

# AN12085

如何使用 i.MX RT 低功耗功能  
Rev. 2 — 10 January 2023

Application note

## Document information

| Information | Content   |
|-------------|---|
| Keywords    | i.MX RT 低功耗功能                                     |
| Abstract    | 本文档介绍了 i.MX RT 系列低功耗应用的设计要点。低功耗是 i.MX RT 的一个重要特性。 |



## 1 介绍

本文档介绍了 i.MX RT 系列低功耗应用的设计要点。低功耗是 i.MXRT 的一个重要特性。本应用笔记将使用 RT1050 为例，但是其基本的思路方法可以适用于 RT1010, RT1020, RT1024, RT1040, RT1050, RT1060, RT1064, RT1060X。

## 2 芯片概述

本节介绍 i.MXRT 芯片的主要特性。

### 2.1 i.MX RT 芯片概述

i.MXRT 是基于 Cortex-M7 的芯片，运行速度高达 600 MHz，可提供高 CPU 性能和最佳实时响应。

- 基于 Cortex-M7 的处理器运行速度高达 600 MHz。
- 具有内部 DCDC 和 LDO 的高级电源管理模块，可降低外部电源的复杂性并简化电源排序。
- 各种存储器接口，包括 SDRAM、RawNAND FLASH、NOR 闪存、SD/eMMC。
- 用于连接外围设备的各种其他接口，例如 WLAN、Bluetooth、GPS、显示器和相机传感器。
- 丰富的音视频功能，包括 LCD 显示、基本 2D 图形、摄像头接口、S/PDIF 和 I2S 音频接口。
- 提供丰富的外设模块，如 SPI、I2C、CAN、以太网、Flex-Timers、ADC 等。
- 针对工业 HMI、电机控制和家用电器领域。

## 3 低功耗概述

以下部分介绍电源、运行模式和低功耗模式。

### 3.1 电源

Table 1 列出了 i.MXRT 的外部电源轨。

Table 1. 外部电源轨

| 电源轨                         | 描述                               |
|-----------------------------|----------------------------------|
| DCDC_IN                     | DCDC 电源                          |
| VDD_HIGH_IN                 | 模拟电源                             |
| VDD_SNVS_IN                 | SNVS 和 RTC 电源                    |
| USB_OTG1_VBUS USB_OTG2_VBUS | USB VBUS 电源                      |
| VDDA_ADC                    | 12 位 ADC 电源                      |
| VDDA_IN                     | LDO 2P5 和 LDO 1P1 电源             |
| NVCC_SD0                    | SDIO1 bank 中 GPIO 的电源 (3.3 V 模式) |
|                             | SDIO1 bank 中 GPIO 的电源 (1.8 V 模式) |
| NVCC_SD1                    | SDIO2 bank 中 GPIO 的电源 (3.3 V 模式) |
|                             | SDIO2 bank 中 GPIO 的电源 (1.8 V 模式) |

**Table 1.** 外部电源轨...continued

| 电源轨       | 描述                       |
|-----------|--------------------------|
| NVCC_GPIO | GPIO bank 里 GPIO 的 IO 电源 |
| NVCC EMC  | EMC bank 里 GPIO 的 IO电源   |

### 3.2 运行模式定义

[Table 2](#)列出运行模式定义。

**Table 2.** 运行模式定义

| 运行模式              | 定义   |
|-------------------|--|
| Overdrive run 模式  | <ul style="list-style-type: none"> <li>• CPU 以 600 MHz 运行，过载电压为 1.275 V</li> <li>• 总线频率为 150 MHz</li> <li>• 所有外设均已启用并以目标频率运行</li> </ul>                                      |
| Full speed run 模式 | <ul style="list-style-type: none"> <li>• CPU 以 528 MHz 运行，满载，电压降至 1.15 V</li> <li>• 总线频率为 132 MHz</li> <li>• 所有外设均已启用并以目标频率运行</li> </ul>                                     |
| Low speed run 模式  | <ul style="list-style-type: none"> <li>• CPU 以 132 MHz 运行，电压降至 1.15 V</li> <li>• 总线频率为 66 MHz</li> <li>• 某些 PLL 掉电</li> <li>• 20% 的外设处于活动状态，其他处于低功耗模式</li> </ul>             |
| Low power run 模式  | <ul style="list-style-type: none"> <li>• CPU 以 24 MHz 运行，电压降至 0.95 V</li> <li>• 总线频率为 12 MHz</li> <li>• 所有 PLL 断电，XTAL 24 M 掉电，OSC RC 24 M 启用</li> <li>• 高速外围设备掉电</li> </ul> |

### 3.3 运行模式配置

[Table 3](#)描述运行模式配置。

**Table 3.** 运行模式配置

|                | 超速运行    | 全速运行    | 低速运行    | 低功耗运行        |
|----------------|---------|---------|---------|--------------|
| CCM LPM Mode   | RUN     | RUN     | RUN     | RUN          |
| CPU Core       | 600 MHz | 528 MHz | 132 MHz | 24 MHz       |
| L1 Cache       | ON      | ON      | ON      | ON           |
| FlexRAM        | ON      | ON      | ON      | ON           |
| SOC Voltage    | 1.275 V | 1.15 V  | 1.15 V  | 0.95 V       |
| Analog LDO     | ON      | ON      | ON      | In Weak Mode |
| 24MHz XTAL OSC | ON      | ON      | ON      | OFF          |
| 24MHz RC OSC   | OFF     | OFF     | OFF     | ON           |
| System PLL     | ON      | ON      | ON      | OFF          |
| Other PLLs     | ON      | ON      | 根据需要开启  | 根据需要开启       |
| Module Clock   | ON      | ON      | 根据需要开启  | 外围时钟关闭       |
| RTC32K         | ON      | ON      | ON      | ON           |

