

习题 7

7-5 解:

帧传输速率钟 = 比特率 ÷ 帧的位数 = $19200 \div 10 = 1920$ (帧/秒)

8 位数据传输时间就是一帧的传输时间。即

$$t = 1 \div (19200 \div 10) = 0.52\text{mS}$$

7-6 解:

要求发送用查询方式, 接收用中断方式。(1) 用汇编语言编写在方式 1, 2, 3 下, 51 机串口发送和接收一帧数据的程序。设发送和接收的数据都在 R5 中。注意: 设系统 focs = 11.0592MHz, 在波特率可变方式下, 波特率取 9600 (Baud), 一律不用 PCON 加速。(2) 用 C 语言程序完成 (1) 的任务。

解 (1):

方式 1 发送

不用 PCON 加速时, 定时器 T1 用方式 2, 查表 5-6 或计算, T1 常数为 FDH

```
ORG      0000H
AJMP     MAIN
ORG      0050H
MAIN:    MOV     SP, #60H
          MOV     SCON, #40H
          MOV     PCON, #00H
          MOV     TMOD, #20H
          MOV     TL1, #0FDH
          MOV     TH1, #0FDH
          SETB    TR1                      ; 启动波特率
          MOV     SBUF, R5
WAIT:    JBC     TI, STOP
          SJMP     WAIT
STOP:    SJMP     $                      ; 动态停机
END
```

方式 1 接收程序为

```
ORG      0000H
AJMP     MAIN
ORG      0023H
LJMP     SERINT
ORG      0050H
MAIN:    MOV     SP, #60H
          MOV     SCON, #50H
          MOV     PCON, #00H
          MOV     TMOD, #20H
          MOV     TL1, #0FDH
          MOV     TH1, #0FDH
          SETB    ES                      ; 允许串口中断
          SETB    EA                      ; 开放总中断
          SETB    TR1                      ; 启动波特率
          SJMP     $
```

```

SERINT:   CLR      RI
          MOV      R5, SBUF
          RETI      ; 中断返回
          END

```

方式 2 发送程序:

不用计算定时器 T1 常数。

```

          ORG      0000H
          AJMP     MAIN
          ORG      0050H
MAIN:     MOV      SP, #60H
          MOV      SCON, #80H      ; SM2=0
          MOV      PCON, #00H
          MOV      SBUF, R5
WAIT:     JBC      TI, STOP
          SJMP     WAIT
STOP:     SJMP     $      ; 动态停机
          END

```

方式 2 接收程序为:

```

          ORG      0000H
          AJMP     MAIN
          ORG      0023H
          LJMP     SERINT
          ORG      0050H
MAIN:     MOV      SP, #60H
          MOV      SCON, #90H      ; SM2=0
          MOV      PCON, #00H
          SETB     ES      ; 允许串口中断
          SETB     EA      ; 开放总中断
          SJMP     $
SERINT:   CLR      RI
          MOV      R5, SBUF
          RETI      ; 中断返回
          END

```

方式 3 发送程序

不用 PCON 加速时，定时器 T1 用方式 2，查表 5-6 或计算，T1 常数为 FDH

```

          ORG      0000H
          AJMP     MAIN
          ORG      0050H
MAIN:     MOV      SP, #60H
          MOV      SCON, #0C0H      ; SM2=0
          MOV      PCON, #00H
          MOV      TMOD, #20H
          MOV      TL1, #0FDH

```

```

MOV TH1, #0FDH
SETB TR1 ; 启动波特率
MOV SBUF, R5
WAIT: JBC TI, STOP
SJMP WAIT
STOP: SJMP $ ; 动态停机
END

```

方式 3 接收程序为:

```

ORG 0000H
AJMP MAIN
ORG 0023H
LJMP SERINT
ORG 0050H
MAIN: MOV SP, #60H
MOV SCON, #0D0H ; SM2=0
MOV PCON, #00H
MOV TMOD, #20H
MOV TL1, #0FDH
MOV TH1, #0FDH
SETB ES ; 允许串口中断
SETB EA ; 开放总中断
SETB TR1 ; 启动波特率
SJMP $
SERINT: CLR RI
MOV R5, SBUF
RETI ; 中断返回
END

```

解 (2)

