QuecOpen EC2X&AG35 低功耗-休眠唤醒方案

www.quectel.com

About the Document

本文档适用于 EC2X 和 AG35 平台

History

Revision	Date	Author	Description
3.0	2018-09-02	Gale	Optimize QuecOpen LPM feature.

目录

1. QuecOpen LPM 概要

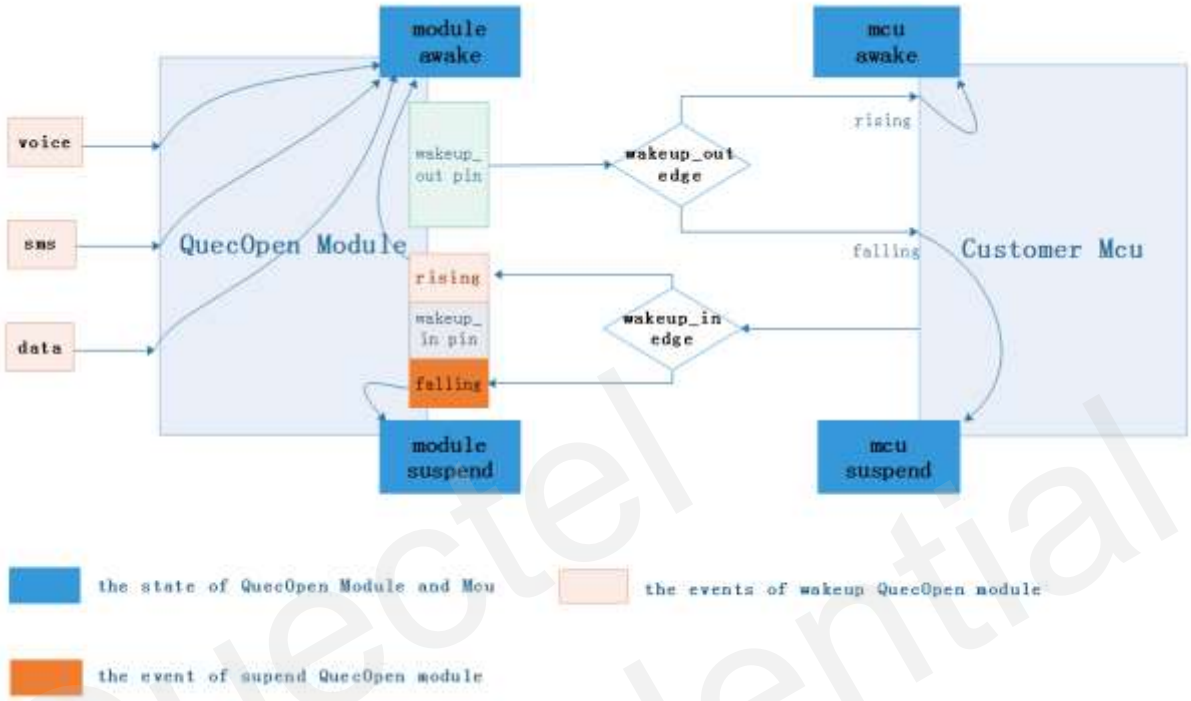
QuecOpen LPM 低功耗方案适用于带有外部 mcu 的产品，典型为车载 T-box：
QuecOpen 低功耗方案基于 Linux 以下几种技术开发：