### 3.4 低功耗模式定义

[Table 4](#) 列出了低功耗模式定义。

**Table 4.** 低功耗模式定义

| 低功耗模式             | 定义  |
|-------------------|---|
| System Idle 模式    | <ul style="list-style-type: none"> <li>• CPU 可以在没有线程运行时自动进入该模式</li> <li>• 所有外围设备都可以保持活动状态</li> <li>• CPU 仅进入WFI模式，它会保留其状态，因此中断响应可以非常短</li> </ul>        |
| Low Power Idle 模式 | <ul style="list-style-type: none"> <li>• 比系统空闲模式功耗低得多，退出所需的时间更长</li> <li>• 所有 PLL 关闭，模拟模块在低功耗模式下运行</li> <li>• 所有高速外围设备均采用电源门控，低速外围设备可保持低频率运行</li> </ul> |
| Suspend 模式        | <ul style="list-style-type: none"> <li>• 最省电、退出时间最长的模式</li> <li>• 关闭所有 PLL，关闭 XTAL，关闭除 32 K 时钟以外的所有时钟</li> <li>• 所有高速外设都是电源门控的，低速外设是时钟门控的</li> </ul>    |
| SNVS模式            | <ul style="list-style-type: none"> <li>• 所有 SOC 数字逻辑、模拟模块都关断，SNVS 域除外</li> <li>• 32 KHz RTC有效</li> </ul>  |

注意：在进入低功耗模式之前，每个模块都被启用或禁用，如配置表。但是，对于唤醒，请确保恢复设置。

### 3.5 低功耗模式配置

[Table 5](#) 描述低功耗模式配置。

**Table 5.** 低功耗模式配置

|                    | 系统空闲   | 低功耗空闲  | 暂停      | SNVS |
|--------------------|--------|--------|---------|------|
| CCM LPM 模式         | WAIT   | WAIT   | STOP    | -    |
| Arm Core (PDM7)    | WFI    | WFI    | 掉电      | OFF  |
| L1 Cache           | ON     | ON     | 掉电      | OFF  |
| FlexRAM (PDRET)    | ON     | ON     | ON      | OFF  |
| FlexRAM (PDRAM0)   | ON     | ON     | 掉电      | OFF  |
| FlexRAM (PDRAM1)   | ON/OFF | ON/OFF | 掉电      | OFF  |
| VDD_SOC_IN Voltage | 1.15 V | 0.95 V | 0.925 V | OFF  |
| SYS PLL            | ON     | 掉电     | 掉电      | OFF  |
| 其它 PLL             | 掉电     | 掉电     | 掉电      | OFF  |
| 24 MHz XTAL OSC    | ON     | OFF    | OFF     | OFF  |
| 24 MHz RC OSC      | OFF    | ON     | OFF     | OFF  |
| LDO 2P5            | ON     | OFF    | OFF     | OFF  |
| LDO 1P1            | ON     | OFF    | OFF     | OFF  |
| WEAK 2P5           | OFF    | ON     | OFF     | OFF  |
| WEAK 1P1           | OFF    | ON     | OFF     | OFF  |
| Band gap           | ON     | OFF    | OFF     | OFF  |
| Low-Power Band gap | ON     | ON     | ON      | OFF  |

**Table 5.** 低功耗模式配置...continued

|               | 系统空闲   | 低功耗空闲  | 暂停  | SNVS |
|---------------|--------|--------|-----|------|
| AHB Clock     | 33 MHz | 12 MHz | OFF | OFF  |
| IPG Clock     | 33 MHz | 12 MHz | OFF | OFF  |
| PER Clock     | 33 MHz | 12 MHz | OFF | OFF  |
| Module Clocks | 根据需要开启 | 根据需要开启 | OFF | OFF  |
| RTC32K        | ON     | ON     | ON  | ON   |

注意：在进入低功耗模式之前，每个模块都已按照配置表中的描述启用或禁用。但是，对于唤醒，请确保恢复成原始设置

### 3.5.1 唤醒源

[Table 6](#)描述唤醒源。

**Table 6.** 唤醒源

|          | 系统空闲 | 低功耗空闲 | 暂停                | SNVS           |
|----------|------|-------|-------------------|----------------|
| GPIO 唤醒  | YES  | YES   | YES               | - YES (仅 1 引脚) |
| RTC 唤醒   | YES  | YES   | YES               | YES            |
| USB 远程唤醒 | YES  | YES   | YES               | NO             |
| 其它唤醒源    | YES  | YES   | ON <sup>[1]</sup> | NO             |

[1] 使用RTC作为时钟源的外设可以作为系统的唤醒源，例如GPT等。

注意：无论模式是系统空闲、低功耗空闲还是暂停，用户都必须先在 GPC 模块中使能唤醒中断，否则唤醒将会失败。SNVS 模式下只能通过 CPIO5\_IO00 引脚来将系统唤醒。本文档不包括关于如何使用 USB 远程唤醒的部分。

## 4 电源模式切换示例

本节提供应用说明，电源模式切换菜单。

设计功耗模式切换示例目的在于模拟客户低功耗应用案例。

特征：

- FreeRTOS 和 tickless 功能支持。
- 支持所有运行和低功耗模式切换。
- 支持 WFI 指令后运行模式和暂停模式复位（由芯片控制）。
- 模拟两个正在运行的任务并在电源模式切换中显示任务状态。

## 4.1 电源模式切换菜单

```
COM58 - PuTTY

CPU wakeup source 0x1...

*****
Power Mode Switch Demo for iMXRT1050
*****

*****
CPU:          600000000 Hz
AHB:          600000000 Hz
SEMC:         100000000 Hz
IPG:          150000000 Hz
OSC:          24000000 Hz
RTC:          32768 Hz
ARMPPLL:      1200000000 Hz
*****

Task 2 is working now
Task 1 is working now

##### Power Mode Switch Demo (build Oct 17 2017) #####

Core Clock = 600000000Hz
Power mode: Over RUN

*****
CPU:          600000000 Hz
AHB:          600000000 Hz
SEMC:         100000000 Hz
IPG:          150000000 Hz
OSC:          24000000 Hz
RTC:          32768 Hz
ARMPPLL:      1200000000 Hz
*****

Select the desired operation

Press A for enter: Over RUN      - System Over Run mode (600MHz)
Press B for enter: Full RUN      - System Full Run mode (528MHz)
Press C for enter: Low Speed RUN - System Low Speed Run mode (132MHz)
Press D for enter: Low Power RUN - System Low Power Run mode (24MHz)
Press E for enter: System Idle   - System Wait mode
Press F for enter: Low Power Idle - Low Power Idle mode
Press G for enter: Suspend       - Suspend mode
Press H for enter: SNVS          - Shutdown the system

Waiting for power mode select..
```

