# The Realization of Serial Communication Between Kingview 6.55 and MCU Based on Modbus-RTU Protocol

Haolin Peng[1, a], Yong Zheng[2,b] ,Guanda Lu[3,c] and Long Luo[4,d]

[1,2,3,4]College of Electrical and Control Engineering, North China University of Technology, Beijing,China

[a]1433583241@qq.com, [b]cntzy@ncut.edu.cn, [c]956602704@qq.com,[d]123456789@qq.com

**Keywords:** Modbus-RTU Communication Protocol; Kingview 6.55; C8051F020; CRC check

**Abstract.** This paper introduces Modbus-RTU Communication Protocol. Then, based on the analysis of the principle of communication and combining with an instance, the paper gives out a specific method to realize the serial port communication between computer and MCU C8051F020 in Kingview 6.55.

## Introduction

With the continuous development of automation technology, the application of MCU has developed from independent single one to networking. And the distributed computer control system composed by PC and multiple MCUs is applied in many areas, which brings the advantage of MCU in actual time data processing and the advantage of PC in human-machine management and data management into full play. In the system like this, configuration software is often used to accomplish the development of PC monitoring program. Configuration software, which has the features of simple programming and short cycle time, solves the problem that control engineers lack the knowledge of computer while computer engineers lack the technology and experience of control and improve efficiency of automation engineering.

## MCU Hardware Circuit Design

### A. CPU Circuit Design

C8051F020 is used as the micro-controller in slave computer, which is a fully integrated mixed-signal system-on-a-chip (SOC), with the MCS-51 instruction kernel, and has high speed pipeline structure. Fig. 1 is CPU circuit and P5.0 - P5.3 connect 4 bit dip switch to set the device address, P5.4- P5.7 connect 4 bit dip switch to choose the communication mode.

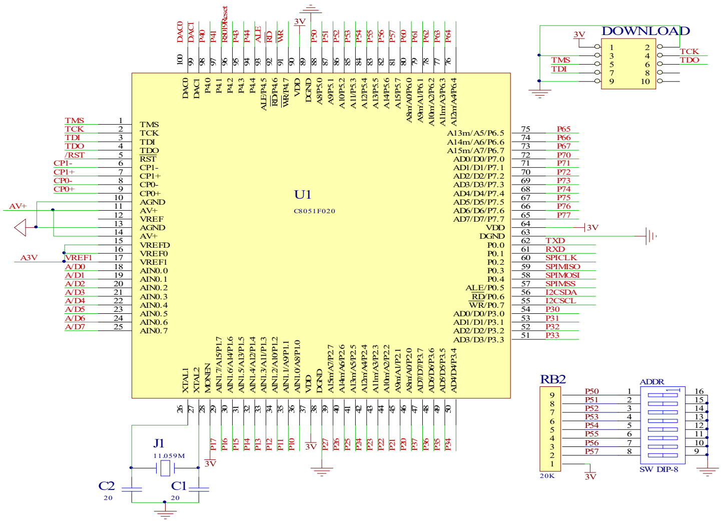


Fig. 1 CPU Circuit

### B. USB Circuit Design

A highly integrated USB-UART bridge chip CP2102 is selected as the communication chip. The circuit is in Fig. 2. The USB interface can be used as a serial port in programming and then the programming is simplified.

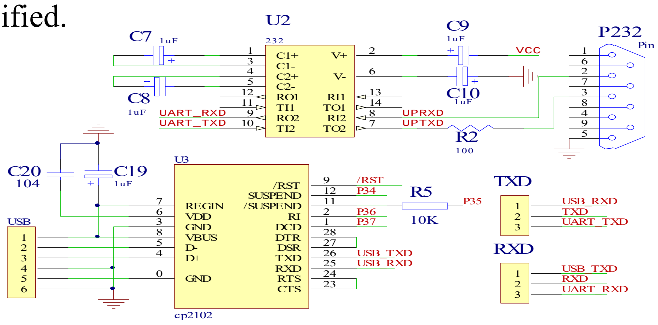


Fig. 2 USB Circuit

### C. AD Conversion Circuit Design

The 3.0V voltage, which is got from AS1117, is linked to external reference voltage input pin VREFO of ADC as the voltage reference of AD conversion. So that the voltage that the ADC0 can convert range from 0 to +3V. The actual input signal voltage range from -5V to +5V processed by the circuit in Fig. 3.

