

✓ 外部中断



■ 参考资料:

《STM32F4开发指南-库函数版本》-第10章 外部中断实验

□ STM32F4xx官方资料:

《STM32F4xx中文参考手册》-第10章 中断和 事件

目录



1

外部中断概述

2

外部中断库函数设置

3

手把手用外部中断写按键实验

✓ 外部中断概述



- ◆ STM32F4的每个IO都可以作为外部中断输入。
- ◆ STM32F4的中断控制器支持22个外部中断/事件请求：

EXTI线0~15：对应外部IO口的输入中断。

EXTI线16：连接到PVD输出。

EXTI线17：连接到RTC闹钟事件。

EXTI线18：连接到USB OTG FS唤醒事件。

EXTI线19：连接到以太网唤醒事件。

EXTI线20：连接到USB OTG HS(在FS中配置)唤醒事件。

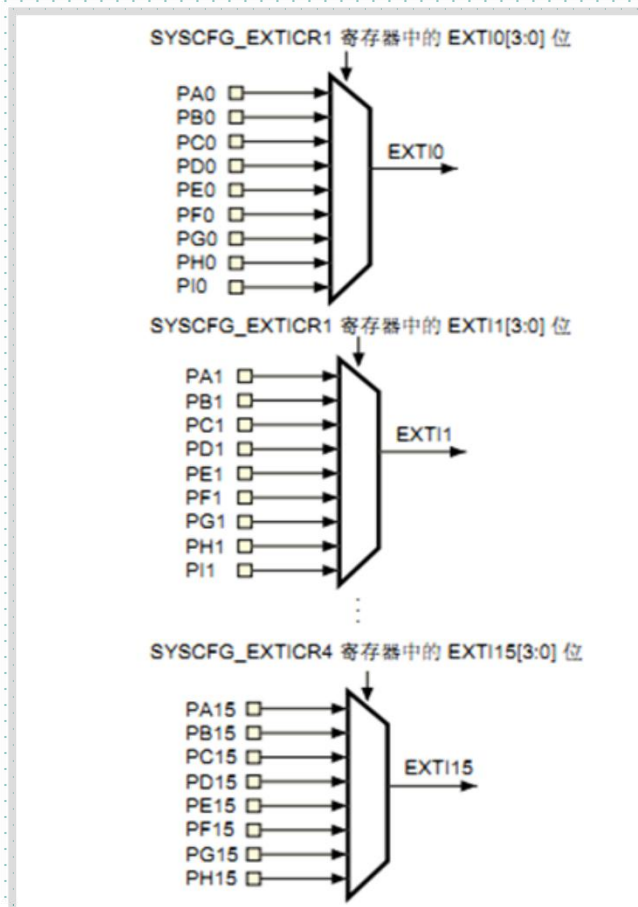
EXTI线21：连接到RTC入侵和时间戳事件。

EXTI线22：连接到RTC唤醒事件。

每个外部中断线可以独立的配置触发方式（上升沿，下降沿或者双边沿触发），触发/屏蔽，专用的状态位。

从上面可以看出，STM32F4供IO使用的中断线只有16个，但是STM32F4xx系列的IO口多达上百个，STM32F103ZGT6(112)，那么中断线怎么跟io口对应呢？

✓ 外部中断概述



GPIOx.0映射到EXTI0

GPIOx.1映射到EXTI1

...

GPIOx.15映射到EXTI15

✓ 外部中断概述



- 对于每个中断线，我们可以设置相应的触发方式（上升沿触发，下降沿触发，边沿触发）以及使能。

✓ 外部中断概述



- ◆是不是16个中断线就可以分配16个中断服务函数呢？
- ◆IO口外部中断在中断向量表中只分配了7个中断向量，也就是只能使用7个中断服务函数

位置	优先级	优先级类型	名称	说明	地址
7	14	可设置	EXTI1	EXTI线1中断	0x0000_005C
8	15	可设置	EXTI2	EXTI线2中断	0x0000_0060
9	16	可设置	EXTI3	EXTI线3中断	0x0000_0064
10	17	可设置	EXTI4	EXTI线4中断	0x0000_0068
23	30	可设置	EXTI9_5	EXTI线[9:5]中断	0x0000_009C
40	47	可设置	EXTI15_10	EXTI线[15:10]中断	0x0000_00E0

- ◆从表中可以看出，外部中断线5~9分配一个中断向量，共用一个服务函数
外部中断线10~15分配一个中断向量，共用一个中断服务函数。

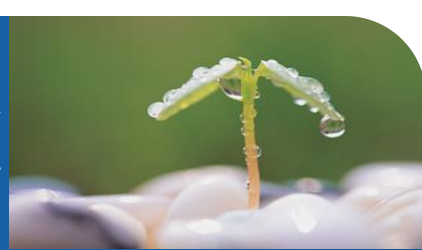
✓ 外部中断概述



■ 中断服务函数列表:

```
EXTI0_IRQHandler  
EXTI1_IRQHandler  
EXTI2_IRQHandler  
EXTI3_IRQHandler  
EXTI4_IRQHandler  
EXTI9_5_IRQHandler  
EXTI15_10_IRQHandler
```

