

STM32 HAL 多中断控制LED项目

本项目使用STM32CubeMX和Keil MDK-ARM开发环境，通过HAL库实现定时器中断、串口中断和外部中断控制LED闪烁功能。

功能特性

- 定时器中断控制LED周期性闪烁
- 串口中断接收命令控制LED状态
- 外部按键中断(PB2)控制LED状态切换
- 所有中断事件通过串口反馈状态信息

硬件要求

- STM32F4xx系列开发板
- LED连接在PF14\PE11\PG1引脚（查看原理图或者用户手册）
- 按键连接在PB2引脚(外部中断)（查看原理图或者用户手册）
- USB转串口模块连接USART6

开发环境配置

- 安装STM32CubeMX
- 安装Keil MDK-ARM
- 安装对应STM32系列的DFP包
- 安装ST-Link驱动(或其他调试器驱动)

CubeMX配置步骤

1. 系统核心配置

- 新建项目选择对应MCU型号
- 在"Pinout & Configuration"选项卡：
 - 配置系统时钟树，确保APB1时钟为90MHz
 - 启用调试接口(SWD)
 - 启用外部高速时钟（RCC）

2. GPIO配置

- PF14\PE11\PG1配置为GPIO_Output(LED)
- PB2配置为GPIO_EXTI2(按键中断) 记得在NVIC把中断打开

Group by peripherals

☒ GPIO

☒ RCC

☒ SYS

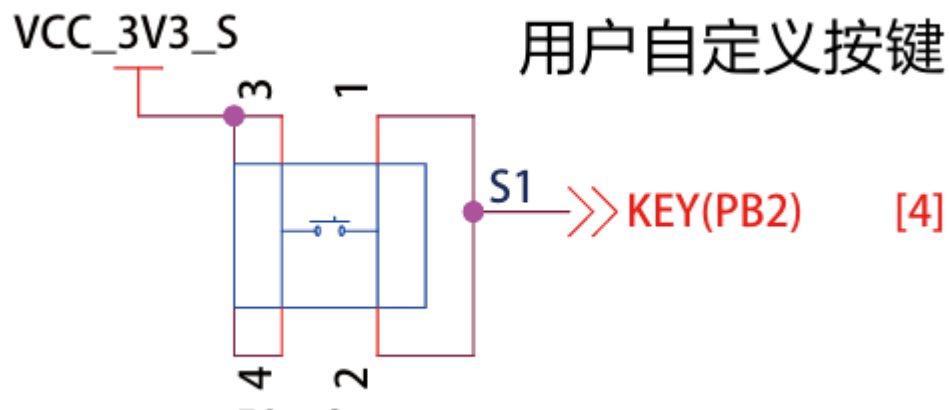
☒ USART

☒ NVIC

| NVIC Interrupt Table | Enabled | Preemption Priority | Sub Priority |
|----------------------|-------------------------------------|---------------------|--------------|
| EXTI line2 interrupt | <input checked="" type="checkbox"/> | 0 | 0 |

因为按键按下是高电平所以外部中断要上升沿触发

| | |
|-----------|--|
| GPIO mode | External Interrupt Mode with Rising edge trigger detection |
|-----------|--|



3. 定时器配置(TIM2)

1. 选择TIM2 (中断频率1s)
2. 参数设置:
 - Prescaler (PSC): 8999
 - Counter Mode: Up
 - Counter Period (ARR): 9999
 - 自动重载预装载: Enable
3. NVIC设置:
 - 启用TIM2全局中断

4. 串口配置(USART1)

1. 模式选择: Asynchronous
2. 基本参数:
 - Baud Rate: 115200
 - Word Length: 8 Bits
 - Stop Bits: 1

- Parity: None

3. NVIC设置：

- 启用USART6全局中断

5. 外部中断配置

1. 在NVIC配置中启用EXTI line2中断
2. 设置合适的中断优先级

6. 生成代码

1. 项目设置：
 - Toolchain: MDK-ARM V5
 - 勾选"Generate peripheral initialization as a pair of .c/.h files"
2. 点击"Generate Code"生成项目

代码实现

中断功能写成了独立的.c和.h文件在APP文件夹里；

main.c修改

```
/* 在Includes部分添加 */
#include "custom_interrupts.h"

/* 在main()函数中初始化后添加 */
Custom_Interrupts_Init();

/* 主循环可以保持简单 */
while (1)
{
    HAL_Delay(1000);
}
```