1. Autosleep： 使能 autosleep 后,软件始终有冻结进程挂起外设以及 CPU 进入休眠的趋势；
2. Wakelock： 内核或者任意应用进程持有一个或多个 wakelock，都会抑制模块休眠；

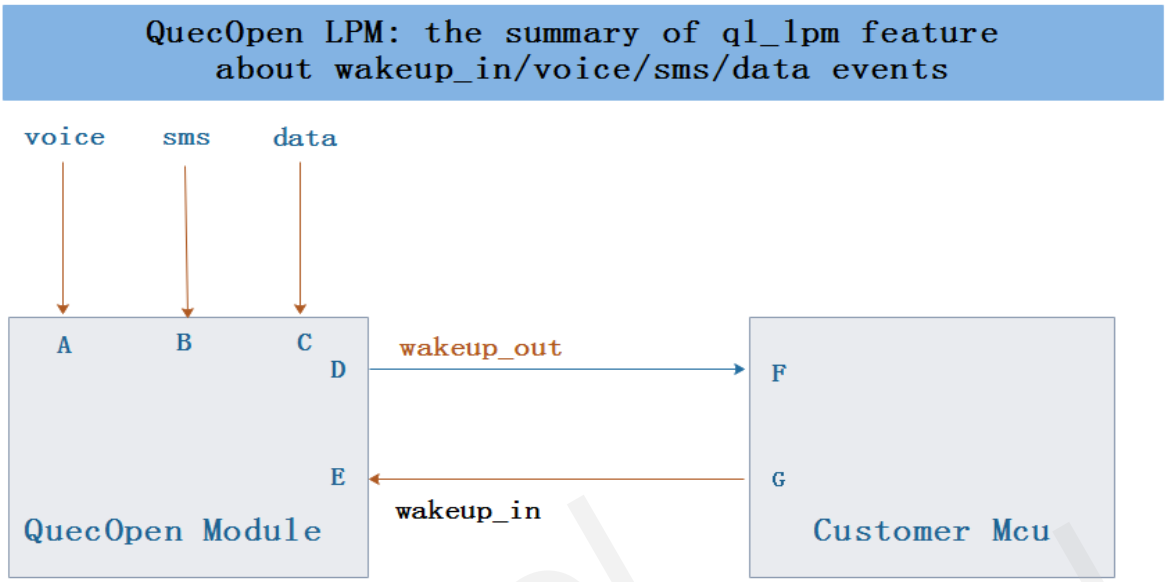
Autosleep 和 wakelock 形成一对矛盾共同体，从而实现了模块的休眠与唤醒功能。

2. QuecOpen LPM 状态介绍

2.1 QuecOpen LPM 状态图



2.2 QuecOpen LPM 方案描述



Initial State:
D:the wakeup_out pin should output high level to keep the "F" pin high level in initial state;
G:the mcu pin should output high level to keep the wakeup_in high level in initial state;

Suspend QuecOpen Module:
E:Unlock the wakelock when receive "falling" edge on wakeup_in pin and the wakeup_out will also output low level automatically when QuecOpen enter sleep mode as a feedback to mcu;

Wakeup QuecOpen Module:
A,B,C,E:Add wakelock when wakeup QuecOpen Module by the four wakeup events;
D:And then control wakeup_out output high level to wakeup/feedback mcu when A,B,C or E event comes in;

如上图：
初始状态：
D: wakeup_out pin 输出高电平，使 mcu 侧“F”pin 初始状态为高电平;
G: mcu 引脚输出高电平，使 wakeup_in 初始状态为高电平;

休眠 QuecOpen 模块：
E:当 wakeup_in 收到下降沿时，app 释放 wakelock,当 QuecOpen 进入睡眠模式，wakeup_out 也会自动输出低电平反馈给 mcu;

唤醒 QuecOpen 模块：
A,B,C,E: 通过四个唤醒事件唤醒 QuecOpen 模块时，app 需要锁定 wakelock;
D:当 A、B、C、E 事件发生时，控制 wakeup_out 输出高电平，唤醒/反馈给 mcu;

3. QuecOpen LPM API

QuecOpen LPM 提供以下应用层接口来进行休眠唤醒控制；

typedef void (*QL_Lpm_Handler_T) (ql_lpm_edge_t lpm_edge);
函数功能：通过 QL_Lpm_Init 注册回调函数，当 wakeup_in 电平发生变化将触发回调；
参数：
lpm_edge: 底层上报的 wakeup_in 电平边沿变化；枚举值：E_QL_LPM_FALLING 或 E_QL_LPM_RISING，参见头文件 ql_lpm.h；