✓ 2. 外部中断常用库函数



◆ 外部中断常用库函数

① `void SYSCFG_EXTILineConfig(uint8_t EXTI_PortSourceGPIOx, uint8_t EXTI_PinSourcex);`
//设置IO口与中断线的映射关系

exp: `SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOE, EXTI_PinSource2);`//区别M3

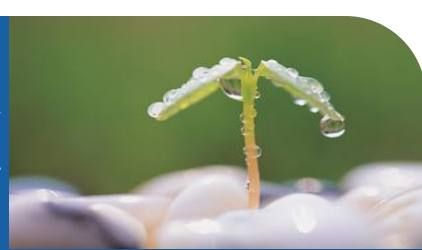
② `void EXTI_Init(EXTI_InitTypeDef* EXTI_InitStruct);`
//初始化中断线：触发方式等

③ `ITStatus EXTI_GetITStatus(uint32_t EXTI_Line);`
//判断中断线中断状态，是否发生

④ `void EXTI_ClearITPendingBit(uint32_t EXTI_Line);`
//清除中断线上的中断标志位

⑤ `RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);`//使能SYSCFG时钟
//这个函数非常重要，在使用外部中断的时候一定要先使能SYSCFG时钟

✓ 2. 外部中断常用库函数



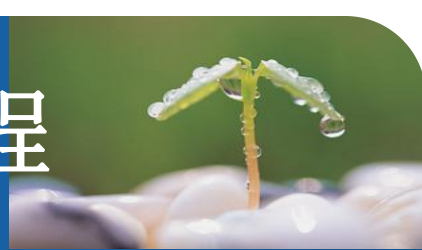
◆ EXTI_Init函数

void EXTI_Init(EXTI_InitTypeDef EXTI_InitStruct);*

```
typedef struct
{
    uint32_t EXTI_Line; //指定要配置的中断线
    EXTI_Mode_TypeDef EXTI_Mode; //模式：事件 OR 中断
    EXTI_Trigger_TypeDef EXTI_Trigger; //触发方式：上升沿/下降沿/双沿触发
    FunctionalState EXTI_LineCmd; //使能 OR 失能
}EXTI_InitTypeDef;
```

```
EXTI_InitStructure.EXTI_Line=EXTI_Line2;
EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
EXTI_InitStructure.EXTI_LineCmd = ENABLE;
EXTI_Init(&EXTI_InitStructure);
```

✓ 2. 外部中断一般配置过程



◆ 外部中断的一般配置步骤:

- ① 使能SYSCFG时钟:
RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
- ② 初始化IO口为输入。
GPIO_Init();
- ③ 设置IO口与中断线的映射关系。
void SYSCFG_EXTILineConfig();
- ④ 初始化线上中断，设置触发条件等。
EXTI_Init();
- ⑤ 配置中断分组（NVIC），并使能中断。
NVIC_Init();
- ⑥ 编写中断服务函数。
EXTIx_IRQHandler();
- ⑦ 清除中断标志位
EXTI_ClearITPendingBit();

✓ 按键硬件连接

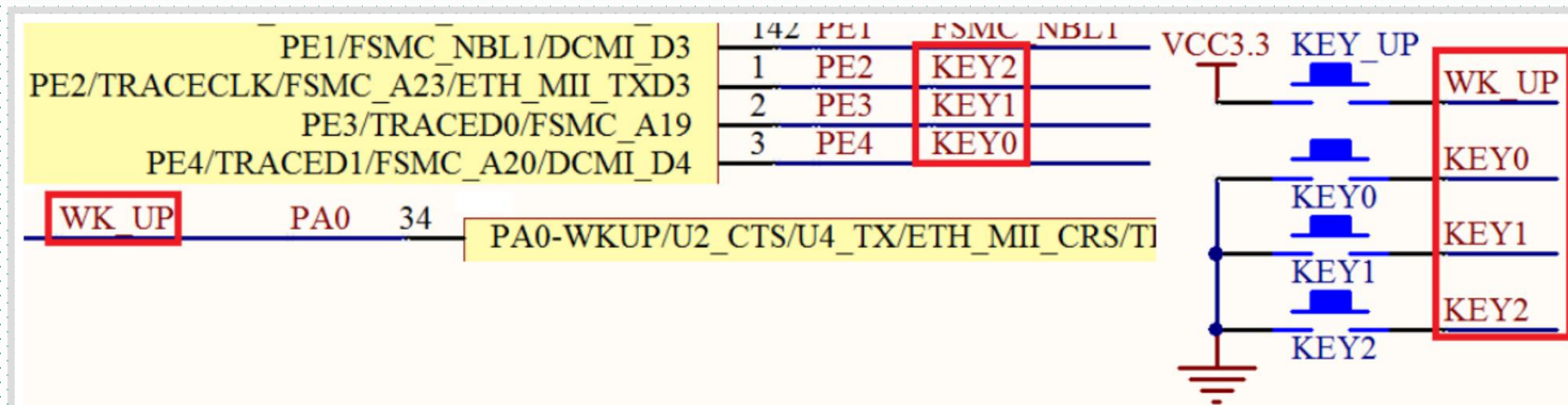


图 8.2.1 按键与 STM32F4 连接原理图

KEY0->PE4 上拉输入
KEY1->PE3 上拉输入
KEY2->PE2 上拉输入
WK_UP->PA0 下拉输入

✓ 3. 手把手写代码



手把手用外部中断知识写按键输入实验

GO!!