方式 1 发送

```

#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
Main()
{
    TMOD=0x20;
    TL1=0xfd;
    TH1=0xfd;
    SCON=0x40; //方式 1 不允许接收, SM2=0
    PCON = 0x00; //PCON = 0 为波特率 9600
    TR1=1;
    SBUF = R5;
    while(TI != 1);
    TI=0; //等待发送完成
}

```

```

    while (1);
}
方式 1 接收程序为:
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
bit emak, errmk; //数据、通信错误标志
void ssio(void)interrupt 4 //串口中断服务程序
{
    RI = 0;
    R5 = SBUF; //得到数据
}
Main()
{
    TMOD=0x20;
    TL1=0xfd;
    TH1=0xfd;
    SCON=0x50; //方式 1 允许接收
    PCON = 0x00; //PCON = 0 为波特率 9600
    TR1=1;
    ES = 1; //允许串口中断
    EA = 1; //开放总中断
    while (1); //等待数据到来
}

```

```

方式 2 发送程序:
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
Main()
{
    SCON=0x80; //方式 2 不允许接收, SM2=0
    PCON = 0x00; //PCON = 0 为波特率 9600
    SBUF = R5;
    while(TI! = 1);
    TI=0; //等待发送完成
    while (1);
}

```

```

方式 2 接收程序为:
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
bit emak, errmk; //数据、通信错误标志
void ssio(void)interrupt 4 //串口中断服务程序

```

```

{
    RI = 0;
    R5 = SBUF; //得到数据
}
Main()
{
    SCON=0x90; //方式 2 允许接收，SM2=0
    PCON = 0x00; //PCON = 0 为波特率 9600
    ES = 1; //允许串口中断
    EA = 1; //开放总中断
    while (1); //等待数据到来
}

```

方式 3 发送程序:

```

#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
Main()
{
    TMOD=0x20;
    TL1=0xfd;
    TH1=0xfd;
    SCON=0xc0; //方式 3 不允许接收，SM2=0
    PCON = 0x00; //PCON = 0 为波特率 9600
    TR1=1;
    SBUF = R5;
    while(TI! = 1);
    TI=0; //等待发送完成
    while (1);
}

```

方式 3 接收程序为:

```

#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar idata R5;
bit emak, errmk; //数据、通信错误标志
void ssio(void)interrupt 4 //串口中断服务程序
{
    RI = 0;
    R5 = SBUF; //得到数据
}
Main()
{
    TMOD=0x20;
    TL1=0xfd;
    TH1=0xfd;
}

```

```

SCON=0xd0;           //方式 3 允许接收
PCON = 0x00;         //PCON = 0 为波特率 9600
TR1=1;
ES = 1;              //允许串口中断
EA = 1;              //开放总中断
while (1);           //等待数据到来
}

```

7-7

答：单片机通信的本质是为了传送数据或命令等信息，是直接发送原形数据还是用编码发送，只是手段或方式。用不同的手段或方式传送数据的效率和益处各不相同，依通信系统的任务和性质决定。如 PC 机通信数据方式默认采用 ASC II 码，而单片机之间的通信常用 16 进制原码，有时也用 ASC II 码。本题中，单片机 A 要先将 5AH 中的“5”和“A”转换成 ASC II 码“35H”和“41H”，再分两次将它们发至单片机 B；单片机 B 在收到这两个字节数据后，要将 ASC II 码“35H”和“41H”还原为 5AH 才能理解单片机 A 发的数据的大小或含义。

7-1 训练题

解 方式 2 波特率固定为 $Baud = \frac{2^{SMOD}}{64} \times f_{osc}$

$$Baud = \frac{2^{SMOD}}{64} \times f_{osc} = \frac{2}{64} \times 6 \times 10^6 = 187.5 KHz$$

