

程序范例：串口指定播放

```
/**
- 实现功能：实现芯片上电分别指定播放第一曲和第二曲，基本的程序供用户测试
- 日期      ：2013-05-06
- 运行环境：STC 晶振：11.0592M 波特率:9600
- 备注      ：在普中科技的 51 开发板上调试 OK --- STC89C516RD+
1、该测试程序必须是模块或者芯片方案中有设备在线，譬如 U 盘、TF 卡、FLASH
**/

#include "REG52.h"

#define COMM_BAUD_RATE 9600 //串口波特率
#define OSC_FREQ 11059200 //运行晶振：11.05926MHZ
static INT8U Send_buf[10] = {0};

void Delay_Ms(INT32U z)
{
    INT32U x=0, y=0;
    for(x=110; x>0; x--)
        for(y=z; y>0; y--);
}

/**
- 功能描述： 串口 1 初始化
- 注：      设置为 9600 波特率
**/

void Serial_init(void)
{
    TMOD = 0x20; // 设置 T1 为波特率发生器
    SCON = 0x50; // 0101,0000 8 位数据位, 无奇偶校验
    PCON = 0x00; //PCON=0;
    TH1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12); //设置为 9600 波特率
    TL1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);
    TR1 = 1; //定时器 1 打开
    REN = 1; //串口 1 接收使能
    ES = 1; //串口 1 中断使能
}

void Uart_PutByte(INT8U ch)
{
    SBUF = ch;
    while(!TI){;}
    TI = 0;
}

/**
- 功能描述： 串口向外发送命令[包括控制和查询]
- 参数说明： CMD:表示控制指令，请查阅指令表，还包括查询的相关指令
              feedback:是否需要应答[0:不需要应答，1:需要应答]
              data:传送的参数
**/

void SendCmd(INT8U len)
{
    INT8U i = 0;
```

```

    Uart_PutByte(0x7E); //起始
    for(i=0; i<len; i++) //数据
    {
        Uart_PutByte(Send_buf[i]);
    }
    Uart_PutByte(0xEF); //结束
}

```

/******

- 功能描述：求和校验

- 和校验的思路如下：

发送的指令，去掉起始和结束。将中间的 6 个字节进行累加，最后取反码。接收端就将接收到的一帧数据，去掉起始和结束。将中间的数据累加，再加上接收到的校验字节。刚好为 0.这样就代表接收到的数据完全正确。

*****/

```

void DoSum( INT8U *Str, INT8U len)
{
    INT16U xorsum = 0;
    INT8U i;
    for(i=0; i<len; i++)
    {
        xorsum = xorsum + Str[i];
    }
    xorsum = 0 -xorsum;
    *(Str+i) = (INT8U)(xorsum >>8);
    *(Str+i+1) = (INT8U)(xorsum & 0x00ff);
}

```

```

void Uart_SendCMD(INT8U CMD ,INT8U feedback , INT16U dat)

```

```

{
    Send_buf[0] = 0xff;    //保留字节
    Send_buf[1] = 0x06;    //长度
    Send_buf[2] = CMD;     //控制指令
    Send_buf[3] = feedback; //是否需要反馈
    Send_buf[4] = (INT8U)(dat >> 8); //datah
    Send_buf[5] = (INT8U)(dat);    //datal
    DoSum(&Send_buf[0],6);        //校验
    SendCmd(8);                    //发送此帧数据
}

```

```

void main()

```

```

{
    Serial_init(); //串口寄存器的初始化设置
    Uart_SendCMD(0x03 , 0 , 0x01) ;//播放第一首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x02) ;//播放第二首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x04) ;//播放第四首
    while(1) ;
}

```

AVR 例程

```
#ifndef _24SS5200DRVS_H_
//如果本程序的头文件还没有补加载
#define _24SS5200DRVS_H_
//加载本程序的头文件
unsigned char MP3_BEFF[11];
//=====
//=====MP3 控制命令串 10 位
// 这 是 24SS5200 驱 动 程 序 部 分
//=====
void MP3_COMMAND(unsigned char CMD,unsigned int InputData)
{
    unsigned char i;
    unsigned int CHECK_TEMP=0; //16 位校验数据
    Delay10ms(1); //命令串之间保证有一个时间间隔
    MP3_BEFF[0]=0x7E; //协议起始
    MP3_BEFF[1]=0xFF; //协议版本 VER/默认为[FF]
    MP3_BEFF[2]=0x06; //数据包长度 LEN
    MP3_BEFF[3]=CMD; //命令字节:如播放/暂停等
    MP3_BEFF[4]=0x00; //命令回复信息:1:要求回复,2:不要回复
    MP3_BEFF[5]=InputData>>8; //指定参数高位
    MP3_BEFF[6]=InputData; //指定参数低位
    for(i=1;i<7;i++){CHECK_TEMP = CHECK_TEMP+MP3_BEFF[i];} //数据包内容相加
    CHECK_TEMP = 0-CHECK_TEMP; //校验数据取反
    MP3_BEFF[7]=CHECK_TEMP>>8; //校验和高位
    MP3_BEFF[8]=CHECK_TEMP; //校验和低位
    MP3_BEFF[9]=0xEF; //协议结束位
    for(i=0;i<10;i++) //发送数据包
    {
        COM0_OUT_BYTE(MP3_BEFF[i]); //串口单字节发送数据
    }
}
#endif
//=====
//=====驱动结束
```

