

Mian.c

```
#include "stm32f10x.h"
#include "delay.h"
#include "relay.h"
#include "di_pb10.h"
#include "exti10.h"
/*-----主程序-----*/
int main(void)
{
    RELAY_Configure();
    di_Configure();
    EXTI10_Configure();
    delay_ms(100);
    RELAY1_Off();
    RELAY2_Off();
    while(1)
    {

    }
}
```

Usart1.c

```
#include "usart1.h"
#include "stdio.h"
uint8_t USART1_RX_Buffer[USART_RX_MAX] = { 0 }; //定义 1.USART1 接收缓存
uint8_t USART1_RX_Index = 0; //定义 2.USART1 接收数组下标
uint8_t USART1_RX_OverFlag = 0; //定义 3.USART1 接收完成标志位
/**
 * @简介： 将 C 库中 printf 重定向到 USART
 * @参数: ch-待发送字符，f-指定文件
 * @返回值: ch
 */
int fputc(int ch, FILE *f)
{
    USART_SendData(USART1, (u8) ch);
    while(!(USART_GetFlagStatus(USART1, USART_FLAG_TXE) == SET))
    {
    }
    return ch;
}
```

```

void USART1_Configure(void)
{
    /* 定义 GPIO 初始化结构体 */
    GPIO_InitTypeDef GPIO_InitStructure;
    /* 定义 USART 初始化结构体 */
    USART_InitTypeDef USART_InitStructure;
    NVIC_InitTypeDef NVIC_InitStructure;
    /* 打开 GPIOA、AFIO 和 USART1 时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA | RCC_APB2Periph_AFIO |
RCC_APB2Periph_USART1, ENABLE);
    /* 配置 PA9(USART_Tx)为复用推挽输出，IO 速度 50MHz */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    /* 完成配置 */
    GPIO_Init(GPIOA, &GPIO_InitStructure);
    /* 配置 PA10(USART1_Rx)为浮空输入 */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    /* 完成配置 */
    GPIO_Init(GPIOA, &GPIO_InitStructure);
    /* 配置 USART 波特率、数据位、停止位、奇偶校验、硬件流控制和模式 */
    USART_InitStructure.USART_BaudRate = 9600;//波特率 115200
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;//8 数据位
    USART_InitStructure.USART_StopBits = USART_StopBits_1;//1 停止位
    USART_InitStructure.USART_Parity = USART_Parity_No;//无奇偶校验
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;//
无硬件流控制
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;//接收和发送模
式
    /* 完成配置 */
    USART_Init(USART1, &USART_InitStructure);
    /* 使能 USART1 */
    USART_Cmd(USART1, ENABLE);
    USART_ITConfig(USART1, USART_IT_RXNE, ENABLE); //开启接收 RXNE 中断
    USART_ITConfig(USART1, USART_IT_IDLE, ENABLE); //开启接收 IDLE 中断
    NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn; //USART1 中断通道
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; //抢占优先级 1
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1; //子优先级 3
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //IRQ 通道使能
    NVIC_Init(&NVIC_InitStructure); //配置生效
}

void USART1_IRQHandler(void)

```

```

{
    uint8_t Res;
    /* 如果发生了接收中断 */
    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
    {
        //Res = USART1->DR; //寄存器方式读取数据
        Res = USART_ReceiveData(USART1); //库函数方式读取接收到的 1 个字节
        if(USART1_RX_Index >= USART_RX_MAX)
            USART1_RX_Index = 0; //防止下标越界
        USART1_RX_Buffer[USART1_RX_Index++] = Res;
        /* 清除接收中断标志位(注:也可以省略, 读 DR 自动清除) */
        USART_ClearFlag(USART1, USART_FLAG_RXNE);
    }
    if(USART_GetITStatus(USART1, USART_IT_IDLE) != RESET)
    {
        USART1_RX_OverFlag = 1; //接收完成标志位置 1
        USART_ClearFlag(USART1, USART_FLAG_IDLE);
        USART_ITConfig(USART1, USART_IT_IDLE, DISABLE); //关闭接收 IDLE 中断
    }
}
}

