

PIC 串口发送

Step1:

学习 Data Sheet，熟悉 PIC 单片机串口各种工作模式。

串口发送相关特殊寄存器为(From Data Sheet):

TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RXB	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

Note 1: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

PIR1:bank0

RCSTA:bank0

TXREG:bank0

PIE1:bank1

TXSTA:bank1

SPBRG:bank1

Step2:

编写简单的入门程序:

A、同步(借鉴自网络,仅完成测试和简单修改测试)

程序规划:

利用串口配合串转并芯片 74LS164 进行静态显示。

程序如下:

```
#include <pic.h>
```

```
void initUSART(void);
```

```

void display(int);
void delay(int d1);
const int disp[]={0xfc,0x60,0xda,0xf2,0x66,0xb6,0xbe,0xe0,0xfe,0xf6,
0xee,0x3e,0x9c,0x7a,0x9e,0x8e};

void main()
{
    int counter=0,d1=50;
    initUSART();
    while(1)
    {
        delay(d1);
        display(counter);
        ++counter;
        if(counter>15) counter=0;
    }//end of while
} //end of main

void initUSART()
{
    TRISC=0x3f;
    SPBRG=0x19;
    TXSTA=0x90;//syn or asyn
    RCSTA=0x80;
    TXEN=1;
} //end of initUSART

void display(int counter)
{
    TXREG=disp[counter];
}

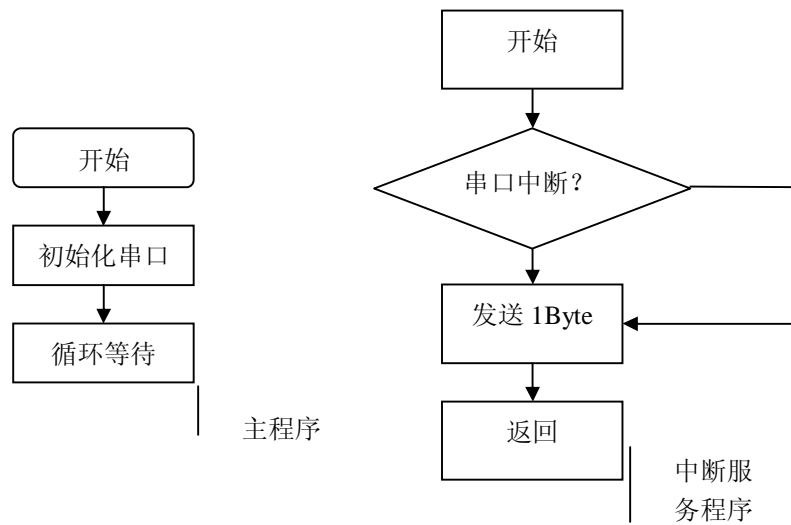
void delay(int d1)
{
    int i,j;
    for(i=0;i<d1;++i)
        for(j=0;j<353;++j);
} //end of delay

```

B、异步

程序规划：

考虑到程序的简单性，且要学习目的单一。所以主程序不做任何工作，循环等待串口中断。串口中断服务程序发送 1byte 到串口，此处对中断的处理很简略，实际上要进行现场保护等诸多工作。流程图如下：



程序为:

```

;Fosc = 4M hz
;Baud = 9.6K
LIST P=16F877A
#include P16F877A.INC

;-----
;reset entry
;-----
    org 0x000
    goto Main
;-----
;interrupt service function
;-----
    org 0x004
;protect w,status,pc ...
    btfss PIR1,TXIF ;
    goto OtherIntChk ;TXIF=0,not usart interrupt

    movlw 0xda
    movwf TXREG
    goto EndOfInt

OtherIntChk
;nop
EndOfInt
;resume w,status,pc ...

```

```

    retfie
Main
    bsf      STATUS,RPO
    movlw   0x19
    movwf   SPBRG
    bcf      TXSTA,SYNC
    bsf      TXSTA,BRGH    ; Baud Rate Mode : High Speed
    bsf      PIE1,TXIE     ; Enable transmit interrupts
    bsf      TXSTA,TXEN
    bcf      STATUS,RPO

;-----
    bsf      RCSTA,SPEN

;-----

    clrf   INTCON
    bsf   INTCON,PEIE
    bsf   INTCON,GIE

wait
    goto  wait

end

```

注：串口初始化的顺序参考自 Data Sheet:

