

# FM33LC0XX 开发注意事项

版本：V0.7

FM33LC0XX 系列芯片是一款 M0 内核的 ARM 芯片，在客户开发中通常会遇到一些普遍的问题，本文将分硬件、编译环境、软件简单阐述这些问题，加快客户的开发过程。详细细节可以参考相关手册、例程。

## 改版记录

2020.7	初版	
2020.8	V0.1	删除 UART0、1 的休眠唤醒功能
2020.9	V0.2	新增 VREF1P2 章节
2020.11	V0.3	1 休眠下建议不要关闭 LPOSC（RCLP） 2 VREF1P2 的建立标志直接使用延时
2020.11	V0.4	VDD15 需要外接 4.7uF
2021.8	V0.5	ADC 模块，在 5V 系统中使用 ADC 前使能 ADSVD_EN
2021.8	V0.6	ADC 模块，在 5V 系统中需要使能 ADSVD_EN，且使能 ADSVD_EN 时 VREF1P2 必须使能
2022.4	V0.7	重新排版，将软硬件中部分注意点集中起来放在文档开头醒目的位置

## 1. 特别注意

开发时需要特别注意以下 3 点，否则可能会影响芯片正常运行：

- 1 NRST 引脚禁止外接电容、硬件看门狗之类的电路
- 2 LDO15 引脚必须外接 4.7uF 接地电容
- 3 休眠时禁止关闭 LPOSC（可参考本文档软件章节中休眠下禁止关闭 LPOSC 的内容）

## 2. 硬件

### 电源

FM33LC0XX 系列芯片电源有 VDD、VDDA 和 VDD15。VDD 为 MCU 的电源，VDDA 是模拟电路电源通常直接和 VDD 相连接，也可以外部接基准源。VDD15 为 VDD 通过芯片内部 LDO 输出的 1.5V 电压用于数字电路的电源。

在硬件设计时 VDD 和 VDD15 不能直接连接，VDD15 必须外接 4.7uF 电容，详细见推荐原理图。

## 编程接口

FM33LC0XX 系列芯片的编程接口使用的是 SWD 接口，芯片在处于低功耗状态时编程口不能正常工作，所以在通常的 SWD 接口的基础上增加了 NRST 引脚。通过 NRST 引脚去复位芯片，才能进行仿真器的操作。

假如客户需要使用 RTC，并对时间的精度要求比较高，还需要增加 FOUT 接口，使用 FOUT 接口可以进行外接晶体的温度调校。

详细见推荐原理图。

## GPIO

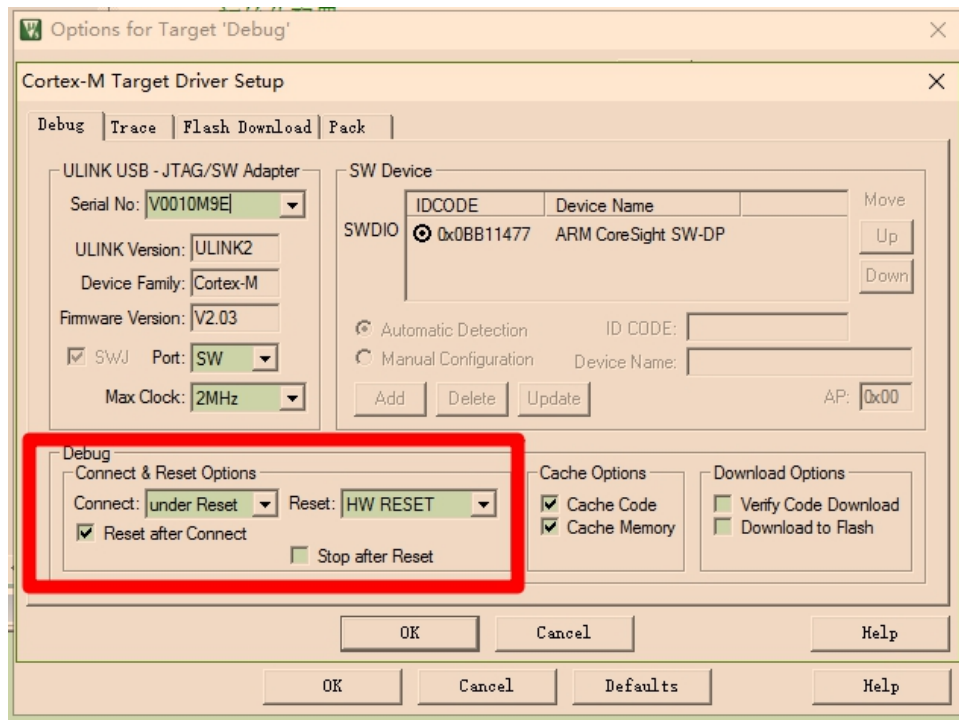
PA11 和 PA12 的 GPIO 引脚是真开漏引脚，假如需要输出高电平必须外接上拉电阻。其余 GPIO 均可设置推挽或者开漏输出。

