

## 前言

BlueNRG-1 是超低功耗蓝牙低能量（BLE）单模片上系统，符合 Bluetooth® 规范。其架构核心为 Cortex-M0 32 位。本应用笔记将介绍 BlueNRG-1 设备的低功耗模式。

---

**目录**

<b>1</b>	<b>BlueNRG-1 HW 低功耗模式</b> .....	<b>3</b>
<b>2</b>	<b>BlueNRG-1 低功耗支持</b> .....	<b>4</b>
<b>3</b>	<b>BlueNRG-1 低功耗模式示例</b> .....	<b>6</b>
	3.1 低功耗模式 SLEEPMODE_RUNNING .....	6
	3.2 低功耗模式 SLEEPMODE_CPU_HALT .....	6
	3.3 低功耗模式 SLEEPMODE_WAKETIMER .....	6
	3.4 低功耗模式 SLEEPMODE_NOTIMER.....	7
<b>4</b>	<b>缩略语列表</b> .....	<b>8</b>
<b>5</b>	<b>版本历史</b> .....	<b>9</b>

# 1 BlueNRG-1 HW 低功耗模式

为了在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡，BlueNRG-1 HW 支持三种低功耗模式：

- CPU-Halt 模式
  - 在该模式下，只有 CPU 停止工作。所有设备外设继续运行并可在发生中断/事件时唤醒 CPU。这是最低级别的节能模式。
- 睡眠模式
  - 在该模式下，CPU 停止且所有外设均被禁用。仅低速振荡器模块和外部唤醒源模块运行。
  - 唤醒源为 *唤醒定时器*或 *IO9*、*IO10*、*IO11*、*IO12* 和 *IO13*。
  - 当上述源触发唤醒时，系统回到运行模式且所有外设启用。应用需等到高速振荡器稳定后，才能从睡眠模式退出。
- 待机模式
  - 在该模式下，CPU 停止且所有外设均被禁用。唤醒源只有 *IO9*、*IO10*、*IO11*、*IO12* 和 *IO13*。
  - 应用需等到高速和低速振荡器稳定后，才能从待机模式退出。
  - 该模式是最高级别的节能模式。

BlueNRG-1 数据表上列出了所有低功耗模式的电流消耗。

## 2 BlueNRG-1 低功耗支持

BlueNRG-1 DK 软件包提供的软件支持所有 BlueNRG-1 HW 低功耗模式。

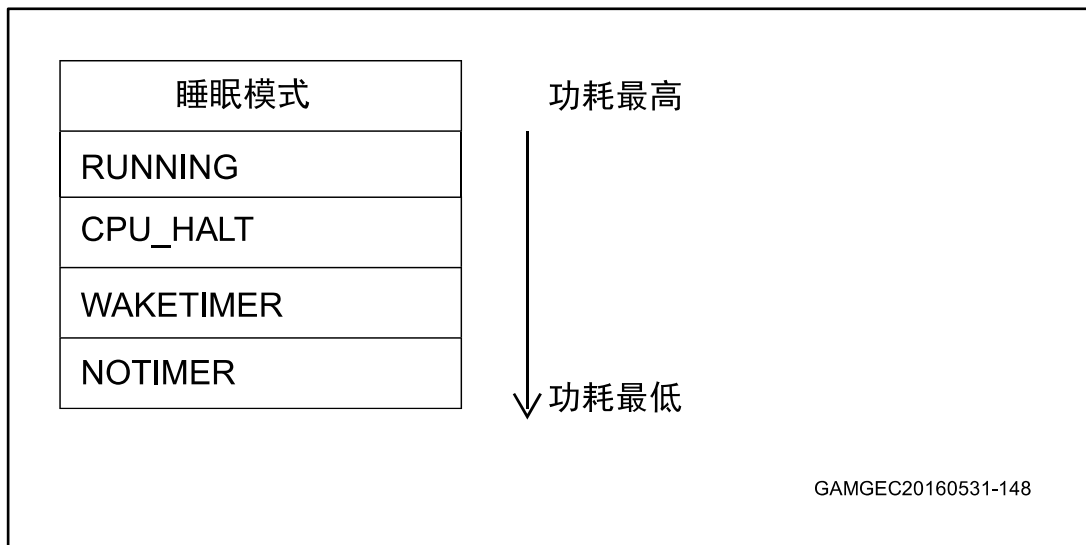
低功耗软件将来自应用的低功耗请求与无线操作模式相结合，选择适用于当前情况的最佳低功耗模式。在无线模块与应用请求之间进行这种协商的目的是避免丢失无线通信数据。由低功耗软件负责此类协商。