When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

初始化程序也可参考 Data Sheet

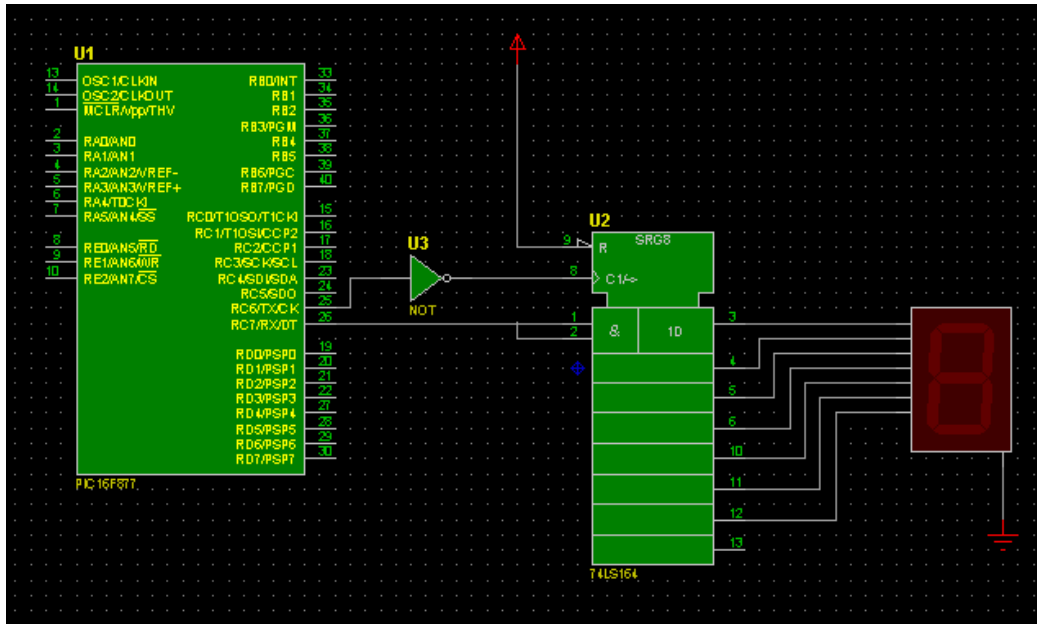
查询法可参照同步程序编写，差别仅在于 SYNC 是置位还是清零。

Step3:

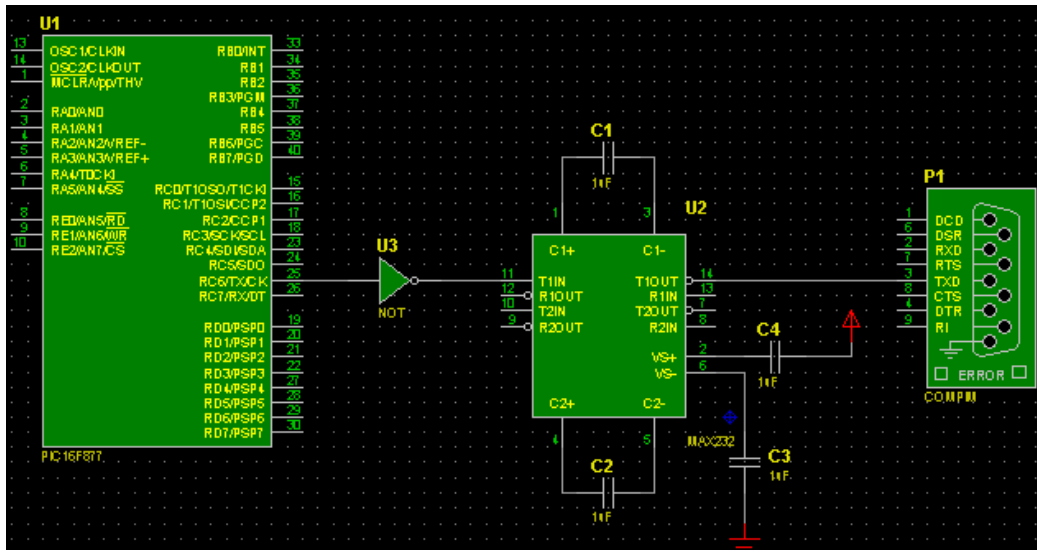
软件仿真，由于 Cuter 的电脑是没有串口的笔记本，所以 Cuter 就参考 EDN 上前辈们的调试经验，利用虚拟串口进行仿真。所参考的资料为：再次感谢！