## 3. 编译器环境配置

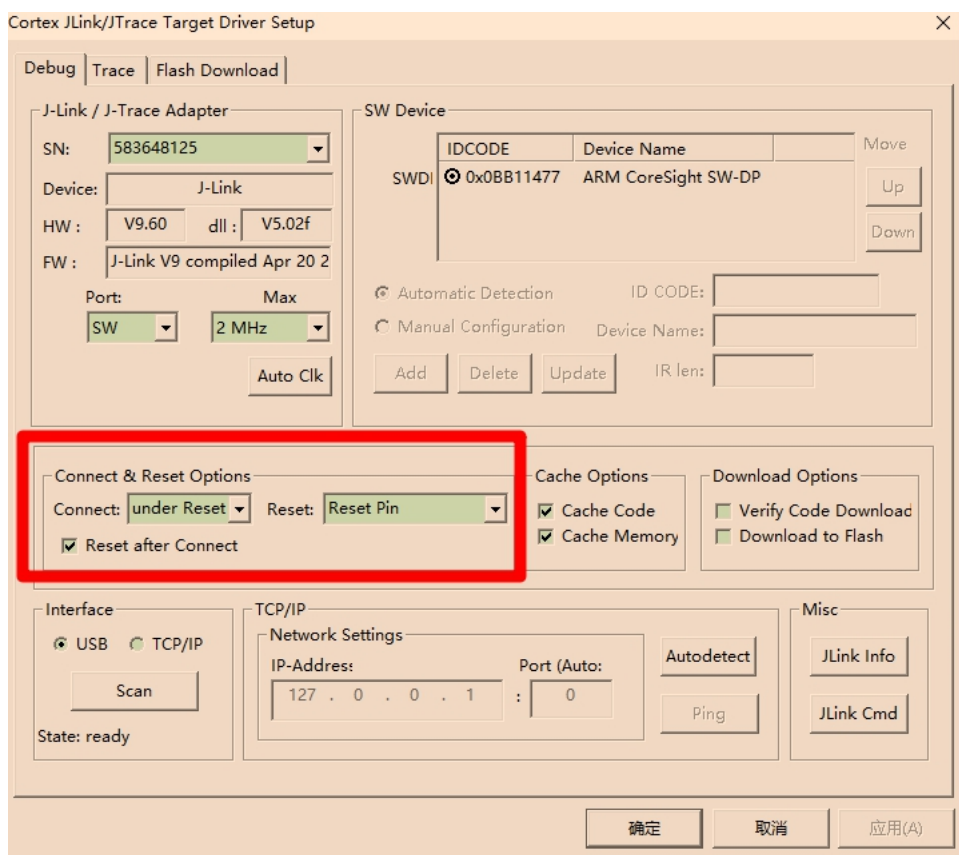
Keil 和 IAR 两种编译器 FM33LC0XX 都可以支持。为了在休眠下能够顺利连接仿真器，除了在硬件上编程口需要连接 RST 脚。在编译器的配置上需要如下设置。

