南京凌鸥创芯电子有限公司

休眠唤醒应用笔记

目 录

表格目录

未找到图形项目表。

图片目录

未找到图形项目表。

1 概述

设置好唤醒源，可以为 GPIO，也可以为内部定时器，定时器需要设置好唤醒间隔，定时唤醒信号不可关闭，芯片会定期从睡眠状态被唤醒。如果希望使用 GPIO 作为唤醒源，而不是定时唤醒信号，可以通过 SYS_RST_CFG.WK_INTV 将定时唤醒间隔设置为最大，并在唤醒中断中做如此判断：如果是定时唤醒立即再次进入睡眠，如果是 GPIO 唤醒则执行相关操作。GPIO 唤醒可以配合 GPIO 外部中断事件 EXTI 使用。

以下程序以 LKS08x 芯片为例，其他型号具体寄存器位置可能不同。但软件配置流程建议相同处理。

切换 PLL 时钟的动作建议封装成一个函数来调用。

```
u32 AFE_REG5_RECORD = 0;
u32 AFE_REG6_RECORD = 0;

void Switch2PLL(void){
    // Turn on PLL/BGP/HRC modules and restore modules state in ANALOG
    SYS_AFE_REG5 = AFE_REG5_RECORD;
    //延时等待 PLL 开启后稳定
    DelayUs(100);
    SYS_AFE_REG6 = AFE_REG6_RECORD;
    // Switch main clock to PLL clock
    SYS_CLK_CFG = 0x1ff;
}
```

2 休眠流程

```
//开启寄存器写使能
SYS_WR_PROTECT = 0x7a83;
//记录配置值
AFE_REG5_RECORD = SYS_AFE_REG5;
AFE_REG6_RECORD = SYS_AFE_REG6;

//关闭不相关 IO，使之进入高阻态
GPIO0_PIE = 0;
GPIO0_POE = 0;
GPIO0_PDO = 0;
GPIO1_PIE = 0;
GPIO1_POE = 0;
GPIO1_PDO = 0;
GPIO2_PIE = 0;
GPIO2_POE = 0;
GPIO2_PDO = 0;
GPIO3_PIE = 0;
GPIO3_POE = 0;
GPIO3_PDO = 0;

//设置 IO 唤醒极性和使能
GPIO_WAKE_POL = 0x0;
GPIO_WAKE_EN    = 0x1;
GPIO0_PIE       |= 0x1;
// tie P0.0 low at board
GPIO0_PUE       |= 0x1;

//  使能 IO 事件用于在唤醒判断是 IO 唤醒还是定时唤醒
// P0.0 negedge trigger
EXTI_CR0 = 0x1;
EXTI_IF = 0x1;

//  使能唤醒中断
NVIC_EnableIRQ(WAKE_IRQn);
__enable_irq();
SYS_WR_PROTECT = PSW_SYS_PROTECT;
```

//此处虽然设置了 PLLPDN，RCHPD，BGPPD，但仅仅这些模块关闭的必要条件。当系统工作在 PLL 时钟上时，PLL 时钟不允许关闭，即使设置了 PLLPDN=0；当 PLL 使用 HRC 作为参考时钟时，HRC 和 BGP 不允许关闭。

```
SYS_AFE_REG5 = 0x0500;     // Prepare to shutdown PLL,HRC,BGP
```

```
SYS_AFE_REG6|=1;            // Turn off power detection module in ANALOG
SYS_CLK_CFG = 0;            // Switch to HRC clock
//此处写入休眠密码以后，芯片内部状态机才会依次关闭高速时钟等相关模块进入休眠状态。
SYS_CLK_SLP = 0xDEAD;       // Sleep password
SYS_WR_PROTECT = 0x0;
//由于使用了 Wait For Interrupt 宏指令，因此必须要使能唤醒中断，芯片唤醒后首先执行唤醒中断代码
__WFI();                                    // Wait for interrupt
```

需要注意的是，由于工程中可能开了很多中断，所以 WFI 指令可能会被其他中断响应，导致软件继续向下执行一段时间才进入休眠。

休眠状态机工作于 LRC 时钟域，相对于芯片主时钟来说很慢。

一个处理方式是休眠之前关闭除 WAKEUP 中断以外的其他中断。

或者保证即使 WFI 被其他中断响应后续执行代码不影响休眠唤醒操作。

3 唤醒

```
Void WAKE_IRQHandle(void){
    //
    Switch2PLL();
    if((EXTI_IF & 0x1) == 0x1){
    //IO 唤醒
    //do something
    }
    else{
    //定时唤醒
    }
    EXTI_IF = 0x1;
}
```