

UM324xF 低功耗配置指南

版本：V1.1



UNICMICRO
广芯微电子

广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

版本修订

版本	日期	描述
V1.0	2022.11.25	初始版
V1.1	2023.9.12	更新所有代码块内容

目录

1	概述.....	1
2	进入低功耗前配置.....	1
3	配置唤醒条件.....	2
3.1	睡眠模式（Sleep mode）下唤醒配置.....	2
3.2	停止模式（Stop mode）下唤醒配置.....	3
3.3	待机模式 0（Standby0 mode）下唤醒配置.....	4
3.4	待机模式 1（Standby1 mode）下唤醒配置.....	5
3.5	深度待机模式 0（DeepStandby0 mode）下唤醒配置.....	5
3.6	深度待机模式 1（DeepStandby1 mode）下唤醒配置.....	6
4	进入低功耗模式.....	7
4.1	睡眠模式（Sleep mode）.....	7
4.2	停止模式（Stop mode）.....	7
4.3	待机模式 0（Standby0 mode）.....	8
4.4	待机模式 1（Standby1 mode）.....	9
4.5	深度待机模式 0（DeepStandby0 mode）.....	9
4.6	深度待机模式 1（DeepStandby1 mode）.....	10
5	被唤醒后的操作.....	12

1 概述

本文主要介绍 UM324xF 的低功耗配置指南，低功耗配置分四步：进入低功耗前配置，配置唤醒条件，进入休眠模式，被唤醒后的操作。

2 进入低功耗前配置

根据实际应用，进入低功耗(Sleep 睡眠模式和 Stop 停止模式)前配置操作步骤如下：

1. 将未使用到的管脚通过功能模式寄存器(GPIOx_MODE)都配置成模拟模式，即设置为高阻态，否则悬空的引脚将会在输入悬空状态下引入电流。在上电的默认状态下，大部分引脚已是模拟模式。若不需要 SWD 功能，也可将该引脚配置为模拟模式。
2. 在 Sleep 睡眠模式下，可根据实际情况关闭不需要使用的 IP 时钟。需要注意的是，部分 IP 在上电后时钟是默认使能的，这些 IP 在不需要时需要手动关闭，否者影响功耗。
注意：强烈建议在调试阶段，进入低功耗模式前加几秒的延时时间，防止唤醒事件配置错误，导致程序无法通过 SWD 改写代码。但是可通过 BootTool 进行代码改写。

3 配置唤醒条件

用户可根据实际应用情况配置唤醒条件。不同模式下支持的唤醒方式如下：

- 睡眠模式（Sleep mode）：所有电源上电，内核高速时钟关闭，保留中断处理功能。其它外设等模块时钟和复位可由软件设置。任何GPIO中断唤醒，任何外设中断唤醒，复位唤醒（RESETN, IWDTR复位）。
- 停止模式（Stop mode）：所有电源上电，系统RCH/XTH/PLL停止工作，（RCL/XTL低速时钟正常工作），保留中断处理功能。其它外设等模块时钟和复位可由软件设置。任何GPIO中断唤醒，PC13(tamper)中断唤醒，RTC闹钟唤醒，IWDTR复位或者中断唤醒，外部LPUART唤醒，LPTIM0~1定时唤醒。
- 待机模式0（Standby0 mode）：CORE domain断电，BBU domain保持上电，内部低速时钟运行，RTC一直工作。BKSRAM/IWDTR/LPTIM/LPUART模块保持上电。外部管脚PA0、PA2、PC0、PC2和PC3边沿唤醒，PC13(Tamper)唤醒，RTC，LPTIM0~1定时唤醒，LPUART(只有PC2管脚唤醒)，IWDTR（复位和中断都支持）功能唤醒。
- 待机模式1（Standby1 mode）：CORE domain断电，BBU domain保持上电，内部低速时钟运行，RTC一直工作。BKSRAM/IWDTR/LPTIM/LPUART模块掉电。外部管脚 PA0、PA2、PC0、PC2、PC3和PC1边沿唤醒，PC13(Tamper)唤醒，RTC中断唤醒。
- 深度待机模式0（DeepStandby0 mode）：CORE domain断电，BBU domain保持上电，内部低速时钟停止工作，备份寄存器保持。BKSRAM/IWDTR/ LPTIM/LPUART模块保持上电。外部管脚PA0、PA2、PC0、PC2、PC3和PC13边沿唤醒。
- 深度待机模式1（DeepStandby1 mode）：CORE domain断电，BBU domain保持上电，内部低速时钟停止工作，备份寄存器保持。BKSRAM/IWDTR/ LPTIM/LPUART模块掉电。外部管脚PA0、PA2、PC0、PC2、PC3和PC13边沿唤醒。