int QL_Lpm_Init(QL_Lpm_Handler_T ql_lpm_handler, QL_Lpm_Cfg_T *ql_lpm_cfg);
函数功能：初始化 Quectel LPM 功能，自动加载 ql_lpm 驱动，同时监听 wakeup_in 状态变化并通知给 ql_lpm_handler；
参数：
ql_lpm_handler: 用户回调，当 wakeup_in 电平发生变化将触发；
ql_lpm_cfg: 用户参数数据结构，推荐此参数传入 NULL 使用移远默认引脚和触发方式即可; 扩展支持用户填充自选引脚以及触发方式；
返回值：0: 成功， -1:失败

int QL_Lpm_Deinit();
函数功能：注销 Quectel LPM 功能，卸载 ql_lpm 内核模块，注销 handler；
返回值：0：成功， -1：失败；

int Ql_Autosleep_Enable(char enable);
函数功能：使能 AutoSleep，使能后系统会在满足条件后自动进入休眠；
参数：
Enable：1：使能 autosleep; 0：取消 autosleep（一般不传参数 0，如果需要保持唤醒，使用下面的 Ql_SLP_WakeLock_Lock 接口进行唤醒后的锁定，即保持唤醒）；
返回值：0：成功；-1：失败；

int Ql_SLP_WakeLock_Create(const char *name, size_t len);
函数功能：创建 wakelock,此接口创建的 wakelock 所有者为当前进程；最多可以创建 512 个 lock；
参数：
name：wakelock 的名字（在/sys/kernel/debug/wakeup_sources 里面可以体现）；
len：wakelock 的长度，名字长度最大允许 28 个字符；
返回值：wakelock 描述符，否则返回-1，通过 errno 可以查询错误码；

int Ql_SLP_WakeLock_Lock(int fd);
函数功能：锁定上面接口创建的 wakelock，锁定后 linux 则无法进入休眠；
参数：fd: wakelock 描述符；
返回值：成功返回 0，否则返回-1；

int Ql_SLP_WakeLock_Unlock(int fd);

函数功能：解锁指定的 wakelock; (如果系统内也没有其他人持有 wakelock,,且 app 使能了 autosleep，那么 linux 就会进入休眠)；
参数：fd: wakelock 描述符;
返回值：成功返回 0，否则返回-1；

int Ql_SLP_WakeLock_Destroy(int fd);
函数功能：销毁 wakelock；
参数：fd: wakelock 描述符；
返回值：成功返回 0，否则返回-1;

注意：Ql_Autosleep_Enable 使能 AutoSleep 后一般不建议使用 Ql_Autosleep_Enable(0)来保持唤醒，而应该使用 Ql_SLP_WakeLock_Lock 使系统保持唤醒，Ql_SLP_WakeLock_Unlock 可以释放 wakelock，放弃休眠锁定。系统休眠后进程会被冻结，唤醒后进程会继续运行。

应用层代码参考示例：
外部引脚中断的低功耗示例：
ql-ol-extsdk/example/low_power_consume_app/example_lpm.c
包含外部中断，电话，短信，数据的低功耗示例：
ql-ol-extsdk/example/low_power_consume_app/example_lpm_all.c

4. QuecOpen LPM 支持的休眠唤醒事件

4.1 休眠事件

4.1.1 引脚下降沿中断事件触发模块休眠

引脚定义：

EC2x		AG35	
wakeup_in	Pin62	wakeup_in	Pin61
wakeup_out	Pin5	wakeup_out	Pin147

Wakeup_in: 模块输入脚，开机初始 mcu 需要控制此管脚为高电平；下降沿触发 app 释放 wakelock 进入休眠；

4.2 唤醒事件

4.2.1 引脚上升沿中断事件

引脚定义：

EC2x		AG35	
wakeup_in	Pin62	wakeup_in	Pin61
wakeup_out	Pin5	wakeup_out	Pin147

Wakeup_in: 模块输入脚，开机初始 mcu 需要控制此管脚为高电平；下降沿模块休眠，上升沿模块唤醒；

Wakeup_out: 模块输出脚，使能 ql_lpm 后默认输出高电平；模块休眠后输出低电平，唤醒后输出高电平；

参考外部引脚中断的低功耗示例：

ql-ol-extsdk/example/low_power_consume_app/example_lpm.c

4.2.2 电话,短信,数据唤醒事件

事先调用 QuecOpen api 注册相应的回调，等待事件触发唤醒模块；

参考示例：

包含外部中断，电话，短信，数据的低功耗示例：

ql-ol-extsdk/example/low_power_consume_app/example_lpm_all.c

5. 模块无法休眠调试

如果客户在 app 中已经释放了 wakelock，但是发现模块仍然没有休眠，那么使用下面指令查看当前系统内持有的唤醒锁；

```
awk '$6 != 0 {print $1"    "$6}' /sys/kernel/debug/wakeup_sources
```

