

实验五 单片机串行通信

杨庆龙
1500012956

2018.4.18

1 实验目的

- 了解串行通信的基本知识
- 掌握用单片机串行口实现串行通信的方法

2 实验原理

2.1 串行通信的异步和同步传送方式

CPU与外设的基本通信方式可分为并行通信和串行通信两类。并行是指要传输的按照二进制位同时传输，串行则是逐位传输的方式。

串行传输所用的传输线远少于并行传输线，也是实际常用的传输线。

单片机使用异步传输的通信方式，其特点为

- 数据是离散发送的
- 通信双方时钟频率相同
- 通信双方按照异步通信协议传输字符

字符帧格式为，一个起始位，N位数据位，一个校验位，一个停止位

异步通信使用波特率=单帧位数*每秒的帧数表示数据传输的速率。并使用高于传输速率的时钟进行采样提高采样的准确性。

2.2 MCS-51的串行通信接口

MCS51内部有一个可编程的全双工串行通信口，可作为通用异步接收和发送器，也可作为同步移位寄存器使用。内部的串行通信口，有两个独立的接收发送缓冲器SBUF，对外也有两条独立的收发信号线RxD和TxD。可以同时发送，接收数据，实现全双工传送。与串行通信有关的寄存器有多个，用SCON控制和监视串行口的工作状态。

- SM0,SM1:00,同步移位寄存器，01,8位UART，10,9位UART，11,9位UART可变波特率
- REN:允许接收控制位，由软件置位或清除
- TB8:模式2，3中的第九位
- RB8:该位是模式2和3中已接收的第九位
- TI:发送中断标识
- RI:接收中断标识

Table 1: 串行控制寄存器SCON

D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI



Figure 1: 8AH信号波形

3 实验内容

3.1 观察UART通信波形

设定串行口工作方式1，用1200bps循环发送一个字节55H或8AH，用示波器观察TxD的电平和信号结构，给出1200bps波特率条件下的TH1计算值，码元宽度的计算值和测量值。

3.2 串口收发实验

编写一个程序是单片机通过键盘输入数据，再把数据送到PC上，PC又将数据发送给单片机，单片机将数据现实到数码管上。

3.3 串口作为STDIO

初始化了UART0后，单片机会将该串口作为STDIO，用户可以直接用printf和scanf从串口中读写数据。编写程序通过串口输入数据，进行四则运算后输出到串口。

4 实验结果

4.1 观察UART通信波形

设定串口工作方式1,用9600bps的波特率，发送8AH，用示波器观察TxD可得1。从图中可以看到，串口输出信号为0010100011,输出波形符合预期，接下来是波特率检查。设置保持串口工作方式不变，波特率不变，输出55H，用示波器观察TxD可得图像如2，使用示波器的measure功能测量得到，信号频率为4.8kHz的方波，即为9600bps的信号波形。

4.2 串口收发实验

按动按键，可以在PC机的串口监视器上看到相应的按键，使用串口监视器向开发板发送0到9的数字，可以在开发板的发光数码管上显示相应的数字，满足设计要求。

4.3 串口作为STDIO

从串口读入两个操作数和操作符，并将计算结果输出到串口上，结果如图4。

5 思考题

- 例如，当接收方时钟为发射方两倍时，发射方每发出一个bit，接收方都会收到两个同样的bit，这就会导致接收方收到的数据有误。

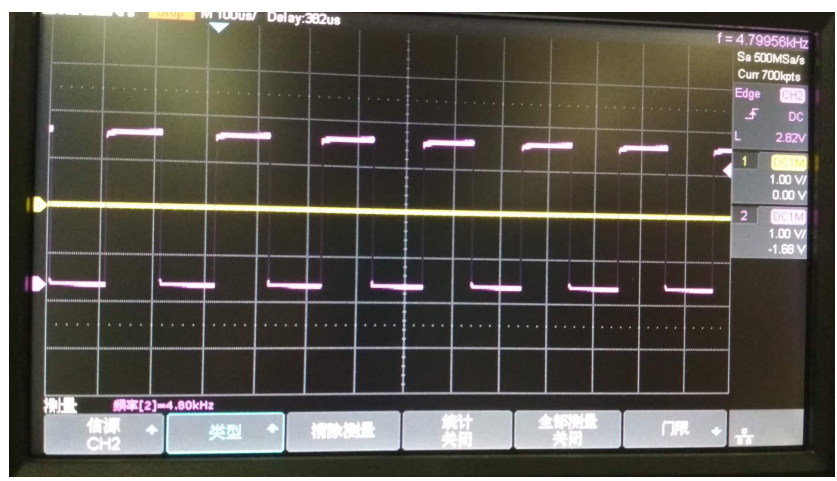
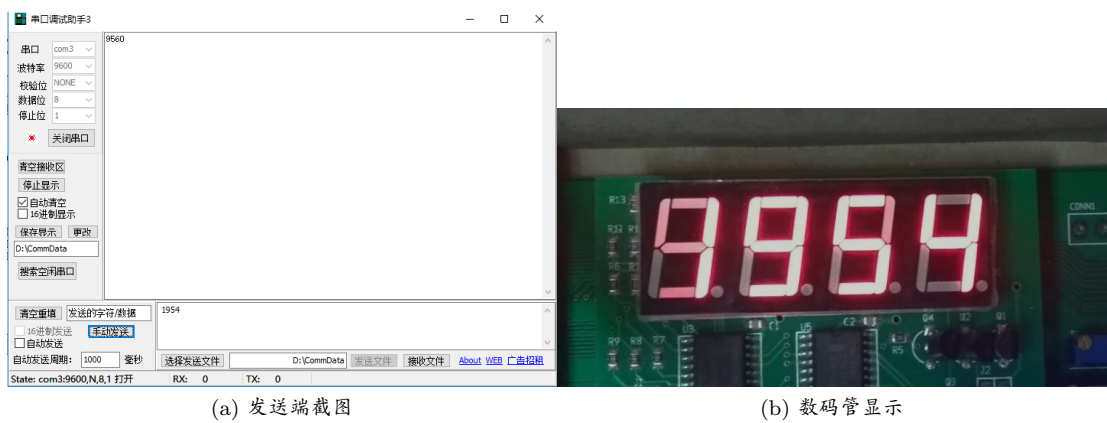