在串口方式 2 下，若要达到波特率匹配，通信双方的 fosc 要么相等，要么相对成 2 的倍数。

7-2 训练题

解：发送方子程序增加的部分用加粗字表示。

```

TRT:      MOV      TMOD, #20H          ; 波特率设置
          MOV      TL1, #0FDH
          MOV      TH1, #0FDH
          SETB     TR1                  ; 启动波特率发生器工作
          MOV      SCON, #50H          ; 方式 1, REN=1
          MOV      PCON, #00H          ; 串口初始化
RPT:      MOV      DPTR, #4000H
          MOV      R6, #N               ; 数据长度
          MOV      R5, #00H
          MOV      A, R6
          MOV      SBUF, A
          ADD      A, R5
          MOV      R5, A                ; 形成累加和送 R5
ML0:      JBC      TI, ML1
          SJMP     ML0
ML1:      MOV      SBUF, R5             ; 发累加和后清零
          MOV      R5, #00H
ML2:      JBC      TI, ML3
          SJMP     ML2
ML3:      JBC      RI, ML4              ; 等 B 机回答
          SJMP     ML3

```

ML4:	MOV	A, SBUF	
	JZ	ML5	; 正确转 ML5
	INC	CWCS	; 错误次数加 1, CWCS 在主程序中定义
	MOV	A, CWCS	
	CLR	C	
	SUBB	A, #03H	; 主程序中检查 CWCS 的值, 判断通信情况
	JNC	ML11	
	LJMP	PRT	; 不够 3 次不正确, 重发
ML5:	MOVB	A, @DPTR	; 读数据写入 BUF
	MOV	SBUF, A	
	ADD	A, R5	; 形成累加和
	MOV	R5, A	
	INC	DPTR	
ML6:	JBC	TI, ML7	; 等待发完一字节
	SJMP	ML6	
ML7:	DJNZ	R6, ML5	; 发完否?
	MOV	SBUF, R5	; 完成, 发累加和
	MOV	R5, #00H	
ML8:	JBC	TI, ML9	
	SJMP	ML8	
ML9:	JBC	RI, ML10	; 等 B 机回答
	SJMP	ML9	
ML10:	MOV	A, SBUF	
	JZ	ML11	; 正确返回
	INC	CWCS	; 错误次数加 1, CWCS 在主程序中定义
	MOV	A, CWCS	
	CLR	C	
	SUBB	A, #03H	
	JNC	ML11	; 超过 3 次不正确, 返回, 由主程序处理
	LJMP	RPT	; 不够 3 次不正确, 重发
ML11:	RET		

7-3 训练题

解: 发送方子程序增加的部分用加粗字表示。

```
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar xdata dsmark , R4, R5, R6, revdata[260], ERR_times;
bit emak, errmk; //数据、通信错误标志
void ssio(void)interrupt 4 //串口中断服务程序
{
    RI = 0;
    switch (dsmark)
    {
        case 0: R6 = SBUF; //得到第一个数据
                R5 = R5+SBUF; //得到校验和
```

```

        dsmark = 1;
        break;
case 1: if (SBUF == R5)                //校验正常， dsmark = 1
        {   dsmark = 2; SBUF = 0x00;   }
        else
        {   dsmark = 0; R6 = 0;
            SBUF = 0xff;
            ERR_times++;
        }
        if (ERR_times > 3)
            break;
        R5 = 0;
        while(TI! = 1);                //发应答
        TI=0;
        break;
case 2: revdata[R4] = SBUF;            //进入收数据过程
        R5 += SBUF; R4++; R6--;
        if(R6 == 0)
        {   while(RI! = 1);            //收校验和
            RI = 0;
            if (R5 != SBUF)
            {   errmk = 1; SBUF = 0xff; ERR_times++; } //校验和错误，置错误标志
            else {SBUF = 0; emak = 1; }   //通信成功
            while(TI! = 1);
            TI=0;                        //发应答
        }
        break;
    }
}
Main()
{   TMOD=0x20;
    TL1=0xfd;
    TH1=0xfd;
    PCON = 0x00;                        //PCON = 0 为波特率 9600
    SCON=0x50;                          TR1=1;
    ES=1; EA=1; dsmark=0; emak=0;
    ERR_times = 0;
    R6 = 0xff; R4 = 0; R5 = 0;
    If(ERR_times>3)
    {
        ERR_times = 0;
        ES = 0;
        while(1);
    }
    while (R6 != 0);
    {if (emak)                          //emak = 1; 通信成功

```



```

        while(1);
    else
    {      emak = 0;   dsmark = 0;
          R6 = 0xff; R4 = 0; R5 = 0;           //重新初始化
    }
}