下面将以GPIO作为唤醒源为例，讲解各个模式下的GPIO唤醒配置。

3.1 睡眠模式（Sleep mode）下唤醒配置

下面以PC13管脚输入下降沿触发中断唤醒为例，阐述软件配置流程：

1. 配置GPIOC PIN13为输入上拉引脚。
2. 配置引脚中断触发为下降沿触发。
3. 使能引脚中断。

```
GPIO_InitTypeDef GPIO_InitStructure;

/* Enable GPIOs clock */
KEY_BUTTON_GPIO_CLK_ENABLE();

GPIO_InitStructure.Mode = GPIO_MODE_IT_EDGE_FALLING;
GPIO_InitStructure.Pull = GPIO_PULL_UP;
GPIO_InitStructure.Speed = GPIO_SPEED_HIGH;
GPIO_InitStructure.Pin = KEY_BUTTON_PIN;
HAL_GPIO_Init(KEY_BUTTON_GPIO_PORT, &GPIO_InitStructure);

/* Enable and set EXTI15_10 Interrupt to the lowest priority */
HAL_NVIC_SetPriority(KEY_BUTTON_EXTI_IRQn, 0x0F, 0x00);
HAL_NVIC_EnableIRQ(KEY_BUTTON_EXTI_IRQn);
```

3.2 停止模式（Stop mode）下唤醒配置

下面以PC13管脚输入下降沿触发中断唤醒为例，阐述软件配置流程：

1. 配置GPIOC PIN13为输入上拉引脚。
2. 配置引脚中断触发为下降沿触发（这里可以不设置，设置触发类型只是为了唤醒后可以正常按照设定的触发类型触发IO中断）。
3. 使能引脚中断。
4. 配置端口中断模式为Stop模式，上电默认为Run模式。
5. 配置EXTI13的唤醒极性。
6. 使能Stop模式下的EXTI13唤醒。

```
GPIO_InitTypeDef GPIO_InitStructure;
```

```
/* Enable GPIOs clock */
KEY_BUTTON_GPIO_CLK_ENABLE();

GPIO_InitStruct.Mode = GPIO_MODE_IT_EDGE_FALLING;
GPIO_InitStruct.Pull = GPIO_PULL_UP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
GPIO_InitStruct.Pin = KEY_BUTTON_PIN;
HAL_GPIO_Init(KEY_BUTTON_GPIO_PORT, &GPIO_InitStruct);

/* Enable and set EXTI15_10 Interrupt to the lowest priority */
HAL_NVIC_SetPriority(KEY_BUTTON_EXTI_IRQn, 0x0F, 0x00);
HAL_NVIC_EnableIRQ(KEY_BUTTON_EXTI_IRQn);

/* EXTI wake up in stop mode needs to be set */
__HAL_EXIT_MODE_STOP();

/* Enable wake up EXTIx and set wake up polarity */
HAL_PMU_EXTIStopMode_EnableWakeUpEXTI(PMU_WAKEUP_EXTI13,
PMU_WAKEUP_POLARITY_DOWN);
```

3.3 待机模式 0（Standby0 mode）下唤醒配置

下面以PA0管脚输入下降沿唤醒为例，阐述软件配置流程：

在掉电唤醒控制寄存器（PMU_PDWKCR）中使能PA0管脚唤醒事件和配置PA0下降沿触发。

```
/* Disable wake up pin */
HAL_PMU_DisableWakeUpPin(PMU_WAKEUP_PA0);

/* Clear wake up flag */
__HAL_PMU_PDWK_CLEAR_FLAG();
```

```
/* Enable wake up pin and set wake up polarity */  
HAL_PMU_EnableWakeUpPin(PMU_WAKEUP_PA0,  
PMU_WAKEUP_POLARITY_DOWN);
```

