



BAT32G137（库函数版本）

Rev 1.0

修订历史

版本	日期	修订人	修订内容
Rev1.1	22.6.22	缪勤文 张刚	

## 目 录

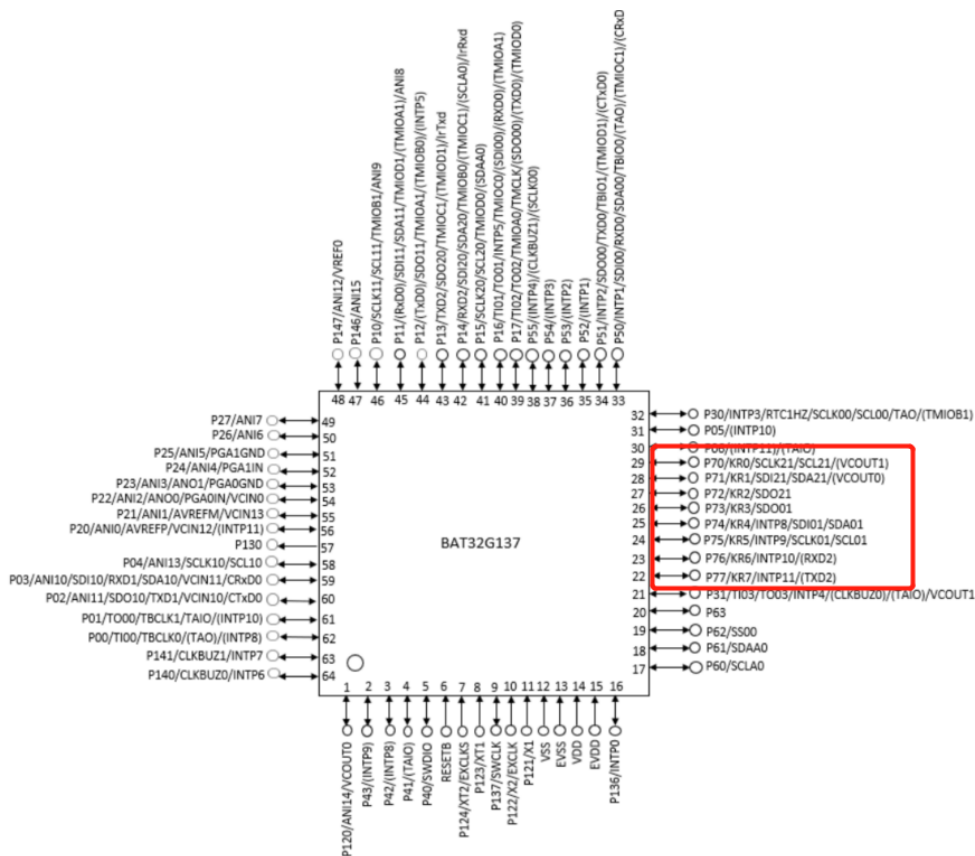
1.前言 .....	3
2.引脚分配 .....	3
3.中微 BAT32G137 按键/外部中断应用库简介 .....	4
3.1.应用例程使用 .....	4
3.1.1. 外部中断初始化 .....	4
3.1.2. 按键中断初始化 .....	5
4.示例演示 .....	6

## 1. 前言

BAT32G137 的外部中断，不同于 STM32 ARM 系列的所有 GPIO 带有外部中断功能；BAT32G137 所有 GPIO 都是支持外部中断的功能，但是同时使用只可以使用 4 个外部中断，也就是说 INTPO~INTP3 可以任意映射到任意引脚。按键中断，输入引脚支持 KR0~KR7 最多 8 个按键中断，按键中断通过输入下降沿产生键中断。

## 2. 引脚分配

BAT32G137 芯片引脚图如下：



按键中断 KR0~KR7 固定引脚为 P70~P77, 因此对于 BAT32G137 来说，使用按键中断则必须使用 P70~P77；

外部按键中断 INTP0~INTP11 从芯片引脚图上可以发现 INTP 和 (INTP)，对于 INTP 字样的，为引脚固定功能；(INTP)为引脚可复用的功能；