1. 类似下面，如果出现 msm_otg 则说明 usb_vbus 处于高电平状态，需要拉低才能休眠；

```
~ # awk '$6 != 0 {print $1"   "$6}' /sys/kernel/debug/wakeup_sources
name   active_since
msm_otg   77464
~ #
```

2. 如果出现 DATA1,则说明 GPS 数据在抑制休眠，休眠前也需要调用 QL_LOC_Stop_Navigation()接口关闭 gps 功能；

```
~ # awk '$6 != 0 {print $1"   "$6}' /sys/kernel/debug/wakeup_sources
name   active_since
DATA1   448
msm_otg   1105442
~ #
```

3. 如果出现 bam_dmux_wakelock，则说明在 apn 口上有数据交互，需要关闭；

```
~ # awk '$6 != 0 {print $1"   "$6}' /sys/kernel/debug/wakeup_sources
name   active_since
bam_dmux_wakelock   1714
msm_otg   226735
~ #
```

4. 如果客户使用 wifi，在进休眠前需要要调用 ql_wifi_disable()接口关闭 wifi；
5. 如果客户使用以太网，在进休眠前需要要调用 ql_sgmii_disable()接口关闭以太网；

6 模块非预期唤醒原因排查

检查是否有非预期的唤醒，可以使用捷安轮电源的捕捉耗流图，简单点也可以使用 debug uart 来捕捉 log;
步骤如下：
执行以下步骤：
~ # echo 1 > /sys/module/printk/parameters/perf_mode_console
~ # echo 1 > /sys/module/msm_show_resume_irq/parameters/debug_mask
~ # echo 0x2 > /sys/module/ipc_router_core/parameters/debug_mask
然后让模块进入休眠，观察串口 log，如果有类似如下 log 则模块有被唤醒：
[113.386694] gic_show_resume_irq: 57 triggered qcom,smd-modem
[113.386694] gic_show_resume_irq: 200 triggered qcom,smd-rpm
[113.386694] resume cycles: 2542257600
[113.388512] [IPCRTR] CLI RX Len:0xd T:0x1 CF:0x0 SVC:<0x3:0x1> SRC:<0x3:0x11> DST:<0x1:0x43> DATA: 51000b04 13000600
[113.388520] PM: noirq resume of devices complete after 0.975 msecs
[113.389994] PM: early resume of devices complete after 1.088 msecs
唤醒 log 分析：

1. gic_show_resume_irq: 57 triggered qcom,smd-modem：irq 57 表示：modem 通过 smd 通道向 ap 侧发送 QMI 消息；（如果是 58 的话表示有 ip 报文发到 ap 这边）
2. [IPCRTR] CLI RX Len:0xd T:0x1 CF:0x0 SVC:<0x3:0x1> SRC:<0x3:0x11> DST:<0x1:0x43> DATA: 51000b04 13000600：
   CLI RX: QMI client 收到消息，可能是 response,也可能是 indication,需要通过 DATA 字段区分；
   SVC:<0x3,0x1>: 这里 0x3 代表 QMI msg id: NAS;
   DATA: 51000b04 13000600: 这个字段逆序看 13000600 51000b04：0x04 表示 indication, 0x0051 根据 8.1 节表示 QMI_NAS_SIG_INFO_IND，即因为信号强度的改变而上报状态变化的 QMI msg

然后确认是否为非预期的唤醒，并进行处理；

7 耗流指标

7.1 耗流指标

模块 3.8V 供电；

休眠耗流：

休眠耗流一般为 4mA 左右；

AT+CFUN=0 关闭射频后，休眠耗流约为 1.x mA;

