



复旦微电子

FM33G0XX

低功耗系列 MCU

应用笔记

低功耗说明

AN011

V1.0



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站 (<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商 标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。

联系方式：

电表产品应用：

邢杰：xingjie@fmsh.com.cn TEL: 13916427310

陈钊：chenzhao@fmsh.com.cn TEL: 18616125501

水气热表及智能家居：

朱发旺：zhufawang@fmsh.com.cn TEL: 17749796664

姜涛：jiangtao@fmsh.com.cn TEL: 18701992908

超高频 900M 及物联网相关：

王晓腾：wangxiaoteng@fmsh.com.cn TEL: 13585663727

王天纵：wangtianzong@fmsh.com.cn TEL: 18221803903

资料下载及交流：

开发者论坛：<http://www.fmdevelopers.com.cn>



目 录

1 说明.....	1
2 原理.....	1
2.1 GPIO 配置.....	1
2.2 休眠配置.....	2
2.2.1 内核休眠.....	2
2.2.2 外设休眠.....	2
2.3 唤醒.....	6
2.3.1 内核休眠唤醒.....	7
2.3.2 芯片整体休眠唤醒.....	8
2.4 编程配置.....	11
2.5 休眠功耗.....	11
版本信息.....	14
附录.....	15
上海复旦微电子集团股份有限公司销售及服	33



图目录

图 2-1 GPIO 配置调用函数	1
图 2-2 SWD 内部上拉调用函数.....	1
图 2-3 看门狗长周期调用函数.....	2
图 2-4 SCR 寄存器	2
图 2-5 对于 V3 开发板 J27 及 J34 位置.....	3
图 2-6 sleep 函数示例	5
图 2-7 deepsleep 函数示例.....	5
图 2-8 RTCBKP 函数示例	6
图 2-9 芯片休眠唤醒源.....	6
图 2-10 普通串口中断唤醒参考例程.....	7
图 2-11 普通串口中断唤醒内核休眠.....	7
图 2-12 休眠程序唤醒时间测试.....	8
图 2-13 示波器测试唤醒时间探头连接实图.....	9
图 2-14 38#中断 (PRIMASK=1) _SLEEP_唤醒时间 5us	10
图 2-15 NMI 中断 (PRIMASK=0) _SLEEP_唤醒时间 8us	10
图 2-16 LPRUN@5V 功耗.....	11
图 2-17 SLEEP@5V 功耗	12
图 2-18 DEEPSLEEP@5V 功耗	12
图 2-19 RTCBKP@5V 功耗	13

表目录

表 2-1 功能模式和外设使能.....	4
表 2-2 各个模式下唤醒时间.....	11

1 说明

本文档为 FM33G0XX 系列低功耗 MCU 的应用笔记，用于说明低功耗配置的方法。FM33G0XX 系列是复旦微电子公司开发的低功耗 MCU 芯片，请联系复旦微电子公司提供更多相关文档支持设计开发。

2 原理

2.1 GPIO 配置

在低功耗应用时，FM33G0XX 的未被使用以及低功耗模式下无需保持的 GPIO，建议配置为开漏输出高（高阻态）。GPIO 不仅包含已经封装出的 GPIO，还包括未封装出的 GPIO。

以 80pin 的 FM33G048 为例，调用以下两个函数。如图 2-1 所示。

```
Close_None_GPIO_80pin();    //关闭80脚芯片不支持的Io
Close_AllIO_GPIO_80pin();    //关闭全部Io
```

图 2-1 GPIO 配置调用函数

值得注意的是这两个函数没有关闭 SWD 的 PG8 和 PG9，仍然为数字功能 IO。

在休眠时，SWD 口的输入假如浮空，可能会感应出电平导致 IO 输入寄存器翻转引起功耗变化。如 SWDTCK/SWDTDO 外部无上拉时(V3.0 开发板设置了外部上拉)，可以调用下图 2-2 所示函数，打开内部上拉电阻。

```
//休眠时SWD的输入脚如果浮空，可能会感应出电平导致Io口输入寄存器翻转引起功耗变大，
//如果SWD口无外部上拉的话，休眠前可开启上拉使能
void SWD_IO_PullUp(FunState NewState)
{
    if( DISABLE == NewState )//关闭SWDTCK，SWDTDO上拉使能
    {
        AltFunIO( GPIOG, GPIO_Pin_8, ALTFUN_NORMAL ); //PG8;//SWDTCK
        AltFunIO( GPIOG, GPIO_Pin_9, ALTFUN_NORMAL ); //PG9;//SWDTDO
    }
    else//打开SWDTCK，SWDTDO上拉使能
    {
        AltFunIO( GPIOG, GPIO_Pin_8, ALTFUN_PULLUP ); //PG8;//SWDTCK
        AltFunIO( GPIOG, GPIO_Pin_9, ALTFUN_PULLUP ); //PG9;//SWDTDO
    }
}
```

图 2-2 SWD 内部上拉调用函数

2.2 休眠配置

在示例或手册中的休眠模式，通常指的是整个芯片的休眠包括内核和外设。在休眠时通常使用 RTC 的中断服务程序来清狗。

在休眠时为降低 RTC 唤醒芯片的频率，可以将看门狗可以配置为 4096S 的长周期。可以调用以下函数来实现，如图 2-3 所示。

```
IWDT_IWDTCFG_IWDTSLP4096S_Setable(ENABLE); //配置休眠时是否启用4096s长周期
```

图 2-3 看门狗长周期调用函数

2.2.1 内核休眠

WFI(Wait for interrupt)和 WFE(Wait for event)是两个让 ARM 核进入休眠模式的指令，WFE 通常用在多处理器应用场景，FM33G 系列作为单核 MCU，推荐使用 WFI。在执行 WFI()函数之前需要先配置 SCR 寄存器:如图 2-4 所示

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											SEVON PEND	Res.	SLEEP DEEP	SLEEP ON EXIT	Res.
											rw		rw	rw	

图 2-4 SCR 寄存器

Sleep/deepsleep : SCB->SCR=0;

RTCBKP: SCB->SCR=0x04(1<<2);

在 SCR 的 bit 2 为 0 的基础上配置 SCB->SCR |=0x02 时，当内核被唤醒后执行完相应的中断处理函数后自动进入休眠模式。

SLEEPONEXIT:

0:被唤醒进入线程模式后不再进入睡眠模式

1:被唤醒后执行完相应的中断处理函数后进入睡眠模式

2.2.2 外设休眠

FM33G0XX 通过配置 LPMCFCG 寄存器的 PMOD 实现外设休眠。通过设置 PMOD 可实现外设的四种状态：Active mode、LPRUN mode、Sleep mode、RTCBKP mode。

注：1、当 PMOD 设置为 sleep mode 时，SLPDP 设置为 1，外设就进入了 deepsleep mode

- 2、Sleep mode 顺利进入 deepsleep 模式的前提是 BOR 和 SVD 处于关闭状态,调用
ANAC_BORCON_OFF_BOR_Setable(ENABLE)及
ANAC_SVDCON_SVDEN_Setable(DISABLE);
- 3、为了更低功耗,休眠前关闭 RCLP、ADC 使能以及 ADC buffer 使能。
RCC_SYSCLKSEL_LPM_RCLP_OFF_Setable(ENABLE)
ANAC_ADCCON_ADC_EN_Setable(DISABLE)
ANAC_ADCINSEL_BUFEN_Setable(DISABLE)
- 4、假如在 deepsleep 下需要使用 SVD,可以使用 SVD 的间歇使能功能,这样平时处于 deepsleep mode,只是在 SVD 使能的那段时间处于 sleep mode。
- 5、针对复旦微提供的 V3 开发板,测试芯片功耗,需要断开 J27(38025T), J34(SVS 电源控制开关),避免引入外部电路造成的功耗,如图 2-5。