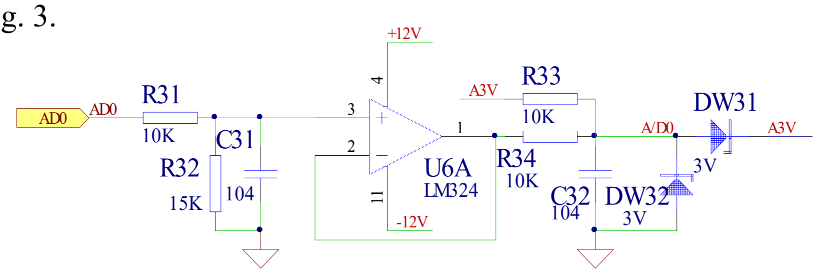


Fig. 3 AD Conversion Signal Process Circuit

## Kingview Monitoring Design

### A. Introduction of Modbus-RTU Communication Protocol

Modbus-RTU communication protocol uses the half-duplex communication mode. The mainframe sends the command signal to the terminal equipment according to the different slave address. After the corresponding operation, the terminal equipment sends the answering signal to the mainframe. The protocol only permits the communication between mainframe and terminal equipment and does not permit it among terminal equipment. Tabel1 shows the data frame format of Modbus-RTU communication protocol.

TABLE 1 THE DATA FRAME FORMAT OF MODBUS-RTU COMMUNICATION PROTOCOL

| Address Code | Function Code | Data Area | Check Code |
|---|---|---|---|
| 8 bits | 8 bits | N*8 bits | 16 bits |

The computer use the query response mode to communicate to the slave computer. The slave computer sends a response message after the computer sends a query command. According to the Modbus-RTU communication protocol and actual requirement, the query command format of the computer is showed in Table 2.

TABLE 2 THE QUERY COMMAND OF THE COMPUTER

| Address Code | Function Code | Starting Channel Number（high-order） | Starting Channel Number（low-order） | The Number of Data（high-order） | The Number of Data（low-order） | CRC Low Byte | CRC High Byte |
|---|---|---|---|---|---|---|---|
| 01-0F | 03 | 00 | 00-07 | 00 | 0X | low byte | high byte |

Address code "01-0F" means that computer obtains by querying the data in 01 to 0F modules in the project; function code "03" represents the operation for modules reading the register values; The computer can query the data of 00 to 07 channel. So the high-order of initial channel is "00" and low-order is 00 to 07; data length "00 0X" means that read the data form the start channel to the 0X channel; CRC check code is calculated by program.

After receiving the command from computer ,the modules executes the corresponding function and return the response message. Table 3 shows the response message format.

TABLE 3 THE RESPONSE MESSAGE FORMAT OF SLAVE COMPUTER

### B. Set Modbus-RTU Communication Protocol

| Device Number | Function Code | The Number of Bytes in the Data | Data1 High Byte | Data 1 Low Byte | …… | CRC Low Byte | CRC High Byte |
|---|---|---|---|---|---|---|---|
| 01-0F | 03 | 02-10 | XX | XX | …… | low byte | High byte |

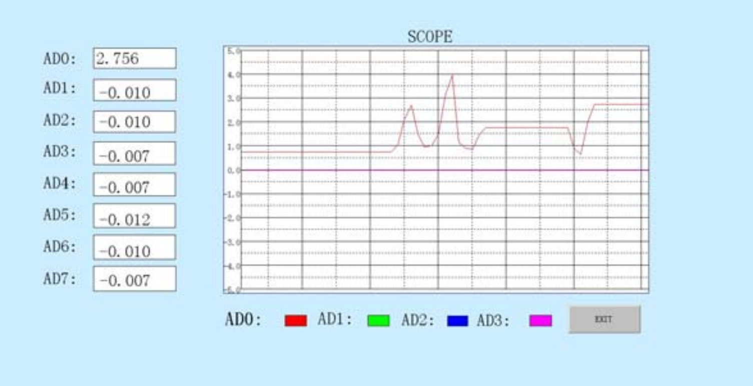
Set the communication protocol with slave computer as Modbus-RTU communication protocol in Kingview project.

