



Welink your smart

ME3630

模块电源管理指导手册

版本: V1.1

日期: 2018-04-29

LTE 模块



Website: www.gosuncnwelink.com

E-mail: welink@gosuncn.com

修订历史

| 版本 | 日期 | 说明 |
|-----|------------|---------|
| 1.0 | 2016-08-30 | 第一次发布版本 |
| 1.1 | 2018-04-28 | 更新文档模板 |

目录

| | |
|---|----|
| 修订历史..... | 2 |
| 目录 | 3 |
| 概述 | 4 |
| 1. 模块产品休眠唤醒机制..... | 6 |
| 1.1 电源管理相关管脚..... | 6 |
| 1.2 时序图 | 6 |
| 1.3 Module 休眠触发条件..... | 7 |
| 1.4 Module 唤醒触发条件..... | 7 |
| 2. 应用场景介绍..... | 8 |
| 2.1 场景 1：使用 USB 和 UART | 8 |
| 2.1.1 概述..... | 8 |
| 2.1.2 时序图及事件说明 | 9 |
| 2.2 场景 2：只使用 USB..... | 11 |
| 2.3 场景 3：使用 WAKEUP_IN、WAKEUP_OUT、USB 和 UART | 12 |
| 2.3.1 概述..... | 12 |
| 2.3.2 硬件连接..... | 12 |
| 2.3.3 时序图及事件说明 | 12 |
| 2.4 场景 4：使用 WAKEUP_IN、WAKEUP_OUT 和 USB | 13 |
| 2.5 场景 5：使用 WAKEUP_IN、WAKEUP_OUT 和 UART..... | 13 |

概述

模块产品的产生，对待机功耗要求越来越高。为了实现模块（Module）+主机（Host）的低功耗，高新兴物联 ME3630 模块支持休眠唤醒功能。该功能可以帮助 PAD 等移动通信终端进行长时间的待机，对其续航能力提供帮助。

模块作为主机的嵌入式设备，消耗着整机功耗。因此，电源管理系统最重要的任务就是通过一定场景下使能模块休眠机制来减少整机功耗。当需要通信的时候，主机及模块可以相互唤醒。故电源管理系统的另一个任务就是提供一个模块和主机的唤醒机制。

电源管理系统包括三个部分：主机软件（USB 驱动、UART 驱动、GPIO 驱动、休眠机制），主机及模块的硬件连接，模块的唤醒机制。

模块软件系统上运行着多个任务，休眠任务是众多任务中优先级最低的，用来探测模块是否能够进入休眠模式。其他任务（如 RF，SIM 卡、USB、UART、WAKEUP_IN）有投票权决定是否可以进入休眠。当没有其他任务运行的时候，休眠任务检测各任务是否投了赞成票，若是则模块能够自动进入休眠模式，如图 1 所示。此时，模块基带芯片减少工作频率，RF 进入 DRX（Discontinuous Reception mode）模式。

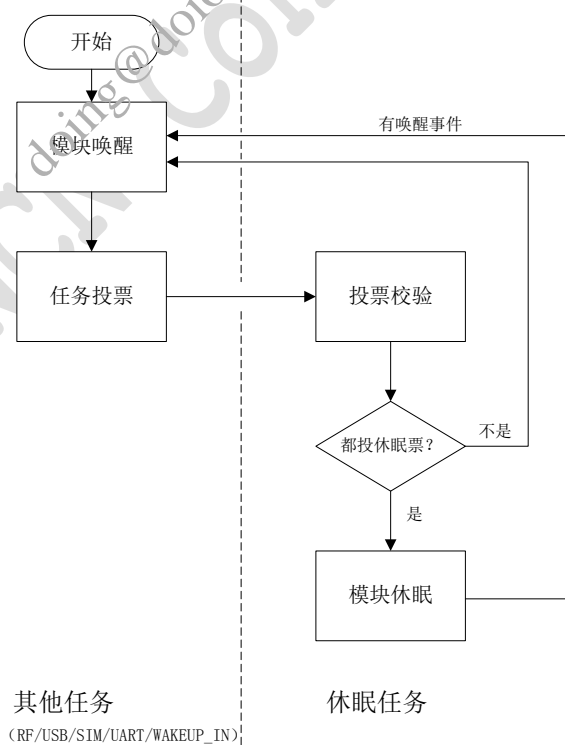


图 1

ME3630 模块电源管理指导手册

该文档不涉及主机侧系统软件。如果主机侧运行在 windows 系统上，高新兴物联提供模块相关的驱动程序。如果模块运行在 Android、Linux、WinCE 等操作系统，我司提供相应的驱动指导文档，指导客户配置 USB 的休眠唤醒功能。因 Linux-2.6.32 以上的内核已自带 selective suspend 电源管理特性，所以只需将电源管理的开关打开，USB 串口驱动即支持 selective suspend 电源管理。