### Keil

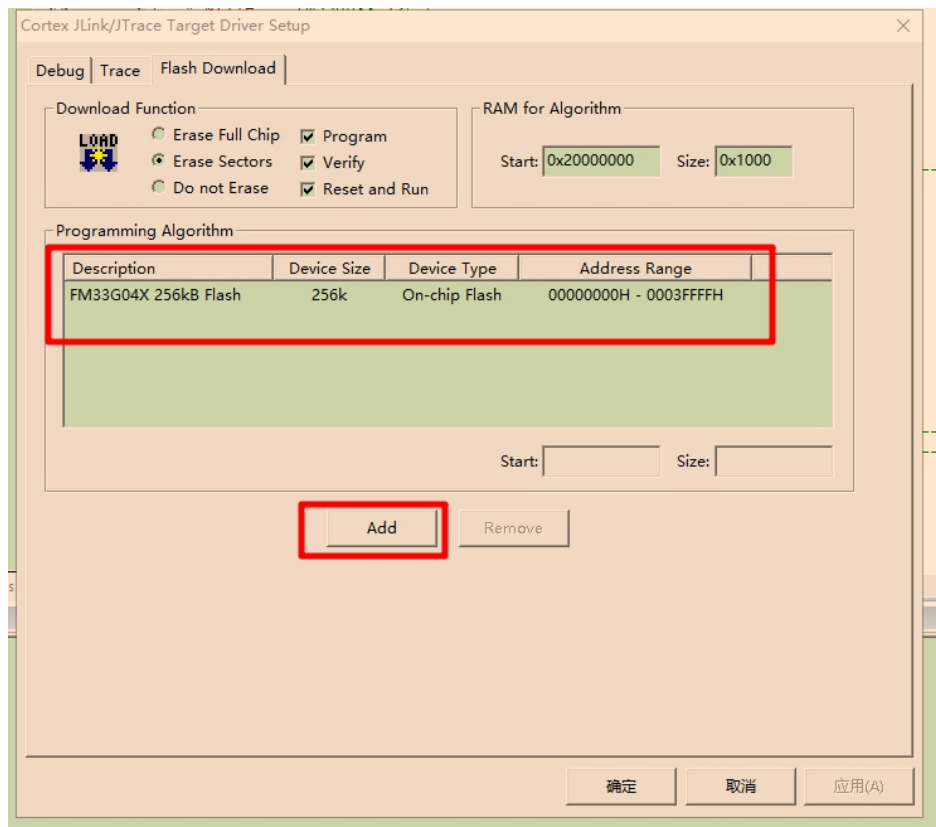
Keil 在使用 ULINK 时 debug 选项中 Connect 和 Reset,建议按下图配置



Keil 在使用 JLINK 时 debug 选项中 Connect 和 Reset,建议按下图配置



在 Flash Download 中最好确认下 Flash 的配置信息和芯片选型是否一致，假如不一致需要点击 Add 手动添加。



## IAR

IAR Reset 选择 Reset pin 如下图



## 4. 软件

### 系统主时钟

FM33LC0XX 系列可以选择多种时钟源做为主时钟，常用的为 RCHF、XTHF 和 PLL。RCHF 的频率为 8MHZ、16MHZ、24MHZ，其中 8MHZ 和 16MHZ 全温区的变化范围小于 $\pm 2\%$ 。XTHF 可外接 4-32MHZ 晶体。PLL 的时钟源必须是 RCHF 和 XTHF 分频后的 1MHZ，PLL 频率的范围 32MHZ-64MHZ。

RCHF 和 PLL(时钟源为 RCHF 或 XTHF)做系统主时钟静电特性较好，XTHF 做系统主时钟的静电特性稍差。从功耗看 RCHF 的功耗较低。

RCHF 的每一档频率出厂前都有校准值，保证在常温时的精度，8MHZ 的 TRIM 为上电自动加载，其余频率需要程序手动加载。可以参看例程。

### 仿真时定时器与看门狗

在仿真状态下定时器和看门狗默认为关闭状态，可以操作 DBG\_CR 来改变状态。可以参看例程。

### 复位

FM33LC0XX 有上电复位和下电复位电路，在程序中下电复位必须要打开。下电复位有 BOR 和 PDR，至少选择打开其中的一个。BOR 阈值比较准，但功耗高大约 2 $\mu$ A。PDR 阈值有 100~200MV 的偏差，但功耗低小于 100nA。

### UART

FM33LC0XX 有 4 路独立 UART，每一路 UART 都具有红外调制输出功能。

UART 的中断标志位有 buffer 空产生和移位寄存器空产生两种，建议使用移位寄存器空。

UART 有 6bit、7bit、8bit、9bit 四种模式

UART0 和 UART1 支持接收超时和发送延时功能

## LPUART

FM33LC0XX 的 LPUART 是低功耗串口最高波特率为 9600。LPUART 的工作时钟为 32K，需要注意的是 LPUART 的波特率的每一个 bit 不是等长的，为了消除波特率的累计误差，波特率的每一 bit 的长度是通过特别调制，可以参看例程。

