

```

/*
(("//项 目 名:MP3 芯片参考代码\n");
(("//功能要求:按键操作播放 MP3 功能和音量设置");
(("//硬件配置:STM32 最小系统和串口通信");
(("//1) MCU 型号:STM32F103");
(("//2) 外部振荡频率为:8M");
(("//软件版本:UART1: ");
(("//公 司 名: 深圳市百为电子科技有限公司\n");
(("//作 者: lanq");
(("// 硬件仿真: Pro");
(("// 软件文件: Code\n");
(("// 开发环境: keil5\n")
(("//技术支持: E-mail:balway@163.com\n");
(("//修改记录:
*/

#include "stm32f10x.h"
#include "USART1.h"
//位带操作,实现 51 类似的 GPIO 控制功能
//具体实现思想,参考<<CM3 权威指南>>第五章(87 页~92 页).
//IO 口操作宏定义
#define BITBAND(addr, bitnum) ((addr & 0xF0000000)+0x20000000+((addr & 0xFFFFF)<<5)+(bitnum<<2))
#define MEM_ADDR(addr) *((volatile unsigned long *) (addr))
#define BIT_ADDR(addr, bitnum) MEM_ADDR(BITBAND(addr, bitnum))
//IO 口地址映射
#define GPIOA_ODR_Addr (GPIOA_BASE+12) //0x4001080C
#define GPIOB_ODR_Addr (GPIOB_BASE+12) //0x40010C0C

#define GPIOA_IDR_Addr (GPIOA_BASE+8) //0x40010808
#define GPIOB_IDR_Addr (GPIOB_BASE+8) //0x40010C08

#define PAout(n) BIT_ADDR(GPIOA_ODR_Addr,n) //输出
#define PAin(n) BIT_ADDR(GPIOA_IDR_Addr,n) //输入

#define PBout(n) BIT_ADDR(GPIOB_ODR_Addr,n) //输出
#define PBin(n) BIT_ADDR(GPIOB_IDR_Addr,n) //输入

#define LED_H GPIO_SetBits(GPIOA, GPIO_Pin_1) // PC1 高电平
#define LED_L GPIO_ResetBits(GPIOA, GPIO_Pin_1) // PC1 低电平

#define KEY_1 PBin(7) //按键使能
#define KEY_2 PBin(10) //按键使能
#define KEY_3 PBin(13) //按键使能
#define KEY_4 PBin(15) //按键使能

```

```

#define KEY_5 PBin(12)          //按键使能
#define KEY_6 PBin(16)          //按键使能

u8 Check = 0;
u8 Play[]={0x7E,0x03,0x01,0x02,0xEF};//播放指令
u8 Call[]={0x7E,0x03,0x0E,0x0D,0xEF};//停止指令

u8 Next[]={0x7E,0x03,0x03,0x00,0xEF};//下一曲指令
u8 Last[]={0x7E,0x03,0x04,0x07,0xEF};//上一曲指令

u8 Add[]={0x7E,0x03,0x05,0x06,0xEF};          //音量+指令
u8 Sub[]={0x7E,0x03,0x06,0x05,0xEF};          //音量减指令

u8 BCCheck(u8 *nDataBuf, u8 nLen)//异或检验函数
{
    u8 nSum = 0;
    u8 i      = 0;
    for(i=0; i<nLen; i++)
    {
        nSum ^= nDataBuf[i];//异或检验函数
    }
    return nSum;
}

void Key_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB, ENABLE);
//按键检测
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_10| GPIO_Pin_13| GPIO_Pin_15;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPD;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
}

void Led_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA, ENABLE);
//按键检测
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;

```

```

    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
}
u8 KEY_Scan(u8 mode)
{
    static u8 key_up=1;//按键按松开标志
    if(mode)key_up=1;  //支持连按
    if(key_up&&(KEY_1==0|KEY_2==0|KEY_3==0|KEY_4==0|KEY_5==0|KEY_6==0))
    {
        key_up=0;
        if(KEY_1==0) return '1';
        else if(KEY_2==0)return '2';
        else if(KEY_3==0)return '3';
        else if(KEY_4==0)return '4';
        else if(KEY_5==0)return '5';
        else if(KEY_6==0)return '6';
    }
    if(KEY_1==1&&KEY_2==1&&KEY_3==1&&KEY_4==1&&KEY_5==1&&KEY_6==1)key_up=1;
    return 0;// 无按键按下
}

int main(void)
{
    u8 t;
    NVIC_PriorityGroupConfig( NVIC_PriorityGroup_2);
    USART1_init(9600);          //初始化串口波特率 9600
    USART1_Send_String("123456789\r\n",11);
    while(1)
    {
        t = KEY_Scan(0);
        if(t)
        {
            switch(t)
            {
                case 1:USART1_Send_String(Play,sizeof(Play));break;
                case 2:USART1_Send_String(Call,sizeof(Call));break;
                case 3:USART1_Send_String(Next,sizeof(Next));break;
                case 4:USART1_Send_String>Last,sizeof>Last));break;
                case 5:USART1_Send_String>Add,sizeof>Add));break;
                case 6:USART1_Send_String(Sub,sizeof(Sub));break;
            }
        }
    }
}

```