The interface of Kingview is ModbusRTU through connector of COM which is visual serial port whose address would be shown as 1/DATA_MSG. They are the steps of Set and Installation for this type of interface.

### C. Kingview Monitoring Program Design

The task of monitoring program in upper computer is that transport the voltage signal in AD0-AD7 collected real-time by MCU to Kingview monitoring program and show on the AD0-AD7 on the interface.

The interface of Kingview monitoring program designed by author is shown in Fig. 4.



AD0: 2.756
AD1: -0.010
AD2: -0.010
AD3: -0.007
AD4: -0.007
AD5: -0.012
AD6: -0.010
AD7: -0.007

Fig. 4 Interface of Kingview Monitoring Program

## MCU Program Design

According to the communication protocol, the first byte of data frame is device address. The program in MCU of slave computer gets different number of device by dial switch in the process of initialization. When receive the data frame from Kingview, MCU compare the device address and local address, only the device which address match respond, and then MCU send response to the Kingview.

### A. Realization of Communication

In MCU program design, use UART interrupt to communicate with upper computer. Judge the type of order in the UART interrupt, then give out the corresponding respond.

1) Initialize Settings

```
if(baud_mode==0xa0)
{
   EA=1;    // interrupt always allowed
   ES0=1;    // UART0 interrupt allowed
   SCON0=0x50;    // receive serial port mode 1
   WorkMode = 5;    // Kingview mode
   PS=1;    // UART0 interrupt priority control
  }
void GetAddr(void)
{
   if(A0) MyAddr+=1;
   if(A1) MyAddr+=2;
   if(A2) MyAddr+=4;
   if(A3) MyAddr+=8;
}
```

2) Program of Receive Serial Port Data by Interrupt

```
if(RI0)
 {
   RI0=0;    // clear the identification of data received
   modbuf =SBUF0;    //copy the data from data buffer register to modbuf
   if(baud_mode == 0xa0)
    {
      modRbuf[modwei]=modbuf;    // save the data from modbuf to the array modRbuf
      modwei++;
      if(modwei == 8)    // finish receiving 8 bit data
      {
        modrecok=1;    // receive standard location 1
        modwei=0;    // restart receive data
        }
     }
    }
```

3) Program of Protocol Processing

Return the asked data according to the function code send by Kingview.

```
 while(WorkMode==5)
 {
  if(modrecok==1)    // finish receiving data
  {
    if(modRbuf[0]==MyAddr&&modRbuf[1]==0x03)
      // compared the device address and local address, send answer to Kingview
     {
      crcr=erheyi(modRbuf[7],modRbuf[6]);    //send the CRC value calculated by device
      if(crc16(modRbuf,6)==crcr)
      {
       for(i=0;i<modRbuf[5];i++)
        {
        ADC_Start(modRbuf[3]+i);
         modTbuf[3+i*2]=(ADC0H<<4)+(ADC0L>>4);
            // send the high byte of data from No.i channel
        modTbuf[4+i*2]=ADC0L<<4; // send the low byte of data from No.i channel
         }
       modTbuf[0]=modRbuf[0];
```

```
        modTbuf[1]=modRbuf[1];
        modTbuf[2]=modRbuf[5]*2;    // the number of data byte is 2X
        crct=crc16(modTbuf,3+modRbuf[5]*2);
        modTbuf[3+modRbuf[5]*2]=crct&0x00ff;    // high byte
        modTbuf[4+modRbuf[5]*2]=crct>>8;    //low byte
        UART0_SendBuf(modTbuf,5+modRbuf[5]*2);    // the data bag sent by UART
        }    //crc
      }    //addr
      modrecok=0;    // clear the standard bit received
    }    //recok
  }    //5
uint erheyi(uchar high,uchar low)
{
 uint he;
 he=(high<<8)|low;
 return he;
}
```