## RTC

如果应用代码中没有对时钟调校寄存器操作，在每次上电时 RTC 计时精度将因为时钟调校寄存器的随机值发生变化。在需要 RTC 时钟精度的应用中注意时钟调校寄存器的复位值问题，或者写 0，或者按温度进行相关的数据填充。

## FLASH

FLASH 的最小编程单位是 32bit。编程前必须对 FLASH 进行擦除，FLASH 支持：全擦、扇区擦、页擦。

如果使用 DMA 通道 7 来对 Flash 进行连续编程，当通道 7 的 DIR 寄存器为 0，只要置位了 DMA 全局使能寄存器(DMA\_GCR.EN)，即使没有置位通道 7 使能寄存器(DMA\_CH7CR.EN)，也能够启动 Flash 编程。即可以在 DIR=0 时，认为通道 7 总是使能的。使用 DMA 时需要注意这一点避免发生 keyerr 错误。用户可以将 DMA 的 7 通道 DIR 改为 1，可以避免这个问题。

在操作 FLASH 时必须注意 RCHF 需要使能。

## LCD

最大支持  $8 \times 28$ 、 $6 \times 30$ 、 $4 \times 32$  的显示段数,可以在 Active 模式、Sleep 模式和 DeepSleep 模式下工作。

选择 4COM 模式时只能选择 COM0-3,6COM 模式选择 COM0-5，8COM 选择 COM0-7。在和 LCD 连接时 COM 和 SEG 可以不必要按照一一对应连接,如在 4COM 模式连接 LCD 的 COM0，MCU 的 COM0-3 都可以和 LCD 的 COM0 连接，SEG 和 COM 一样。

特别需要注意的是，没有和 LCD 连接的 COM 和 SEG，GPIO 不能配置为 LCD 功能，相应的 COM 和 SEG 也不能使能，否则会影响 LCD 的正常显示。

## ADC

FM33LC0XX 带有 1Msps 12bit SAR-ADC，可实现温度、电池电压或其他直流信号的测量功能。使用方法和注意点可以参看相应 AN。

## 低功耗模式

FM33LC0XX 的低功耗有 SLEEP 和 DEEPSLEEP 两种模式，SLEEP 模式为 6ua，DEEPSLEEP 模式(RTC 走时+全部 RAM 保持+CPU 内核保持)为 1ua。

内核和外设的都进入低功耗，MCU 才能得到理想的低功耗，PMU\_CR.PMOD 的配置使得外设进入低功耗，SCR 和 WFI 的执行使得内核进入低功耗。具体的实现方法可以参看例程。

获得理想功耗的注意点：

- 1 在休眠时不使用的 GPIO 配置高阻态，高阻态的配置有两种方式：GPIO 配置为输入模式并且输入禁止，或者 GPIO 配置为输出模式，开漏输出高电平；
  - 2 休眠时 SWD 的 SWCLK 和 SWIO 配置数字功能加上拉；
  - 3 注意 ADC 和 VREF 的 BUFFER 使能是否关闭；
  - 4 低功耗下不需要工作的电路，设计为无电或者电源可控状态；
- 休眠时外设和 GPIO 的寄存器都能保持不变。

FM33LC0XX 使用中断事件唤醒休眠，MCU 休眠唤醒后默认的主时钟为 RCHF，可以通过 PMU\_CR.WKFDEL 配置唤醒后的频率为 8MHZ、16MHZ 或 24MHZ。

唤醒后先进入中断，再进入主循环。也可以通过屏蔽中断(void \_\_disable\_irq(void);)，唤醒后直接执行 WFI 后的第一条指令。

## VREF1P2

VREF1P2 主要用于 ADC 的参考电压、SVD 和 BOR 下电复位基准电压。VREF1P2 在 3V 下约 1.8ua，建立时间约 1.8ms。

3 个标志可以查询 VREF1P2 的建立时间：FLAG\_B、RDY、IF。FLAG\_B 是模拟信号直接输出的信号，0 代表建立完成。RDY 是数字信号 1 代表建立完成。FLAG\_B 和 RDY 都是只读标志。IF 是数字输出，1 表示建立完成，可读可写并可产生中断。