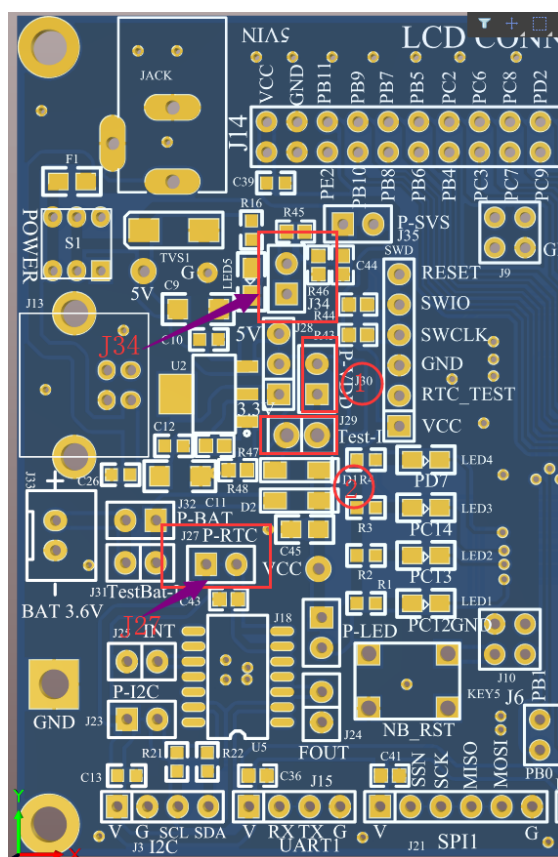


图 2-5 对于 V3 开发板 J27 及 J34 位置

注：测试芯片休眠电流时，断开图 2-5 中 1 位置(J30)的短接帽，将万用表电流档通过 2 位置(J29)连入电路中测量。



功耗模式和外设使能，如表 2-1

表 2-1 功能模式和外设使能

	ACTIVE	LPRUN	SLEEP	DEEPSLEEP	RTCBKP
CPU clock	On	On	-	-	-
RCHF	On	-	-	-	-
PLL	On	-	-	-	-
RCLF	On	On	-	-	-
RCLP	On	On	On	On	On
XTLF	On	On	On	On	On
UART	On	On	-	-	-
SPI	On	On	-	-	-
I2C	On	On	-	-	-
7816	On	-	-	-	-
AES	On	On	-	-	-
TRNG	On	On	-	-	-
CRC	On	On	-	-	-
DMA	On	On	-	-	-
LPUART	On	On	On	On	-
RTC	On	On	On	On	On
BSTIM	On	On	-	-	-
EXTIM	On	On	-	-	-
LPTIM	On	On	On	On	On
LCD	On	On	On	On	-
Comparator	On	On	On	On	On
ADC	On	On	On	On	On
TempSensor	On	On	On	On	On
SVD	On	On	On	On	On
WWDT	On	On	-	-	-
IWDT	On	On	On	On	-
RAM	On	On	On	On	-
Retention Registers	On	On	On	On	On
GPIO interrupt	On	On	On	On	-
WKUPx	On	On	On	On	On
BOR	On	On	On	On	On
PDR	On	On	On	On	On
NRST	On	On	On	On	On
GPIO state retention	On	On	On	On	-

配置程序示例，如图 2-6、2-7、2-8 所示

Sleep:

```
void Test_Sleep(void)
{
    PMU_SleepCfg_InitTypeDef SleepCfg_InitStruct;

    /*下电复位配置*/
    //pdr和bor两个下电复位至少要打开一个
    //当电源电压低于下电复位电压时，芯片会被复位住
    //pdr电压档位不准但是功耗极低（几乎无法测量）
    //bor电压档位准确但是需要增加2uA功耗
    ANAC_PDRCON_PDREN_Setable(ENABLE); //打开PDR
    ANAC_PDRCON_PDRCFG_Set(ANAC_PDRCON_PDRCFG_1P35V); //pdr电压调整到1.35V
    ANAC_BORCON_OFF_BOR_Setable(ENABLE); //关闭BOR 3uA

    RCC_SYSCCLKSEL_LPM_RCLP_OFF_Setable(ENABLE); //关闭rclp 0.2uA
    ANAC_ADCCON_ADC_EN_Setable(DISABLE); //ADC关闭
    ANAC_ADCINSEL_BUFEN_Setable(DISABLE);

    SleepCfg_InitStruct.PMOD = PMU_LPMC_CFG_PMOD_SLEEP; //功耗模式配置
    SleepCfg_InitStruct.SLPDP = PMU_LPMC_CFG_SLPDP_SLEEP; //DeepSleep控制
    SleepCfg_InitStruct.CVS = DISABLE; //内核电压降低控制
    SleepCfg_InitStruct.XTOFF = PMU_LPMC_CFG_XTOFF_DIS; //保持XTLF
    SleepCfg_InitStruct.SCR = 0; //MO系统控制寄存器，一般配置为0即可

    PMU_SleepCfg_Init(&SleepCfg_InitStruct); //休眠配置

    IWDI_Clr();
    __WFI(); //进入休眠
    IWDI_Clr();
}
```

图 2-6 sleep 函数示例

Deepsleep:

```
void Test_Sleep(void)
{
    PMU_SleepCfg_InitTypeDef SleepCfg_InitStruct;

    /*下电复位配置*/
    //pdr和bor两个下电复位至少要打开一个
    //当电源电压低于下电复位电压时，芯片会被复位住
    //pdr电压档位不准但是功耗极低（几乎无法测量）
    //bor电压档位准确但是需要增加2uA功耗
    ANAC_PDRCON_PDREN_Setable(ENABLE); //打开PDR
    ANAC_PDRCON_PDRCFG_Set(ANAC_PDRCON_PDRCFG_1P35V); //pdr电压调整到1.35V
    ANAC_BORCON_OFF_BOR_Setable(ENABLE); //关闭BOR 3uA

    RCC_SYSCCLKSEL_LPM_RCLP_OFF_Setable(ENABLE); //关闭rclp 0.2uA
    ANAC_ADCCON_ADC_EN_Setable(DISABLE); //ADC关闭
    ANAC_ADCINSEL_BUFEN_Setable(DISABLE);

    SleepCfg_InitStruct.PMOD = PMU_LPMC_CFG_PMOD_SLEEP; //功耗模式配置
    SleepCfg_InitStruct.SLPDP = PMU_LPMC_CFG_SLPDP_DEEPSLEEP; //DeepSleep控制
    SleepCfg_InitStruct.CVS = DISABLE; //内核电压降低控制
    SleepCfg_InitStruct.XTOFF = PMU_LPMC_CFG_XTOFF_DIS; //保持XTLF
    SleepCfg_InitStruct.SCR = 0; //MO系统控制寄存器，一般配置为0即可

    PMU_SleepCfg_Init(&SleepCfg_InitStruct); //休眠配置

    IWDI_Clr();
    __WFI(); //进入休眠
    IWDI_Clr();
}
```

图 2-7 deepsleep 函数示例

RTCBKP:

```

void Test_Sleep(void)
{
    PMU_SleepCfg_InitTypeDef SleepCfg_InitStruct;

    /*下电复位配置*/
    //pdr和bor两个下电复位至少要打开一个
    //当电源电压低于下电复位电压时，芯片会被复位住
    //pdr电压档位不准但是功耗极低（几乎无法测量）
    //bor电压档位准确但是需要增加2uA功耗
    ANAC_PDRCON_PDREN_Setable(ENABLE); //打开PDR
    ANAC_PDRCON_PDRCFG_Set(ANAC_PDRCON_PDRCFG_1P35V); //pdr电压调整到1.35V
    ANAC_BORCON_OFF_BOR_Setable(ENABLE); //关闭BOR 3uA

    RCC_SYSCLSSEL_LPM_RCLP_OFF_Setable(ENABLE); //关闭rclp 0.2uA

    ANAC_ADCCON_ADC_EN_Setable(DISABLE); //ADC关闭
    ANAC_ADCINSEL_BUFEN_Setable(DISABLE);

    SleepCfg_InitStruct.PMOD = PMU_LPMCFG_PMOD_RTCBKP; //功耗模式配置
    SleepCfg_InitStruct.XTOFF = PMU_LPMCFG_XTOFF_DIS; //保持XTLFF
    SleepCfg_InitStruct.SCR = 1<<2; //使能cpu深度睡眠特性

    PMU_SleepCfg_Init(&SleepCfg_InitStruct); //休眠配置

    IWDI_Clr();
    __WFI(); //进入休眠
    IWDI_Clr();
}

