

Fibocom

完美无线体验

FIBOCOM L610 & LC610N & LG610N 系列 休眠测试指南

文档版本：V1.0.7

更新日期：2020-07-14



适用型号

序号	产品型号	说明
1	L610 系列	NA
2	LC610N 系列	NA
3	LG610N 系列	NA

FIBOCOM
Confidential

版权声明

版权所有©2020 深圳市广和通无线股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

商标申明



为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。

版本记录

文档版本	编写人	主审人	批准人	更新日期	说明
V1.0.7	王海龙	邓利斌	龙忠友	2020-07-14	文档内容修改
V1.0.6	王海龙	邓利斌	龙忠友	2020-06-16	文档格式内容修改
V1.0.5	王海龙	邓利斌	龙忠友	2020-05-31	文档格式内容修改
V1.0.4	王海龙	邓利斌	龙忠友	2020-05-28	文档格式修改，按照公司要求整理
V1.0.3	王海龙	邓利斌	龙忠友	2020-05-15	添加 VBUS 休眠控制，添加尾末备注
V1.0.2	王海龙	邓利斌	龙忠友	2020-05-08	添加 USB 休眠的描述
V1.0.1	王海龙	邓利斌	龙忠友	2020-03-30	去掉一些与项目无关的描述
V1.0.0	王海龙	邓利斌	龙忠友	2020-02-28	初始版本

目录

1	前言	5
2	休眠	6
2.1	UART 控制下的 sleep	6
2.1.1	GTLPMODE 为 0	6
2.1.2	GTLPMODE 不为 0	6
2.1.2.1	GTLPMODE=1,X	6
2.1.2.2	GTLPMODE=2,X	6
2.2	USB 控制下的休眠	7
2.2.1	USB_SUSPEND/RESUME 控制的休眠	7
2.2.2	VBUS 控制下的休眠	7
3	唤醒	8
3.1	AT 唤醒	8
3.2	外部 GPIO 中断唤醒	8
3.3	SMS/CALL/DATA 唤醒	8
3.3.1	唤醒信号示例	8

1 前言

L610 系列、LC610N 系列、LG610N 系列模块（以下简称模块）分 AP 和 CP，不同的应用都有自己的控制资源（PM_SOURCE），在所有的应用都释放掉自己的 PM_SOURCE 之后，模块会进入 sleep，如果有任一应用当前资源没有释放，就会导致当前模块不能直接进入休眠，一直到被占有资源释放为止。类似于投票机制，所有应用都投票进入休眠，模块才能进入。模块是通过控制 UART 和 USB 的资源释放来尝试进入休眠的（一般来讲，UART 是模块在其他应用释放资源后还一直占有资源的应用）。

2 休眠

模块的休眠分为两种情况：

- 1) UART 控制下的 sleep
 - 2) USB 控制下的 sleep
- 并且要求是 DC 供电。

2.1 UART 控制下的 sleep

UART 控制下的 sleep 的先决条件是 GTLPMMODE 指令（USB 不插入）。

2.1.1 GTLPMMODE 为 0

使用 AT+GTLPMODE 指令来设置进入 sleep 等待的时长。

举例：

```
AT+GTLPMODE=0
```

```
AT+GTLPMODE=3 //3 秒之后放开 UART 控制资源，尝试休眠
```

2.1.2 GTLPMMODE 不为 0

2.1.2.1 GTLPMMODE=1,X

由 WAKEUP_IN 脚来控制模块进入或者退出休眠。当 SUB_MODE 为 0 时，是高电平休眠，低电平唤醒；为 1 时，是低电平休眠，高电平唤醒。需要下发 AT+CSCLK=1。

举例：

```
AT+GTLPMODE=1,1 // WAKEUP_IN 拉低进入休眠，拉高唤醒
```

```
AT+GTLPMODE=1,0 // WAKEUP_IN 拉高进入休眠，拉低唤醒
```

2.1.2.2 GTLPMMODE=2,X

由 DTR 脚来控制模块进入或者退出休眠。当 SUB_MODE 为 0 时，是高电平休眠，低电平唤醒；为 1 时，是低电平休眠，高电平唤醒。需要下发 AT+CSCLK=1

举例：

```
AT+GTLPMODE=2,1 // DTR 拉低进入休眠，拉高唤醒
```

```
AT+GTLPMODE=2,0 // DTR 拉高进入休眠，拉低唤醒
```

2.2 USB 控制下的休眠

2.2.1 USB_SUSPEND/RESUME 控制的休眠

(USB 设备需要插入)

1, 首先要下发 ATS24 指令, 先关闭 UART 的资源 (参考 2.1.1)。注意, 当前 USB_SUSPEND 对 PC 端有要求, 只能是 Linux 系统, Windows 不支持。

