



## GR551x功耗模式及功耗测量说明

版本： 2.3

发布日期： 2025-06-06

版权所有 © 2025 深圳市汇顶科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得对本手册内的任何部分擅自摘抄、复制、修改、翻译、传播，或将其全部或部分用于商业用途。

## 商标声明

**GOODiX** 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人持有。

## 免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

深圳市汇顶科技股份有限公司（以下简称“GOODiX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODiX对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经GOODiX书面批准，不得将GOODiX的产品用作生命维持系统中的关键组件。在GOODiX知识产权保护下，不得暗或以其他方式转让任何许可证。

深圳市汇顶科技股份有限公司

总部地址：深圳市福田区梅康路1号汇顶科技总部大厦26楼

电话：+86-755-33338828      邮编：518000

网址：[www.goodix.com](http://www.goodix.com)

# 前言

## 编写目的

本文档将介绍GR551x系列芯片的功耗模式、功耗管理模式以及示例应用详解，以指导用户快速配置系统的功耗管理模式。

## 读者对象

本文适用于以下读者：

- 芯片用户
- 开发人员
- 测试人员
- 开发爱好者
- 文档工程师

## 版本说明

本文档为第11次发布，对应的产品系列为GR551x。

## 修订记录

版本	日期	修订内容
1.0	2019-12-08	首次发布
1.3	2020-03-16	更新“IO管脚配置”章节描述
1.5	2020-05-30	优化“睡眠模式”和“IO管脚配置”章节描述
1.6	2020-06-30	<ul style="list-style-type: none"><li>• 删除“IO管脚配置”章节代码中msio的声明</li><li>• 优化“睡眠流程验证”章节描述及示意图</li><li>• 删除“软件配置”章节中io_table的配置说明</li></ul>
1.7	2020-09-25	“环境搭建”章节，补充开始功耗测量前需先移除J5上跳线帽的说明、增加开发板硬件布局图
1.8	2021-08-09	更新“准备工作”章节
1.9	2022-02-20	基于SDK修改固件名称
2.0	2022-06-24	全文优化描述
2.1	2022-09-30	<ul style="list-style-type: none"><li>• 更新“配置示例工程”和“实现功耗管理模式”章节代码名称、代码路径和代码内容</li><li>• 删除“配置功耗管理模式”章节重复的说明</li></ul>
2.2	2025-02-17	更新“功耗模式”和“固件烧录”章节
2.3	2025-06-06	删除“IO管脚配置”章节

# 目录

前言.....	I
<b>1 简介.....</b>	<b>1</b>
1.1 功耗模式.....	1
1.2 唤醒源.....	1
1.3 功耗模式切换流程.....	2
1.4 参考文档.....	3
<b>2 Demo测试以及验证.....</b>	<b>4</b>
2.1 准备工作.....	4
2.2 睡眠模式唤醒流程验证.....	4
2.2.1 配置示例工程.....	4
2.2.2 固件烧录.....	6
2.2.3 输出波形验证.....	6
<b>3 功耗管理模式配置.....</b>	<b>7</b>
3.1 实现功耗管理模式.....	7
3.2 配置功耗管理模式.....	7
<b>4 测量功耗.....</b>	<b>9</b>
4.1 测量原理.....	9
4.2 环境搭建.....	9
4.3 进行测量.....	10
<b>5 常见问题.....</b>	<b>12</b>
5.1 热启动外设无法正常操作.....	12
5.2 J-Link断开.....	12
5.3 J-Link无法链接.....	12

# 1 简介

本章介绍了GR551x芯片的功耗模式、低功耗模式的唤醒源及其切换流程。

## 1.1 功耗模式

GR551x芯片支持三种功耗模式：工作模式、空闲模式、睡眠模式。

- 工作模式（Active Mode）

此模式下CPU处于全速运行状态，典型的情况是：

- MCU子系统（包括ARM处理器、SRAM和外设等）一直处于工作状态。
- 蓝牙子系统（包括RF收发机、通信内核等）一直处于工作状态。
- PMU子系统（包括DC/DC、LDO、RTC等）一直处于工作状态。

- 空闲模式（Idle Mode）

此模式下通过ARM系列自带的WFI/WFE（Wait For Interrupt/Event）指令实现。当调用此指令时，PC指针停留在WFI/WFE指令处，当发生IRQ或者Event来唤醒CPU时，PC指针指向下一条指令并继续执行。系统会根据外设/蓝牙子系统/Timer的使用情况自动判断是否进入该状态。