```

图 2-8 RTCBKP 函数示例

2.3 唤醒

芯片休眠唤醒——在芯片整体(内核和外设)处于休眠时，如图 2-9 中断源可唤醒芯片，唤醒后 RCLF 会自动打开，当芯片由 sleep 和 deepsleep 唤醒后时钟为 RCHF8M。其中 RTCBKP 唤醒后芯片会复位。

唤醒源	应用	可唤醒模式		
		Sleep	DeepSleep	RTCBKP
停振检测	可屏蔽，32786Hz 晶振停振时唤醒芯片	√	√	√
SVD	可屏蔽，在电源电压跌落至阈值以下或升高至阈值以上时唤醒芯片	√	√	√
比较器	可屏蔽，用于外部事件唤醒	√	√	√
RTC	可屏蔽，根据需要的唤醒周期设置	√	√	√
IO 引脚中断	可屏蔽，用于外部事件唤醒	√	√	×
Debug	不可屏蔽，用于 debug 唤醒	√	√	×
LPUART	可屏蔽，接收数据唤醒	√	√	×
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√	√
NRST	不可屏蔽，用于全局复位	√	√	√
LPTIM	可屏蔽，用于定时唤醒	√	√	√

注：SVD的外部检测引脚（SVS）、比较器输入引脚、RTC时间戳引脚、WKUPx，如配置为以上功能，则在RTCBKP模式下保持输入，以实现RTCBKP唤醒。

通过PRIMASK能够实现事件唤醒芯片，但是不进入中断服务程序，省去进出中断的时间。

图 2-9 芯片休眠唤醒源

2.3.1 内核休眠唤醒

当 FM33G0XX 只有内核处于 sleep/deepsleep 状态时，大部分外设处于激活状态时，处于激活的外设都可以使用中断唤醒内核。

以 UART1 内核休眠唤醒为例，参考例程如图 2-10 所示

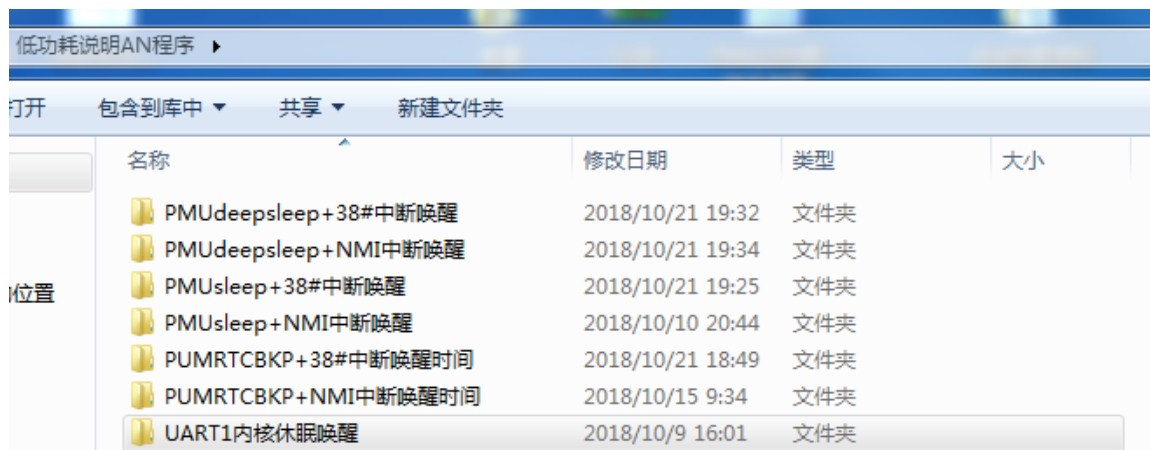


图 2-10 普通串口中断唤醒参考例程

具体程序如图 2-11:

```
int main (void)
{
    Init_System();           //系统初始化
    //UARTx测试
    RTC_Cfg();
    Test_Uartx(UART1);
    for(;;)
    {
        SCB->SCR=0;
        IWDT_Clr();
        __WFI();
        IWDT_Clr();
        if (SET == UARTx_RXBUFSTA_RXFF_Chk(UART1))
        {
            IWDT_Clr();
            tmp08 = UARTx_RXREG_Read(UART1); //接收中断标志仅可通过读取rxreg寄存器清除
            break;
        }
    }

    for(;;)
    {
        IWDT_Clr();           //清系统看门狗

        LED0_TOG;             //LED0闪烁
        TicksDelayMs( 50, NULL ); //软件延时
    }
}
```

图 2-11 普通串口中断唤醒内核休眠

程序解释：当仅仅内核休眠时，看门狗溢出周期 4096s 无法工作(即芯片休眠时（内核+外设休眠）溢出周期 4096 才能工作）

所以采用 RTC 秒中断进行唤醒清狗—>休眠—>秒中断进行唤醒清狗

当串口接收到数据，内核唤醒执行—>执行应用程序。

2.3.2 芯片整体休眠唤醒

以 WKUP 唤醒为例，WKUP 有两个中断入口一个是不可屏蔽中断，一个是可屏蔽中断#38。唤醒时间以触发唤醒源到进入中断服务程序的所需时间，但 RTCBKP 由于会执行复位所以唤醒时间的计算以触发唤醒源到 main 函数初始化。其中 RTCBKP 唤醒后的可编程延迟设置为 16 us。

注：选择#38 中断入口时，PRIMASK 寄存器置 1 即 __disable_irq()，唤醒后可不执行唤醒源中断服务程序。可以通过多种方法编程 PRIMASK 寄存器，

●使用 C 语言以及 CMSIS 库，可以通过以下函数来设置和清除 PRIMASK;（函数被编译为 CPS 指令，<core_cmFunc.h>）

```
void __enable_irq(void);           //清除 PRIMASK
```

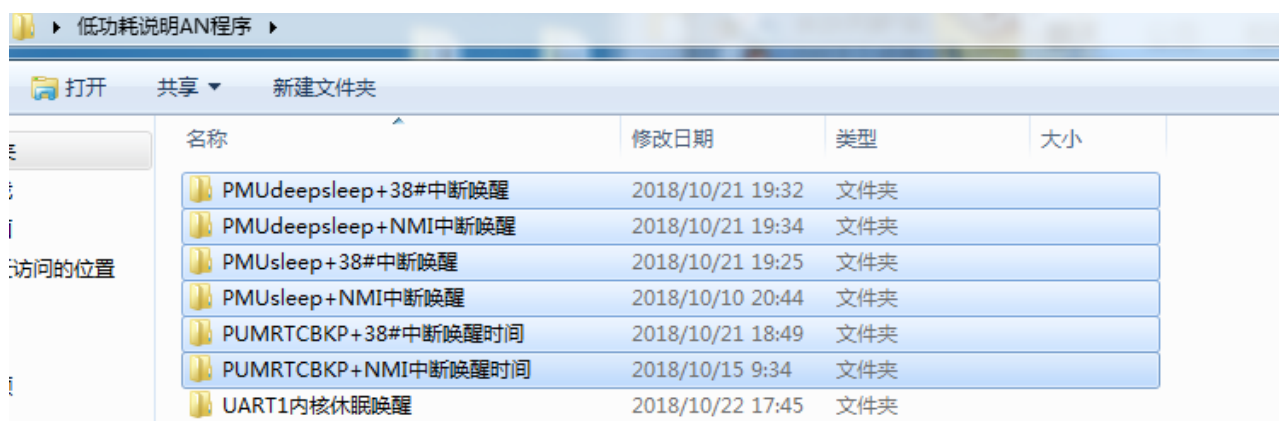
```
void __disable_irq(void);          //设置 PRIMASK
```