```

```

Usart1.h
#ifndef __USART1_H
#define __USART1_H
#include "stm32f10x.h"
#define USART_RX_MAX 255 //定义最大接收字节数为 255
extern uint8_t USART1_RX_Buffer[USART_RX_MAX]; //定义 1.USART1 接收缓存
extern uint8_t USART1_RX_Index; //定义 2.USART1 接收数组下标
extern uint8_t USART1_RX_OverFlag; //定义 3.USART1 接收完成标志位
void USART1_Configure(void);
#endif

```

Relay.c

```
#include "relay.h"
```

```
void RELAY_Configure(void)
```

```
{
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD|RCC_APB2Periph_GPIOC,ENABLE);
    GPIO_InitStructure.GPIO_Pin =GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;
    GPIO_Init(GPIOD,&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin =GPIO_Pin_12;
    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;
    GPIO_Init(GPIOC,&GPIO_InitStructure);
    GPIO_ResetBits(GPIOD,GPIO_Pin_2);
    GPIO_ResetBits(GPIOC,GPIO_Pin_12);
}
```

```
void RELAY1_On(void)
```

```
{
    GPIO_SetBits(GPIOC,GPIO_Pin_12);
}
```

```
void RELAY1_Off(void)
```

```
{
    GPIO_ResetBits(GPIOC,GPIO_Pin_12);
}
```

```
void RELAY2_On(void)
```

```
{
    GPIO_SetBits(GPIOD,GPIO_Pin_2);
}
```

```
void RELAY2_Off(void)
```

```
{
    GPIO_ResetBits(GPIOD,GPIO_Pin_2);
}
```

```

Relay.h
#ifndef _RELAY_H
#define _RELAY_H
#include "stm32f10x.h"
#include "delay.h"
void RELAY_Configure(void); //LED 引脚初始化
void RELAY1_On(void);
void RELAY1_Off(void);
void RELAY2_On(void);
void RELAY2_Off(void);
#endif

```

```

Exti10.c
#include "exti10.h"
#include "delay.h"
#include "relay.h"
#include "usart1.h"
void EXTI10_Configure(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;
    EXTI_InitTypeDef EXTI_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_AFIO,ENABLE);
    GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource10);
    GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource11);
    EXTI_InitStructure.EXTI_Line =EXTI_Line10 | EXTI_Line11;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger =EXTI_Trigger_Rising_Falling;
    EXTI_InitStructure.EXTI_LineCmd    =ENABLE;
    EXTI_Init(&EXTI_InitStructure);
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
    NVIC_InitStructure.NVIC_IRQChannel = EXTI15_10_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority =2;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority    =0;
    NVIC_InitStructure.NVIC_IRQChannelCmd    =ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}

void EXTI15_10_IRQHandler(void)
{
    if((EXTI_GetITStatus(EXTI_Line10)) || (EXTI_GetITStatus(EXTI_Line11)) != RESET) //确保是否

```

产生了 EXTI Line 中断

```
{
    if(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_10)==1)
    {
        RELAY1_On();
    }
    if(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_11)==1)
    {
        RELAY2_On();
    }

    if((GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_10)==0)&&(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_11)==0))
    {
        RELAY1_Off();
        RELAY2_Off();
    }
    EXTI_ClearITPendingBit(EXTI_Line10);    //清除中断标志位
    EXTI_ClearITPendingBit(EXTI_Line11);    //清除中断标志位
}
}
```

Exti10.h

```
#ifndef __EXTI10_H
#define __EXTI10_H
#include "stm32f10x.h"
void EXTI10_Configure(void);
#endif
```

Di_pb10.c

```
/*-----头文件包含-----*/
#include "di_pb10.h"
/** * @简介： 按键初始化 */

void di_Configure(void)
{
    /*定义 GPIO 初始化结构体*/
```

```

    GPIO_InitTypeDef GPIO_InitStructure;
    /*打开 GPIOB 时钟*/
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    /*配置 PB10 输入*/
    GPIO_InitStructure.GPIO_Pin=GPIO_Pin_10 | GPIO_Pin_11;
    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_IN_FLOATING;
    /*完成配置*/
    GPIO_Init(GPIOB,&GPIO_InitStructure);
}

```

```

Di_pb10.h
/*****宏定义防止重复包含*****/
#ifndef _DI_PB10_H
#define _DI_PB10_H
/*****头文件包含*****/
#include "stm32f10x.h"
/*****函数声明*****/
void di_Configure(void);
#endif

```