- 睡眠模式（Sleep Mode）

此模式下HFXO\_32M时钟停止运行，MCU子系统（Retention SRAM除外）和蓝牙子系统均处于断电状态，系统只有Always-on（AON）模块的电源运行，以保证Retention SRAM中存储的数据不丢失，同时为具有唤醒功能的模块（如Bluetooth LE（BLE）Timer、Sleep Timer、Real Time Calendar、AON GPIO）供电。当系统没有任务处理时，可以进入Sleep Mode。

系统功耗模式切换过程如下图所示，系统会自动根据当前的状态进行切换。

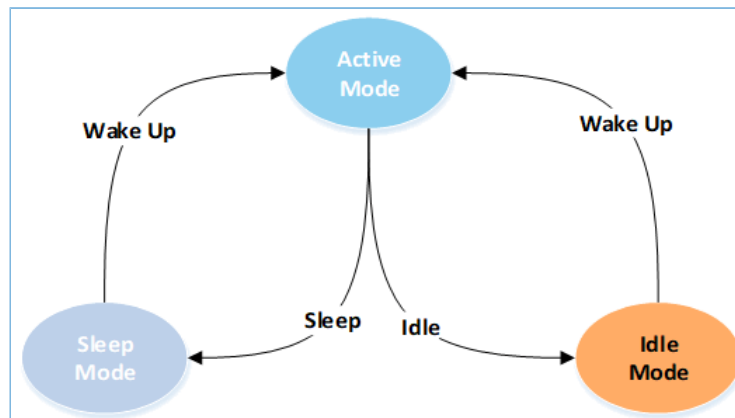


图 1-1 系统功耗模式切换过程

## 1.2 唤醒源

唤醒源可将系统从低功耗模式中唤醒，包括从Idle Mode唤醒为Active Mode、从Sleep Mode唤醒为Active Mode。

Idle Mode唤醒源：

- Reset
- NVIC Event
- Debug Event

Sleep Mode唤醒源:

- BLE Timer
- Sleep Timer
- Real Time Calendar
- AON GPIO
- LPCOMP
- Reset

用户可以调用SDK的API来配置唤醒源:

```
void pwr_mgmt_wakeup_source_setup(uint32_t wakeup_source);
```

唤醒源配置详情, 请参考《GR551x Datasheet》。

### 1.3 功耗模式切换流程

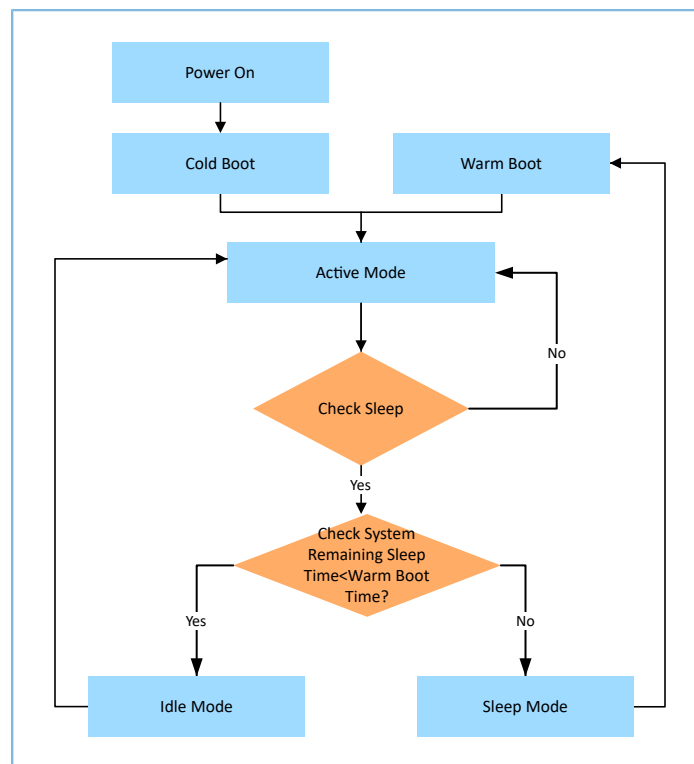


图 1-2 功耗模式切换流程图

一个典型的功耗模式切换流程（比如用户同时使用BLE和外设）描述如下：

1. 系统上电，冷启动。

2. 进入Active Mode。
3. 当无业务逻辑处理时，系统进入Check Sleep单元，检查当前系统是否空闲。
4. 如果Check Sleep单元输出为YES，则判断系统空余时间（如BLE Timer、Sleep Timer、RTC等timer的到期时间）。若小于热启动耗时，则系统会进入Idle Mode；若剩余系统空闲时间大于热启动耗时，则系统进入Sleep Mode，等待被唤醒。
5. 如果Check Sleep单元输出为No，则系统保持在Active Mode。