●使用内核封装函数来设置和清除 PRIMASK;（编译后，通过 MSR 指令将值（0 或 1）写入 primask 内核寄存器，<core_cmFunc.h>）

```
__set_PRIMASK(1);                 //设置 PRIMASK
```

```
__set_PRIMASK(0);                 //清除 PRIMASK
```

唤醒时间测试参考例程如图 2-12



名称	修改日期	类型	大小
PMUdeepsleep+38#中断唤醒	2018/10/21 19:32	文件夹	
PMUdeepsleep+NMI中断唤醒	2018/10/21 19:34	文件夹	
PMUsleep+38#中断唤醒	2018/10/21 19:25	文件夹	
PMUsleep+NMI中断唤醒	2018/10/10 20:44	文件夹	
PUMRTCBKP+38#中断唤醒时间	2018/10/21 18:49	文件夹	
PUMRTCBKP+NMI中断唤醒时间	2018/10/15 9:34	文件夹	
UART1内核休眠唤醒	2018/10/22 17:45	文件夹	

图 2-12 休眠程序唤醒时间测试

以 SLEEP 为例

➤ 配置为 38#中断时，设置 PRIMASK，则中断可以被 PRIMASK 屏蔽掉，唤醒后，CPU 先执行__WFI()的下一条语句(翻转 IO)，而非响应 38#中断，所以缩短了唤醒时间。

GPIO 的 PINWKEN 寄存器 BIT31 选择 38#中断入口，#38 中断使能

```
void Wakeup_Cfg(void)
{
    //唤醒源：外设中断打开+NVIC中断打开(可关闭全局中断使能，这样唤醒后不会进中断服务程序；此时需手动清除挂起标志，否则：
    //避免：外设中断打开+NVIC中断关闭，会导致芯片唤醒但是cpu依然休眠的情况
    // RCC_PERCLK_SetableEx(RTCCLK, ENABLE); //RTC总线时钟使能
    // RTC_RTCIE_SetableEx(DISABLE, 0xFFFFFFFF); //关闭所有RTC中断
    // RTC_RTCIE_SetableEx(ENABLE, RTC_RTCIE_SEC_IE_Msk); //打开RTC秒中断
    // RTC_RTCIF_ClrEx(RTC_RTCIE_SEC_IE_Msk); //清除秒中断标志
    //
    // NVIC_ClearPendingIRQ(RTC_IRQn);
    // NVIC_DisableIRQ(RTC_IRQn);
    // NVIC_SetPriority(RTC_IRQn,2);
    // NVIC_EnableIRQ(RTC_IRQn);

    RCC_PERCLK_SetableEx( EXTI2CLK, ENABLE ); //EXTI外部引脚中断采样时钟，IO数字滤波时钟使能
    RCC_PERCLK_SetableEx( EXTI1CLK, ENABLE ); //EXTI外部引脚中断采样时钟，IO数字滤波时钟使能
    RCC_PERCLK_SetableEx( EXTI0CLK, ENABLE ); //EXTI外部引脚中断采样时钟，IO数字滤波时钟使能
    RCC_PERCLK_SetableEx( PDCCLK, ENABLE ); //IO控制时钟寄存器使能

    InputtIO( GPIOF, GPIO_Pin_5, IN_PULLUP ); //PF5//DISPKEY,带上拉输入
    GPIO_EXTI_Init(GPIOF, GPIO_Pin_5, EXTI_DISABLE); //关闭IO中断

    NVIC_DisableIRQ(GPIO_IRQn);
    NVIC_ClearPendingIRQ(GPIO_IRQn);
    NVIC_DisableIRQ(WKUP_IRQn);
    NVIC_SetPriority(WKUP_IRQn,2);
    NVIC_EnableIRQ(WKUP_IRQn);
    //NVIC_SetPriority(WKUP_IRQn,2); //NVIC_SetPriority(WKUP_IRQn,2);
    GPIO->PINWKEN |= 0x80000001;
    //GPIO_PINWKEN_SetableEx(PINWKEN_PF5, ENABLE); //使能PF5的NWKUP1功能
}
```

示波器探头一个接到 PA0，一个接到按键 PF5，如图 2-13 所示



图 2-13 示波器测试唤醒时间探头连接实图

通过示波器观察唤醒时间，如图 2-14

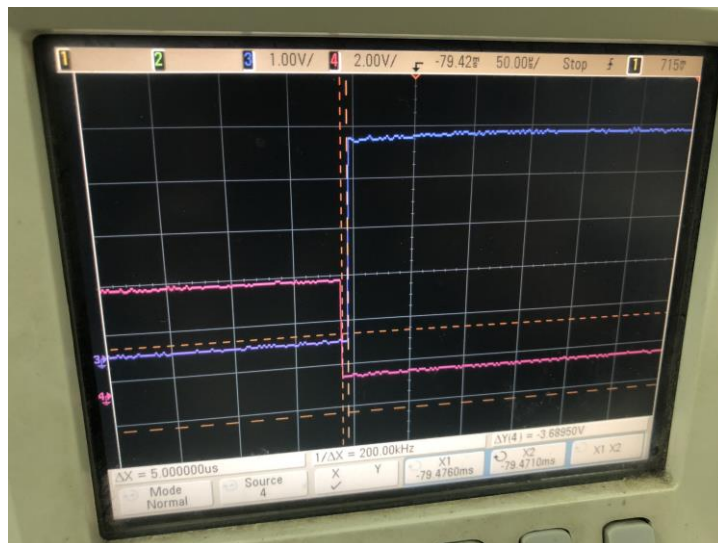


图 2-14 38#中断 (PRIMASK=1) _SLEEP_唤醒时间 5us

➤ 配置为 NMI 中断时，CPU 先执行 NMI 中断函数处理(函数开始 IO 翻转)，然后再执行 __WFI() 的下一条语句，所以唤醒时间时间较长。

GPIO 的 PINWKEN 寄存器的 BIT31 选择 NMI 中断入口，NMI 中断处理函数中直接翻 IO。

示波器探头一个接到 PA0，一个接到按键 PF5，通过示波器观察唤醒时间如图 2-15

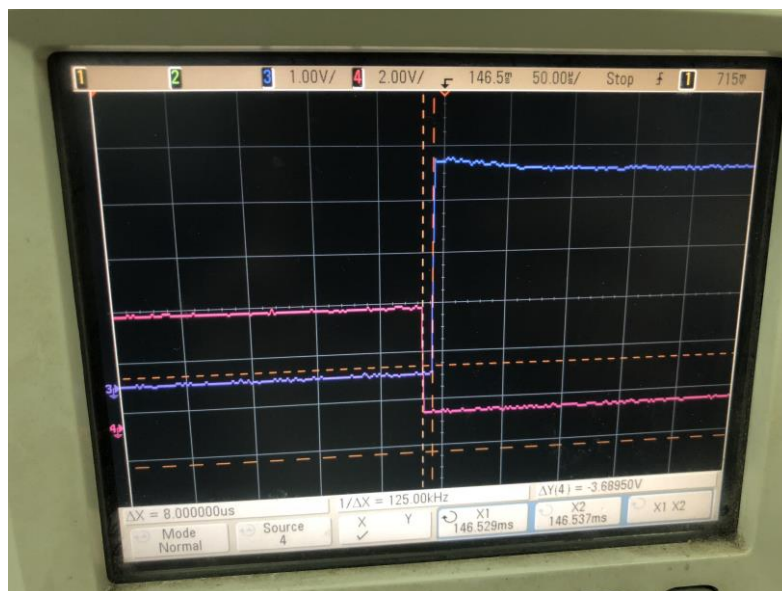


图 2-15 NMI 中断 (PRIMASK=0) _SLEEP_唤醒时间 8us

注：对于 RTCBKP 唤醒时间测试，芯片进入 RTCBKP 后，CPU 会掉电，唤醒后，程序从 main() 函数开始执行，所以在 main 函数中，先配置 GPIO 时钟，紧接着翻 IO(PA0)。各个模式下唤醒时间如表二

表 2-2 各个模式下唤醒时间

中断入口	sleep	deepsleep	RTCBKP
#38 (PRIMASK 置 1)	5 (5.2us)	5 (5.2us)	28.3us
NMI	8 (7.9us)	8 (7.88us)	33.8us

