学院

单片机原理及应用 实验报告

|  |  |
| --- | --- |
| 成绩 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 专业班级 |  | 实验日期 |  |
| 姓 名 |  | 学 号 |  |
| 实验名称 | UART串口收发双向通信 | 指导教师 |  |

（报告内容包括实验目的、实验设备及器材、实验内容和要求、实验步骤、实验小结等）

1. **实验目的**
2. 熟悉89C52全双工异步串行通信方式；
3. 熟悉SCON、PCON设置方法及其对串口的控制作用；
4. 熟悉串口初始化程序、及收发程序的写法。
5. 掌握串口调试助手的使用方法。
6. **实验内容及原理**
7. USB转串口模块

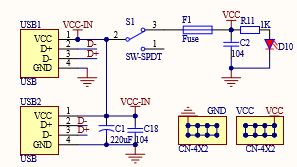
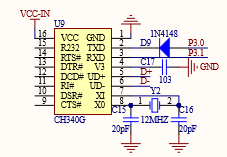
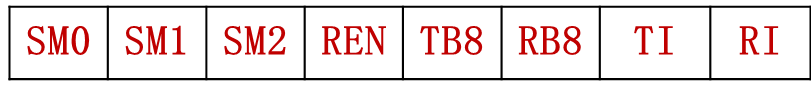


图1 USB转串口电路

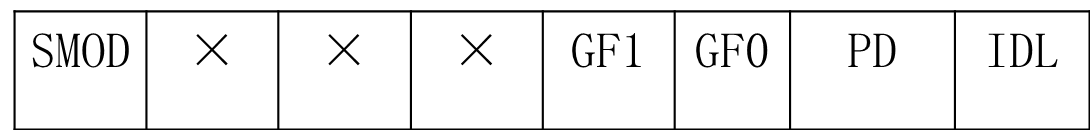
USB接口数据通过CH340G转换为TTL连接至单片机异步串行口P3.0（RXD）和P3.1（TXD）

1. **与串口通信有关的SFR**

**SCON**（98H）

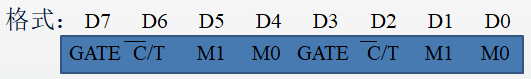
****

**PCON**（87H）

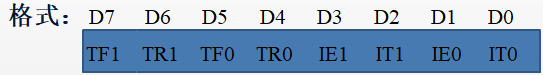


**3、T1方式2常用于串口波特率发生器，与T1有关的SFR**

**TMOD**（89H）



**TCON**（88H）

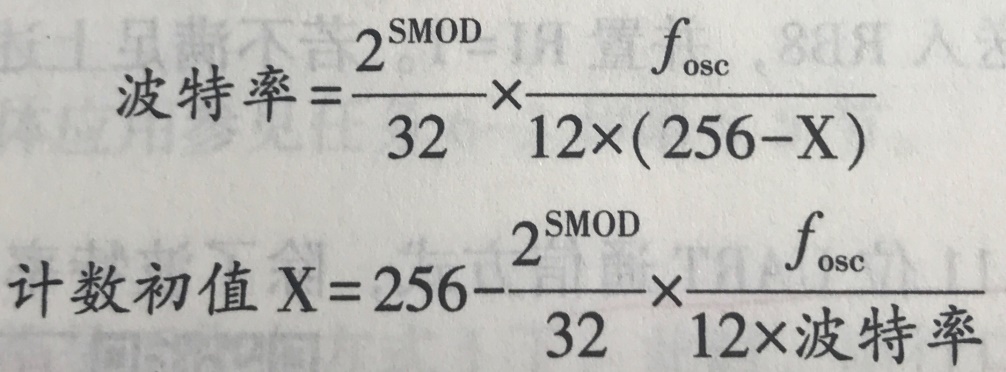


**4、**主频fosc、波特率、计数器初值N 之间的关系：

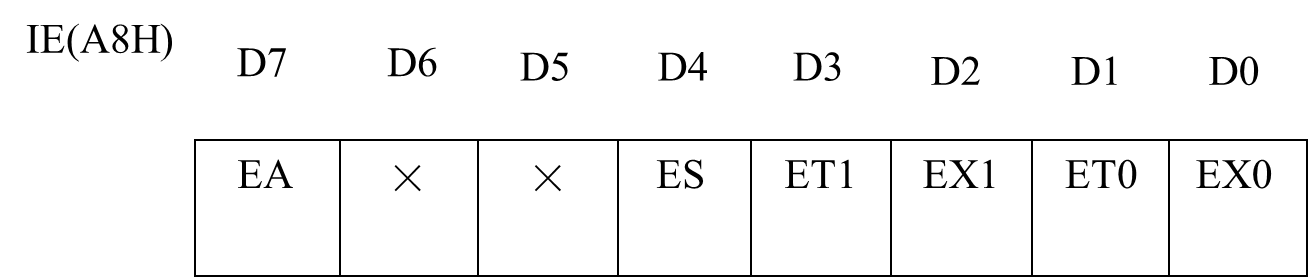
* + - 波特率=2SMOD×T1溢出率/32
    - T1溢出率=fosc/[12\*(2n-N)]