在此菜单中，用户可以切换电源模式以及从空闲、暂停模式中唤醒。

## 4.2 任务状态显示

当用户选择转换到新的电源模式时，任务状态将会被显示出来。

### 1. 开始工作

```
Task 2 is working now
Task 1 is working now
```

### 2. 电源模式转换

例如，  
Overrun 模式 => System Idle 模式

```
Waiting for power mode select..
WorkingTask 2: Transfer from Over RUN to System Idle
WorkingTask 1: Transfer from Over RUN to System Idle
```

### 3. 唤醒后转回 Overrun 模式

```
Waiting for key press..
Switch SW8 from off to on to wake up.
WorkingTask 2: Transfer from System Idle to Over RUN
WorkingTask 1: Transfer from System Idle to Over RUN
Next loop
```

## 4.3 唤醒源

有两个唤醒源可供系统空闲、低功耗空闲和暂停模式选择使用，分别是SW8键（在 EVK板上）和 GPT 定时器。

```
Waiting for power mode select..
WorkingTask 2: Transfer from Over RUN to System Idle
WorkingTask 1: Transfer from Over RUN to System Idle
Select the wake up source:
Press T for GPT - GPT Timer
Press S for switch/button SW8.
```

- SW8 用户键唤醒。
- GPT 定时器唤醒。

当选择 GPT 定时器作为唤醒源时，另一个菜单会要求用户输入 1 - 9 秒作为唤醒前的等待时间。

```
Select the wake up timeout in seconds.
The allowed range is 1s ~ 9s.
Eg. enter 5 to wake up in 5 seconds.
Waiting for input timeout value...
3
Will wakeup in 3 seconds.
WorkingTask 2: Transfer from Low Power Idle to Over RUN
WorkingTask 1: Transfer from Low Power Idle to Over RUN
Next loop
```

## 5 低功耗应用设计

本章将提供一些设计低功耗应用的指南。此外，我们将给出一些关于低功耗模块启用和禁用的代码示例。

### 5.1 总体描述

进入低功耗模式是一组启用/禁用模块的操作。应注意低功耗模式配置。

Table 7. 低功耗配置

|                    | 系统空闲   | 低功耗空闲  | 暂停      | SNVS |
|--------------------|--------|--------|---------|------|
| CCM LPM 模式         | WAIT   | WAIT   | STOP    | -    |
| Arm Core (PDM7)    | WFI    | WFI    | 掉电      | OFF  |
| L1 Cache           | ON     | ON     | 掉电      | OFF  |
| FlexRAM (PDRET)    | ON     | ON     | ON      | OFF  |
| FlexRAM (PDRAM0)   | ON     | ON     | 掉电      | OFF  |
| FlexRAM (PDRAM1)   | ON/OFF | ON/OFF | 掉电      | OFF  |
| VDD_SOC_IN Voltage | 1.15 V | 0.95 V | 0.925 V | OFF  |
| SYS PLL            | ON     | 掉电     | 掉电      | OFF  |
| Other PLL          | 掉电     | 掉电     | 掉电      | OFF  |
| 24 MHz XTAL OSC    | ON     | OFF    | OFF     | OFF  |
| 24 MHz RC OSC      | OFF    | ON     | OFF     | OFF  |
| LDO 2P5            | ON     | OFF    | OFF     | OFF  |
| LDO 1P1            | ON     | OFF    | OFF     | OFF  |
| WEAK 2P5           | OFF    | ON     | OFF     | OFF  |
| WEAK 1P1           | OFF    | ON     | OFF     | OFF  |
| Band gap           | ON     | OFF    | OFF     | OFF  |
| Low Power Band gap | ON     | ON     | ON      | OFF  |
| AHB Clock          | 33 MHz | 12 MHz | OFF     | OFF  |
| IPG Clock          | 33 MHz | 12 MHz | OFF     | OFF  |
| PER Clock          | 33 MHz | 12 MHz | OFF     | OFF  |
| Module Clocks      | 根据需要开启 | 根据需要开启 | OFF     | OFF  |
| RTC32K             | ON     | ON     | ON      | ON   |

## 5.2 低功耗模式进入和退出顺序

在 i.MX RT 上，芯片可以进入每个低功耗模式并退出回到运行模式。



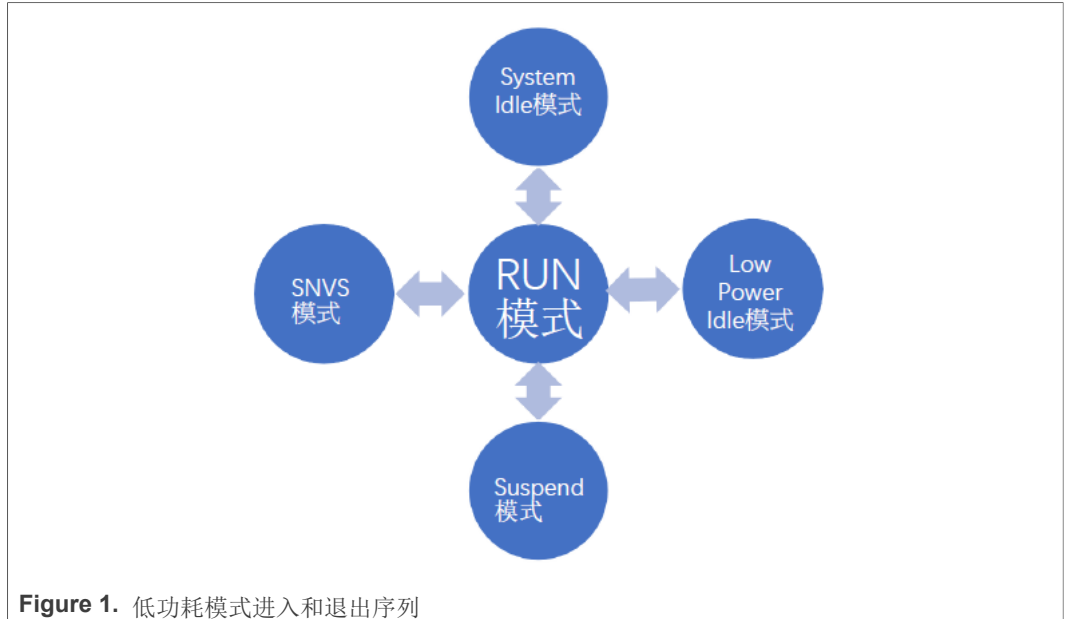


Figure 1. 低功耗模式进入和退出序列

### 5.2.1 进入 system idle 模式

System Idle 模式是一种简单的模式，大多数模块的默认设置不需要更改。

#### 5.2.1.1 如何进入

要进入 system idle 模式，应首先更改核频率，然后关闭未使用的 PLL。最后，核电压也需要做相应的改变。在执行 WFI 指令之前，需要设置所使用模块的 DOZE 位。



