The Communication Protocol of Cabinet Air Conditioner

Version	Date	Modify content	Change type	Production	Sign	Review
V1.00	2018/3/2	First draft	New	Will		
V2.00	2018/7/25	Increase high and low voltage, compressor alarm content	Change logic	Will		
V3.00	2019/6/12	Add description of variable	Update	Will		

Communication Protocol

B.1 Realization of ModBus RTU communication protocol

Cabinet air conditioner controller can support ModBus RTU communication protocol （ModBus is a registered trademark of Modicon Corporation). The communication protocol describes in detail the input and output commands, information, and data of the controller for use and development by third parties.

B.2 Physical interface

Connecting the upper PC with Industrial Standard Serial RS485 Communication Port.Data transmission method is asynchronous, Start bit 1, Data bit 8， and Stop bit 1. No check and the rate of data transmission is 9600b/s.

B.3 Data byte format description

ModBus RTU adopts master-slave structure to effectively transfer information and data between upper computer and controller, allowing upper computer to access relevant data of air conditioner controller and send control commands.

The main communication process is shown in Figure B.1：

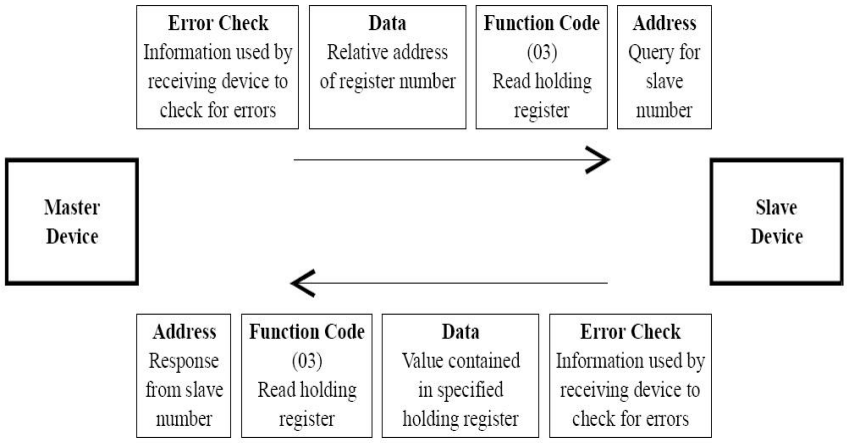

Figure B.1 The main communication process

Data frame composition is shown in the table B.1(RTU mode)

Table B.1 Data frame composition

Address Code	Function Code	Data Info	CRC Check
8 bit	8 bit	N*8 bit	16bit

B.3.1 Address Code（Device Number）

Address code (Device number) is the first byte transmitted by communication. This byte

indicates that the slave, whose device number is set by the user, will receive the message sent by the master. And each slave has a unique device number, and the response echo starts with its own device number. The device number sent by the master indicates the slave address to be sent, and the device number sent by the slave indicates the slave address to be sent back. Address 0 is the broadcast address. All slave machines receive but do not reply.

B.3.2 Function Code

The function code is the second byte of each data frame transfer，The ModBus function code that can be defined by the communication protocol is 1，2，3，4， 5，6，7，11，12，15，16，17，20，21，22，23，24. And only the 03H and 06H function codes are used in the cabinet air conditioner controller. The master sends the request and tell the slave what action to perform through the function code. And the slave machine returns the same function code as the one sent from the master, indicating that the slave machine has responded to the host and done the related operation. Shown as the Table B.2.

Table B.2 ModBus function code used in the computer room aircon

Item	Function code	Definition	Description
1	03H	Read Holding Registers	Read hold register value
2	06H	Preset Single Register	Rewrite a hold register value
3	01H	Read Coil state	Read coil status
4	05H	Preset Single Coil	Rewrite a coil

B.3.3 Data area

Data areas include what information needs to be returned from the machine or what actions need to be performed. This information can be data, reference addresses, etc.

B.3.4 Error check（16-bit CRC check）

The master or slave can use the check code to determine whether the received information is correct. Due to electronic noise or some other interference，information sometimes generates errors during transmission. Error check code（CRC) can check whether the information of the master or slave during the transmission of the communication data is incorrect. Wrong data can be abandoned（no matter sending or receiving), which increases the security 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.CRC code calculated by the transmitting device (master) and placed at the end of the transmitted message frame (CRC High byte are in front). The device receiving the information (slave) recalculates the CRC of the received message and compares whether the calculated CRC matches the received one. If these two do not match, it indicates an error.