#### 说明:

从Sleep Mode进入热启动流程需要一段启动时间且启动电流较大。睡眠时间较短时（如小于2 ms），系统进入Sleep Mode后功耗降低不明显。从系统综合性能考虑，当睡眠时间较长时，系统进入Sleep Mode的平均功耗更低。

## 1.4 参考文档


在进行操作前，可参考以下文档。

表 1-1 文档参考

名称	描述
GR551x开发者指南	介绍GR551xSDK以及基于SDK的应用开发和调试
Keil用户指南	Keil详细操作说明： <a href="http://www.keil.com/support/man/docs/uv4/">www.keil.com/support/man/docs/uv4/</a>
Bluetooth GATT Spec	Bluetooth Profile和Service的详细信息查看地址： <a href="http://www.bluetooth.com/specifications/gatt">www.bluetooth.com/specifications/gatt</a>
GR551x Datasheet	介绍GR551x芯片引脚分布、内存、PMU、时钟、外设、安全核心、通信子系统和封装信息。

## 2 Demo测试以及验证

本章介绍如何使用和验证带睡眠模式的示例工程，SDK\_Folder\projects\ble\ble\_peripheral\目录下的示例工程均支持睡眠模式。

 说明:

SDK\_Folder为SDK的根目录。

### 2.1 准备工作

应用工程示例之前，需要完成以下准备工作。

- 硬件准备

表 2-1 硬件准备

名称	描述
开发板	GR5515 Starter Kit开发板（以下简称“开发板”）
连接线	Micro USB 2.0数据线、USB 2.0数据线
逻辑分析仪	Saleae逻辑分析仪、KeySight测试仪

- 软件准备

表 2-2 软件准备

名称	描述
Windows	Windows 7/Windows 10操作系统
J-Link Driver	J-Link驱动程序，下载网址： <a href="https://www.segger.com/downloads/jlink/">https://www.segger.com/downloads/jlink/</a>
Keil MDK5	IDE工具，支持MDK-ARM 5.20 及以上版本，下载网址： <a href="https://www.keil.com/download/product/">https://www.keil.com/download/product/</a>


### 2.2 睡眠模式唤醒流程验证

当系统处于Sleep Mode时，一旦产生BLE timer中断，系统将被唤醒并进入热启动流程。在热启动流程代码中，通过添加GPIO初始化模块（比如GPIO翻转），可实现GPIO在系统Sleep Mode和热启动下输出相应的消息，以验证睡眠模式与热启动流程之间转换是否正常。

下面章节将对睡眠流程验证的各步骤进行详细描述。

#### 2.2.1 配置示例工程

本文档以ble\_app\_pcs示例工程为例进行配置说明。

 说明:


- Ultra Deep Sleep是一种特殊的Deep Sleep模式。在该模式下，除Always-on（AON）模块的电源运行外，其他电源域均关闭电源，Retention SRAM电源也会被关闭，所以通过AON GPIO或Sleep Timer唤醒后，程序将执行冷启动流程。用户必须通过调用SDK层的pwr\_mgmt\_ultra\_sleep()才可以进入Ultra Deep Sleep模式。
- ble\_app\_pcs开始运行时，系统会调用pwr\_mgmt\_ultra\_sleep()进入Ultra Deep Sleep，并设置AON\_GPIO为唤醒源。长按开发板上的OK键3秒以上可以将系统从Ultra Deep Sleep中唤醒，并发起持续时长为30秒的广播。

1. 用Keil打开ble\_app\_pcs工程，路径为：SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_pcs\Keil\_5\ble\_app\_pcs.uvprojx
2. 在热启动流程的代码（main\_init函数）中，添加GPIO初始化函数。