doing@doiot.net 47.104.50.32 2019/1/7 20:49:59
GOSUNCN Confidential

1. 模块产品休眠唤醒机制

1.1 电源管理相关管脚

模块（Module）主要通过 USB、UART 与主机（Host）进行通信，如下表格描述了电源管理相关引脚，这些管脚对应的管脚号及管脚外围设计等详细信息请参考各模块的硬件设计指导。

表 1-1 电源管理相关引脚

| 管脚 | I/O | 描述 | 备注 | 时序图 |
|------------|-----|--|--|-----|
| WAKEUP_IN | I | 模块休眠唤醒信号；默认内部下拉，电平跳变时才触发动作； 高电平：模块唤醒； 低电平：模块可以休眠。 | 若 Host 支持 USB 挂起和唤醒功能，WAKEUP_IN 请勿使用，悬空即可。 | 图 1 |
| WAKEUP_OUT | O | 唤醒主机信号； 在模块处于休眠状态下，如果收到外部唤醒事件，如接收到短信、电话、网络数据时，该管脚输出一个低 1s 电平脉冲信号唤醒主机。 | 若模块处于唤醒态，接收到短信、电话、网络数据时，无信号输出，保持高电平。 | 图 2 |

注：I/O 为信号方向，I 表示该信号方向为模块侧输入，O 表示为模块侧输出。

1.2 时序图

模块运行状态与 WAKEUP_IN，WAKEUP_OUT 的关系，如下图。

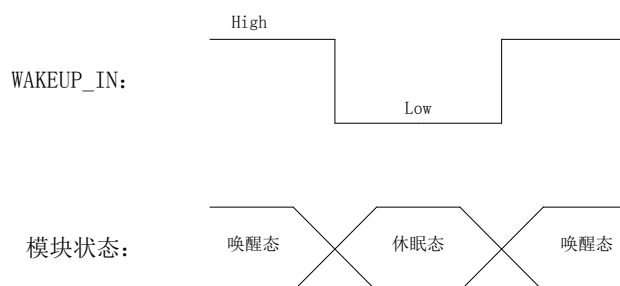


图 2

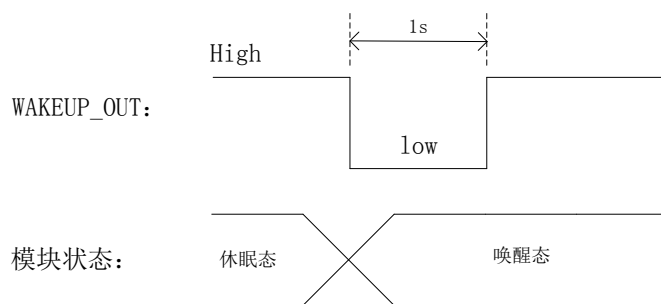


图 3

1.3 Module 休眠触发条件

- 1) Host 断开 USB 物理连接。
- 2) Host USB 挂起。
- 3) Host 拉低 WAKEUP_IN。

1.4 Module 唤醒触发条件

- 1) Host 重新连接 USB。
- 2) Host 唤醒 USB。
- 3) Host 拉高 WAKEUP_IN。
- 4) Module 收到外部唤醒事件, 如接收到短信、电话、网络数据等。
- 5) Module 收到非正常事件, 如天线断开、SIM 卡松动等。