Idle 耗流：

耗流约为 25mA；

数据传送或者语音通话：

耗流一般为几百毫安，不同制式下有所不同；

具体参见下表或者参考硬件手册；

参数	描述	条件	典型值	单位
I_{VBAT}	关机模式	模块关机时	12	uA
	睡眠模式	**AT+CFUN=0**（USB 断开）	1.11	mA
		EGSM @DRX=2（USB 断开）	2.21	mA
		EGSM @DRX=5（USB 断开）	1.67	mA
		EGSM @DRX=5（USB 挂起）	1.91	mA
		EGSM @DRX=9（USB 断开）	1.51	mA
		DCS @DRX=2（USB 断开）	2.02	mA
		DCS @DRX=5（USB 断开）	1.45	mA
		DCS @DRX=5（USB 挂起）	1.64	mA
		DCS @DRX=9（USB 断开）	1.32	mA
		TD-SCDMA Band A @PF=64（USB 断开）	2.03	mA
		TD-SCDMA Band A @PF=128（USB 断开）	1.67	mA
		TD-SCDMA Band A @PF=256（USB 断开）	1.56	mA
		TD-SCDMA Band A @PF=512（USB 断开）	1.42	mA
		BC0 @SCI=1（USB 断开）	3.45	mA

	BC0 @SCI=1（USB 挂起）	3.74	mA
	WCDMA @PF=64（USB 断开）	2.02	mA
	WCDMA @PF=64（USB 挂起）	2.17	mA
	WCDMA @PF=128（USB 断开）	1.71	mA
	WCDMA @PF=256（USB 断开）	1.42	mA
	WCDMA @ PF=512（USB 断开）	1.33	mA
	LTE-FDD @PF=32（USB 断开）	3.37	mA
	LTE-FDD @PF=64（USB 断开）	2.27	mA
	LTE-FDD @PF=64（USB 挂起）	2.53	mA
	LTE-FDD @PF=128（USB 断开）	1.86	mA
	LTE-FDD @PF=256（USB 断开）	1.52	mA
	LTE-TDD @PF=32（USB 断开）	3.41	mA
	LTE-TDD @PF=64（USB 断开）	2.27	mA
	LTE-TDD @PF=64（USB 挂起）	2.51	mA
	LTE-TDD @PF=128（USB 断开）	1.71	mA
	LTE-TDD @PF=256（USB 断开）	1.42	mA
空闲模式	EGSM @DRX=5（USB 断开）	17.54	mA
	EGSM @DRX=5（USB 连接）	27.67	mA
	BC0 @SCI=1（USB 断开）	18.92	mA
	BC0 @SCI=1（USB 连接）	29.08	mA
	TD-SCDMA Band A @PF=64（USB 断开）	17.61	mA
	TD-SCDMA Band A @PF=64（USB 连接）	27.60	mA
	WCDMA @PF=64（USB 断开）	17.92	mA
	WCDMA @PF=64（USB 连接）	28.00	mA
	LTE-FDD @PF=64（USB 断开）	17.84	mA

	LTE-FDD @PF=64（USB 连接）	27.94	mA
	LTE-TDD @ PF=64（USB 断开）	18.11	mA
	LTE-TDD @ PF=64（USB 连接）	28.08	mA
GPRS 数据传送（GNSS 关闭）	GSM900 4DL/1UL @32.62dBm	246.8	mA
	GSM900 3DL/2UL @32.45dBm	418.3	mA
	GSM900 2DL/3UL @30.73dBm	513.2	mA
	GSM900 1DL/4UL @29.75dBm	594.3	mA
	DCS1800 4DL/1UL @29.57dBm	170.8	mA
	DCS1800 3DL/2UL @29.45dBm	274.9	mA
	DCS1800 2DL/3UL @29.28dBm	374.8	mA
	DCS1800 1DL/4UL @29.11dBm	475.5	mA
EDGE 数据传送（GNSS 关闭）	GSM900 4DL/1UL @27.24dBm	157.3	mA
	GSM900 3DL/2UL @27.14dBm	258.8	mA
	GSM900 2DL/3UL @27.01dBm	358.3	mA
	GSM900 1DL/4UL @26.91dBm	461.0	mA
	DCS1800 4DL/1UL @25.85dBm	143.4	mA
	DCS1800 3DL/2UL @25.57dBm	235.2	mA
	DCS1800 2DL/3UL @25.55dBm	323.7	mA
	DCS1800 1DL/4UL @25.22dBm	415.7	mA
CDMA/TD-SCDMA 数据传送（GNSS 关闭）	BC0 @23.98dBm	600.7	mA
	TD-SCDMA Band A @23.42dBm	130.6	mA
	TD-SCDMA Band F @23.32dBm	131.9	mA
WCDMA 数据传送（GNSS 关闭）	WCDMA B1 HSDPA @21.06dBm	503.8	mA
	WCDMA B1 HSUPA @20.56dBm	500.6	mA
	WCDMA B8 HSDPA @21.16dBm	469.5	mA