用户可使用GPIO输出信息，该GPIO应该是当前应用工程中未被使用的GPIO（如AON\_GPIO\_3）。

```
void main_init(void)
{
    uint32_t boot_flag = get_wakeup_flag();
    if ( COLD_BOOT == boot_flag )
    {
        extern void __main(void);
        __main();
    }
    else
    {
        ll_aon_gpio_set_pin_mode(LL_AON_GPIO_PIN_3, LL_AON_GPIO_MODE_OUTPUT);
        ll_aon_gpio_disable_it(LL_AON_GPIO_PIN_3);
        ll_aon_gpio_toggle_pin(LL_AON_GPIO_PIN_3);
        ll_aon_gpio_toggle_pin(LL_AON_GPIO_PIN_3);

        warm_boot_process();
        while (1);
    }
    // Never execute here
}
```

 说明:

名称: gr\_platform.c

路径: SDK\_Folder\platform\soc\common

需要在该文件开始处添加#include "gr55xx\_ll\_aon\_gpio.h"，防止编译错误。

3. 配置系统功耗管理模式为睡眠自动管理模式：将pwr\_mgmt\_mode\_set()函数的入参设置为：PMR\_MGMT\_SLEEP\_MODE。

```
void app_periph_init(void)
```

```
{  
    SYS_SET_BD_ADDR(s_bd_addr);  
    wkup_key_init();  
    pwr_mgmt_mode_set(PMR_MGMT_SLEEP_MODE);  
}
```

#### 说明:

名称: *user\_periph\_setup.c*

路径: SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_pcs\Src\platform\

## 2.2.2 固件烧录

*ble\_app\_pcs*工程的源码位于SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_pcs。

用户可使用GProgrammer将*ble\_app\_pcs.bin*烧录至开发板。GProgrammer烧录固件的具体操作方法, 请参考《GProgrammer用户手册》。

#### 说明:

- *ble\_app\_pcs.bin*位于SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_pcs\Keil\_5\Listings\。
- GProgrammer安装文件可从[汇顶官网](#)获取。

## 2.2.3 输出波形验证

使用逻辑分析仪器捕捉GPIO口 (AON\_GPIO\_3) 的输出波形。

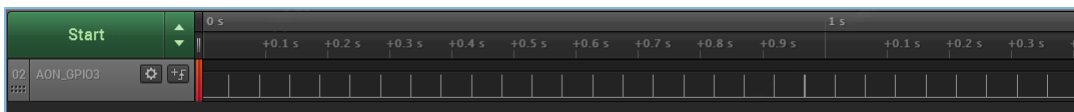


图 2-1 逻辑分析仪捕捉GPIO口的输出波形

AON\_GPIO\_3输出周期性的脉冲波 (见图 2-1), 表示系统的睡眠模式与热启动流程正常配合。其中, 两次脉冲之间代表系统进入Sleep Mode。当BLE Timer中断到来即刻唤醒整个系统, 使系统进入热启动。热启动流程中, GPIO被初始化, AON\_GPIO\_3翻转为高电平。

#### 说明:

系统处于Sleep Mode时J-Link会断开, 用户无法进行在线调试。推荐使用GPIO方式输出外部信息进行Sleep Mode调试。

## 3 功耗管理模式配置

本章主要介绍如何配置功耗管理模式：睡眠自动管理模式（PMR\_MGMT\_SLEEP\_MODE）、Idle自动管理模式（PMR\_MGMT\_IDLE\_MODE）和Active管理模式（PMR\_MGMT\_ACTIVE\_MODE）。

### 3.1 实现功耗管理模式


以ble\_app\_pcs工程为例，其main函数基本能满足常规的低功耗应用。用户可通过pwr\_mgmt\_schedule函数自动实现各种功耗管理模式。该main函数的代码片段如下所示。

```
int main(void)
{
    // Initialize user peripherals.
    app_periph_init();

    if (is_enter_ultra_deep_sleep())
    {
        pwr_mgmt_ultra_sleep(0);
    }

    // Initialize ble stack.
    ble_stack_init(ble_evt_handler, &heaps_table);

    // Loop
    while (1)
    {
        pwr_mgmt_schedule();
    }
}
```

 说明:

名称: *main.c*

路径: SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_pcs\Src\user\

### 3.2 配置功耗管理模式

用户可通过pwr\_mgmt\_mode\_set()函数配置系统的功耗管理模式。

```
/**
*****
 * @brief Set the specified sleep mode. When the setting is completed, the system will
 *         automatically enter the specified sleep mode through the strategy.
 * @param[in] pm_mode : sleep level
 * @retval : void
*****
 */
void pwr_mgmt_mode_set(pwr_mgmt_mode_t pm_mode);
```

