

## 一起学习 CC3200 系列教程之外部中断

阿汤哥

序:

能力有限, 英语不怎么好, 难免有错, 有问题请联系我,

QQ1519256298 [hytga@163.com](mailto:hytga@163.com)

Pdf 下载 <http://pan.baidu.com/s/1hqjWB56>

现在我们学习怎么设置外部中断,

外部中断设置步骤:

- 1、设置引脚为 GPIO (不介绍了)
- 2、使能时钟 (必须的啊)
- 3、设置引脚的电流强度, 上拉或者下拉等等 (不介绍了)
- 4、设置 GPIO 为输入模式 (不介绍了)
- 5、设置成边沿触发
- 6、编写中断处理函数: 1、判断哪个 GPIO 触发, 2、清除中断标志
- 7、使能中断触发

汗, 没看到优先级的设置, 真是奇怪,,,

CC3200 有 4 组 GPIO, 每组的 GPIO 共用一个中断处理函数, 既然有多个 GPIO 共用一个中断处理函数, 当然要在中断函数查询哪个 GPIO 触发中断的(可以查询中断状态寄存器),

现在着重介绍中断配置需要的库函数:

```
GPIOIntTypeSet(GPIOA1_BASE,GPIO_PIN_5,GPIO_RISING_EDGE);
```

配置 GPIO A1 组的 pin 5 设置成上升边沿触发中断, 类似的还有下拉,

```
GPIOIntRegister(GPIOA1_BASE,GPIO13_handle);
```

配置 GPIOA1 组的中断处理函数 GPIO13\_handle

```
GPIOIntEnable(GPIOA1_BASE,GPIO_INT_PIN_5);
```

使能 GPIOA1 的 pin5 触发中断

中断函数需要调用的函数

```
GPIOIntStatus(GPIOA1_BASE,true);
```

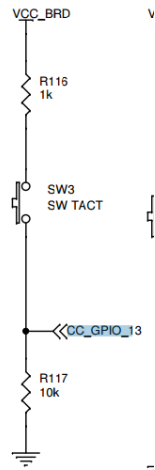
查询 GPIOA1 的中断状态寄存器: 注意查询的是组内的 GPIO 的中断状态。获得的值跟 GPIO\_INT\_PIN\_5 相与, 就可以获取是不是 pin 5 触发中断的。

```
GPIOIntClear(GPIOA1_BASE,GPIO_INT_PIN_5);
```

清除 GPIOA1 的 pin5 的中断标志

硬件图

GPIO13 SW TACT



软件：设置按键输入，上升沿触发中断，在中断处理函数读取 GPIO 的值，并打印，这里我们直接用以前的串口程序。

主函数

```

void main()
{
    char cString[MAX_STRING_LENGTH+1];
    char cCharacter;
    int iStringLength = 0;
    //
    // Initailizing the board
    //
    BoardInit();
    uart0_Init();
    //使能时钟
    PRMPeripheralClkEnable(PRCM_GPIOA1,PRCM_RUN_MODE_CLK);
    //设置引脚 4 为 GPIO 模式，标准模式，2ma
    PinModeSet(PIN_04,PIN_MODE_0);
    PinConfigSet(PIN_04,PIN_STRENGTH_2MA,PIN_TYPE_STD);
    //设置成输入，引脚 4 对应于 GPIOA1 的 pin5
    GPIODirModeSet(GPIOA1_BASE,GPIO_PIN_5,GPIO_DIR_MODE_IN);
    //设置成上升沿触发中断
    GPIOIntTypeSet(GPIOA1_BASE,GPIO_PIN_5,GPIO_RISING_EDGE);
    //设置中断处理函数，
    GPIOIntRegister(GPIOA1_BASE,GPIO13_handle);
    //使能中断
    GPIOIntEnable(GPIOA1_BASE,GPIO_INT_PIN_5);
    while(1) {
        delay(0xfffff);

        //my_printf("%d\r\n",(GPIOPinRead(GPIOA1_BASE,GPIO_PIN_5)&GPIO_PIN_5)?1:0);
    }
}
  
```

```

    }

}

中断处理函数
//中断处理函数
void GPIO13_handle(void) {
    int i,k,j;
    #if 1
    //获取经过屏蔽的中断状态
    i = GPIOIntStatus(GPIOA1_BASE,true);
    //判断是不是 GPIOA1 的 pin5 触发的中断的
    if(i&GPIO_INT_PIN_5) {
        //打印 GPIOA1 的 pin5 的值
        my_printf("%d\r\n", (GPIOPinRead(GPIOA1_BASE,GPIO_PIN_5)&GPIO_PIN_5)?1:0);

        //清除中断标志位
        GPIOIntClear(GPIOA1_BASE,GPIO_INT_PIN_5);

    }
    #endif
}

```

这里我没去讲寄存器了,,, 英语实在是差