### 3. 中微 BAT32G137 按键/外部中断应用库简介

用户只需要对软件接口相关参数进行简单配置、以及封装接口函数调用即可实现所需功能，节约时间，提高开发效率。

使用方式：

使用按键中断需要将应用层 key\_demo.c key\_demo.h 驱动层 key.c key.h、gpio.c gpio.h、isr.c isr.h 加入到工程中去；

使用外部中断需要将应用层 intp\_demo.c intp\_demo.h 驱动层 intp.c intp.h、gpio.c gpio.h、isr.c isr.h 加入到工程中去；

#### 3.1.应用例程使用

##### 3.1.1. 外部中断初始化

```
1. /*****
2. * Function Name: Intp_Init
3. * @brief config gpio exit trigger-interrupt
4. * @param   :
5. * @return
6. *****/
7. void Intp_Init(void)
8. {
9.
10. GPIO_InitTypeDef  GPIO_InitStruct ={0};
11. ISR_InitTypeDef_t ISR_InitStructure={0};
12. INTP_InitTypeDef INTP_InitStructure;
13.
14. GPIO_PinAFConfig(GPIO_PORT7,GPIO_Pin_0,GPIO_P70,GROUP_AF_INTP0);//P
    70 使用INTP0 外部中断功能
15.
16. GPIO_InitStruct.GPIO_Pin    = GPIO_Pin_0;
17. GPIO_InitStruct.GPIO_Mode   = GPIO_Mode_IN;
18. GPIO_InitStruct.GPIO_Ctrl   = GPIO_Control_DIG;
19. GPIO_InitStruct.GPIO_PuPd   = GPIO_PuPd_UP;
20. GPIO_Init(GPIO_PORT7,&GPIO_InitStruct);
21.
22. INTP_InitStructure.INTP_Select  = INTP0 ;           //选择外部中断
    INTP0
23. INTP_InitStructure.EXTI_Trigger = Trigger_Falling; //设置外部中断,
    下降沿触发
24. INTP_Init(&INTP_InitStructure);
```

```

25.
26. ISR_Register(INTP0_IRQn,intp_interrupt);    // 中断服务路径注册
27.
28. INTP_Start(INTP0);                          // Enable INTP0 Interrupt
29.}

```

- 配置外部中断输入的 GPIO，数字功能 INTP0 可以任意映射到 GPIO
- 配置 GPIO 为输入，上拉功能、数字功能
- 按键中断可以选择 INTP0~INTP3，需与 GPIO 实际配置的向匹配
- 外部中断下降沿触发
- 注册中断服务函数

### 3.1.2. 按键中断初始化

```

1. /*****
2. * Function Name: Key_Init
3. * @brief  config key interrupt
4. * @param   None
5. * @return
6. *****/
7. void Key_Init(void)
8. {
9.     volatile uint8_t w_count;
10.    GPIO_InitTypeDef GPIO_InitStructure = {0};
11.    ISR_InitTypeDef_t ISR_InitStructure={0};
12.    KEY_InitTypeDef KEY_InitStructure;
13.
14.
15.    GPIO_PinAFConfig(GPIO_PORT7,GPIO_Pin_0,GPIO_P70,GROUP_AF_ODEFAULT);
        //P70 默认功能 KR0
16.
17.    GPIO_InitStructure.GPIO_Pin    = GPIO_Pin_0;
18.    GPIO_InitStructure.GPIO_Mode   = GPIO_Mode_IN;
19.    GPIO_InitStructure.GPIO_Ctrl   = GPIO_Control_DIG;
20.    GPIO_InitStructure.GPIO_PuPd   = GPIO_PuPd_UP;
21.    GPIO_Init(GPIO_PORT7,&GPIO_InitStructure);
22.
23.    KEY_InitStructure.INT_Select = KR0_INT;    //选择检测键中断信号 KR0
24.    KEY_Init(&KEY_InitStructure);
25.

```

```
26. ISR_Register(KEY_IRQn,key_interrupt);    // 中断服务路径注册
27. Key_Start();        // Enable KEY Interrupt
28.
29. }
```

对于按键中断，P70~P77 分别对应 KR0~KR7 按键中断，因此配置按键中断也只能在 GPIO P70 到 P77 中选择。

## 4. 示例演示

代码示例例程包含了按键中断和外部中断，P70 分别引脚分别设置为按键中断和外部中断输入。