2. 应用场景介绍

模块电源管理实际应用中可分为 5 种使用场景，按照数据通路方式可分类为 USB 和 UART 两种，根据休眠唤醒控制方式可分为 USB 热插拔、USB 挂起唤醒 和 WAKEUP_IN 一键休眠唤醒 三种。具体如下表所示：

| 分类 | 数据通路 | 休眠唤醒控制方式（主动） | 场景 |
|----|------------|------------------|-----------------|
| 1 | USB 和 UART | USB 热插拔 | 场景 1 USB 热插拔方式 |
| | | USB 挂起、唤醒 | 场景 1 USB 挂起唤醒方式 |
| | | WAKEUP_IN 一键休眠唤醒 | 场景 3 |
| 2 | USB | USB 热插拔 | 场景 2 |
| | | USB 挂起、唤醒 | 场景 2 |
| | | WAKEUP_IN 一键休眠唤醒 | 场景 4 |
| 3 | UART | WAKEUP_IN 一键休眠唤醒 | 场景 5 |

注：实际应用中，若 Host 支持 USB 挂起、唤醒功能，请勿使用 WAKEUP_IN 控制方式。休眠唤醒控制方式只能使用一种，不能多种混用。

2.1 场景 1：使用 USB 和 UART

2.1.1 概述

在该应用场景下，USB 和 UART 作为数据通路，Module 休眠唤醒方式可进一步分为两种：USB 热插拔 和 USB 挂起唤醒 两种休眠唤醒方式。

- USB 热插拔方式

USB 物理上连接后，Module 进入唤醒状态并保持；USB 物理上断开连接，Module 进入休眠状态，当 Module 收到外部唤醒事件（如接收到短信、电话、网络数据）时，WAKEUP_OUT 会输出唤醒信号。

注意：管脚连接要求：USB、UART（TX、RX 等）、WAKEUP_OUT。

（建议 Host 使用带中断（下降沿或低电平）功能的输入型管脚检测 WAKEUP_OUT）

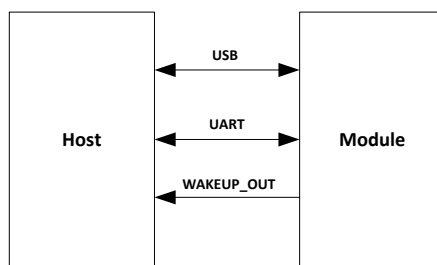


图 4

- USB挂起唤醒方式

Host端的USB控制器必须支持USB挂起（Suspend）和远程唤醒（remote wakeup）功能。Host配置请参考《GOSUNCN Wireless Module Driver User Guide for EmbeddedLinux》文档。

