an accuracy of 0.001A, and the display accuracy of DPM8616 and DPM8624 products is 0.01A. The third decimal place is automatically ignored.

Communication protocol introduction

Information transmission is asynchronous, Modbus-RTU mode uses 11-bit bytes as the unit

Word format (serial data)	10-bit binary
Start bit	1-bit
Data bit	8-bit
Parity bit	None
End bit	1-bit

Data frame structure:

Data frame interval	Address code	Function code	Data area	CRC check
3.5 bytes or more	1 byte	1 byte	N byte	2 bytes

Before sending data, it is required that the data bus static time is greater than 3.5 (for example: 5ms when the baud rate is 9600). The message transmission must start at least with a pause interval of 3.5 bytes. The entire message frame must be used as a continuous data transmission stream, if there is a pause of more than 3.5 bytes before the frame is completed, the receiving device will refresh the incomplete message and assume that the next byte is the address field of a new message. Similarly, if a new message starts after the previous message in less than 3.5 characters, the receiving device will consider it a continuation of the previous message.

1. Address code

The address code is the first byte (8 bits) of each communication information frame from 1 to 255. This byte indicates that the slave with the address set by the user will receive the information sent by the master. Each slave must have a unique address code, and only the slaves that match the address code can respond to the return message. When the slave sends back information, the data sent back starts with their respective address codes. The address code sent by the master indicates the address of the slave to be sent, and the address code returned by the slave indicates the address of the returned slave. The corresponding address code indicates where the information comes from.

2. Function code

The function code is the second byte transmitted in each communication information frame. The function codes that can be defined in the ModBus communication protocol are 1 to 127. Send as a host request, tell the slave what action should be performed through the function code. As a response from the slave, the function code returned by the slave is the same as the function code sent from the master, and indicates that the slave has responded to the master and has performed related operations. This machine only supports 0x03, 0x06, 0x10 function codes.

Function code	Definition	Operation (binary)
0x03	Read register data	Read the data of one or more registers
0x06	Write a 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

3. Data area

The data area includes what kind of information needs to be returned by the slave or what action to perform. This information can be data (such as switch input/output, analog input/output, register, etc.), reference address, etc. For example, if the master tells the slave to return the value of the register through the function code 03 (including the starting address of the register to be read and the length of the read register), the returned data includes the data length and data content of the register.

0x03 Read function host format

Address code	Function code	Register start address	Number of register addresses n	CRC check code
1 byte	1 byte	2 bytes	2 bytes	2 bytes

0x03 Read function slave return format

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

0x06 Write single register function host format

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

0x06 Write a single register function slave return format

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

0x10 Write function host format

Address code	Function code	Register start address	Number of register addresses n	Number of bytes written 2*n	Register data	CRC check code
1 byte	1 byte	2 bytes	2 bytes	1 byte	2*n bytes	2 bytes

0x10 Write function slave return format

Address code	Function code	Register start address	Number of register addresses n	CRC check code
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Introduction to protocol registers (data in a single register address is double-byte data)

Item	Value	Description	Number of bytes	Range	Decimal point	Unit	Read and write	Address	Remarks
1	Set-U	Voltage setting value	2	0-Umax	2	V	r/w	0000H	Note 1
2	Set-I	Current setting value	2	0-Imax	3	A	r/w	0001H	Note 1
3	SW	Output switch	2	0, 1	0	-	r/w	0002H	Note 2
4	CCCV	Output state	2	0,1,2	0	-	r	1000H	Note 3
5	U	Voltage display value	2	Actual value	2	V	r	1001H	Note 4
6	I	Current display value	2	Actual value	3	A	r	1002H	Note 4
7	T	Temperature display value	2	Actual value	0	°C	r	1003H	Note 5

Note:

- Set-U, Set-I are two set values, used to control the voltage and current set values of the power supply.
- Control whether the output is turned on, write 0 to turn off, write 1 to turn on, and the status can also be read.
- This is a read-only register. Reading 0 means no output, 1 means CV status, and 2 means CC status.
- U, I are two real-time measurement values, used to read back the voltage and current values of the power supply.
- T is the temperature measurement value, used to read back the internal

temperature of the power supply.

- The current accuracy in the agreement is unified to 0.001A

4. Error check code (CRC check)

The master or slave can use the check code to judge whether the received information is correct. Due to electronic noise or some other interference, errors sometimes occur during the transmission of information. The error check code (CRC) can check whether the information in the communication data transmission process of the master or slave is wrong, and the wrong data can be discarded (whether it is sending or receiving), which increases the safety and efficiency of the system. The CRC (Redundant Cyclic Code) of the MODBUS communication protocol contains 2 bytes, which is a 16-bit binary number. The CRC code is calculated by the sending device (host) and placed at the end of the sending information frame. The device (slave) receiving the information recalculates the CRC of the received information, and compares whether the calculated CRC is consistent with the received one. If the two do not match, it indicates an error. When the CRC check code is sent, the low bit is first and the high bit is behind.

CRC code calculation method

- (1) Preset a 16-bit register as hexadecimal FFFF (that is all 1), call this register a CRC register
- (2) XOR the first 8-bit binary data (that is, the first byte of the communication information frame) with the lower 8 bits of the 16-bit CRC register and put the result in the CRC register
- (3) Shift the contents of the CRC register one bit to the right (toward the lower bit) and fill the highest with 0 Bit, and check the shifted bit after right shift;
- (4) If the shifted bit is 0: repeat step 3 (shift right one bit again); if the shifted bit is 1: XOR CRC register and polynomial A001
- (5) Repeat steps 3 and 4 until right shift 8 times, so that the entire 8-bit data has been processed
- (6) Repeat steps 2 to 5 to process the next byte of the communication information frame
- (7) After all the bytes of the communication information frame are calculated according to the above steps, the high and low bytes of the obtained 16-bit CRC register are exchanged
- (8) The content of the finally obtained CRC register is the CRC code.

Communication examples

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

The message format sent by the host:

Host send	Number of bytes	Information sent	Remarks
Slave address	1	01	Send to the slave with address 01
function code	1	03	Read register
Register start address	2	0000H	Register start address
Number of register addresses	2	0002H	2 bytes in total
CRC code	2	C40BH	CRC code calculated by the host

For example, if the current display value is 05.00V, 5.000A, the message format returned by the slave in response:

Slave response	Number of bytes	Information returned	Remarks
Slave address	1	01	From slave 01
Function code	1	03	Read register
Number of bytes read	1	04	1 byte total
The contents of the register at address 0002H	2	01F4H	Output voltage display value
The contents of the register at address 0003H	2	1388H	Output current display value
CRC code	2	B76BH	CRC code calculated by slave

Example 2: The host wants to set the voltage 24.00V

The message format sent by the host:

Host send	Number of bytes	Information sent	Remarks
Slave address	1	01H	From slave 01
Function code	1	06H	Write a single register
Register address	2	0000H	Register address
The address is the contents of the register 0000H	2	0960H	Set output voltage value
CRC code	2	8FB2H	CRC code calculated by the host

Example 3: The host wants to set the voltage to 24.00V and the current to 1.500A.

The message format sent by the host:

Host send	Number of bytes	Information sent	Remarks
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
Number of bytes written	1	04H	1 byte total
The address is the contents of the register 0000H	2	0960H	Set output voltage value
The address is the contents of the register 0001H	2	05DCH	Set output current value
CRC code	2	F2E4H	CRC code calculated by the host

The message format returned in response after receiving the slave:

Slave response	Number of bytes	Information returned	Remarks
Slave address	1	01H	Send to the slave with address 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
CRC code	2	41C8H	CRC code calculated by slave