当 BlueNRG-1 退出任何低功耗模式时，发生复位。此复位导致所有外设配置和应用上下文丢失。低功耗软件采用一种机制，在节能程序被调用时保存并恢复所有外设配置和应用上下文。因此，从应用的角度来看，从低功耗程序退出是完全透明的，这意味着从低功耗模式唤醒时，CPU 继续执行低功耗函数被调用后的下一个指令。

低功耗软件执行下列节能模式：

- **SLEEPMODE\_RUNNING**
  - 在该低功耗模式下，一切处于激活并运行的状态。事实上不会使用该模式，定义它的目的是确保信息的完整性：它不是真正的节能模式。
- **SLEEPMODE\_CPU\_HALT**
  - 该节能模式执行 *HW CPU-Halt* 低功耗模式。
- **SLEEPMODE\_WAKETIMER**
  - 该节能模式执行 *硬件睡眠* 低功耗模式。
- **SLEEPMODE\_NOTIMER**
  - 该节能模式执行 *硬件待机* 低功耗模式。

图 1: BlueNRG-1 节能模式



为了启用任意低功耗模式，应用应调用函数“BlueNRG\_Sleep()”。

```
uint8_t BlueNRG_Sleep(SleepModes sleepMode,
                     uint8_t gpioWakeBitMask,
                     uint8_t gpioWakeLevelMask,
                     uint32_t sleep_time);
```

其中：

- **sleepMode** 是低功耗模式，将启用：

- SLEEPMODE\_RUNNING
- SLEEPMODE\_CPU\_HALT
- SLEEPMODE\_WAKETIMER
- SLEEPMODE\_NOTIMER
- **gpioWakeBitMask** 是可将芯片从低功耗模式唤醒的 IO 的位掩码：
  - WAKEUP\_IO9
  - WAKEUP\_IO10
  - WAKEUP\_IO11
  - WAKEUP\_IO12
  - WAKEUP\_IO13
 如果该字段为零，将忽略 **gpioWakeLevelMask** 参数。
- **gpioWakeLevelMask** 是用于设置每个唤醒源的激活唤醒级别的位掩码：
  - WAKEUP\_IO\_LOW, 当 IO 为低电平时唤醒系统
  - WAKEUP\_IO\_HIGH, 当 IO 为高电平时唤醒系统
- **sleep\_time** 是深度睡眠超时，以毫秒为单位。唤醒源也可以是 IO。如果此超时等于 0，节能软件将自动按照当前无线操作模式设置超时。

函数返回状态：

- **BLUENRG\_SLEEP\_CONFIGURATION\_ERROR**
- **SUCCESS**

低功耗软件导出对应用有用的其他函数：

- **BlueNRG\_WakeupSource.**
  - 该函数返回低功耗模式的上一个唤醒源。
- **BlueNRG\_IdleSleep.**
  - 该函数等待中断指令“WFI”，并暂停内核的执行，直至发生一次中断。这是使 BlueNRG-1 进入 *HW CPU-Halt* 低功耗模式的另一种方式。
- **App\_SleepMode\_Check.**
  - 该函数允许应用在特殊情况下设置其他低功耗模式。这意味着应用可以使用函数 **App\_SleepMode\_Check()** 修改在主循环中设置的低功耗模式。在调用 **BlueNRG\_Sleep()** 函数时，低功耗软件根据无线状态机、应用在主循环中请求的低功耗模式和从 **App\_SleepMode\_Check()** 请求的低功耗模式，与要应用的睡眠模式协商。
  - 例如，假设应用在主循环中设置了睡眠模式 SLEEPMODE\_WAKETIMER。中断向应用发送传感器准备提供一些测量值的信号。为避免进入睡眠模式，应用可以将 **App\_SleepMode\_Check()** 设置为 SLEEPMODE\_RUNNING，因此，当主循环执行 **BlueNRG\_Sleep()** 时，该函数与低功耗模式协商以应用 SLEEPMODE\_RUNNING。