**所以，T1工作在方式2时：**

* **28-N=2SMOD\*fosc/(32\*12)**

****

**5、**串口通信可以工作在查询方式，也可以工作在中断方式。与串口中断有关的中断允许控制寄存器IE：



1. **实验要求**
2. 编写程序实现以下功能：
3. 单片机与计算机之间串行异步通信，波特率9600bps。串口工作方式3，偶校验。定时器1工作方式2，作为串口波特率发生器。
4. 甲机发送字符串：甲机每按下一次独立按键S2键，松开后均会向串口发送字符串“HELLO”；
5. 当乙机接收到甲机发过来的字符串“HELLO”时，若校验正确则回送字符串“HI”，同时将收到的字符串HELLO用数码管显示；若校验错误，回送“NO”。
6. 甲机收到乙机的回送字符串HI或NO后，用数码管显示。

**2、**用USB转串口方式连接计算机与单片机开发板；

**3、**调试、编译并下载程序，用串口助手调试并验证程序执行结果。

1. **实验步骤**
2. 实验所用硬件原理图如下：



根据波特率、单片机时钟频率，及工作方式，计算SMOD及T1初值N：SMOD= 0xd0

N=0xFD

1. **设置串口波特率步骤：**
2. 写TMOD，设置定时器T1的工作方式；
3. TH1、TL1赋值，设置T1初值；
4. TR=1，启动T1，即启动波特率发生器。
5. **串口收/发过程：**
6. 串口初始化：设置串口工作方式（帧格式）、设置T1工作方式、设置波特率（传输速率）、启动波特率发生器。

若以中断方式传输，则开ES和EA，若以查询方式传输无需开中断。

初始化程序如下：

TMOD=0x20； //T1工作在方式2

TH1=0xFD; //T1初值

TL1=0xFD;

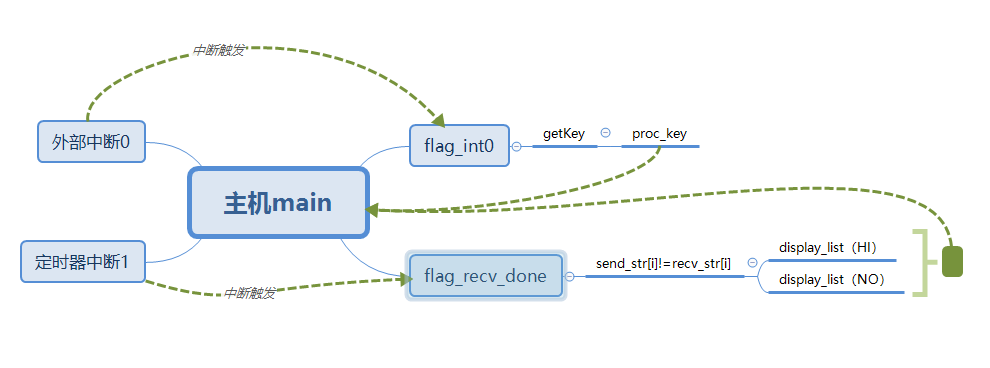
TR1=1; //启动T1

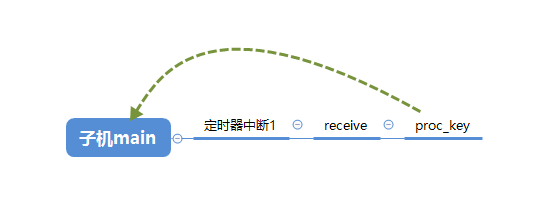
PCON=0x00; //SMOD=0，波特率不翻倍

SCON=0xd0; //串口工作在方式3，允许接收

1. 发送数据/接收数据，即读写SBUF；
2. 判断一帧是否发送完毕，即判断TI是否为1。接收时检测RI标志；
3. TI/RI清零；
4. 跳转到（2），收/发下一帧数据。
5. 编写流程图及实验程序。