若Host不支持远程唤醒功能时，硬件设计上必须使用WAKEUP_OUT。

（建议在Host和Module适配调试初期，硬件上预留WAKEUP_OUT连接。）

管脚连接要求：USB、UART（TX、RX等）、WAKEUP_OUT（选用）。

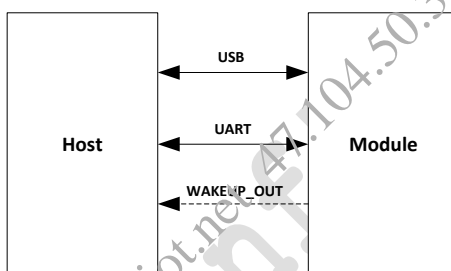


图 5

2.1.2 时序图及事件说明

- USB热插拔方式：

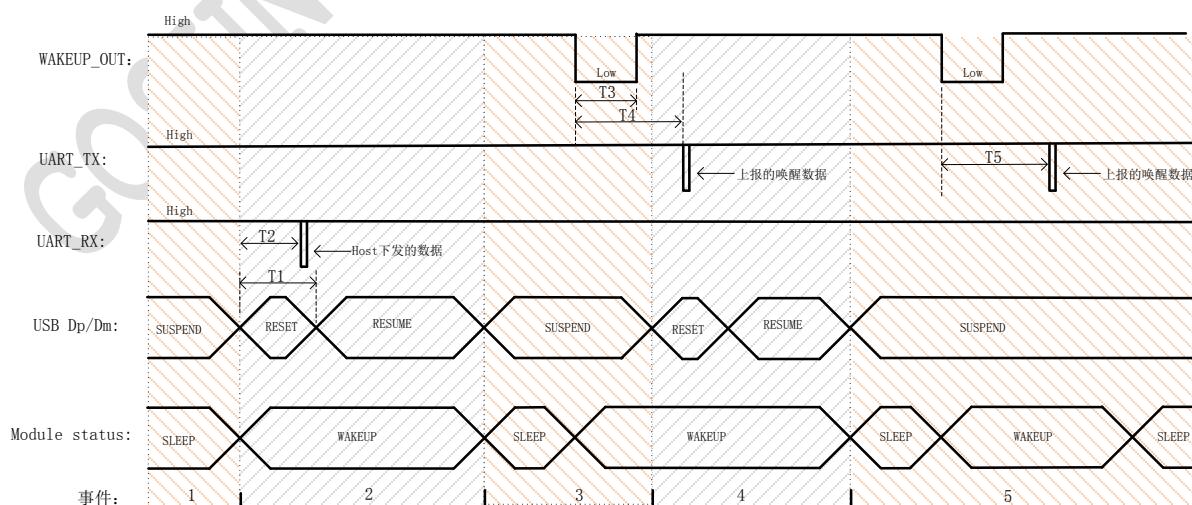


图 6 USB 热插拔方式

图 6 中五个事件的说明：

1) USB 热拔

Host 断开与 Module USB 物理连接后，Module USB 进入 SUSPEND，进而 Module 进入休眠。

2) USB 热插

Host 与 Module USB 物理连接动作完成后，Module 唤醒，Module 的 USB 会重新枚举，需经过 T1 时间后才能正常工作；Module UART 需经过 T2 时间才能正常工作。

3) USB 热拔后，Module 收到唤醒事件

Host 断开与 Module USB 物理连接后，Module 进入休眠；Module 在休眠期间收到唤醒事件，WAKEUP_OUT 输出 T3 低电平信号。

4) Module 收到唤醒事件后，USB 热插

接 3)，Host 必须在 T4 时间内 UART 准备好，才能接收 Module UART 上报的数据；USB 物理连接好后，才能接收 Module USB 上报的数据。

5) Module 收到唤醒事件后，USB 不热插

接 3)，Host 必须在 T5 时间内 UART 准备好，才能接收 Module UART 上报的数据。

图 6 中各个 T 的说明：（单位秒）

| 间隔标号 | 说明 | Min | Typ | Max |
|------|---|-----|-----|-----|
| T1 | USB 热插动作到 Module USB 可以正常工作的时间间隔。 | | 1.5 | |
| T2 | USB 热插动作到 Module UART 可以正常工作的时间间隔。 | | 1 | |
| T3 | WAKEUP_OUT 低电平持续的时间间隔。 | | 1 | |
| T4 | Module 收到的唤醒事件后，USB 热插时，WAKEUP_OUT 下降沿到 Module UART 上报唤醒数据的时间间隔。 | | 3.3 | |
| T5 | Module 收到的唤醒事件后，USB 不热插时，WAKEUP_OUT 下降沿到 Module UART 上报唤醒数据的时间间隔。 | | 3.3 | |

- USB挂起唤醒方式:

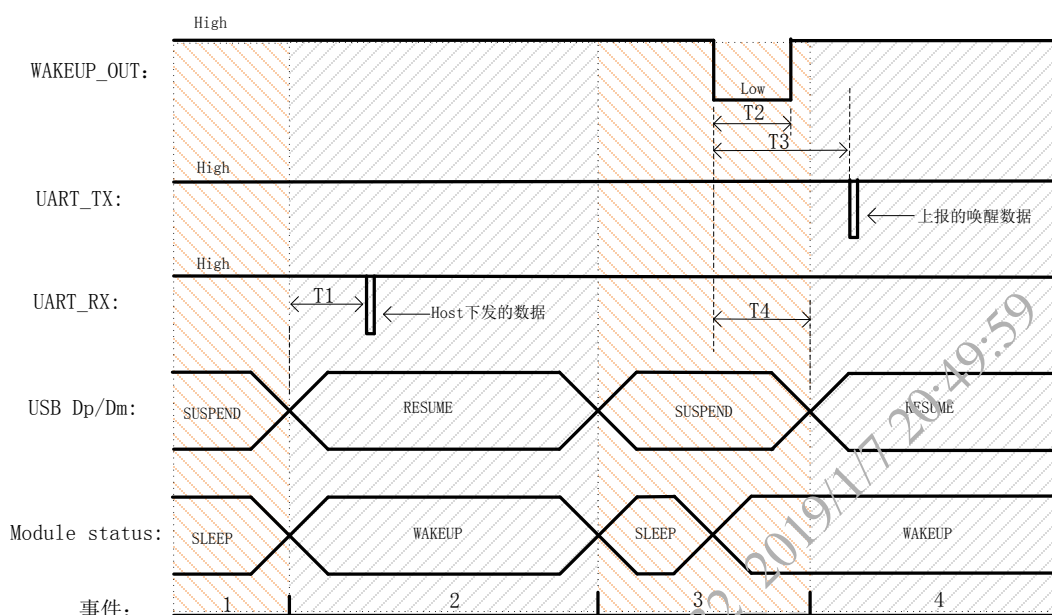


图 7 USB 挂起唤醒方式

图 7 中四个事件的说明：

- 1) USB suspend
Host USB 挂起，Module 进入休眠。
- 2) USB resume
Host USB 唤醒后，Module 的 UART 需经过 T1 时间才可以正常工作。
- 3) USB suspend，Module 休眠过程中收到唤醒事件
Host USB 挂起，Module 休眠期间收到唤醒事件，WAKEUP_OUT 输出 T2 低电平信号，Module UART 将在 T3 后上报唤醒数据。
- 4) USB resume，Module USB 唤醒 Host
Module WAKEUP_OUT 发出唤醒信号后，Host USB 必须在 T4 时间内唤醒，才能接收 Module USB 上报的唤醒数据。

图 7 中各个 T 的说明：（单位秒）

| 间隔标号 | 说明 | Min | Typ | Max |
|------|---|-----|-----|-----|
| T1 | Host USB 唤醒动作到 Module 的 UART 可以正常工作的时间间隔。 | 0.6 | 0.8 | |
| T2 | WAKEUP_OUT 低电平持续的时间间隔。 | 0.8 | 1 | |
| T3 | WAKEUP_OUT 下降沿到 Module UART 上报唤醒数据的时间间隔。 | | 3.3 | |
| T4 | WAKEUP_OUT 下降沿到 Module USB resume 的时间间隔。 | 0 | | 1.5 |

2.2 场景 2：只使用USB

该应用场景下，模块休眠唤醒机制与 2.1 场景 1 相同，忽略 UART 部分即可。

2.3 场景 3：使用 WAKEUP_IN、WAKEUP_OUT、USB 和 UART

2.3.1 概述

该应用场景，USB 和 UART 作为数据通路，在 USB 物理连接不断开、Host 不支持 USB 挂起和远程唤醒功能情况下，Host 可以通过拉高、拉低 WAKEUP_IN 来控制 Module 唤醒、休眠，WAKEUP_OUT 用于 Module 唤醒 Host。

2.3.2 硬件连接

管脚连接要求：USB、UART（TX、RX 等）、WAKEUP_IN、WAKEUP_OUT。
（建议 Host 使用带中断（下降沿或低电平）功能的输入型管脚检测 WAKEUP_OUT）