		WCDMA B8 HSUPA @20.83dBm	527.2	mA
	LTE 数据传送（GNSS 关闭）	LTE-FDD B1 @22.04dBm	709.7	mA
		LTE-FDD B3 @22.87dBm	717.1	mA
		LTE-FDD B5 @22.11dBm	609.6	mA
		LTE-FDD B8 @22.40dBm	609.4	mA
		LTE-TDD B38 @22.75dBm	434.4	mA
		LTE-TDD B39 @22.90dBm	336.5	mA
		LTE-TDD B40 @23.04dBm	360.5	mA
		LTE-TDD B41 @22.95dBm	403.8	mA
	GSM 语音通话	GSM900PCL=5 @32.71dBm	244.4	mA
		GSM900PCL=12 @19.53dBm	111.7	mA
		GSM900PCL=19 @5.69dBm	81.2	mA
		DCS1800 PCL=0 @29.64dBm	165.6	mA
		DCS1800 PCL=7 @16.66dBm	126.4	mA
		DCS1800 PCL=15 @0.41dBm	105.0	mA
	CDMA 语音通话	BC0 @24.09dBm	686.3	mA
		BC0 @-60.12dBm	114.3	mA
	WCDMA 语音通话	WCDMA B1 @23.01dBm	607.9	mA
		WCDMA B8 @22.57dBm	542.3	mA

7.2 休眠耗流偏高分析

如果确认模块休眠了，但是平均耗流偏高，有三个方向的原因：非预期的唤醒，底电流，RF 因素（制式，band 等）；

排查步骤：

1. 检查是否有非预期的唤醒，可以使用捷安轮电源的捕捉耗流图，简单点也可以使用 debug uart 来捕捉 log;

   步骤如下：

   执行以下步骤：

   ~ # echo 1 > /sys/module/printk/parameters/perf_mode_console

   ~ # echo 1 > /sys/module/msm_show_resume_irq/parameters/debug_mask

   ~ # echo 0x2 > /sys/module/ipc_router_core/parameters/debug_mask

   然后让模块进入休眠，观察串口 log，如果有类似如下 log 则模块有被唤醒：

   [113.386694] gic_show_resume_irq: 57 triggered qcom,smd-modem

   [113.386694] gic_show_resume_irq: 200 triggered qcom,smd-rpm

   [113.386694] resume cycles: 2542257600

   [113.388512] [IPCRTR] CLI RX Len:0xd T:0x1 CF:0x0 SVC:<0x3:0x1> SRC:<0x3:0x11> DST:<0x1:0x43> DATA: 51000b04 13000600

   [113.388520] PM: noirq resume of devices complete after 0.975 msecs

   [113.389994] PM: early resume of devices complete after 1.088 msecs

   然后需要检查是否为异常的唤醒，并进行处理；

2. 若第 1 步骤中，未发现异常的唤醒，那么需要检查底电流是否偏高；

   AT+CFUN=0 关闭射频干扰，观察底电流是否偏高，1.5mA 以下为正常；若底电流偏高则需要检查休眠时的管脚配置是否与外部电路形成漏电回路；

3. 如果前面两步均正常，但是休眠耗流仍然偏高，则使用捷安轮电源捕捉精确的耗流图，类似下图：

这里看出是 DRX(Discontinuous Reception)周期过小，还是 DRX 峰值过大导致的；