RTCBKP 两种中断入口时间差别是因为，NMI 唤醒复位后会执行相应的中断服务程序，但 #38 被屏蔽，所以时间上略有差别。

2.4 编程配置

FM33G0XX 上电默认主时钟为 RCHF8MHZ，SWD 为数字功能。为了能顺利下载程序，在主程序里执行休眠、切换主时钟、配置 SWD 口之前需要延时至少 10ms。保证在下载程序时，主时钟和 SWD 口符合编程需求。

2.5 休眠功耗

FM33G0XX 在各模式下的功耗如图 2-16、2-17、2-18、2-19：

LPRUN:

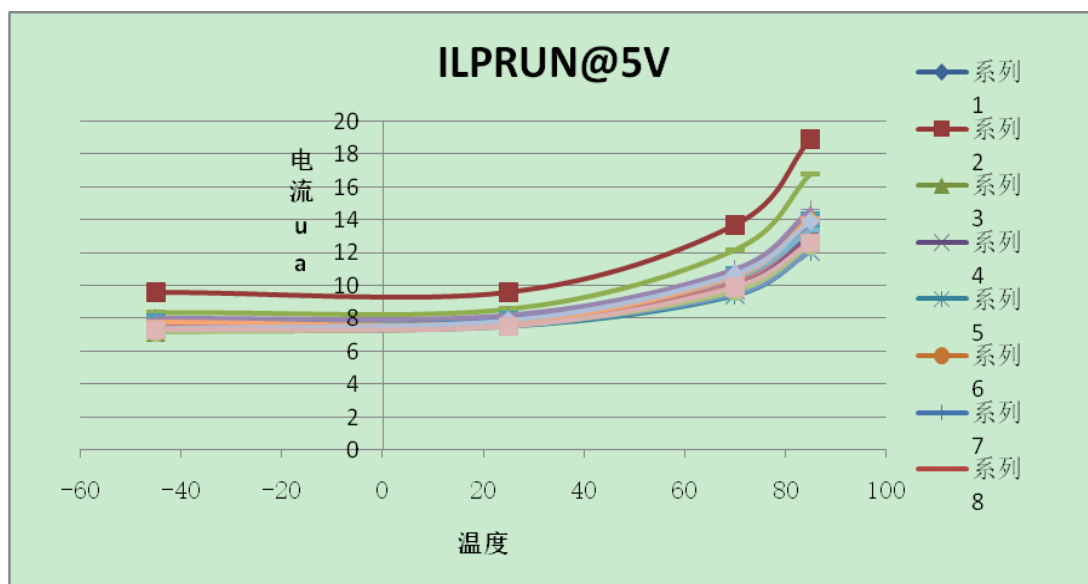


图 2-16 LPRUN@5V 功耗

Sleep:

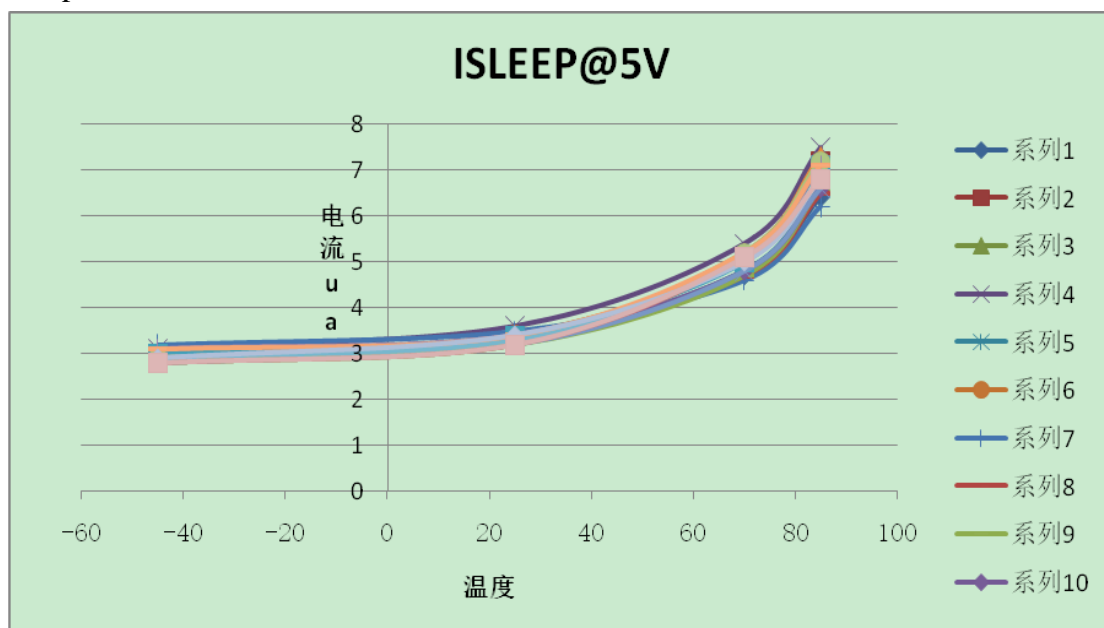


图 2-17 SLEEP@5V 功耗

Deepsleep:

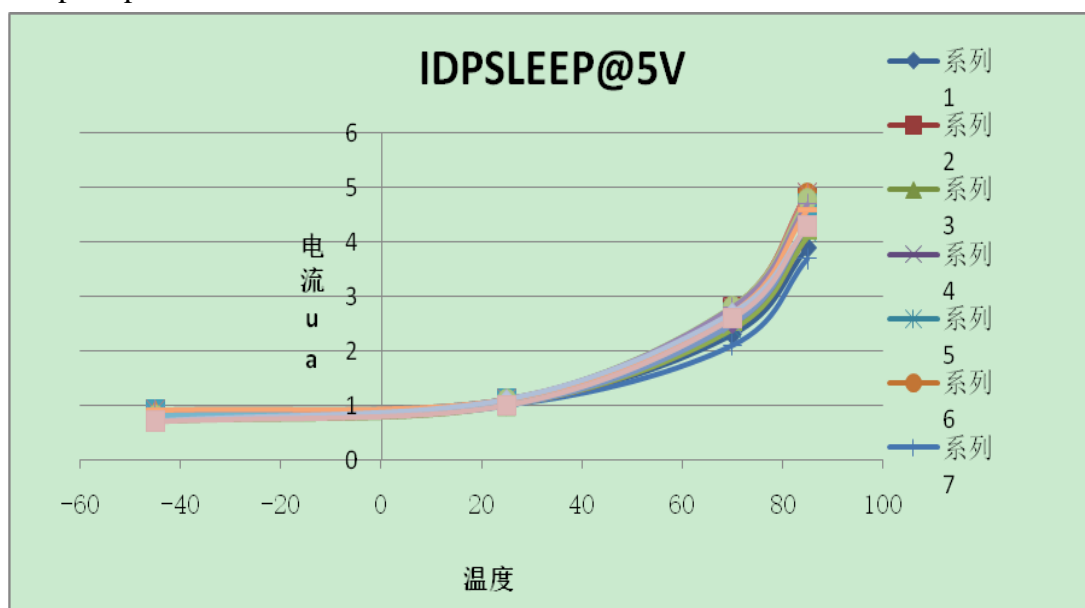


图 2-18 DEEPSLEEP@5V 功耗



RTCBKP:

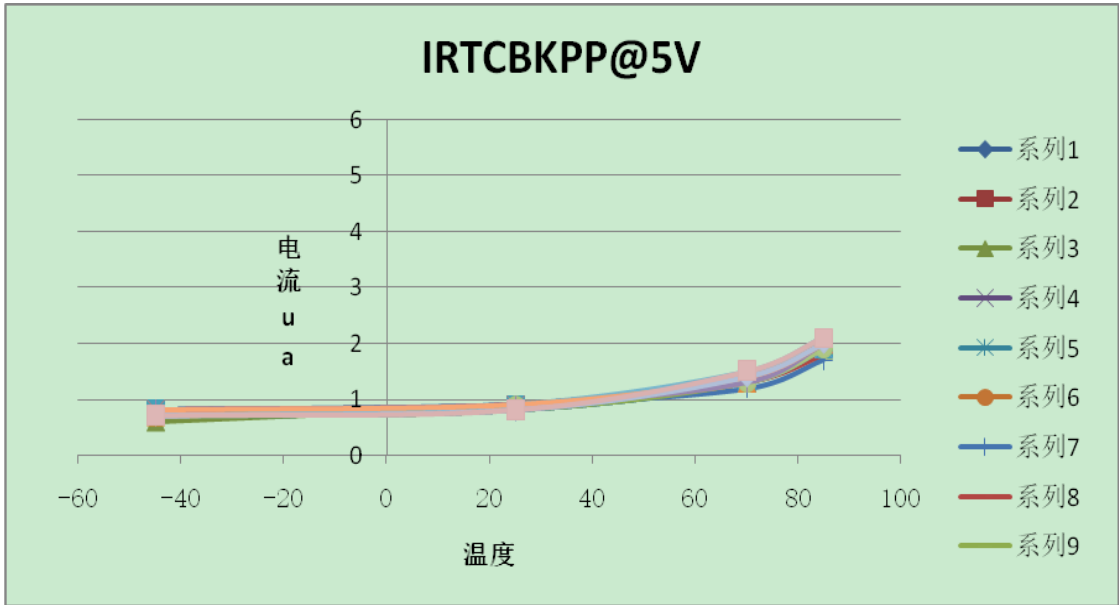


图 2-19 RTCBKP@5V 功耗



版本信息

版本号	发布日期	更改说明
1.0	2018.10	首次发布



附录

1. 芯片寄存器介绍

附 1-1: 电源管理寄存器

地址	名称	符号
0x40000100	低功耗控制寄存器	LPMCFCG
0x40000104	唤醒时间控制寄存器	WKDLYCON
0x40000108	引脚唤醒标志寄存器	WKPFLAG
0x4000010C	PMU 中断使能寄存器	LPREIE
0x40000110	PMU 中断标志寄存器	LPREIF
0x40000D08	WKUP 使能寄存器	PINWKEN

低功耗控制寄存器

名称	LPMCFCG							
地址	0x40000100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							FLSDPS EN
位权限	U-0							R/W-1
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				XTOFF	XTOFF_ B	LDO15E N	LDO15E N_B
位权限	U-0				R/W/Dy-0	R/W/Dy-1	R/Dy-1	R/Dy-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						SLPDP	CVS
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PMOD	
位权限	U-0						R/W-00	

Bit	助记符	功能描述
31:25	--	未实现, 读为 0
24	FLSDPS EN	低功耗模式下 Flash DeepStandby 使能 1: 低功耗模式下 Flash 进入 DPSTB 0: 低功耗模式下 Flash 不进入 DPSTB 通常情况下建议用户不要改写这个 bit



Bit	助记符	功能描述
23:20	--	未实现，读为 0
19	XTOFF	关闭 XTLP，仅在 SLEEP/DEEPSLEEP 下起作用 1: 关闭 XTLP 0: 保持 XTLP 开启 若为无晶振产品，无论 XTOFF 是什么值，均关闭 XTLP
18	XTOFFB	XTOFF 反码校验位
17	LDO15EN	LDO15 使能标志位 1: LDO15 处于工作状态 0: LDO15 被关闭
16	LDO15EN_B	LDO15 使能标志反码校验位
15:10	--	未实现，读为 0
9	SLPDP	DeepSleep 控制寄存器 1: DeepSleep 模式使能，下关闭基准电压源 0: 常规 Sleep 模式 在 Sleep 下，如果置位了 SLPDP 位即为 DeepSleep 模式； 该位仅在 Sleep 下有效
8	CVS	CoreVoltageScaling 配置 0: 低功耗模式下不使能内核电压调整 1: 低功耗模式下降低内核电压 该位仅在 Sleep/DeepSleep/RTCBKP 下起作用
7:2	--	未实现，读为 0
1:0	PMOD	低功耗模式配置寄存器 00: Active mode 01: LPRUN mode 10: Sleep mode 11: RTCBKP mode



唤醒时间控制寄存器

名称	WKDLYCON							
地址	0x40000104							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						T1a	
位权限	U-0						R/W-01	

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1:0	T1a	RTCBKP 模式唤醒过程中的可编程延迟，不建议用户程序修改 00: 14us 01: 16us 10: 24us 11: 32us

引脚唤醒标志寄存器

名称	WKPFLAG							
地址	0x40000108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							DBGWK F



位权限	U-0							R/W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKP7F	WKP6F	WKP5F	WKP4F	WKP3F	WKP2F	WKP1F	WKP0F
位权限	R/W1C-00000000							

Bit	助记符	功能描述
31:9	--	未实现，读为 0
8	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零
7:0	WKPxF	WKUPx Pin 唤醒标志寄存器，仅 Sleep/DeepSleep 模式下有效，对应引脚上的有效唤醒事件将置位寄存器，软件写 1 清零

PMU 中断使能寄存器

名称	LPREIE							
地址	0x4000010C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RTCBKP EIE	SLPEIE	LPREIE
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	未实现，读为 0
2	RTCBKPEIE	RTCBKP 错误中断使能 1: 使能 RTCBKP 错误中断 0: 禁止 RTCBKP 错误中断
1	SLPEIE	SLEEP 错误中断使能 1: 使能 SLEEP 错误中断



Bit	助记符	功能描述
		0: 禁止 SLEEP 错误中断
0	LPREIE	LPRUN 错误中断使能 1: 使能 LPRUN 错误中断 0: 禁止 LPRUN 错误中断

PMU 中断标志寄存器

名称	LPREIF							
地址	0x40000110							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RTCBKP EIF	SLPEIF	LPREIF
位权限	U-0					R/W1C-0	R/W1C-0	R/W1C-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
2	RTCBKPEIF	RTCBKP 错误中断标志，硬件置位，软件写 1 清零 1: 在 PMOD=2'h3 时，CPU 内核 SLEEPDEEP 寄存器没有置位 0: 在 PMOD=3'h3 时，CPU 内核 SLEEPDEEP 寄存器置位
1	SLPEIF	SLEEP 错误中断标志，硬件置位，软件写 1 清零 1: 在 PMOD=2'h2 后，CPU 执行 WFI/WFE 指令前置位了 SLEEPDEEP 寄存器时置位 0: 在 PMOD=2'h2 后，CPU 正确进入 SLEEP
0	LPREIF	LPRUN 错误中断标志，硬件置位，软件写 1 清零；软件进入 LPRUN 模式时如果触发了 LPREIF，则芯片仍将停留在 ACTIVE 模式



Bit	助记符	功能描述
		1: LPRUN Condition Error, 即进入 LPRUN 时满足如下情况: 1) HSCLK 选择不是 LSCLK 或 RCLP 2) RCHF、PLL、ADC 使能未关闭 0: LPRUN 正常进入

WKUP 使能寄存器

名称	Name: PINWKEN							
地址	0x40000D08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKISEL							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PINWKSEL[7:0]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN[7:0]							
位权限	R/W-00000000							

Bit	助记符	功能描述
31	WKISEL	WKUPx 中断入口选择 0: NMI 中断 1: #38 中断
30:16	--	RFU: 未实现, 读为 0
15:8	PINWKSEL	WKUP 边沿选择 1: 对应的 WKUP 引脚为上升沿唤醒 0: 对应的 WKUP 引脚为下降沿唤醒
7:0	PINWKEN	WKUP 引脚使能信号 1: 对应的 WKUP 引脚功能有效 0: 对应的 WKUP 引脚功能无效 PINWKEN[x]控制 WKUPx 引脚的使能



2 库函数介绍

寄存器	描述
LPMCFG	低功耗控制寄存器
WKDLYCON	唤醒时间控制寄存器
WKPFLAG	引脚唤醒源标志寄存器
LPRIE	PMU 中断使能寄存器
LPRIF	PMU 中断标志寄存器