假设核心频率降低到 132 MHz。如果核心频率从 24 MHz 切换到 132 MHz，应该先提高核电压。

#### 5.2.1.2 如何退出

退出 system idle 模式也很简单。首先调整核电压，然后启用 PLLs。



## 5.2.2 进入 low power idle 模式

### 5.2.2.1 如何进入

要进入 Low Power Idle 模式，请参阅下图。



### 5.2.2.2 如何退出

要退出 Low Power Idle 模式，请参考下图。



### 5.2.3 进入 Suspend 模式

在 Suspend 模式下，当将 CLPCR[LPM] 设置为 STOP 时，系统将关闭 PLL，无需调整频率。

#### 5.2.3.1 如何进入

要进入 Suspend 模式，请参阅下图。



5.2.3.2 如何退出

要退出 Suspend 模式，请参阅下图。



## 5.2.4 SNVS 模式

### 5.2.4.1 如何进入

可以选择通过硬件或者软件这两种方式来进入 SNVS 模式。若要从软件方式进入 SNVS 模式，将 SNVS\_LPCR[TOP] 设置为 1。

示例代码：

```
SNVS->LPCR |= SNVS_LPCR_TOP_MASK;
while (1) /* Shutdown */
{
}
```

若要从硬件方式进入 SNVS 模式，长按开/关按钮（EVK 板上的 SW2）。

### 5.2.4.2 如何退出

长按开/关按钮（EVK 板上的 SW2）或使用唤醒引脚。

该芯片支持使用 WAKEUP 引脚（GPIO5\_IO00ALT5）来使主 SoC 上电从而退出 SNVS 模式。要使用它，请按照以下提示操作：

- SNVS 必须处于 Dumb PMIC 模式（默认，必须是对于芯片上的 DCDC）
- 配置 IOMUXC\_SNVS，选择 WAKEUP 引脚 ALT5 将其复用到 GPIO5\_IO00
- 配置 GPIO5 ICR，让它能被低电平或高电平触发
- 设置 GPIO5IMR bit0，使能 GPIO5\_IO00 中断
- 通过 SW 或 ON/OFF 按钮进入 SNVS 模式，然后芯片上的 DCDC 将会被关闭，PMIC\_ON\_REQ 引脚将变成低电平状态以此作为信号通知外部

- 将 WAKEUP 引脚置位至少两个 32kHz 周期（高电平或低电平取决于 GPIO5ICR 配置），以便 SNVS 模块对 GPIO5 中断进行采样
- SNVS 将 PMIC\_ON\_REQ 引脚置为高电平，片上 DCDC 开始攀升
- Soc 的上电流程（SNVS 模式除外）将开始执行（ROM boot, ...）

## 5.3 低功耗设计要点

### 5.3.1 如何改变核频率

当核的频率需要改变时，核的电压也需要改变。改变核的率和电压是有规律的。

- 提高核的率，首先是电压，然后是频率。
- 降低核的频率，先降低频率，再降低电压。

### 5.3.2 如何改变 PLL 频率

改变 PLL 频率顺序如下：

- 设置 BYPASS 和 ENABLE 位以先将时钟信号暂设为旁路。
- 设置 PLL 时钟分频器。
- 清除 PLL BYPASS 位。
- 通过测试检查 CCM\_CDHIPR 寄存器来确认 CCM 操作是否完成。

示例代码：

```
/* Wait CCM operation finishes */
while (CCM->CDHIPR != 0)
{
}
```

注意：当需要改变 PLL 的频率时，先改变 BYPASS 时钟，然后改变 PLL 频率，最后切换回 PLL。

### 5.3.3 如何更改 SPIFlash 和 SDRAM 的时钟源

通常在任何 RUN 模式下，如果使用 SPIFlash 或 SDRAM 来存储代码或数据，则不建议更改根时钟的频率。更改时钟源和频率对 R/W SPI Flash 或 SDRAM 来说是很危险的。如果必须执行这些操作，代码应该在 On-Chip RAM 中运行。

### 5.3.4 如何在 FreeRTOS 中启用 tickless 低功耗功能

FreeRTOS tickless 空闲模式在空闲时段（没有执行的应用程序任务的时段）将会停止周期性 tick 中断，然后在 tick 中断重新启动时对 RTOS 时钟内核计数值进行更正调整。

停止 tick 中断可以使得微控制器保持深度省电状态，直到发生其他中断将其唤醒。

注意：MCU-SDK 已经提供了 fsl\_tickless\_systick.c 和 fsl\_tickless\_gpt.c 来实现 Tickless 低功耗功能。用户需要在 FreeRTOSConfig.h 中定义 configUSE\_TICKLESS\_IDLE。

#### 5.3.4.1 选择一个 FreeRTOS 滴答计时器

通常，系统 tick 模块用作 FreeRTOS 中的滴答计时器。在 i.MXRT 系列上，Cortex-M7 核中的系统 tick 模块可以支持两个定时器源，核时钟和来自 24M 振荡器并分频为 100KHz 的时钟。

在 FreeRTOS 中，在 tickless 停止时所消耗的时间将得到补偿。tickless 停止消耗的时间不应超过最大持续时间（否则 FreeRTOS 将无法知道要补偿多少时间）。计算最大持续时间计数的代码在 tickless 代码中的函数 vPortSetupTimerInterrupt() 中。

```
ulTimerCountsForOneTick = ( configSYSTICK_CLOCK_HZ / configTICK_RATE_HZ );
xMaximumPossibleSuppressedTicks = portMAX_32_BIT_NUMBER /
ulTimerCountsForOneTick;
```