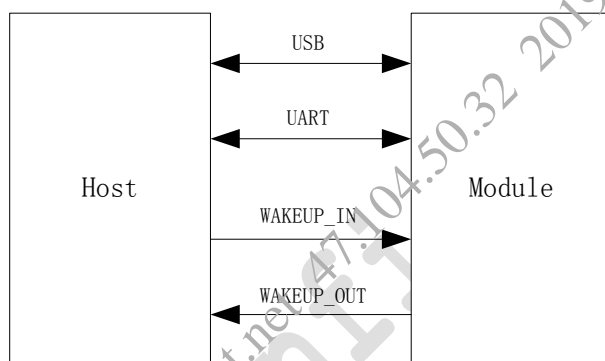


图 8

2.3.3 时序图及事件说明

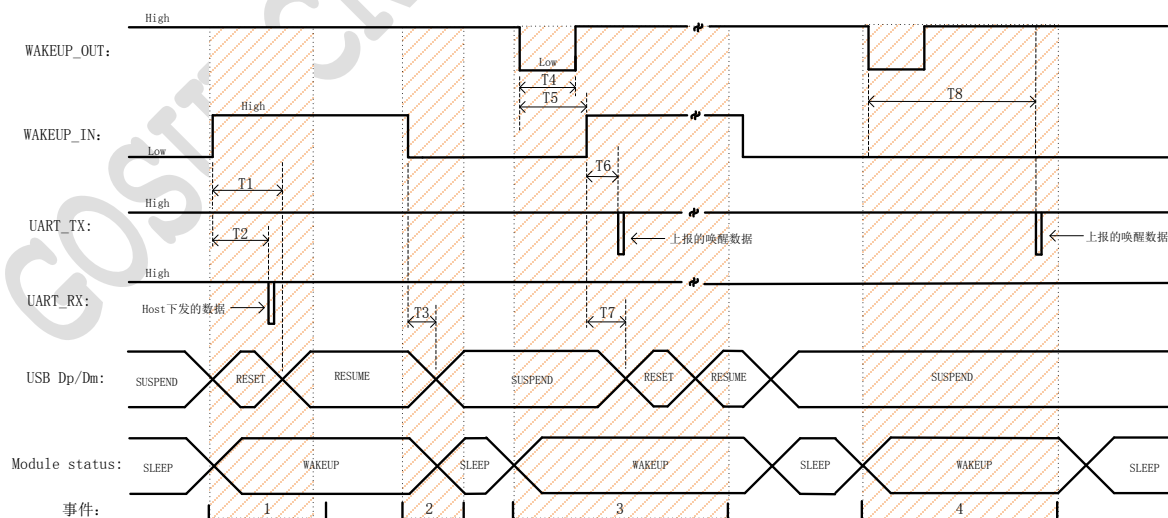


图 9

图 9 中四个事件的说明：

1) Host 主动唤醒 Module

Host 通过拉高 WAKEUP_IN 可以唤醒 Module, USB 会重新枚举, 必须在 T1 时间后才可正常使用, UART 必须在 T2 时间后才可正常使用。

2) Host 主动让 Module 休眠

当 Host 与 Module 没有数据业务, 需要让 Module 进入休眠时, 可以通过拉低 WAKEUP_IN, Module 的 USB 经过 T3 时间后进入 USB suspend, 进而 Module 整机休眠。

3) Module 收到外部唤醒事件, Host 及时响应事件

Module 收到外部唤醒事件时 (如接收到短信、电话、网络数据等), WAKEUP_OUT 将输出一个 T4 低电平信号, Host 检测到信号后自身唤醒, 需在 T5 时间内拉高 WAKEUP_IN 响应事件, 在 T6 时间内 UART 准备好才能接收到唤醒数据, USB 在 T7 时间后会重新枚举。

4) Module 收到外部唤醒事件, Host 不响应事件

Module 收到外部唤醒事件时 (如接收到短信、电话、网络数据等), WAKEUP_OUT 将输出一个 T4 低电平信号, 若 Host 不拉高 WAKEUP_IN 响应事件, T8 时间超时后 Module 将直接上报唤醒数据。

图 9 中各个 T 的说明: (单位秒)

| 间隔标号 | 说明 | Min | Typ | Max |
|------|--|-----|-----|-----|
| T1 | Host 主动唤醒 Module, WAKEUP_IN 上升沿到 USB resume 的间隔。 | | 2 | |
| T2 | Host 主动唤醒 Module, WAKEUP_IN 上升沿到 Module UART 可以接收处理数据的间隔。 | 0.8 | 1 | |
| T3 | Host 主动让 Module 休眠时, WAKEUP_IN 下降沿到 USB suspend 的间隔。 | | 1 | |
| T4 | WAKEUP_OUT 低电平持续的时间间隔。 | | 1 | |
| T5 | Host 拉高 WAKEUP_IN 及时响应外部唤醒事件的时间间隔。 | 0 | 1 | 3 |
| T6 | WAKEUP_IN 上升沿到 Module UART 上报唤醒数据的时间间隔。 | | 0.8 | |
| T7 | Host 拉高 WAKEUP_IN 响应唤醒事件时, WAKEUP_IN 上升沿到 USB resume 的时间间隔。 | | 1.2 | |
| T8 | Host 不拉高 WAKEUP_IN 响应唤醒事件时, WAKEUP_OUT 下降沿到 Module UART 上报唤醒数据的时间间隔。 | | 3.3 | |

2.4 场景 4: 使用 WAKEUP_IN、WAKEUP_OUT 和 USB

该应用场景下, 模块休眠唤醒机制与 2.3 场景 3 相同, 忽略 UART 部分即可。

2.5 场景 5: 使用 WAKEUP_IN、WAKEUP_OUT 和 UART

该应用场景下, 模块休眠唤醒机制与 2.3 场景 3 相同, 忽略 USB 部分即可。