	函数名	描述
1.	PMU_SleepCfg_Init	低功耗模式初始化
2.	PMU_Deinit	PMU 初始化为缺省值
3.	PMU_LPMCFG_PMOD_Set	设置低功耗模式
4.	PMU_LPMCFG_PMOD_Get	获取低功耗模式设置
5.	PMU_LPMCFG_CVS_Setable	设置低功耗模式内核电压开关
6.	PMU_LPMCFG_CVS_Getable	获取低功耗模式内核电压开关设置
7.	PMU_LPREIE_LPREIE_Setable	设置 LP Run 错误中断使能
8.	PMU_LPREIE_LPREIE_Getable	获取 LP Run 错误中断使能状态
9.	PMU_LPREIF_LPREIF_Clr	清除 LP Run 错误中断标志
10.	PMU_LPREIF_LPREIF_Chk	获取 LP Run 错误中断标志
11.	PMU_LPMCFG_SLPDP_Set	设置是否发生 DeepSleep
12.	PMU_LPMCFG_SLPDP_Get	获取是否发生 DeepSleep 的设置
13.	MU_LPREIE_SLPEIE_Setable	设置 Sleep 错误中断使能
14.	MU_LPREIE_SLPEIE_Getable	获取 Sleep 错误中断使能状态
15.	PMU_LPREIF_SLPEIF_Clr	清除 Sleep 错误中断标志
16.	PMU_LPREIF_SLPEIF_Chk	获取 Sleep 错误中断标志
17.	PMU_LPREIE_RTCBKPEIE_Setable	设置 RTCBK 错误中断使能
18.	PMU_LPREIE_RTCBKPEIE_Getable	获取 RTCBK 错误中断使能标志
19.	PMU_LPREIF_RTCBKPEIF_Clr	清除 RTCBK 错误中断标志
20.	PMU_LPREIF_RTCBKPEIF_Chk	获取 RTCBK 错误中断标志
21.	PMU_WKDLYCON_T1a_Set	设置从 RTCBKP 唤醒的可编程延时时间
22.	PMU_WKDLYCON_T1a_Get	从 RTCBKP 唤醒的可编程延时时间
23.	PMU_WKPFLAG_DBGWKF_Clr	清除 CPU Debugger 唤醒标志
24.	PMU_WKPFLAG_DBGWKF_Chk	获取 CPU Debugger 唤醒标志



25.	PMU_WKPFLAG_WKPxF_ClrEx	清除 NWKUPx Pin 唤醒标志
26.	PMU_WKPFLAG_WKPxF_ChkEx	获取 NWKUPx Pin 唤醒标志
27.	PMU_LPMCFG_XTOFF_Set	设置 XTLP 使能
28.	PMU_LPMCFG_XTOFF_Get	获取 XTLP 使能状态
29.	PMU_LPMCFG_LDO15EN_Chk	获取 LDO15 使能状态

PMU_SleepCfg_Init

函数名	PMU_SleepCfg_Init
函数原型	void PMU_SleepCfg_Init(PMU_SleepCfg_InitTypeDef* SleepCfg_InitStruct)
功能描述	低功耗模式初始化
输入参数 1	<p>低功耗模式参数：</p> <p>低功耗模式设置：</p> <p>00: Active mode</p> <p>01: LPRUN mode</p> <p>10: Sleep mode</p> <p>11: RFU</p> <p>DeepSleep 控制：</p> <p>0: 常规 Sleep 模式</p> <p>1: DeepSleep 模式</p> <p>DeepSleep 模式下 RAM 数据保持设置：</p> <p>0: DeepSleep 模式下保持所有 RAM 数据</p> <p>1: DeepSleep 模式下保持 16KB RAM 数据</p> <p>CVS (CoreVoltageScaling) 设置：</p> <p>0: 低功耗模式下不能能内核电压调整</p> <p>1: 低功耗模式下降低内核电压</p> <p>禁止 XTLP，仅在 Sleep/DeepSleep 下起作用：</p> <p>0: 保持 XTLP 开启</p> <p>1: 禁止 XTLP</p> <p>MO 系统控制寄存器：一般设置为 0</p>
输出参数	无
返回值	无



PMU_Deinit

函数名	PMU_Deinit
函数原型	void PMU_Deinit(void)
功能描述	PMU 初始化为缺省值
输入参数 1	无
输出参数	无
返回值	无

PMU_LPMCFG_PMOD_Set

函数名	PMU_LPMCFG_PMOD_Set
函数原型	void PMU_LPMCFG_PMOD_Set(uint32_t SetValue)
功能描述	设置低功耗模式
输入参数 1	低功耗模式： 00: Active mode 01: LPRUN mode 10: Sleep mode 11: RFU
输出参数	无
返回值	无

PMU_LPMCFG_PMOD_Get

函数名	PMU_LPMCFG_PMOD_Get
函数原型	uint32_t PMU_LPMCFG_PMOD_Get(void)
功能描述	获取低功耗模式设置
输入参数 1	无
输出参数	无
返回值	低功耗模式： 00: Active mode 01: LPRUN mode 10: Sleep mode 11: RFU



PMU_LPMC_CFG_CVS_Setable

函数名	PMU_LPMC_CFG_CVS_Setable
函数原型	void PMU_LPMC_CFG_CVS_Setable(FunState NewState)
功能描述	设置低功耗模式内核电压开关
输入参数 1	CVS (CoreVoltageScaling) : 0: 低功耗模式下不能能内核电压调整 1: 低功耗模式下降低内核电压
输出参数	无
返回值	无

PMU_LPMC_CFG_CVS_Getable

函数名	PMU_LPMC_CFG_CVS_Getable
函数原型	FunState PMU_LPMC_CFG_CVS_Getable(void)
功能描述	获取低功耗模式内核电压开关设置
输入参数 1	无
输出参数	无
返回值	CVS (CoreVoltageScaling) 设置: 0: 低功耗模式下不能能内核电压调整 1: 低功耗模式下降低内核电压

PMU_LPREIE_LPREIE_Setable

函数名	PMU_LPREIE_LPREIE_Setable
函数原型	void PMU_LPREIE_LPREIE_Setable(FunState NewState)
功能描述	设置 LP Run 错误中断使能
输入参数 1	LP Run 错误中断使能状态: DISABLE: 禁止 LP Run 错误中断 ENABLE: 使能 LP Run 错误中断
输出参数	无
返回值	无



PMU_LPREIE_LPREIE_Getable

函数名	PMU_LPREIE_LPREIE_Getable
函数原型	FunState PMU_LPREIE_LPREIE_Getable(void)
功能描述	获取 LP Run 错误中断使能状态
输入参数 1	无
输出参数	无
返回值	LP Run 错误中断使能状态： DISABLE: 禁止 LP Run 错误中断 ENABLE: 使能 LP Run 错误中断

PMU_LPREIF_LPREIF_Clr

函数名	PMU_LPREIF_LPREIF_Clr
函数原型	void PMU_LPREIF_LPREIF_Clr(void)
功能描述	清除 LP Run 错误中断标志
输入参数 1	无
输出参数	无
返回值	无

PMU_LPREIF_LPREIF_Chk

函数名	PMU_LPREIF_LPREIF_Chk
函数原型	FlagStatus PMU_LPREIF_LPREIF_Chk(void)
功能描述	获取 LP Run 错误中断标志
输入参数 1	无
输出参数	无
返回值	LP Run 错误中断使能标志： 0: 未发生 LP Run 错误中断 1: 发生 LP Run 错误中断



PMU_LPMC_CFG_SLPDP_Set

函数名	PMU_LPMC_CFG_SLPDP_Set
函数原型	void PMU_LPMC_CFG_SLPDP_Set(uint32_t SetValue)
功能描述	设置是否发生 DeepSleep
输入参数 1	DeepSleep 控制: 0: 常规 Sleep 模式 1: DeepSleep 模式
输出参数	无
返回值	无