3.4 待机模式 1（Standby1 mode）下唤醒配置

下面以PA2管脚输入下降沿唤醒为例，阐述软件配置流程：

在掉电唤醒控制寄存器（PMU_PDWKCR）中使能PA2管脚唤醒事件和配置PA2下降沿触发。

```
/* Disable wake up pin */  
HAL_PMU_DisableWakeUpPin(PMU_WAKEUP_PA2);  
  
/* Clear wake up flag */  
__HAL_PMU_PDWK_CLEAR_FLAG();  
  
/* Enable wake up pin and set wake up polarity */  
HAL_PMU_EnableWakeUpPin(PMU_WAKEUP_PA2,  
PMU_WAKEUP_POLARITY_DOWN);
```

3.5 深度待机模式 0（DeepStandby0 mode）下唤醒配置

下面以PC0管脚输入下降沿唤醒为例，阐述软件配置流程：

在掉电唤醒控制寄存器（PMU_PDWKCR）中使能PC0管脚唤醒事件和配置PC0下降沿触发。

```
/* Disable wake up pin */  
HAL_PMU_DisableWakeUpPin(PMU_WAKEUP_PC0);  
  
/* Clear wake up flag */
```



```
__HAL_PMU_PDWK_CLEAR_FLAG());

/* Enable wake up pin and set wake up polarity */
HAL_PMU_EnableWakeUpPin(PMU_WAKEUP_PC0,
PMU_WAKEUP_POLARITY_DOWN);
```

3.6 深度待机模式 1（DeepStandby1 mode）下唤醒配置

下面以PC2管脚输入下降沿唤醒为例，阐述软件配置流程：

在掉电唤醒控制寄存器（PMU_PDWKCR）中使能PC2管脚唤醒事件和配置PC2下降沿触发。

```
/* Disable wake up pin */
HAL_PMU_DisableWakeUpPin(PMU_WAKEUP_PC2);

/* Clear wake up flag */
__HAL_PMU_PDWK_CLEAR_FLAG());

/* Enable wake up pin and set wake up polarity */
HAL_PMU_EnableWakeUpPin(PMU_WAKEUP_PC2,
PMU_WAKEUP_POLARITY_DOWN);
```

4 进入低功耗模式

UM324xF 的 6 种休眠模式配置如下。

4.1 睡眠模式 (Sleep mode)

通过调用以下函数接口进入 Sleep 睡眠模式。

```
/* Request to enter SLEEP mode */  
HAL_PMU_EnterSLEEPMode();
```

该接口实现如下：

```
/* Clear SLEEPDEEP bit of Cortex System Control Register */  
CLEAR_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));  
  
/* Request Wait For Interrupt */  
__WFI();
```

4.2 停止模式 (Stop mode)

通过调用以下函数接口进入 Stop 停止模式。

```
/* Request to enter STOP mode */  
HAL_PMU_EnterSTOPMode();
```

该接口实现如下：

```
/* Unlock PMU protection register */  
__HAL_PMU_UNLOCK_REGISTER();  
  
/* Keep the same system clock after Stop mode wake up */  
SET_BIT(PMU->MR, PMU_MR_STOP_CLK_SEL);  
  
/* Config PMU Mode: Stop Mode*/  
MODIFY_REG(PMU->MR, PMU_MR_PMU_MODE, PMU_STOP_MODE);  
  
/* Lock PMU protection register */  
__HAL_PMU_LOCK_REGISTER();  
  
/* Set SLEEPDEEP bit of Cortex System Control Register */
```

```

SET_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));

/* Request Wait For Interrupt */

__WFI();

```

4.3 待机模式 0 (Standby0 mode)

通过调用以下函数接口进入 Standby0 待机模式 0。

```

/* Request to enter STANDBY0 mode */

HAL_PMU_EnterSTANDBY0Mode();

```

该接口实现如下：

```

/* Unlock PMU protection register */

__HAL_PMU_UNLOCK_REGISTER();

/* Config PMU mode: Standby Mode*/

MODIFY_REG(PMU->MR, PMU_MR_PMU_MODE, PMU_STANDBY_MODE);

/* BKSRAM, IWDG, LPUART, LPTimer 0~1 keep power on */

CLEAR_BIT(PMU->MR, PMU_MR_BKSRAMOFF);

/* Standby mode valid */

SET_BIT(PMU->MR, PMU_MR_STDBY_EN);

/* Flash Power Down */

WRITE_REG(EFC->LPCR, 0xa5000001);

/* Lock PMU protection register */

__HAL_PMU_LOCK_REGISTER();

/* Set SLEEPDEEP bit of Cortex System Control Register */

SET_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));

/* Request Wait For Interrupt */

__WFI();