对于系统 tick 模块，portMAX\_32\_BIT\_NUMBER 是 0x00FFFFFF 和 configTICK\_RATE\_HZ 正常设为 1000，意味着单个 tick 持续时间为 1ms。

因此，如果选择核时钟 600MHz 作为系统 tick 模块时钟源，ulTimerCountsForOneTick 将会是 600000，xMaximumPossibleSuppressedTicks 将会是 27.962025，也就是 27.962025ms。

27.962025ms 对于 tickless 持续时间来说太短了。意思是在 tickless 停止时，系统需要每 27.962025ms 唤醒一次来补偿时间。

因此，当考虑在 FreeRTOS 应用程序中使用低功耗 tickless 时，我们最好不要使用核心时钟作为系统时钟源，而是使用 100KHz 时钟源。在这种情况下，ulTimerCountsForOneTick 将会是 100

xMaximumPossibleSuppressedTicks 将会是 167772.15 ms，差不多 168 s。这是一个可以接受的持续时间。要使用 100KHz 时钟，只需在 FreeRTOSConfig.h 中定义 configSYSTICK\_CLOCK\_HZ

```
#define configSYSTICK_CLOCK_HZ (10000U)
```

#### 5.3.4.2 选择一个无滴答计时器

在 tickless 模式中

FreeRTOS 也需要补偿消耗的时间，但我们需要选择一个不能超过 FreeRTOS 可以支持的最大时间限度的计时器。

Systick 是此计时器的首选。但是在 i.MXRT 中，GPC 模块中没有包含 systick 中断。所以 systick 中断不能唤醒系统，只能唤醒 Arm 核。我们需要选择另一个计时器作为无滴答计时器。

无滴答计时器的要求：

- 计时器。
- 被包含在 GPC 模块中，可唤醒系统。
- 在系统空闲、低功耗空闲和暂停模式下工作，即使在核和所有 PLL 掉电时也可运行。

GPT 模块可以使用 32KRTC（通过设置 GPT\_CR[CLKSRC] = 4）作为时钟源，可以工作在系统空闲、低功耗空闲和暂停模式。

#### 5.3.4.3 WFI 指令前后的 hook 函数

FreeRTOS tickless 空闲模式提供了两种方法。

- vPortSuppressTicksAndSleep 功能。

在 tickless 模式，系统将在空闲任务中调用 vPortSuppressTicksAndSleep 函数，并将预期的睡眠时间（以微秒为单位）作为参数传递。

用户可以实现自定义的 vPortSuppressTicksAndSleep 并且空闲任务将定期调用此函数。

- 优点：可以根据用户需要处理 WFI 指令。



例如，在暂停模式下，内核是处于掉电状态，因此唤醒后会将内核复位。但是用户可以在复位后从 **WFI** 处继续运行后续代码的。为了达到此目的，用户需要设置一个还原点，以便能在复位后进行还原。

在这种情况下，用户更容易处理客户函数中的 **WFI** 指令。

- 缺点：用户函数需要计算 **tickles** 周期并调用 **vTaskStepTick** 来补偿消耗的时间。  
注意：用户可以参考 **fsl\_tickless\_systick.c** 和 **fsl\_tickless\_gpt.c** 中的 **vPortSuppressTicksAndSleep** 函数作为例子来学习如何使用时间补偿。
- 在 **WFI** 指令前后调用的两个 **hook** 函数， **configPRE\_SLEEP\_PROCESSING** 和 **configPOST\_SLEEP\_PROCESSING**。

如果用户想在 **fsl\_tickless\_systick.c** 和 **fsl\_tickless\_gpt.c** 中重复使用 **vPortSuppressTicksAndSleep**，可以用 **configPRE\_SLEEP\_PROCESSING** 和 **configPOST\_SLEEP\_PROCESSING** 来更改和恢复模块以降低功耗。

- 优点：不需要在意时间补偿。 **fsl\_tickless\_systick.c** 和 **fsl\_tickless\_gpt.c** 中的 **vPortSuppressTicksAndSleep** 函数将为您完成。
- 缺点：不是很灵活。用户不能在自己的函数中执行 **WFI** 指令。

在 **Power Mode Switch** 示例中，我们使用第二种方法来重复使用 **vPortSuppressTicksAndSleep**。

### 5.3.5 Doze 和 IPGSTOP 信号

为了降低功耗，应该将 **Doze** 和 **Stop** 位置位，并设置将各个模块在进入低功耗模式之前进入 **Doze of Stop** 状态。在示例代码中，所有支持的外设的 **Doze** 的位都被使能，但在应用程序中，开发人员应该知道需要哪些模块并使能相应模块的 **Doze** 的位。

#### 5.3.5.1 Doze 模式

对于处于休眠模式的各个模块，AHB时钟和串行时钟域将在内部被关闭，但 **IPS** 总线时钟不会被关闭。通过设置 **IOMUXC\_GPR\_GPR8** 和 **IOMUXC\_GPR\_GPR12** 中相应的 **DOZE** 位进入该模式，并通过清除 **IOMUXC\_GPR\_GPR8** 和 **IOMUXC\_GPR\_GPR12** 中的相应位退出该模式。

注意：**Doze** 模式适用于所有低功耗模式。因此，应在进入任何低功耗模式之前设置 **Doze** 位。

示例代码：

在进入低功耗模式前：

```
IOMUXC_GPR->GPR8 = 0xaaaaaaaa; IOMUXC_GPR->GPR12 = 0x0000000a;
```

在唤醒后：

```
IOMUXC_GPR->GPR8 = 0x00000000; IOMUXC_GPR->GPR12 = 0x00000000;
```

#### 5.3.5.2 Stop 模式

当系统要求外围设备进入 **Stop** 模式时，它将等待所有事务完成并向系统返回 **ACK** 握手信号。ACK 握手信号返回后，系统可以在系统级别对 **AHB** 总线时钟、**IPG** 总线时钟和串行时钟进行门控操作。内部没有时钟门控操作。通过设置 **IOMUXC\_GPR\_GPR8** 和 **IOMUXC\_GPR\_GPR12** 中相应的 **DOZE** 和 **STOP** 位进入该模式，系统应等待 **IOMUXC\_GPR\_GPR4** 和 **IOMUXC\_GPR\_GPR7** 中相应的位被置位。暂停模式从属于 **Stop** 模式。因此在进入暂停模式之前应该设置 **Stop** 和 **Doze** 位。