PMU_LPMC_CFG_SLPDP_Get

函数名	PMU_LPMC_CFG_SLPDP_Get
函数原型	uint32_t PMU_LPMC_CFG_SLPDP_Get(void)
功能描述	获取是否发生 DeepSleep 的设置
输入参数 1	无
输出参数	无
返回值	DeepSleep 控制: 0: 常规 Sleep 模式 1: DeepSleep 模式

PMU_LPREIE_SLPEIE_Setable

函数名	MU_LPREIE_SLPEIE_Setable
函数原型	void PMU_LPREIE_SLPEIE_Setable(FunState NewState)
功能描述	设置 Sleep 错误中断使能
输入参数 1	Sleep 错误中断使能状态: DISABLE: 禁止 Sleep 错误中断 ENABLE: 使能 Sleep 错误中断
输出参数	无
返回值	无



PMU_LPREIE_SLPEIE_Getable

函数名	MU_LPREIE_SLPEIE_Getable
函数原型	FunState PMU_LPREIE_SLPEIE_Getable(void)
功能描述	获取 Sleep 错误中断使能状态
输入参数 1	无
输出参数	无
返回值	Sleep 错误中断使能状态： DISABLE: 禁止 LSleep 错误中断 ENABLE: 使能 Sleep 错误中断

PMU_LPREIF_SLPEIF_Clr

函数名	PMU_LPREIF_SLPEIF_Clr
函数原型	void PMU_LPREIF_SLPEIF_Clr(void)
功能描述	清除 Sleep 错误中断标志
输入参数 1	Sleep 错误中断使能标志： 0: 未发生 Sleep 错误中断 1: 发生 Sleep 错误中断
输出参数	无
返回值	无

PMU_LPREIF_SLPEIF_Chk

函数名	MU_LPREIF_SLPEIF_Chk
函数原型	FlagStatus PMU_LPREIF_SLPEIF_Chk(void)
功能描述	获取 Sleep 错误中断标志
输入参数 1	无
输出参数	无
返回值	Sleep 错误中断使能标志： 0: 未发生 Sleep 错误中断 1: 发生 Sleep 错误中断



PMU_LPREIE_RTCKPEIE_Setable

函数名	PMU_LPREIE_RTCKPEIE_Setable
函数原型	void PMU_LPREIE_RTCKPEIE_Setable(FunState NewState)
功能描述	设置 RTCK 错误中断使能
输入参数 1	RTCK 错误中断使能： DISABLE: 禁止 RTCK 错误中断 ENABLE: 使能 RTCK 错误中断
输出参数	无
返回值	无

PMU_LPREIE_RTCKPEIE_Getable

函数名	PMU_LPREIE_RTCKPEIE_Getable
函数原型	FunState PMU_LPREIE_RTCKPEIE_Getable(void)
功能描述	读取 RTCK 错误中断使能的设置
输入参数 1	无
输出参数	无
返回值	RTCK 错误中断使能： DISABLE: 禁止 RTCK 错误中断 ENABLE: 使能 RTCK 错误中断

PMU_LPREIF_RTCKPEIF_Clr

函数名	PMU_LPREIF_RTCKPEIF_Clr
函数原型	void PMU_LPREIF_RTCKPEIF_Clr(void)
功能描述	清除 RTCK 错误中断标志
输入参数 1	无
输出参数	无
返回值	无



PMU_LPREIF_RTCBKPEIF_Chk

函数名	PMU_LPREIF_RTCBKPEIF_Chk
函数原型	FlagStatus PMU_LPREIF_RTCBKPEIF_Chk(void)
功能描述	读取 RTCBK 错误中断标志的设置
输入参数 1	无
输出参数	无
返回值	RTCBK 错误中断标志： RESET: 未产生 RTCBK 错误中断 SET: 产生 RTCBK 错误中断

PMU_WKDLYCON_T1a_Set

函数名	PMU_WKDLYCON_T1a_Set
函数原型	void PMU_WKDLYCON_T1a_Set(uint32_t SetValue)
功能描述	设置从 RTCBKP 唤醒的可编程延时时间
输入参数 1	DeepSleep 唤醒时，从 BG 使能到开启 LDO 并开始读取 Flash 的时间间隔： 00: 12us 01: 16us 10: 24us 11: 32us
输出参数	无
返回值	无

PMU_WKDLYCON_T1a_Get

函数名	PMU_WKDLYCON_T1a_Get
函数原型	uint32_t PMU_WKDLYCON_T1a_Get(void)
功能描述	获取从 RTCBKP 唤醒的可编程延时时间
输入参数 1	无
输出参数	无
返回值	DeepSleep 唤醒时，从 BG 使能到开启 LDO 并开始读取 Flash 的时间间隔： 00: 12us 01: 16us 10: 24us 11: 32us



PMU_WKPFLAG_DBGWKF_Clr

函数名	PMU_WKPFLAG_DBGWKF_Clr
函数原型	void PMU_WKPFLAG_DBGWKF_Clr(void)
功能描述	清除 CPU Debugger 唤醒标志
输入参数 1	无
输出参数	无
返回值	无

PMU_WKPFLAG_DBGWKF_Chk

函数名	PMU_WKPFLAG_DBGWKF_Chk
函数原型	FlagStatus PMU_WKPFLAG_DBGWKF_Chk(void)
功能描述	获取 CPU Debugger 唤醒标志
输入参数 1	无
输出参数	无
返回值	CPU Debugger 唤醒标志： 0: CPU Debugger 未唤醒 1: CPU Debugger 唤醒

PMU_WKPFLAG_WKPxF_ClrEx

函数名	PMU_WKPFLAG_WKPxF_ClrEx
函数原型	void PMU_WKPFLAG_WKPxF_ClrEx(uint32_t NWKPinDef)
功能描述	清除 NWKUPx Pin 唤醒标志
输入参数 1	NWKUPx Pin 唤醒标志： 0: 不清除 NWKUPx Pin 唤醒标志 1: 清除 NWKUPx Pin 唤醒标志
输出参数	无
返回值	无



PMU_WKPFLAG_WKPxF_ChkEx

函数名	PMU_WKPFLAG_WKPxF_ChkEx
函数原型	FlagStatus PMU_WKPFLAG_WKPxF_ChkEx(uint32_t NWKPinDef)
功能描述	获取 NWKUPx Pin 唤醒标志
输入参数 1	无
输出参数	无
返回值	NWKUPx Pin 唤醒标志： 0: 不清除 NWKUPx Pin 唤醒标志 1: 清除 NWKUPx Pin 唤醒标志

PMU_LPMCFG_XTOFF_Set

函数名	PMU_LPMCFG_XTOFF_Set
函数原型	void PMU_LPMCFG_XTOFF_Set(uint32_t SetValue)
功能描述	设置 XTTF 使能
输入参数 1	禁止 XTTF, 仅在 Sleep/DeepSleep 下起作用： 0: 保持 XTTF 开启 1: 禁止 XTTF
输出参数	无
返回值	无

PMU_LPMCFG_XTOFF_Get

函数名	PMU_LPMCFG_XTOFF_Get
函数原型	uint32_t PMU_LPMCFG_XTOFF_Get(void)
功能描述	获取 XTTF 使能状态
输入参数 1	无
输出参数	无
返回值	禁止 XTTF, 仅在 Sleep/DeepSleep 下起作用： 0: 保持 XTTF 开启 1: 禁止 XTTF



PMU_LPMCFG_LDO15EN_Chk

函数名	PMU_LPMCFG_LDO15EN_Chk
函数原型	FlagStatus PMU_LPMCFG_LDO15EN_Chk(void)
功能描述	获取 LDO15 使能状态
输入参数 1	无
输出参数	无
返回值	LDO15 使能状态: DISABLE: 禁止 LDO15 ENABLE: 使能 LDO15



上海复旦微电子集团股份有限公司销售及服 务 网 点

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：（86-10）8418 6608

传真：（86-10）8418 6211

深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888

新加坡办事处

地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929

电话：(65) 6472 3688

传真：(65) 6472 3669

北美办事处

地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话：(480) 857-6500 ext 18

公司网址：<http://www.fmsh.com/>