Figure 2: 55H信号波形



(a) 发送端截图 (b) 数码管显示

Figure 3: 串口通信实验结果图

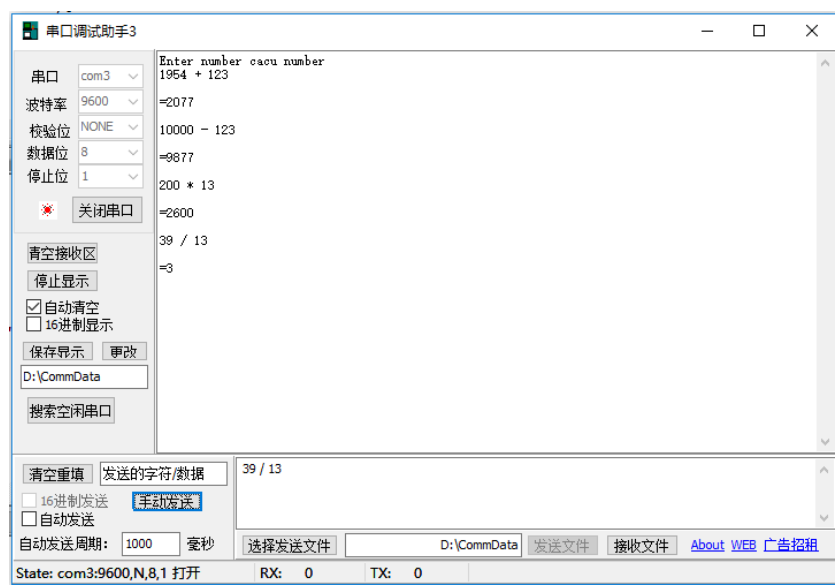


Figure 4: 计算结果

2. 使用系统时钟对230400bps分频，再依据采样率计算得到计数器溢出数字即可。例如对于16倍采样率有 $22118400/230400/16=4608$ ，对于300bps可以得到相同的结论，计数器赋值为 $22118400/300/16=4608$ 。

6 源码

6.1 观察UART通信波形

```
1 #include <C8051F020.h>
2 #include "../includes/time.h"
3 #include "../includes/communicate.h"
4 void main() {
5     WDTCN = 0xDE;
6     WDTCN = 0xAD;
7     sysclk_init();
8     uart0_port_init();
9     uart0_init();
10    while (1) {
11        SBUF0 = 0x55; //0x8A
12        while(!TIO);
13        TIO = 0;
14    }
15 }
```

6.2 串口收发实验

```
1 #include <C8051F020.h>
2 #include "../includes/communicate.h"
3 #include "../includes/keyboard.h"
4 #include "../includes/display.h"
5 #include "../includes/storage.h"
6 #define RX_LEN 4
7 #define EMPTY 255
8
9 int RxBuf[RX_LEN];
10 char TxBuf;
11 uchar keyResult;
12
13 void main() {
14     WDTCN = 0xDE;
15     WDTCN = 0xAD;
16     sysclk_init();
17     uart0_port_init();
18     uart0_init();
19     storage_port_init();
20     int0_init(SYSCLK / 800);
21     ES0 = 1;
22     EA = 1;
23     TxBuf = '\0';
24     while(1) {
25         delay(1000);
26         keyResult = getKey();
27         if(keyResult != NOKEY) {
28             SBUF0 = keyResult + '0';
29             keyResult = NOKEY;
30             TIO = 1;
31         }
```

```

32     }
33 }
34
35 void uart0_int() interrupt 4 {
36     char c;
37     if(RIO == 1) {
38         RIO = 0;
39         c = SBUF0;
40         c -= '0';
41         RxBuf[3] = RxBuf[2];
42         RxBuf[2] = RxBuf[1];
43         RxBuf[1] = RxBuf[0];
44         RxBuf[0] = c;
45     }
46     else if(TIO == 1)
47         TIO = 0;
48 }
49
50 void time0_int() interrupt 1 {
51     static int index = 0;
52     index ++;
53     index %= 4;
54     digital_selecte = digital_index[index];
55     if(RxBuf[index] < 16)
56         digital_number = digital_trans[RxBuf[index]];
57     else
58         digital_number = digital_trans[16];
59 }

```

6.3 串口作为STDIO

```

1  #include <C8051F020.h>
2  #include <stdio.h>
3  #include "../includes/communicate.h"
4  #include "../includes/time.h"
5
6  int main() {
7      int a,c,result;
8      char b;
9      WDTCN = 0xDE;
10     WDTCN = 0xAD;
11     sysclk_init();
12     uart0_port_init();
13     uart0_init();
14     TIO = 1;
15     printf("Enter number cacu number\n");
16     while(1) {
17         if(scanf("%d%c%d",&a,&b,&c)) {
18             switch(b) {
19                 case '+':
20                     result = a+c;
21                     break;
22                 case '-':
23                     result = a-c;
24                     break;
25                 case '*':
26                     result = a*c;

```

```

27         break;
28     case '/':
29         result = a/c;
30     break;
31     default:
32         result = -1;
33 }
34 printf("\r\n=%d\r\n",result);
35 }
36 }
37 }

```