**（流程图附下图，实验程序以电子文档提交）；**





3、串行方式连接单片机与计算机；

4、下载程序

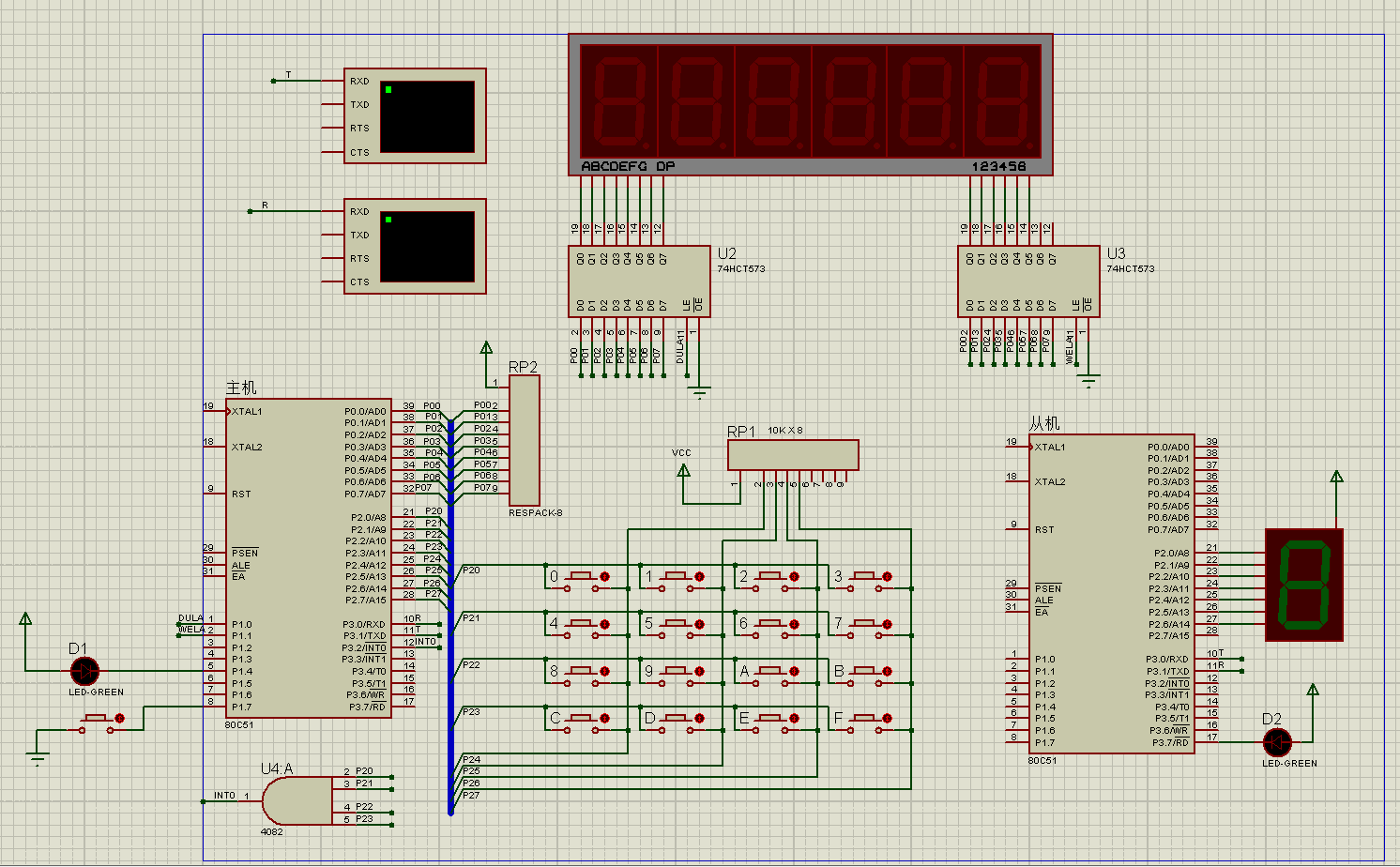
5、在串口助手设置：打开串口、设置相应的串口号( )、波特率(9600)、校验方式(偶校验)、停止位(1)。

6、按下松开开发板S2，在串口助手接收区以字符方式接收并验证收到的数据。

7、在串口助手发送区以16进制数形式发送1字节，如55H、80H、01H等，然后按下并松开S3，观察开发板8位LED是否为对应的数据。

**五、实验结果**

（截图贴于下方）



**附录**

1. 主机源码

/\*  
proj:多机通信（主机）程序  
desc:基于80c51通过串口实现多机通信，主机具备校验返回信息功能。主机接收信息使用定时器中断完成。  
 主机利用中断服务扫描矩阵键盘获取子机序列，子机数量可拓展。  
author:mrskye  
website:https://www.mrskye.cn/  
encoding:utf-8  
\*/  
  