在关注功耗的应用下可能会有关闭 VREF1P2 的需求，如休眠时关闭 VREF1P2，唤醒后再



使能。

重新使能 VREF1P2 时需要注意 RDY 标志的清除和建立需要 LPOSC 的一个 CLOCK。即关闭后立即使能的时间小于一个 LPOSC 的 CLOCK，RDY 没有被清除，这时通过查询 RDY 确定 VREF1P2 是否建立将会判断错误。假如关闭 VREF1P2 时又关闭 LPOSC，RDY 也不会被清除，唤醒后 LPOSC 自动使能，但 LPOSC 的建立时间大约 150us，在 LPOSC 建立前查询 RDY 也会判断错误。

在实际使用中可以在 VREF1P2 使能后直接延时 2ms 即可以认为建立完成。

## 低功耗定时器

FM33LC0XX 除了 LPTIME 可以作为低功耗定时器，BSTIM 把工作时钟设置 LSCLK 或 RC4M 同样可以在低功耗下定时。

## 休眠下禁止关闭 LPOSC

LPOSC 在有些版本中可能写为 RCLP，都是指内部 32K 振荡器。

在旧版本 LL 驱动库 V0.11 包括 V0.11 的休眠例程，以及魔方 FL 驱动库 V0.01 例程中 PMU 例程，在休眠函数中关闭了 LPOSC。

在休眠函数中关闭了 LPOSC 的情况下，假如关闭和唤醒的频率过快，有极低的风险可能会导致 LPOSC 唤醒后不能起振。休眠下禁止关闭 LPOSC，可以更新到最新的例程。

具体修改的内容如下：

LL 库中删除红框中关闭 RCLP 的语句：

```

// Sleep
void Sleep(void)
{
    LL_RCC_SetSleepModeRCLPWorkMode(LL_RCC_RCLP_UNDER_SLEEP_CLOSE); //休眠下关闭RCLP
    LL_PMU_EnablePowerDownReset(RMU); //打开PDR
    LL_RMU_DisableBORPowerDownReset(RMU); //关闭BOR 2uA

    LL_ADC_Disable(ADC); //关闭ADC
    LL_VREF_DisableVREF(VREF); //关闭VREF1p2
    WRITE_REG(VREF->BUFCR, 0); //关闭全部VREFbuf

    LPM_InitStruct.DeepSleep = LL_PMU_SLEEP_MODE_NOMAL;
    LPM_InitStruct.PowerMode = LL_PMU_POWER_MODE_SLEEP AND DEEPSLEEP;
    LPM_InitStruct.WakeupFrequency = LL_PMU_SLEEP_WAKEUP_FREQ_RCHF_8MHZ;
    LPM_InitStruct.WakeupDelay = LL_PMU_WAKEUP_DELAY_TIME_2US;
    LPM_InitStruct.CoreVoltageScaling = DISABLE;
    LL_PMU_Sleep_Init(PMU, &LPM_InitStruct);
}

//DeepSleep
void DeepSleep(void)
{
    LL_PMU_SleepInitTypeDef LPM_InitStruct;
    LL_RCC_SetSleepModeRCLPWorkMode(LL_RCC_RCLP_UNDER_SLEEP_CLOSE); //休眠下关闭RCLP
    LL_PMU_EnablePowerDownReset(RMU); //打开PDR
    LL_RMU_DisableBORPowerDownReset(RMU); //关闭BOR 2uA

    LL_ADC_Disable(ADC); //关闭ADC
    LL_VREF_DisableVREF(VREF); //关闭VREF1p2
    WRITE_REG(VREF->BUFCR, 0); //关闭全部VREFbuf

    LPM_InitStruct.DeepSleep = LL_PMU_SLEEP_MODE_DEEP;
    LPM_InitStruct.PowerMode = LL_PMU_POWER_MODE_SLEEP AND DEEPSLEEP;
    LPM_InitStruct.WakeupFrequency = LL_PMU_SLEEP_WAKEUP_FREQ_RCHF_8MHZ;
    LPM_InitStruct.WakeupDelay = LL_PMU_WAKEUP_DELAY_TIME_2US;
    LPM_InitStruct.CoreVoltageScaling = DISABLE;
    LL_PMU_Sleep_Init(PMU, &LPM_InitStruct);
}