仿真图为：

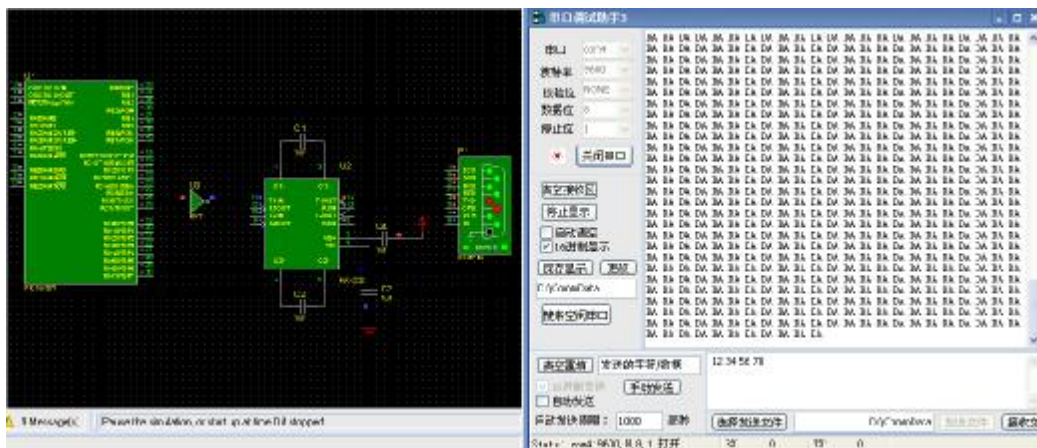
同步：



仿真结果：略，现象是数码管循环显示 0~9。
异步：



仿真结果：



Step4:

硬件测试。电路用万用板焊的，测试成功。Cuter 是穷人一个，没有相机，就没法上图了。

Step5:

说一下测试时候碰到的问题吧：

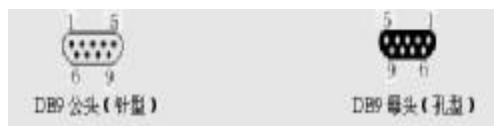
Cuter 把整个项目分两个模块，一个是 PIC16F877A 最小系统，一个是串口模块。

I 最小系统

最小系统仅将单片机的 40 个引脚用排针扩展出来，辅以震荡电路和按键复位电路，不做任何其它工作。

I 串口模块

Cuter 在制作串口模块的时候，忽略了一个细节：DB9 串口引脚定义问题，忘记了公母头之分，只是想当然地认为只有一种定义，最后调试出问题的时候才注意到这个细节，所幸的是，Cuter 用的 DB9 母头，用的也是母头的引脚定义。下次要避免此类错误。DB9 串口引脚定义为：

**1. RS-232 端 (DB9 母头/孔型) 引脚定义**

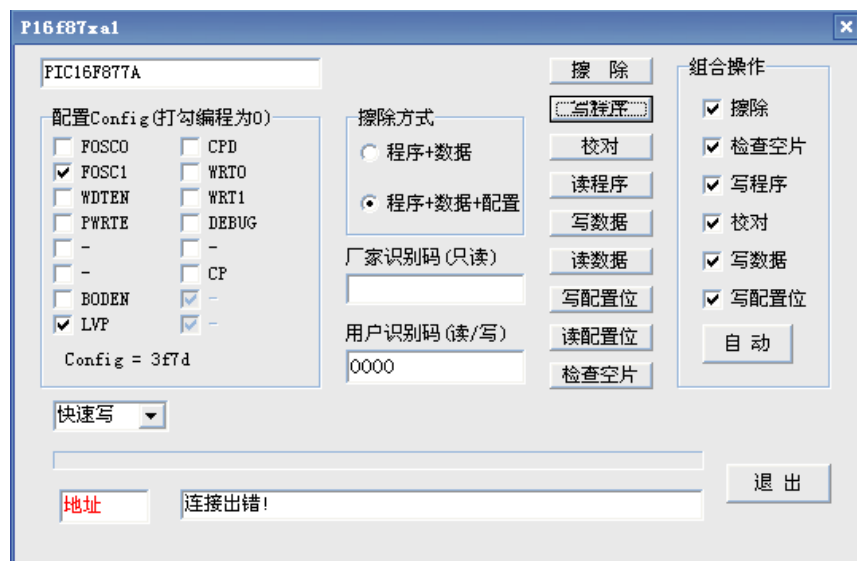
引脚序号	2	3	5	1、4、6	7、8
信号定义	TXD	RXD	地	内部相连	内部相连

2. RS-232 端 (DB9 公头/针型) 引脚定义

引脚序号	2	3	5	1、4、6	7、8
信号定义	RXD	TXD	地	内部相连	内部相连

除了这个细节呢，还有一个就是程序的烧写问题：

PIC16F877A 是低电压编程的，因此，在使用 top 系列编程器编程时，需要在配置 Configuration 一栏下作如下设置，即选中 Fosc1 和 LVP。



刚开始的时候，没注意到这个问题，单片机总是不工作，捣鼓了很长时间。