The Calculation formula of 16-bit CRC checksum. CRC-16 $= x^{16} + x^{12} + x^5 + x^0$

The calculation step：

a) The preset 16-bit register is hexadecimal FFFF (that is, all 1), which is called the CRC register;

b) The first 8 bits of data are distinguished from the lower bits of the 16-bit CRC register, and the result is placed in the CRC register;

c) Shift the contents of the register to the right by one bit (toward the lower bit), fill the highest bit with 0, and check the lowest bit (Note: the lowest bit at this time refers to the lowest bit before shifting, not the lowest bit after shifting);

d) If the least significant bit is 0: Repeat step 3 (shift again), if the least significant bit is 1: the CRC register is XORed with the polynomial A001H (1010000000000001B);

e) Repeat steps 3 and 4 until the right shift is 8 times, so that the entire 8-bit data is processed;

f) Repeat steps 2 through 5 for the next 8-bit data processing;

g) The resulting CRC register is the CRC code.

B.3.5 Command message format

B.3.5.1 The Function code "03H" can access all input registers which is Mainly used to read the parameters of the device.

Sending format：

Address	Function code	Starting Address		Number of data		CRC	
01	03	High 8 bit	Low 8bit	High 8 bit	Low 8bit	High 8 bit	Low 8 bit

Return format：

Address	Function code	Number of bytes	Data	CRC	
01	03	N	data（8 bit）1..N	High 8 bit	Low 8 bit

B.3.5.2 Function code "06H" can modify a register which is mainly used to set a variable parameter in this protocol.

Sending format：

Address	Function code	Register address		Data		CRC	
01	06	High 8 bit	Low 8bit	High 8 bit	Low 8bit	High 8 bit	Low 8 bit

Return format：

Address	Function code	Register address		Data		CRC	
01	06	High 8 bit	Low 8bit	High 8 bit	Low 8bit	High 8 bit	Low 8 bit

B.3.6 Communication address list

The analog data in the table below should be multiplied by 10 when transferred.

Name	Modbus address	Variable type	Description
Refrigeration set point	41(29H)	analog (write&read） unsinged	Refrigeration start point = Refrigeration set point + Refrigeration deviation Refrigeration stop point = Refrigeration set point
Refrigeration deviation	42(2AH)	analog (write&read） unsinged	
High temperature alarm point	38(26H)	analog (write&read） unsinged	
Low temperature alarm point	40(28H)	analog (write&read） unsinged	
Heating set point	13(0DH)	analog (write&read） unsinged	Heating start=Heating set point Heating stop= Heating set point + Heating deviation
Heating deviation	14(0EH)	analog (write&read） unsinged	
Current cabinet temperature	102(66H)	analog (write&read） unsinged	
switching machine	64(40H)	Digital (write&read） char	1: switch on， 0: shut down
High and low pressure alarm	41(29H)	Digital (read only） char	1 : alarm， 0 : no alarm
High and low temperature alarm	51(33H)	Digital (read only） char	1 : alarm， 0 : no alarm
Sensor failure	46(2EH)	Digital (read only） char	1 : alarm， 0 : no alarm
Compressor failure	47(2FH)	Digital (read only） char	1 : alarm， 0 : no alarm
High and low voltage alarm	50(32H)	Digital (read only） char	1 : alarm， 0 : no alarm

Reading analog：Temperature

[sending]01 03 00 66 00 01 64 15

[receiving]01 03 02 00 FA 38 07

Analysis：temperature 00 FA= 250（Decimal /10 ,Actual value:25）

Setting the analog: Refrigeration set point is 35℃

[sending]01 06 00 29 01 5E D8 6A

[receiving]01 06 00 29 01 5E D8

Analysis：temperature 01 5E= 350（Decimal /10）

Close:（Setting switching value)

[sending]01 05 00 40 00 00 CC 1E

[receiving]01 05 00 40 00 00 CC 1E

Analysis：close 00 00

Open:（Setting switching value)

[sending]01 05 00 40 00 01 0D DE

[receiving]01 05 00 40 00 01 0D DE

Analysis：open 00 01

Reading high temperature alarm

[sending]01 01 00 33 00 01 0D C5

[receiving]01 01 01 01 90 48

Analysis： 01 High temperature alarm