```

7-4

解 定时 2ms 可用软件，也可用硬件实现。

主机用硬件定时：选用定时器 T0。主机参考程序如下：

```

        ORG      0000H
        SJMP     START
        ORG      000BH
        SETB     F0
        CLR      TR0
        MOV      TH0, #0F8H
        MOV      TL0, #0CDH
        RETI                      ; 中断服务程序少于等于 8 字节
        ORG      0040H
START:   MOV      SP, #60H
        MOV      TMOD, #21H
        MOV      TH0, #0F8H
        MOV      TL0, #0CDH      ; 2ms 定时初值
        MOV      TH1, #0FDH
        MOV      TL1, #0FDH
        MOV      PCON, #00H
        MOV      SCON, #0C0H    ; 方式 3
        SETB     REN
        SETB     TR1
        SETB     TB8            ; 准备发(地址)命令
        MOV      SBUF, B        ; 被呼叫的从机号存于 B 中
WAIT1:   JBC      TI, CON1      ; 等待发送完成
        SJMP     WAIT1
        SETB     TR0            ; 开始 2ms 定时
CON1:    JBC      RI, CON4      ; 等待从机应答
        JBC      F0, CON3      ; 超时，结束本次通信过程
        SJMP     CON1
CON4:    MOV      A, SBUF
        XRL      A, B
        JZ       CON5
        SJMP     CON3          ; 通信错误，结束本次通信过程
CON5:    CLR      TB8          ; 发送命令

```

```

MOV      A, R5                ; 命令存于 R5 中, 00H 为写 01H 为读
MOV      SBUF, A
WAIT2:   JBC      TI, CON2
         SJMP     WAIT2
CON2:    CJNE     A, #00H, READ ; 转读命令处理
         MOV      A, R6        ; 要发送的数据存于 R6 中
         MOV      SBUF, A
WAIT3:   JBC      TI, CON3
         SJMP     WAIT3
CON3:    SJMP     STOP
READ:    JBC      RI, STOP1
         SJMP     READ
STOP1:   MOV      R7, SBUF     ; 接收从机的数据存于 R7 中
STOP:    .....              ; 待命
         SJMP     STOP
END

```

从机用软件定时，参考程序如下：

```

ORG      0000H
SJMP     START
ORG      0023H
LJMP     SCOM
ORG      0050H
START:   MOV      SP, #60H
         MOV      R0, #01H    ; R0 为从机号, 设为 1 号
         CLR      F0
         MOV      TMOD, #20H
         MOV      TH1, #0FDH
         MOV      TL1, #0FDH
         MOV      PCON, #00H
         MOV      SCON, #0E0H ; 方式 3, SM2=1(地址)
         SETB     REN
         SETB     ES
         SETB     TR1
         SETB     EA
CHECK:   JBC      F0, STOP
         SJMP     CHECK
STOP:    SJMP     CHECK        ; 调试程序用, 再通信多次
SCOM:    CLR      RI
         CLR      EA
         MOV      A, SBUF
         SUBB     A, R0
         JNZ      CON4        ; 不是本机, 退出中断

```

	CLR	SM2	; 是本机，准备接收命令
TRANS:	MOV	A, R0	; 本机地址号在 R0 中
	MOV	SBUF, A	
TRAN1:	JBC	TI, WAIT	
	SJMP	TRAN1	
	MOV	R7, #0E6H	; 11MHz 下 2ms 延时的循环数
WAIT:	JBC	RI, CON2	
	MUL	AB	; 没有实际意义，只为延时 4 个机器周期
	DEC	R7	
	CJNE	R7, #00H, WAIT	
	SJMP	CON4	; 超时，中断通信过程，直接返回
CON2:	MOV	A, SBUF	
	JNZ	READ	; 不是写命令，转读处理
WAIT3:	JBC	RI, CON3	
	SJMP	WAIT3	
CON3:	MOV	R1, SBUF	; (R1) 为从机发给主机的内容
	SJMP	CON4	
READ:	MOV	SBUF, R1	
WAIT5:	JBC	TI, CON5	
	SJMP	WAIT5	
CON5:	SETB	F0	
CON4:	SETB	EA	
	SETB	SM2	
	RETI		
	END		

刘焕成 2010 年 8 月 18 日修改
2011 年 2 月 21 日修改 2
2011 年 03 月 29 日修改 3
2011 年 11 月 02 日修改