```

FL 库中删除红框中的关闭 RCLP 的语句：

```

// Sleep
void Sleep(void)
{
    FL_RCC_RCMF_Disable(); //关闭RCMF
    FL_RCC_LPOSC_DisableSleepModeWork(); // 休眠下关闭RCLP
    FL_PMU_PDR_EnableReset(RMU); // 打开PDR
    FL_RMU_BORPowerDown_Disable(RMU); // 关闭BOR 2uA

    FL_VREF_Disable(VREF); // 关闭VREF1p2
    FL_PMU_SetLowPowerMode(PMU, FL_PMU_POWER_MODE_SLEEP_OR_DEEPSLEEP);
    __WFI();
}

```

## ADC

系统电源为 5V 的用户需要检查下自己的程序，建议使用 ADC 之前使能 ADSVD\_EN。假如不使能，可能会影响 ADC 的长期可靠性。

需要注意的是 ADSVD 使能时，VREF1P2 必须使能

ADSVD\_EN。寄存器位置见下图：

13.7.1 SVD 配置寄存器 (SVD\_CFGR)

名称	SVD_CFGR							
offset	0x00000000							
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	-				ADSVD_SEL			ADSVD_EN
位权限	U-0				R/W-110			R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LVL				DFEN	MOD	-	
位权限	R/W-0000				R/W-1	R/W-0	U-0	

位号	助记符	功能描述
31:12	-	RFU: 未实现, 读为 0
11:9	ADSVD_SEL	ADC 电源检测档位配置 (ADC supply monitor select) 000: 3.300V 001: 3.514V 010: 3.729V 011: 3.943V 100: 4.157V 101: 4.371V 110: 4.586V 111: 4.800V
8	ADSVD_EN	ADC 电源检测功能, 在 5V 系统供电方案下建议在使用 ADC 之前打开这个功能 (ADC supply monitor enable) 使能 ADC 电源检测功能不需要使能 SVDEN

在 FL 驱动中 FL\_ADC\_Init()函数中已经包含这个功能，见下图：

```
ErrorStatus FL_ADC_Init(ADC_Type *ADCx, FL_ADC_InitTypeDef *ADC_InitStruct)
{
    ErrorStatus status = PASS;
    uint32_t i=0;
    /* 入口合法性检查 */
    assert_param(IS_FL_ADC_INSTANCE(ADCx));
    assert_param(IS_FL_ADC_CONTINUOUSCONVMODE(ADC_InitStruct->conversionMode));
    assert_param(IS_FL_ADC_AUTO_MODE(ADC_InitStruct->autoMode));
    assert_param(IS_FL_ADC_SCANDIRECTION(ADC_InitStruct->scanDirection));
    assert_param(IS_FL_ADC_EXTERNALTRIGCONV(ADC_InitStruct->externalTrigConv));
    assert_param(IS_FL_ADC_OVERSAMPNCOFIG(ADC_InitStruct->oversamplingMode));
    assert_param(IS_FL_ADC_OVERSAMPINGRATIO(ADC_InitStruct->oversamplingMultiplier));
    assert_param(IS_FL_ADC_OVERSAMPINGSHIFT(ADC_InitStruct->oversamplingShift));

    /* 使能 ADC 电源检测 */
    FL_RCC_EnableGroup1BusClock(FL_RCC_GROUP1_BUSCLK_ANAC);
    FL_SVD_EnableADCMonitor(SVD);

    /* 使能 VREF1P2 */
    FL_VREF_ClearFlag_Ready(VREF);
    FL_VREF_Enable(VREF); //置位VREF_EN寄存器, 使能VREF1p2模块

    FL_VREF_EnableTemperatureSensor(VREF); //置位PTAT_EN寄存器
    while(FL_VREF_IsActiveFlag_Ready(VREF) == 0)
    {
        if(i>=128000)
        {
            break;
        }
        i++;
    }
}
```

另外使能 ADSVD 是有功耗的，休眠时需要关闭。唤醒后使用 ADC 时再使能 ADSVD，VREF1P2 同样确保已经使能