**B. CRC Check**

CRC value is calculated by the sending device and saved in the end of the data frame. The equipment of receiving information recalculates CRC value. And take the CRC value to compare with the new CRC value that calculated by the sending device based on the information that returned by the receiving device. It means that there is something wrong with the communication if the two values are inconsistent.

(1) Set the register 0FFFFH which is CRC register.

(2) Xor the first byte of the data frame with low byte of the CRC register and save the result in the CRC register.

(3) move the data to right for a bit, and fill the high-order with ‘0’, then detect the removing order which is lowest order.

(4) If the removing order is ‘0’, return to step (3);if it is ‘1’, xor CRC register with a fixed value (0A001H).

(5) Repeat step (3) and (4) until 8 shifts. Then a 8 bits data can be gained.

(6) Repeat step (2) to (5) to deal with the next 8 bits data until all the bytes are finished to dispose.

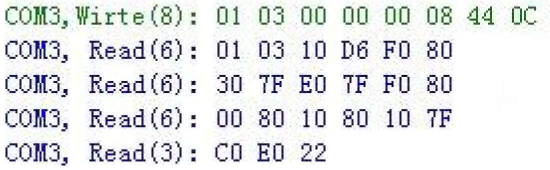
(7) Now CRC register value is CRC value.

According to the above rules, the program of CRC check code is as follows:

```
uint crc16(uchar*str,uint num)
{
 uint i,j,c,crc;
 crc=0xffff;
 for(i=0;i<num;i++)
  {
   c=str[i]&0x00ff;
   crc^=c;
   for(j=0;j<8;j++)
   {
    if(crc&0x0001)
    {
    crc>>=1;
    crc^=0xa001;
     }
    else
    {crc>>=1;}
    }
  }
return(crc);
}
```

### C. Debugging and Analyzing

The serial port monitoring software is used to analysis the Modbus-RTU Communication Protocol in Kingview 6.55 in debugging, which is shown in Fig. 8.



```
COM3,Wirte(8): 01 03 00 00 00 08 44 0C
COM3, Read(6): 01 03 10 D6 F0 80
COM3, Read(6): 30 7F E0 7F F0 80
COM3, Read(6): 00 80 10 80 10 7F
COM3, Read(3): C0 E0 22
```

Fig. 5 Serial Port Monitoring Software

It can be known from the figure that MCU can send the corresponding order actually according to the order sent by the upper, which means that it can accomplish the communicate task.

## Summary

From the above, we can figure out that the Kingview software supports the communication protocol of Modbus-RTU which provides convenience for measure and control unit networking communicating made by user. The workspace for this project is shown in Fig. 6.

Fig. 6 The Workspace for the Project Discussed above

## References

[1] Changfei Tong, *C8051F Series MCU Development and C Programming* (Beijing BUAA Press, Beijing, 2005).

[2] Jiangquan Li, Weibin Cao, Yao Zheng, Zhong Zheng, *The Application of Computer Monitoring and Control and Serial Port Communication Development Software* ( The People's Posts and Telecommunications Press, Beijing,2008).

[3] Shuhong Wang, Jiansheng Wei, *The Technology and Application of Configuration Software Control* (China Electric Power Press, Beijing, 2011).

[4] Zhuojin Pan,Guojun Shi,*The Principle and Application of High Speed C8051 SOC Single Chip* (Beijing BUAA Press, Beijing, 2002).

[5] Shuchun Du,Singlechip, *C Language and Assemble Language Programming Example Explanation*(Beijing BUAA Press, Beijing, 2006).