 说明:

名称: *gr55xx\_pwr.h*

路径: SDK\_Folder\components\sdk

`pwr_mgmt_mode_set()`的形参`pm_mode`，用于设置系统的功耗管理模式。

表 3-1 `pwr_mgmt_mode_t`的取值

pm_mode值	说明
PMR_MGMT_ACTIVE_MODE	Active管理模式，系统保持在Active Mode，全速运行。
PMR_MGMT_IDLE_MODE	Idle自动管理模式，系统会自动在Active Mode与Idle Mode之间切换。 <ul style="list-style-type: none"><li>有待处理任务时，系统处于Active Mode。</li><li>无待处理任务时，系统处于Idle Mode，等待中断唤醒后进入Active Mode。</li></ul>
PMR_MGMT_SLEEP_MODE	睡眠自动管理模式，系统会自动在Active Mode、Idle Mode、Sleep Mode之间切换。 <ul style="list-style-type: none"><li>当有待处理任务时，系统处于Active Mode。</li><li>无待处理任务时，系统根据可预测的睡眠时间（AON Timer 与BLE Timer均可预测睡眠时间）判断进入Idle Mode或Sleep Mode。</li></ul>

## 4 测量功耗

本节以开发板作为待测板，简述如何使用功耗测试仪（KeySight N6705C DC Power Analyzer）对其进行功耗测试的方式和概要分析测试结果。

### 4.1 测量原理

使用功耗测试仪器为芯片供电时，芯片在运行过程中消耗的电流将经过功耗测试仪器，仪器将电流信号记录下来。

根据公式：

$$P = U * I \text{ (P为功耗；U为供电电压；I为电流；)}$$

U为电压保持不变，P的值与I值成正比，因此功耗仪器记录的电流曲线，便可以认为是功耗的曲线。

### 4.2 环境搭建

按照下图连接开发板（已下载2.2.2 固件烧录的固件）与KeySight功耗测试仪器（以KeySight N6705C为例），将功耗测试仪与开发板进行物理连接。

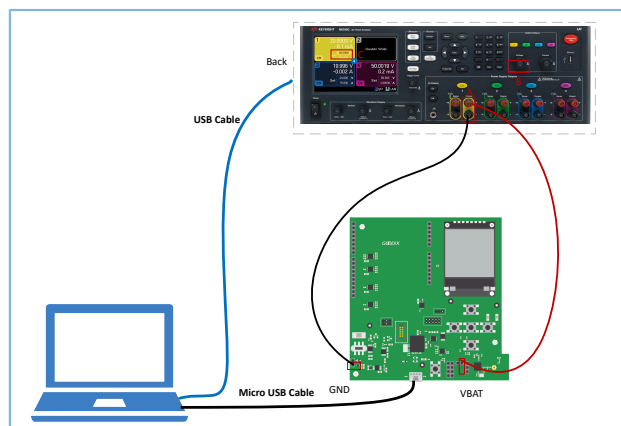


图 4-1 开发板和Keysight板连接图

1. 打开KeySight，用KeySight B按钮将通道的电压输出调整到3.3 V，即KeySight C处显示3.3 V。
2. KeySight的A+连接到开发板的VBAT端（开发板上的J10 Pin 2，如下图所示），为开发板上GR551x芯片供电。