#include <reg51.h>  
#define uchar unsigned char  
#define NODE0\_ADDR 0 //0#主机地址  
#define NODE1\_ADDR 1 //1#子机地址  
  
/\*测试引脚\*/  
sbit P1\_4=P1^4;  
sbit P1\_7=P1^7;  
  
/\*矩阵键盘全局变量\*/  
unsigned char flag\_int0 = 0;  
//行列式键盘键值  
unsigned char key\_buf[] = {0xee, 0xde, 0xbe, 0x7e,0xed, 0xdd, 0xbd, 0x7d,0xeb, 0xdb, 0xbb, 0x7b,0xe7, 0xd7, 0xb7, 0x77};  
  
/\*led显示全局变量\*/  
sbit DULA = P1^0;  
sbit WELA = P1^1;  
//7led显示字形码(阴极)  
unsigned char wordlist[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x00};  
//led灯位  
unsigned char ledlist[] = {0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};  
//默认显示列表  
unsigned char display\_list[] = {0x40,0x40};  
  
  
/\*串口通信全局变量\*/  
uchar KeyValue=0; //键值  
uchar send\_str[]="HELLO"; //发送字符存储列表  
uchar recv\_str[]="AAAAA"; //接受字符存储列表  
uchar flag\_sms = 0; //收发信息正确标志  
uchar pointer\_recv = 0; //主机当前接收字符指针  
uchar flag\_recv\_done = 0; //接收完成标志  
  
void delay(unsigned int i) //可控时间延迟函数  
{  
 unsigned int j;  
 unsigned char k;  
 for(j=i;j>0;j--)  
 {  
 for(k=125;k>0;k--);  
 }  
}  
  
void display(unsigned char display\_list[],unsigned int time)//led显示函数  
{  
 unsigned char i;  
 for(i=0;\*(display\_list+i);i++)  
 {  
 //选择显示位  
 P0=\*(ledlist+i);  
 WELA=1;  
 WELA=0;  
 //输出字符码  
 P0=\*(display\_list+i);  
 DULA=1;  
 DULA=0;  
 delay(time);  
 }  
}  
  
void service\_int0(void) interrupt 0 //外部中断0：启动键盘扫描  
{  
 flag\_int0 = 1;  
}  
  
unsigned char getKey() //矩阵键盘取值函数  
{ unsigned char key\_scan[] = {0xef, 0xdf, 0xbf, 0x7f}; //键扫描码  
 unsigned char i = 0, j = 0;  
 for (i = 0;key\_scan[i]; i++)   
 {   
 P2 = key\_scan[i]; //P2送出键扫描码  
 if ((P2 & 0x0f) != 0x0f) //取P2低四位判断有无键闭合  
 {   
 delay(100);//消抖  
 if ((P2 & 0x0f) != 0x0f) //再次判断键闭合  
 {  
 for (j = 0 ; j < 16 ;j++) //低效而方便地遍历查找闭合键键号  
 {   
 if (key\_buf[j]==P2)  
 {  
 while ((P2 & 0x0f) != 0x0f);//松开按键  
 return j;  
 }   
 }   
 }  
 }  
 }  
 return -1;//无键闭合   
}  
  
void receive(void) interrupt 4 //定时器中断1：串口接收  
{   
 RI=0;  
 if(RB8==1)  
 {  
 if(SBUF==NODE0\_ADDR)  
 {  
 SM2=0;  
 }  
 return;  
 }  
 //存储到列表  
 \*(recv\_str+pointer\_recv)=SBUF;  
 if(++pointer\_recv>4)  
 {  
 flag\_recv\_done=1;  
 pointer\_recv=0;  
 }  
 P1\_4=!P1\_4;  
 SM2=1;  
}  
  
void proc\_key(uchar node\_number,uchar str[])//发送程序  
{  
 switch(node\_number) //切换子机  
 {  
 case 1:   
 {  
 uchar pointer\_1; //子机当前发送字符指针  
 for(pointer\_1=0;\*(str+pointer\_1);pointer\_1++)  
 {  
   
