

Communication Protocol

Communication Method: RS485 ModBus RTU

Baud Rate: 9600 bps

Start Bit: 1 bit

Data Bits: 8 bits

Parity Check: None

Stop Bits: 1 bit

Read Device Status Frame Format

Device Address	Function Code	Register Address	Register Quantity	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes
01—200	0x03	0x00 0x00	0x00 0x08	CRC High CRC Low

Device Return Code

Device Address	Function Code	Register Quantity	Data	CRC16
01—200	0x03	0x10	16 bytes	CRC High CRC Low

Data Parsing:

00 - Power On/Off Status (0000: Unit Off; 0001: Unit On)

01 - Set Humidity (16-bit data)

02 - Ambient Humidity (16-bit data)

03 - Ambient Temperature (16-bit data)

04 - Operating Mode (0000: Automatic; 0001: Ventilation; 0002: Constant Temperature; 0003: Constant Humidity)

05 - System Fault (High byte: 80=Temperature and Humidity Fault; 40=Coil Fault; 20=High Pressure Alarm; 10=Low Pressure Alarm; 08=Compressor Overload; 04=Three-Phase Alarm; 02=High Temperature Alarm; 01=Low Water Level Alarm; Low byte: 80=High Water Level Alarm)

06 - Set Temperature (16-bit data)

07 - Auto/Manual (0000: Manual Mode; 0001: Automatic Mode)

Control Command Frame Format

Device Address	Function Code	Register Address	Register Value	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes
01—200	0x06	0x00 0x01—0xFF	XX XX	CRC High CRC Low

Device Return Code

Original Command Response

Control Command Parsing

Register Address	Register Address and Values (High Byte First, Low Byte Second)
0x00 0x00	0001 - Unit On 0000 - Unit Off
0x00 0x01	Set Humidity: 00XX (XX=Set Value in Hexadecimal)
0x00 0x04	0000: Automatic; 0001: Ventilation; 0002: Constant Temperature; 0003: Constant Humidity
0x00 0x06	Set Temperature: 00XX (XX=Set Value in Hexadecimal)
0x00 0x07	0000=Manual Mode; 0001=Automatic Mode

CRC16 Check C Language Program Code

```
-----  
uncharCRC_data[40];           //The data group for CRC check can be changed according to  
                               the data volume (maximum 255 data)  
  
unchar  CRC_H;                //CRC Check High  
unchar  CRC_L;                //CRC Check Low  
unchar  temp;                 //N data to be checked  
unint    CRC_16;              //Check preset value  
#define  CRC_DXS              0xa001  
-----  
  
modbus_CRC16(temp);           //CRC16 check calculation subroutine  
void modbus_CRC16(unchartemp)  
{  
    unchar  t_1;  
    unchart_2;  
    unchar  data;  
    CRC_16=0xffff;            //Preset value  
  
    for(t_1=0;t_1<temp;t_1++)  
    {  
        CRC_16^=CRC_data[t_1];  
        for(t_2=0;t_2<8;t_2++)  
        {  
            data=CRC_16&0x0001;  
            CRC_16>>=1;  
            if(data==1)  
            {  
                CRC_16^=CRC_DXS;  
            }  
        }  
    }  
    CRC_H=CRC_16;              //CRC Check High  
    CRC_16>>=8;  
    CRC_L=CRC_16;              //CRC Check Low  
}
```