2, 然后进入 /sys/bus/usb/devices# 下先下发 dmesg 查看是否属于 usb 1-1, 再 cd 1-1 进入 power 目录

如下图:

```

1-0:1.0 1-1:1.0 1-1:1.2 1-1:1.4 1-1:1.6 1-1:1.8 2-1 2-2 usb1
1-1 1-1:1.1 1-1:1.3 1-1:1.5 1-1:1.7 2-0:1.0 2-1:1.0 2-2:1.0 usb2
root@ubuntu:/sys/bus/usb/devices# cd 1-1
root@ubuntu:/sys/bus/usb/devices/1-1# ls
1-1:1.0 authorized bMaxPower driver product
1-1:1.1 avoid_reset_quirk bNumConfigurations ep_00 quirks
1-1:1.2 bcdDevice bNumInterfaces idProduct removable
1-1:1.3 bConfigurationValue busnum idVendor remove
1-1:1.4 bDeviceClass configuration ltm_capable speed
1-1:1.5 bDeviceProtocol descriptors manufacturer subsystem
1-1:1.6 bDeviceSubClass dev maxchild uevent
1-1:1.7 bmAttributes devnum port urbnum
1-1:1.8 bMaxPacketSize0 devpath power version
root@ubuntu:/sys/bus/usb/devices/1-1# cd power/
root@ubuntu:/sys/bus/usb/devices/1-1/power# ls
active_duration runtime_active_kids wakeup_active
async runtime_active_time wakeup_active_count
autosuspend runtime_enabled wakeup_count
autosuspend_delay_ms runtime_status wakeup_expire_count
connected_duration runtime_suspended_time wakeup_last_time_ms
control runtime_usage wakeup_max_time_ms
level wakeup wakeup_total_time_ms
persist wakeup_abort_count
    
```

3, 设置 level 为 auto 状态, 因为没有那个 off 设置, auto 就相当于关闭了

```

root@ubuntu:/sys/bus/usb/devices/1-1/power# echo auto >level
root@ubuntu:/sys/bus/usb/devices/1-1/power# echo auto >control
root@ubuntu:/sys/bus/usb/devices/1-1/power#
    
```

此时查看程控电源的电流值已经直线下降了, 可以监测休眠后的电流值了。

备注: USB_SUSPEND 休眠因为部分器件并没有像 USB 不插入那样掉电, 导致休眠电流会比 VBUS (不插入 USB 设备) 休眠电流偏大 1.4mA。

2.2.2 VBUS 控制下的休眠

通过操作 USB_VBUS 电平控制 USB 休眠或唤醒。模块检测 USB_VBUS 中断以确定是否需要睡眠。如果 USB_VBUS 为低电平, 则释放 USB 的资源, 准备进入休眠。如果 USB_VBUS 为高电平, 则模块处于唤醒状态 (典型体现为插拔 USB)。在 USB 释放资源后, 再释放其他应用的资源 (比如 UART, timer, mutex, semaphore 等), 以进入休眠状态。

3 唤醒

在模块进入 sleep 之后，模块可以接收外部 GPIO 中断，AT，incoming call，SMS，data 等方式唤醒，下面将详细介绍。

3.1 AT 唤醒

通过串口工具，发送 AT 指令，可以唤醒模块，唤醒后模块如果没有继续有抢占资源的动作，模块会继续休眠下去。（如果采用 ATS24 方式休眠，再次进入休眠时间为 ATS24 设置时长。如果为非 ATS24 方式休眠，默认 500ms 后再次进入休眠）。

3.2 外部 GPIO 中断唤醒

模块在 sleep 状态下，GPIO 均可以响应外部中断，并唤醒模块。默认软件版本中，WAKEUP_IN 和 DTR 脚支持响应外部中断。

具体内容参考 2.1 章节。

3.3 SMS/CALL/DATA 唤醒

L610 在模块休眠之前没有执行 AT+CFUN=4/0 情况下，休眠之后可以通过 SMS/call/data 来唤醒模块，唤醒模块之后，根据 AT+GTWAKE，AT+GTPMTIME，AT+WRIM 三条指令来决定模块唤醒后的动作。

3.3.1 唤醒信号示例

command	Example	使用说明
AT+GTWAKE	AT+GTWAKE=0	RI 脚默认为高，当接收到 SMS/incoming call/data 时候，会发出低电平波形信号， 输出波形长度由 AT+WRIM 指令决定
	AT+GTWAKE=2,0	RI 脚默认为低电平，当接收到 SMS/incoming call/data 时候，会输出高电平波形信号， 输出波形长度由 AT+GTPMTIME 决定
	AT+GTWAKE=2,1	RI 脚默认为高电平，当接收到 SMS/incoming call/data 时候，会输出低电平波形信号， 输出波形长度由 AT+GTPMTIME 决定
AT+GTPMTIME	AT+GTPMTIME= 200,200,1000	设置 RI 脚信号宽度为 1.2 秒

command	Example	使用说明
AT+WRIM	AT+WRIM=0,1000	默认值, RI 脚在接收来电的时候波形长度为 1000 毫秒
	AT+WRIM=1,150	默认值, RI 脚在接收新短信息的时候波形长度为 150 毫秒
	AT+WRIM=2,500	RI 脚在接收数据的时候波形长度为 500 毫秒

小结:

当 AT+GTWAKE 设置值非 0 时, 输出波形由 GTPMTIME 设置。

当 AT+GTWAKE 设置值为 0 时, 输出波形由 WRIM 设置。



备注:

如果在释放了 UART 和 USB 资源后, 还不能进入休眠状态, 需要去查看一下是否有其他应用占住了资源没释放, 比如, PWM, I2C, timer, mutex, semaphore 等等。