 TB8=1; //发送地址帧  
 SBUF=node\_number; //子机地址  
 while(TI==0); //等待地址帧发送结束  
 TI=0; //清TI标志  
 TB8=0; //准备发送数据帧  
 SBUF=\*(str+pointer\_1); //1#子机字符帧  
 while(TI==0); //等待地址帧发送结束  
 TI=0; //清TI标志  
 delay(1200); //发送延时  
 }  
 }  
 default: break;  
 }  
}  
  
void main()  
{  
 /\*中断服务初始化\*/  
 IE = 0x81; //10000001|总中断、外部中断0  
 TCON = 0x00; //00000000|中断触发方式  
 /\*串口服务初始化\*/  
 SCON=0xf0; //串口方式3、多机通信、允许接收、中断标志清零  
 TMOD=0x20; //T1定时方式2  
 TH1=TL1=0xfd; //9600bps  
 TR1=1; //启动T1  
 ES=1;EA=1; //开中断  
 /\*其余服务初始化\*/  
 P1=0xff; //复位关闭全部LED  
 P2=0x0f; //键盘待触发状态  
 while (1)  
 {  
 if(flag\_int0)  
 {  
 unsigned char key;  
 key = getKey(); //获取子机序号  
 switch(key) //切换子机  
 {   
 case 0x00: proc\_key(NODE1\_ADDR,send\_str);break;  
 default: break;  
 }  
 P2=0x0f; //键盘待触发状态  
 flag\_int0 = 0; //flag\_int0复位  
 }  
 if(flag\_recv\_done)  
 {  
 unsigned char i; //校验指针  
 for(i=0;send\_str[i];i++)  
 {  
 if(send\_str[i]!=recv\_str[i])//收发不一致  
 {  
 display\_list[0] = 0x54;  
 display\_list[1] = 0x5c;  
 flag\_recv\_done = 0; //标志位置零  
 break;  
 }  
 }  
 display\_list[0] = 0x76; //收发一致  
 display\_list[1] = 0x06;  
 flag\_recv\_done = 0; //标志位置零  
 }  
 display(display\_list,20);  
 }   
}

1. 子机源码

/\*  
proj:多机通信（1#子机）  
desc:基于80c51通过串口实现多机通信，从机具备收发功能。通过定时器中断服务  
 函数先接收信息，后调用发送函数返还接收信息给主机  
author:mrskye  
website:https://www.mrskye.cn/  
encoding:utf-8  
\*/  
  
#include <reg51.h>  
#define NODE0\_ADDR 0  
#define NODE1\_ADDR 1  
#define uchar unsigned char  
  
//测试引脚  
sbit P3\_7=P3^7;   
//delay全局变量  
uchar i,j;   
//7led显示字形码(阳极)  
uchar code table[16]={0x88,0x83,0xc6,0xa1,0x86,0x8e,0x00,0x89,0x00,0x00,0x00,0xc7,0x00,0x00,0xc0};  
  
  
void delay(unsigned int i) //可控时间延迟函数  
{  
 unsigned int j;  
 unsigned char k;  
 for(j=i;j>0;j--)  
 {  
 for(k=125;k>0;k--);  
 }  
}  
  
void display(uchar ch)  
{  
 if((ch>=65)&&(ch<=79)) P2=table[ch-65];  
}   
  
void proc\_key(uchar node\_number,uchar str)//发送程序  
{   
 delay(200);   
 switch(node\_number)//切换主机  
 {  
 case 0:   
 {  
 uchar pointer\_0; //主机当前发送字符指针  
 TB8=1; //发送地址帧  
 SBUF=node\_number;  
 while(TI==0); //等待地址帧发送结束  
 TI=0; //清TI标志  
 TB8=0; //准备发送数据帧  
 SBUF=str; //0#主机字符帧  
 while(TI==0); //等待地址帧发送结束  
 TI=0; //清TI标志  
 }  
 default: break;  
 }  
}  
  
void main()  
{  
 SCON=0xd0; //串口方式3、多机通信、允许接收、中断标志清零  
 TMOD=0x20; //T1定时方式2  
 TH1=TL1=0xfd; //9600bps  
 TR1=1; //启动T1  
 ES=1;EA=1; //开中断  
 while(1);  
}  
  
void receive(void) interrupt 4   
{  
 uchar message;  
 RI=0;   
 if(RB8==1)  
 {  
 if(SBUF==NODE1\_ADDR)  
 {  
 SM2=0;  
 P3\_7=!P3\_7;  
 }  
 return;  
 }  
 message = SBUF;  
 display(SBUF); //显示接收字符  
 SM2=1;  
 proc\_key(NODE0\_ADDR,message);//返回消息给主机  
}