



中微 M0 系列

GPIO中断唤醒MCU应用笔记

Rev. 1.0.0

请注意以下有关CMS知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

目录

1. 概述	3
1.1 目的.....	3
1.2 内容提要.....	3
2. 功能概述	4
3. 功能描述	4
3.1 低功耗模式.....	4
3.2 GPIO 中断.....	4
3.3 寄存器说明.....	4
4. 操作实例	5
4.1 实现功能.....	5
4.2 操作步骤.....	5
4.3 流程图.....	5
4.4 样例代码.....	6
4.5 测试结果.....	7
5. 应用注意事项	2
6. 更多信息	3
7. 版本修订说明	4

1. 概述

1.1 目的

本文以 CMS32M57xx 系列产品为例，介绍了 GPIO 中断唤醒 MCU 的功能配置以及应用注意事项。

1.2 内容提要

本文档包含以下内容：

第 2 章：功能概述。

第 3 章：功能描述。

第 4 章：操作实例。

第 5 章：应用注意事项。

2. 功能概述

芯片可以配置为低功耗模式，以适应不同应用的功耗需求。芯片进入低功耗模式后，GPIO 中断的产生可唤醒系统，使 MCU 从低功耗模式恢复到正常运行模式。

3. 功能描述

3.1 低功耗模式

芯片有不同工作模式，可适应不同应用的功耗需求。不同工作模式的芯片状态及唤醒源如表 3-1 所示。CMS32M57xx 系列芯片低功耗模式包括睡眠模式、深度睡眠模式、停止模式。三种模式均能被 GPIO 中断唤醒。唤醒后，MCU 恢复到正常模式，程序继续执行。

表 3-1: CMS32M57xx 系列芯片不同工作模式对比

	正常模式	睡眠模式	深度睡眠模式	停止模式
定义	MCU 处于正常工作状态，外设正常运行，LDO 开启。	MCU 处于睡眠状态，CPU 停止工作，外设正常运行，LDO 开启。	MCU 处于深度睡眠模式，CPU 除 WDT 外的外设停止。LDO 状态由 LDODS 配置决定。	MCU 处于停止模式，所有模块停止工作，LDO 状态由 LDODS 配置决定。
唤醒源	-	所有中断	I/O 中断，WDT 中断	I/O 中断

3.2 GPIO 中断

每个 GPIO 管脚都可以设置成芯片的中断源。并且有以下五种中断触发条件可以设置：

- (1) 低电平触发
- (2) 高电平触发
- (3) 下降沿触发
- (4) 上升沿触发
- (5) 上升或下降沿触发

在边沿触发中，用户可以通过使能输入信号去抖功能来阻止由噪声引起的意外中断。当芯片进入睡眠/深度睡眠/停止模式时，GPIO 可以唤醒系统。唤醒触发的条件由 GPIOxIVAL 决定，需要注意如下：

- (1) 下降沿唤醒，需要在进入低功耗状态之前将端口电平拉高。
- (2) 上升沿唤醒，需要在进入低功耗状态之前将端口电平拉低。

3.3 寄存器说明

详情参考芯片手册。

4. 操作实例

4.1 实现功能

GPIO 中断唤醒低功耗模式。

4.2 操作步骤

- (1) 初始化 LED 连接 IO 口为推挽输出模式；
- (2) 配置 GPIO 中断为下降沿触发方式，开启 GPIO 中断；
- (3) 配置芯片为低功耗模式，等待唤醒；
- (4) 唤醒后执行程序为闪烁 LED 灯。