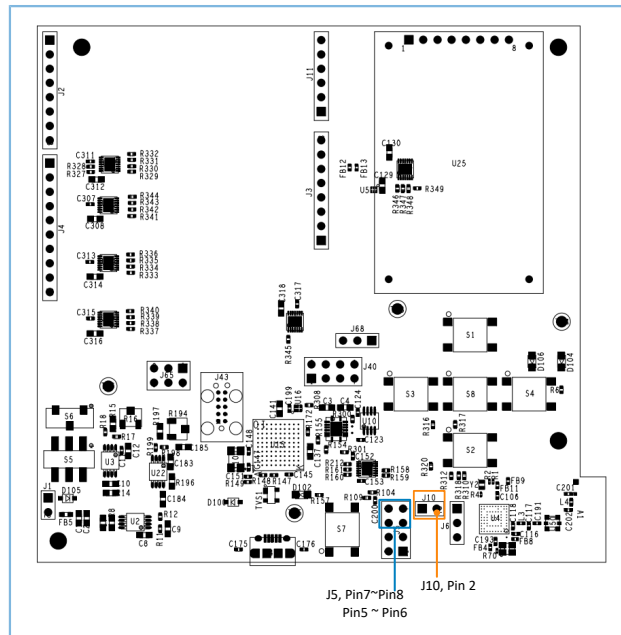


图 4-2 GR5515 Starter Kit开发板的硬件布局（顶层图）

3. Keysight的A-连接到开发板的GND端，形成供电环路。
4. 将开发板Micro USB接口接到PC机的任意USB端口，为开发板芯片外围电路供电。
5. 通过USB线将Keysight与PC机连接。

#### 说明:

在开发板测量功耗前，需要先将如图 4-2所示J5上Pin 5 ~ Pin 6、Pin 7 ~ Pin 8上的跳线帽移除，并取下开发板右上方的LCD屏幕。避免额外电流从VDDIO电源泄露，导致测量睡眠功耗异常。详细说明请参考《GR5xx IO引脚配置说明》。

## 4.3 进行测量

基本环境搭建完成后，使用Keysight与PC机的配套软件可以进行功耗的测试。

下图为一个典型的广播业务电流波形图：系统在Sleep模式下唤醒后，进行37、38、39信道的TX/RX业务，业务完毕后，再次进入了Sleep模式。



图 4-3 开发板广播业务功耗测试波形

下图为一个典型的连接业务电流波形图，也有同广播业务类似的电流时序。

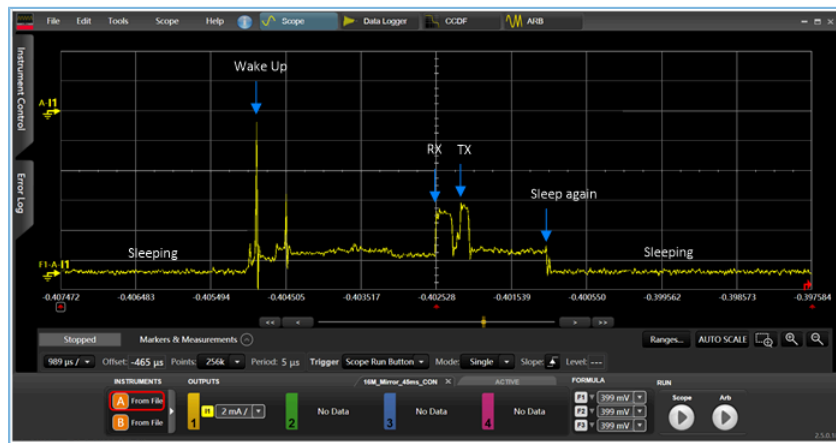


图 4-4 开发板连接业务功耗测试波形

## 5 常见问题

本章列出了系统在热启动、睡眠模式时可能出现的问题和解决办法。

### 5.1 热启动外设无法正常操作

- 问题描述

热启动唤醒后，外设不能正常运行。

- 问题分析

由于外设属于MCU Subsystem域，睡眠期间会关闭电压，导致原先的寄存器配置丢失。

- 处理方法

推荐用户采用App Drvier外设驱动接口，当热启动唤醒之后，系统会自动初始化外设模块。不推荐使用HAL Driver外设驱动接口，否则会出现唤醒后，在使用HAL层接口时会因外设寄存器未恢复而造成操作失败的情况。

### 5.2 J-Link断开

- 问题描述

使能Sleep自动管理模式后，应用程序会进入Sleep Mode。此时J-Link断开，无法继续调试。

- 问题分析

Sleep Mode下会关闭除Always-on域外的所有电源（包括J-Link电源）。J-Link不能保持正常连接。

- 处理方法

在开发初期，建议选择Active管理模式或者Idle自动管理模式进行开发。在开发后期，仅需验证睡眠功能是否正常使用。

### 5.3 J-Link无法链接

- 问题描述

将使能Sleep自动管理模式的应用下载到开发板，再次使用J-Link链接时，链接失败。

- 问题分析

开发板大部分时间处于Sleep Mode导致J-Link无法与PC端正常交互，无法链接。

- 处理方法

使用*custom\_config.h*中的BOOT\_LONG\_TIME配置，使能该字段后，在cold boot阶段，系统会增加500 ms的delay时间，用户可以利用这一段时间完成J-Link的链接，及时的下载代码刷新固件。