```

4.4 待机模式 1 (Standby1 mode)

通过调用以下函数接口进入 Standby1 待机模式 1。

```
/* Request to enter STANDBY1 mode */  
HAL_PMU_EnterSTANDBY1Mode();
```

该接口实现如下：

```
/* Unlock PMU protection register */  
__HAL_PMU_UNLOCK_REGISTER();  
  
/* Config PMU mode: Standby Mode*/  
MODIFY_REG(PMU->MR, PMU_MR_PMU_MODE, PMU_STANDBY_MODE);  
  
/* BKSRAM, IWDG, LPUART, LPTimer 0~1 keep power down */  
SET_BIT(PMU->MR, PMU_MR_BKSRAMOFF);  
  
/* Standby mode valid */  
SET_BIT(PMU->MR, PMU_MR_STDBY_EN);  
  
/* Flash Power Down */  
WRITE_REG(EFC->LPCR, 0xa5000001);  
  
/* Lock PMU protection register */  
__HAL_PMU_LOCK_REGISTER();  
  
/* Set SLEEPDEEP bit of Cortex System Control Register */  
SET_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));  
  
/* Request Wait For Interrupt */  
__WFI();
```

4.5 深度待机模式 0 (DeepStandby0 mode)

通过调用以下函数接口进入 DeepStandby0 待机模式 0。

```
/* Request to enter DEEPSTANDBY0 mode */  
HAL_PMU_EnterDEEPSTANDBY0Mode();
```

该接口实现如下：

```
/* Unlock PMU protection register */
```

```

__HAL_PMU_UNLOCK_REGISTER();

/* Config PMU mode: DeepStandby Mode*/
MODIFY_REG(PMU->MR, PMU_MR_PMU_MODE,
PMU_DEEPSTANDBY_MODE);

/* BKSRAM, IWDT, LPUART, LPTimer 0~1 keep power on */
CLEAR_BIT(PMU->MR, PMU_MR_BKSRAMOFF);

/* Standby mode valid */
SET_BIT(PMU->MR, PMU_MR_STDBY_EN);

/* Flash Power Down */
WRITE_REG(EFC->LPCR, 0xa5000001);

/* Lock PMU protection register */
__HAL_PMU_LOCK_REGISTER();

/* Set SLEEPDEEP bit of Cortex System Control Register */
SET_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));

/* Request Wait For Interrupt */
__WFI();

```

4.6 深度待机模式 1（DeepStandby1 mode）

通过调用以下函数接口进入 DeepStandby1 待机模式 1。

```

/* Request to enter DEEPSTANDBY1 mode */
HAL_PMU_EnterDEEPSTANDBY1Mode();

```

该接口实现如下：

```

/* Unlock PMU protection register */
__HAL_PMU_UNLOCK_REGISTER();

/* Config PMU mode: DeepStandby Mode*/
MODIFY_REG(PMU->MR, PMU_MR_PMU_MODE,
PMU_DEEPSTANDBY_MODE);

/* BKSRAM, IWDT, LPUART, LPTimer 0~1 keep power down */
SET_BIT(PMU->MR, PMU_MR_BKSRAMOFF);

```

```
/* Standby mode valid */  
SET_BIT(PMU->MR, PMU_MR_STDBY_EN);  
  
/* Flash Power Down */  
WRITE_REG(EFC->LPCR, 0xa5000001);  
  
/* Lock PMU protection register */  
__HAL_PMU_LOCK_REGISTER();  
  
/* Set SLEEPDEEP bit of Cortex System Control Register */  
SET_BIT(SCB->SCR, ((uint32_t)SCB_SCR_SLEEPDEEP_Msk));  
  
/* Request Wait For Interrupt */  
__WFI();
```

5 被唤醒后的操作

Sleep 睡眠模式和 Stop 停止模式唤醒后系统回到 Run 模式，代码继续往下执行。

Stop 模式唤醒后，需要将端口中断模式配置回 Run 模式。配置如下所示：

```
/* After the stop mode wakes up, switching to run mode will be interrupted using  
EXTI */  
__HAL_EXIT_MODE_RUN();
```

由于 Standby0 待机模式 0、Standby1 待机模式 1、DeepStandby0 深度待机模式 0 和 DeepStandby1 深度待机模式 1 四种低功耗模式的内核掉电，因此唤醒后系统会复位重新运行。