## 3 BlueNRG-1 低功耗模式示例

下面几节提供了一些低功耗模式应用用例。

### 3.1 低功耗模式 SLEEPMODE\_RUNNING

在该低功耗模式下，一切处于激活并运行的状态：这实际上不是低功耗模式。典型源代码如下：

```
while(1) /*main loop*/
{
  /* BLE 协议栈时间片 */
  BTLE_StackTick();
  /* 应用程序时间片 */
  APP_Tick();
  /* 低功耗管理 */
  BlueNRG_Sleep(SLEEPMODE_RUNNING, 0, 0, 0);
}
```

该低功耗模式无需唤醒源。

### 3.2 低功耗模式 SLEEPMODE\_CPU\_HALT

在该模式下，只有 CPU 停止工作。所有外设继续运行并可在发生中断时唤醒 CPU。典型源代码如下：

```
while(1) /*main loop*/
{
  /* BLE 协议栈时间片 */
  BTLE_StackTick();
  /* 应用程序时间片 */
  APP_Tick();
  /* 低功耗管理 */
  BlueNRG_Sleep(SLEEPMODE_CPU_HALT,
                WAKEUP_IO13| WAKEUP_IO11,
                (WAKEUP_IO_LOW<< WAKEUP_IO11)|(WAKEUP_IO_LOW<< WAKEUP_IO13),
                0);
}
```

唤醒源为 IO13 和 IO11，它们对低电平触发。所有外设中断也可以唤醒 BlueNRG-1。

如果无线操作模式正在执行不能停止的操作，从而不允许此低功耗请求，则低功耗软件内部自动将此低功耗模式转换为 SLEEPMODE\_RUNNING。

### 3.3 低功耗模式 SLEEPMODE\_WAKETIMER

在该模式下，CPU 停止且所有外设均被禁用。仅低速振荡器模块和外部唤醒源模块运行。

典型源代码如下：

```
while(1) /*main loop*/
{
    /* BLE 协议栈时间片 */
    BTLE_StackTick();
    /* 应用程序时间片 */
    APP_Tick();
    /* 低功耗管理 */
    BlueNRG_Sleep(SLEEPMODE_WAKETIMER,
        WAKEUP_IO13,
        WAKEUP_IO_LOW<< WAKEUP_IO13,
        2000);
}
```

唤醒源为对低电平触发的 IO13 和 2 秒超时。

在本例中，应用想要启用低功耗模式 *WAKETIMER*。如果无线操作模式处于连接状态或广播状态，则栈接受应用请求的低功耗模式，但是如果有必要，将更改低功耗软件内部的超时，以符合连接间隔时间配置文件或广播间隔时间配置文件。

### 3.4 低功耗模式 SLEEPMODE\_NOTIMER

在该低功耗模式下，CPU 停止且所有外设均被禁用。仅外部唤醒源模块运行。典型源代码如下：

```
while(1) /*main loop*/
{
    /* BLE 协议栈时间片 */
    BTLE_StackTick();
    /* 应用程序时间片 */
    APP_Tick();
    /* 低功耗管理 */
    BlueNRG_Sleep(SLEEPMODE_NOTIMER,
        WAKEUP_IO13,
        WAKEUP_IO_HIGH << WAKEUP_IO13,
        0);
}
```

唤醒源是高电平触发的 IO13。

在本例中，应用想要启用低功耗模式 *NOTIMER*。如果无线模块处于连接状态，在与无线栈协商后，低功耗软件将低功耗模式更改为 *WAKETIMER*，以符合连接时间配置文件。如果无线模块处于空闲状态，无线栈将接受应用请求的低功耗模式 *NOTIMER*，并且低功耗软件不更改模式。

## 4 缩略语列表

表 1: 文档中所用缩略语的列表

术语	意义
BLE	低功耗蓝牙
HW	硬件



## 5 版本历史

表 2: 文档版本历史

日期	版本	变更
2016 年 6 月 29 日	1	初始版本。

表 3: 中文文档版本历史

日期	版本	变更
2017 年 6 月 17 日	1	中文初始版本。

## 重要通知 - 请仔细阅读

意法半导体公司及其子公司 (“ST”) 保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利