4.3 流程图

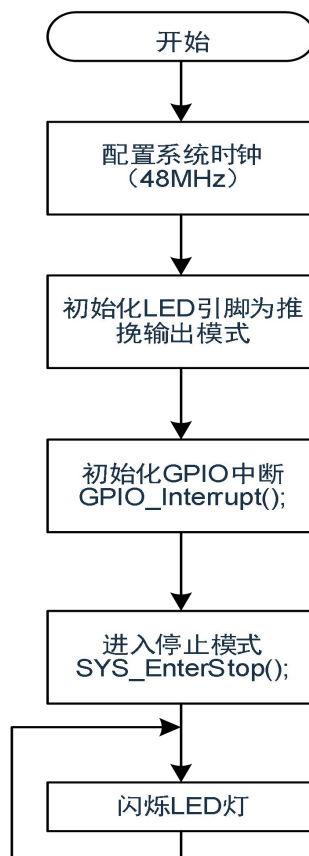


图 4-1：应用实例流程图

4.4 样例代码

```
int main(void)
{
    uint32_t j;
    SYS_ConfigHSI(SYS_CLK_HSI_48M); /*设置内部高速时钟为 48Mhz*/
    SYS_EnableHSI(); /*开启高速时钟*/
    SYS_ConfigAHBClock(SYS_CLK_SEL_HSI,SYS_CLK_DIV_1); /*设置 AHB 时钟为高速时钟的 1 分频*/
    SYS_ConfigAPBClock(AHB_CLK_DIV_1); /*设置 APB 时钟为 AHB 时钟的 1 分频*/
    SystemCoreClockUpdate(); /*刷新 SystemCoreClk、SystemAPBClock 变量值*/

    /*初始化 LED 端口为推挽输出模式*/
    SYS_SET_IOCFG(IOP45CFG,SYS_IOCFG_P45_GPIO); /*设置 P45 为 GPIO 模式*/
    GPIO_CONFIG_IO_MODE(GPIO4,GPIO_PIN_5,GPIO_MODE_OUTPUT); /*设置 P45 为推挽输出*/
    GPIO4->DO_f.P5 = 0; /*P45 输出低电平*/

    /*初始化 GPIO 中断*/
    GPIO_Interrupt();
    /*进入停止模式*/
    SYS_EnterStop();

    while(1)
    {
        /*设置延时
        for(j=0;j<5000000;j++);
        //P4_5 翻转
        GPIO4->DO ^= 1<<5;
        */
    }
}

void GPIO_Interrupt(void)
{
    /*(1) 设置 P02 为上拉输入模式*/
    SYS_SET_IOCFG(IOP02CFG,SYS_IOCFG_P02_GPIO); /*设置 P02 为 GPIO 模式*/
    GPIO_CONFIG_IO_MODE(GPIO0,GPIO_PIN_2,GPIO_MODE_INPUT);
    /*(2) 设置 P02 中断为下降沿中断模式*/
    GPIO_EnableInt(GPIO0,GPIO_PIN_2_MSK,GPIO_INT_EDGE_FALLING);
    /*(3) 设置 P02 IO 口输入滤波*/
    GPIO_EnableFilter(GPIO0,GPIO_PIN_2_MSK,GPIO_FILCLK_DIV_8);
    /*(4) 设置 GPIO0 中断*/
    GPIO_ClearIntFlag(GPIO0,GPIO_PIN_2); /*清除 P02 中断标志位*/
    NVIC_EnableIRQ(GPIO0_IRQn); /*开启 GPIO0 中断*/
    /*(5) 设置 GPIO0 中断优先级*/
    NVIC_SetPriority(GPIO0_IRQn,3); /* 0~3, 0 最高, 3 最低*/
}

void GPIO0_IRQHandler(void)
{
    /*清除中断标志位*/
    if(GPIO_GetIntFlag(GPIO0,GPIO_PIN_2))
    {
        GPIO_ClearIntFlag(GPIO0,GPIO_PIN_2);
    }
}
```

4.5 测试结果

芯片上电后，等待一段时间，芯片进入休眠模式并稳定运行后，给 P02 输入下降沿信号触发 GPIO 中断。观察到 LED 持续闪烁，MCU 恢复到正常模式，程序继续执行。

5. 应用注意事项

- 1、采用下降沿触发 I/O 中断唤醒芯片低功耗模式时，应特别注意如下前提：
在进入低功耗模式前，需将端口电平拉高，否则无法唤醒 MCU。
- 2、采用上升沿触发 I/O 中断唤醒芯片低功耗模式时，应特别注意如下前提
在进入低功耗模式前，需将端口电平拉低，否则无法唤醒 MCU。
- 3、当触发方式选择为双沿触发（上升沿和下降沿都可触发）时，I/O 中断不支持 MCU 低功耗模式唤醒功能。
- 4、本应用注意事项适用于以下产品型号：
 - 1) CMS32F033
 - 2) CMS32M55xx
 - 3) CMS32F035
 - 4) CMS32MEBIKE
 - 5) CMS32M57xx

6. 更多信息

更多信息，请登录中微半导体网站查看 <http://www.mcu.com.cn>

7. 版本修订说明

版本号	时间	作者	修改内容
V1.0.0	2022 年 3 月	Renlq	初始版本