在示例代码中，所有支持的外设的 STOP 和 DOZE 位都已使能，但在应用程序中，开发人员应了解需要哪些模块并使能相应模块的 STOP 和 DOZE 位。

示例代码：

在进入暂停模式之前：

```
IOMUXC_GPR->GPR4 = 0x00000011;
while ((IOMUXC_GPR->GPR4 & 0x00110000) != 0x00110000) {};
IOMUXC_GPR->GPR4 = 0x000036ff;
IOMUXC_GPR->GPR7 = 0x0000ffff;
IOMUXC_GPR->GPR8 = 0xffffcffff;
IOMUXC_GPR->GPR12 = 0x0000000a;
while ((IOMUXC_GPR->GPR4 & 0x36f90000) != 0x36f90000) {};
while ((IOMUXC_GPR->GPR7 & 0xffff0000) != 0xffff0000) {};
```

在唤醒后：

```
IOMUXC_GPR->GPR4 = 0x00000000;
IOMUXC_GPR->GPR7 = 0x00000000;
IOMUXC_GPR->GPR8 = 0x00000000;
IOMUXC_GPR->GPR12 = 0x00000000;
```

查看上面的代码，STOP 请求首先发送到 EDMA 和 ENET。

```
IOMUXC_GPR->GPR4 = 0x00000011;
while ((IOMUXC_GPR->GPR4 & 0x00110000) !=
0x00110000) {};
```

原因在于EDNA和NET是两个模块，可以作为 BUS 上的master，发起传输。如果代码在 SPI Flash 上运行，则stop请求不应发送到 SPI Flash。

注意：确保在进入暂停模式之前，如果相应模块的 STOP 请求被确认，系统应等待确认位被置位。否则系统可能无法进入暂停模式。

如果代码在 SDRAM 中运行并且用户想要进入暂停模式。SDRAM 需要等待 STOP 信号才能进入自刷新模式。在这里，如果 STOP 请求被发送到 SDRAM 并使用轮询的方法来确认是否被置位，有可能永远轮询不到。因为代码在 SDRAM 上运行，SDRAM 上的事务不能停止。在这种情况下，代码或数据应在 SPI Flash 或芯片上 RAM 中运行。

注意：根据时钟门控和 IP 模块的使能位是否设置，每个 IP 模块在发送停止信号时可能有不同的反应。

- 一些模块即使时钟门关闭也可以确认stop信号，例如 PIT。
- 当时钟门关闭时，某些模块无法确认stop信号。但是当时钟门打开时，它可以确认stop信号，例如 EDMA。
- 当时钟门控打开且模块的使能位为ON时，某些模块可以确认stop信号。目前只有CAN属于这种类型。

### 5.3.6 如何启用弱模拟 LDO

i.MX RT 上有两种类型的模拟 LDO，常规 LDO 和低功耗弱 LDO。1.1 V 和 2.5 V LDO 包括一个备用设备、自偏置、低精度的弱稳压器，可用于需要在低功耗模式下保持 1.1 V 和 2.5 V 输出电压的应用，其中主稳压器及其相关的全局带隙参考模块被禁用。使用弱 LDO 可以降低低功耗空闲模式下的功耗。

- 启用弱模拟 LDO

```
/* Enable regular 2P5 and wait it stable */
```

```

PMU->REG_2P5_SET = PMU_REG_2P5_ENABLE_LINREG_MASK;
while ((PMU->REG_2P5 & PMU_REG_2P5_OK_VDD2P5_MASK) == 0)
{
}
/* Turn off weak 2P5 */
PMU->REG_2P5_CLR = PMU_REG_2P5_ENABLE_WEAK_LINREG_MASK;

/* Enable regular 1P1 and wait for stable */
PMU->REG_1P1_SET = PMU_REG_1P1_ENABLE_LINREG_MASK;
while ((PMU->REG_1P1 & PMU_REG_1P1_OK_VDD1P1_MASK) == 0)
{
}
/* Turn off weak 1P1 */
PMU->REG_1P1_CLR = PMU_REG_1P1_ENABLE_WEAK_LINREG_MASK;
    
```

- 禁止使用弱模拟 LDO

```

/* Enable weak 2P5 and turn off regular 2P5 */
PMU->REG_2P5 |=
PMU_REG_2P5_ENABLE_WEAK_LINREG_MASK; PMU->REG_2P5
&= ~PMU_REG_2P5_ENABLE_LINREG_MASK;
/* Enable weak 1P1 and turn off regular 1P1 */
PMU->REG_1P1 |=
PMU_REG_1P1_ENABLE_WEAK_LINREG_MASK; PMU->REG_1P1
&= ~PMU_REG_1P1_ENABLE_LINREG_MASK;
    
```

5.3.7 如何禁用 OGRAM 电源

i.MX RT1050 芯片有芯片上的 FlexRAM，这是由 I-TCM、D-TCM 和通用芯片上 RAM (OCRAM) 划分出来的。FlexRAM 管理一个大的芯片上 RAM 阵列。FlexRAM 有 16 个 RAM bank，分为 Bank0、FlexRAM0 (Bank 1 - 7) 和 FlexRAM1 (Bank 8 -15)。下面列出了对 OGRAM 组的描述。

Table 8. 对 OGRAM bank 的描述

|             | Bank 0                | Bank 1-7   | Bank 8-15   |
|-------------|-----------------------|--|---|
| 别名          | Bank0                 | FlexRAM0   | FlexRAM1  |
| 电源域         | SNVS                  | PDRAM0   | PDRAM1  |
| 电源门位        | N/A                   | GPC_CNTR[PDRAM0_PGE]                                   | PGC_MEGA_CTRL[PCR]  |
| 电源门默认值      | N/A                   | 1 (将在低功耗模式下掉电)   | 0 (不会掉电)  |
| LPM 模式开/关状态 | ON                    | 可控   | 可控  |
| 掉电时刻        | 在 SNVS 模式, bank0 会掉电. | GPC_CNTR[PDRAM0_PGE] 被置位, 并且当核执行 WFI 指令时, Bank 1-7 将关闭 | PGC_MEGA_CTRL[PCR] is set and Bank 8-15 will be off immediately |

注意：当电源门位被置位时，FlexRAM0 不会立即关闭，除非核执行 WFI 指令。但 FlexRAM1 将立即关闭。

代码示例：

- 关闭 FlexRAM0 和 FlexRAM1 电源

```

/* Turn off FlexRAM1 */
PGC->MEGA_CTRL |= PGC_MEGA_CTRL_PCR_MASK;
    
```

```
/* Turn off FlexRAM0 */
GPC->CNTR |= GPC_CNTR_PDRAM0_PGE_MASK;
```

- 打开 FlexRAM0 和 FlexRAM1 电源

```
/* Turn on FlexRAM1 */
PGC->MEGA_CTRL &= ~PGC_MEGA_CTRL_PCR_MASK;
/* Turn on FlexRAM0 */
GPC->CNTR &= ~GPC_CNTR_PDRAM0_PGE_MASK;
```

### 5.3.8 带隙切换

在 i.MX RT 中，有两个带隙模块，常规带隙和低功耗带隙。在低功耗应用中，应在进入低功耗 Wait 或 Stop 状态之前使用低功率带隙。

注意：在暂停模式下，STOP 模式将在 CLPCR 中被设置，这将有助于在低功耗 Stop 状态自动将带隙切换到低功耗带隙。

代码示例：

将带隙切换到低功耗带隙：

```
/* Switch band gap */
PMU->MISC0_SET = 0x00000004;
XTALOSC24M->LOWPWR_CTRL_SET =
XTALOSC24M_LOWPWR_CTRL_LPBG_SEL_MASK;
PMU->MISC0_SET = CCM_ANALOG_MISC0_REFTOP_PWD_MASK;
Switch band gap back to regular band gap:
/* Restore band gap */
/* Turn on regular band gap and wait for stable */
CCM_ANALOG->MISC0_CLR = CCM_ANALOG_MISC0_REFTOP_PWD_MASK;
while ((CCM_ANALOG->MISC0 &
CCM_ANALOG_MISC0_REFTOP_VBGUP_MASK) == 0)
{
}
/* Low power band gap disable */
XTALOSC24M->LOWPWR_CTRL_CLR =
XTALOSC24M_LOWPWR_CTRL_LPBG_SEL_MASK;
PMU->MISC0_CLR = 0x00000004;
```

### 5.3.9 启用 DCDC 的 DCM 模式

DCDC 模块提供断续导通模式 (DCM) 和连续导通模式 (CCM)。它可以在运行模式下使用 CCM 模式或 DCM 模式下运行。使用哪种模式取决于负载电流的水平。通常芯片处于运行模式时采用 CCM 模式，芯片处于低功耗模式时采用 DCM 模式。如果在运行模式下功耗较低，也可以使用 DCM 模式。DCM 模式可以减少 DCDC\_IN 的电流，但 DCDC\_OUT 电流不会改变。

注意：由于不连续导通模式 (DCM) 可以在低电流负载的情况下提高 DCDC 的效率，所以推荐优先选择该模式。

应在进入低功耗模式之前设置 DCM 模式。详细步骤如下：

```
pwd_zcd=0x0
pwd_cmp_offset=0x0
loopctrl_en_rcscale=0x4
DCM_SET_CTRL=1'b1
```

唤醒时不要忘记恢复到 CCM 模式。并启用 CCM 模式：

```
pwd_zcd=0x1
pwd_cmp_offset=0x0
loopctrl_en_rcscale=0x3
```

### 5.3.10 启用和禁用 RBC 位

在暂停模式下，RBC 位需要在执行 WFI 指令前置位，唤醒后清零。RBC 位在 CCM\_CCR 寄存器中。CCM\_CLPCR 中有一个 VSTBY 位，它将电压改变为待机电压（DCDC 将进入省电模式）。在低功耗空闲模式和暂停模式下需要对该位置位。

启用 RBC\_EN 和 VSTBY 将会使模拟部分在 STOP 模式中进入低功耗。

启用和禁用 RBC\_EN 位需要注意两点：

- 需要在设置位之前屏蔽 GPC 中断
- 设置位后需要延迟（3 us）。

示例

- 启用 RBC\_EN:

```
/* Mask all GPC interrupts before enabling the RBC counters to
   avoid the counter starting too early if an interrupt is
   already pending.
   */
for (i = 0; i < LPM_GPC_IMR_NUM; i++)
{
    gpcIMR[i] = GPC->IMR[i];
    GPC->IMR[i] = 0xFFFFFFFFFU;
}

/* Enable the RBC bypass counter here to hold off the
interrupts. RBC counter
* = 32 (lms). Minimum RBC delay should be 400us.
* /
CCM->CCR = (CCM_CCR_RBC_EN_MASK | CCM_CCR_COSC_EN_MASK |
CCM_CCR_OSCNT(0xAF));

/* Recover all the GPC interrupts. */
for (i = 0; i < LPM_GPC_IMR_NUM; i++)
{
    GPC->IMR[i] = gpcIMR[i];
}

/* Now delay for a short while (3usec) Arm is at 528MHz
at this point
* so a short loop should be enough. This delay is
required to ensure that
* the RBC counter can start counting in case an
interrupt is already pending
* or in case an interrupt arrives just as Arm is about
to assert DSM_request.
* /
for (i = 0; i < 22 * 24; i++)
{
    __NOP();
}
}
```

- 禁用 RBC\_EN:

```

/* Mask all GPC interrupts before disabling the RBC counters
 */
for (i = 0; i < LPM_GPC_IMR_NUM; i++)
{
    gpcIMR[i] = GPC->IMR[i];
    GPC->IMR[i] = 0xFFFFFFFFU;
}
/* Disable the RBC bypass counter */
CCM->CCR &= ~CCM_CCR_RBC_EN_MASK;
CCM->CCR &= ~CCM_CCR_REG_BYPASS_COUNT_MASK;

/* Now delay for 2 CKIL cycles (61usec). Arm is at 528MHz
at this point
 * so a short loop should be enough.
 */
for (i = 0; i < 528 * 22; i++)
{
    __NOP();
}

/* Recover all the GPC interrupts. */
for (i = 0; i < LPM_GPC_IMR_NUM; i++)
{
    GPC->IMR[i] = gpcIMR[i];
}
    
```

### 5.3.11 ERR007265 的解决方法

Errata ERR007265 描述了 SoC 可能无法进入低功耗模式的情况。

ERR007265: CCM: 当使用不正确的低功耗序列时，SoC在 Arm核执行 WFI之前进入低功耗模式。

软件解决方法:

1. 软件应通过设置 IOMUX\_GPR1\_GINT 来触发 GPR\_IRQ 始终挂起。
2. 在设置 CCM 低功耗模式之前，软件应在 GPC 中取消屏蔽 GPR\_IRQ。
3. 软件应在设置 CCM 低功耗模式后（设置 CCM\_CLPCR 的第0和1位）屏蔽 GPR\_IRQ。

### 5.3.12 Suspend 模式复位后如何运行

#### 5.3.12.1 介绍

Suspend 模式是一种功耗最低的低功耗模式。但随着核掉电，唤醒其实是复位唤醒。在 System idle 和 low power idle 下，唤醒时代码可以在 WFI 指令后继续运行，这很方便，但无法获得最小的功耗。在 Suspend 模式下可以获得最小的功耗。复位后，核可以检测是否是因 Suspend 模式唤醒而导致的复位（通过 PGC → CPU\_SR[PSR]）。

如果系统想在 WFI 指令后运行代码，恢复数据可以保存在 OCRAM 上，起始地址的偏移量为 32 KB。这意味着 FlexRAM0 不能在暂停模式下掉电。

#### 5.3.12.2 SDRAM 数据存储建议

在大多数应用中，用户可能希望使用 SDRAM 来存储数据。SDRAM 可以在 suspend 模式中自动进入自刷新模式，数据不会丢失。

警告：在进入暂停模式之前，应向 SDRAM 发送一个 STOP 请求，要求 SDRAM 进入自刷新模式。避免访问 SDRAM 以及在 OCRM 中使用变量是很重要的。

## 6 参考文献

1. [i.MX RT 1050 Reference Manual](#)
2. [Arm Cortex M7 Reference Manual](#)

## 7 修订记录

| 版本号 | 发布日期       | 说明   |
|-----|------------|------|
| 0   | 2023年1月10日 | 初次发布 |

## 8 Legal information

### 8.1 Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

### 8.2 Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Suitability for use in non-automotive qualified products** — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**Translations** — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

### 8.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

**NXP** — wordmark and logo are trademarks of NXP B.V.



## Contents

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | 介绍 .....                              | <b>2</b>  |
| <b>2</b> | 芯片概述 .....                            | <b>2</b>  |
| 2.1      | i.MX RT 芯片概述 .....                    | 2         |
| <b>3</b> | 低功耗概述 .....                           | <b>2</b>  |
| 3.1      | 电源 .....                              | 2         |
| 3.2      | 运行模式定义 .....                          | 3         |
| 3.3      | 运行模式配置 .....                          | 3         |
| 3.4      | 低功耗模式定义 .....                         | 4         |
| 3.5      | 低功耗模式配置 .....                         | 4         |
| 3.5.1    | 唤醒源 .....                             | 5         |
| <b>4</b> | 电源模式切换示例 .....                        | <b>5</b>  |
| 4.1      | 电源模式切换菜单 .....                        | 6         |
| 4.2      | 任务状态显示 .....                          | 6         |
| 4.3      | 唤醒源 .....                             | 7         |
| <b>5</b> | 低功耗应用设计 .....                         | <b>7</b>  |
| 5.1      | 总体描述 .....                            | 7         |
| 5.2      | 低功耗模式进入和退出顺序 .....                    | 8         |
| 5.2.1    | 进入 system idle 模式 .....               | 9         |
| 5.2.1.1  | 如何进入 .....                            | 9         |
| 5.2.1.2  | 如何退出 .....                            | 9         |
| 5.2.2    | 进入 low power idle 模式 .....            | 10        |
| 5.2.2.1  | 如何进入 .....                            | 10        |
| 5.2.2.2  | 如何退出 .....                            | 11        |
| 5.2.3    | 进入 Suspend 模式 .....                   | 12        |
| 5.2.3.1  | 如何进入 .....                            | 12        |
| 5.2.3.2  | 如何退出 .....                            | 13        |
| 5.2.4    | SNVS 模式 .....                         | 14        |
| 5.2.4.1  | 如何进入 .....                            | 14        |
| 5.2.4.2  | 如何退出 .....                            | 14        |
| 5.3      | 低功耗设计要点 .....                         | 15        |
| 5.3.1    | 如何改变核频率 .....                         | 15        |
| 5.3.2    | 如何改变 PLL 频率 .....                     | 15        |
| 5.3.3    | 如何更改 SPIFlash 和 SDRAM 的时钟源 .....      | 15        |
| 5.3.4    | 如何在 FreeRTOS 中启用 tickless 低功耗功能 ..... | 15        |
| 5.3.4.1  | 选择一个 FreeRTOS 滴答计时器 .....             | 15        |
| 5.3.4.2  | 选择一个无滴答计时器 .....                      | 16        |
| 5.3.4.3  | WFI 指令前后的 hook 函数 .....               | 16        |
| 5.3.5    | Doze 和 IPGSTOP 信号 .....               | 17        |
| 5.3.5.1  | Doze 模式 .....                         | 17        |
| 5.3.5.2  | Stop 模式 .....                         | 17        |
| 5.3.6    | 如何启用弱模拟 LDO .....                     | 18        |
| 5.3.7    | 如何禁用 OCRAM 电源 .....                   | 19        |
| 5.3.8    | 带隙切换 .....                            | 20        |
| 5.3.9    | 启用 DCDC 的 DCM 模式 .....                | 20        |
| 5.3.10   | 启用和禁用 RBC 位 .....                     | 21        |
| 5.3.11   | ERR007265 的解决方法 .....                 | 22        |
| 5.3.12   | Suspend 模式复位后如何运行 .....               | 22        |
| 5.3.12.1 | 介绍 .....                              | 22        |
| 5.3.12.2 | SDRAM 数据存储建议 .....                    | 22        |
| <b>6</b> | 参考文献 .....                            | <b>23</b> |
| <b>7</b> | 修订记录 .....                            | <b>23</b> |
| <b>8</b> | Legal information .....               | <b>24</b> |

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.