

# ECx00U&EGx00U 系列

## QuecOpen 低功耗

## API 参考手册

**LTE Standard 模块系列**

版本：1.0

日期：2021-06-28

状态：受控文件



上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司  
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233  
电话：+86 21 51086236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：  
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。因未能遵守有关操作或设计规范而造成的损害，上海移远通信技术股份有限公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

## 免责声明

上海移远通信技术股份有限公司尽力确保开发中功能的完整性、准确性、及时性或效用，但不排除上述功能错误或遗漏的可能。除非其他有效协议另有规定，否则上海移远通信技术股份有限公司对开发中功能的使用不做任何暗示或明示的保证。在适用法律允许的最大范围内，上海移远通信技术股份有限公司不对任何因使用开发中功能而遭受的损失或损害承担责任，无论此类损失或损害是否可以预见。

## 保密义务

除非上海移远通信技术股份有限公司特别授权，否则我司所提供文档和信息的接收方须对接收的文档和信息保密，不得将其用于除本项目的实施与开展以外的任何其他目的。未经上海移远通信技术股份有限公司书面同意，不得获取、使用或向第三方泄露我司所提供的文档和信息。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，上海移远通信技术股份有限公司有权追究法律责任。

## 版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2021，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2021.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2020-11-11	Kevin WANG	文档创建
1.0	2021-06-28	Kevin WANG	正式版本

## 目录

文档历史 .....	2
目录 .....	3
表格索引 .....	4
<b>1 引言 .....</b>	<b>5</b>
1.1. 适用模块 .....	5
<b>2 低功耗 API 介绍 .....</b>	<b>6</b>
2.1. 头文件 .....	6
2.2. 函数概览 .....	6
2.3. 低功耗相关 API .....	7
2.3.1. ql_autosleep_enable .....	7
2.3.1.1. ql_errcode_sleep .....	7
2.3.1.2. QL_SLEEP_FLAG_E .....	8
2.3.2. ql_autosleepex_enable .....	8
2.3.3. ql_lpm_wakelock_create .....	9
2.3.4. ql_lpm_wakelock_delete .....	9
2.3.5. ql_lpm_wakelock_lock .....	10
2.3.6. ql_lpm_wakelock_unlock .....	10
2.3.7. ql_sleep_register_cb .....	11
2.3.8. ql_wakeup_register_cb .....	11
<b>3 应用程序示例操作及调试 .....</b>	<b>12</b>
3.1. 应用程序示例介绍 .....	12
3.2. 低功耗功能调试 .....	12
3.2.1. 休眠失问题分析 .....	12
3.2.2. 休眠注意事项 .....	13
<b>4 附录 参考文档和术语缩写 .....</b>	<b>14</b>

## 表格索引

表 1: 适用模块 .....	5
表 2: 函数概览 .....	6
表 3: 参考文档 .....	14
表 4: 术语缩写 .....	14

# 1 引言

移远通信 LTE Standard ECx00U 和 EGx00U 模块支持 QuecOpen®方案；QuecOpen®是基于 RTOS 的嵌入式开发平台，可简化 IoT 应用的软件设计和开发过程。有关 QuecOpen®的详细信息，请参考文档 [1]。

本文档主要介绍在 QuecOpen®方案下，ECx00U 和 EGx00U 模块的低功耗功能相关 API 的使用及调试方法。

## 1.1. 适用模块

表 1：适用模块

模块系列	模块
ECx00U 系列	EC200U 系列
	EC600U 系列
EGx00U 系列	EG500U-CN
	EG700U-CN

## 2 低功耗 API 介绍

### 2.1. 头文件

低功耗相关 API 头文件 `ql_power.h` 在位于 SDK 包中的 `components\ql-kernel\inc` 目录下。

### 2.2. 函数概览

表 2：函数概览

函数	说明
<code>ql_autosleep_enable()</code>	允许/禁止系统进入休眠
<code>ql_autosleepex_enable()</code>	使能增强型休眠
<code>ql_lpm_wakelock_create()</code>	创建唤醒锁
<code>ql_lpm_wakelock_delete()</code>	删除创建的唤醒锁
<code>ql_lpm_wakelock_lock()</code>	锁定唤醒锁
<code>ql_lpm_wakelock_unlock()</code>	解锁唤醒锁
<code>ql_sleep_register_cb()</code>	注册休眠回调函数
<code>ql_wakeup_register_cb()</code>	注册唤醒回调函数

## 2.3. 低功耗相关 API

### 2.3.1. ql\_autosleep\_enable

该函数用于允许/禁止系统进入休眠。

- 函数原型

```
ql_errcode_sleep ql_autosleep_enable(QL_SLEEP_FLAG_E sleep_flag)
```

- 参数

*sleep\_flag*:

[In] 允许/禁止系统进入休眠的标志。详见第 2.3.1.2 章。

- 返回值

错误码详见第 2.3.1.1 章。

#### 2.3.1.1. ql\_errcode\_sleep

```
typedef enum
{
    QL_SLEEP_SUCCESS                = QL_SUCCESS,
    QL_SLEEP_INVALID_PARAM          = (QL_COMPONENT_PM_SLEEP << 16) | 1000,
    /* invalid input param          */
    QL_SLEEP_LOCK_CREATE_FAIL       = (QL_COMPONENT_PM_SLEEP << 16) | 1001,
    /* failed to create wake lock   */
    QL_SLEEP_LOCK_DELETE_FAIL       = (QL_COMPONENT_PM_SLEEP << 16) | 1002,
    /* failed to delete wake lock   */
    QL_SLEEP_LOCK_LOCK_FAIL         = (QL_COMPONENT_PM_SLEEP << 16) | 1003,
    /* failed to lock the wake lock */
    QL_SLEEP_LOCK_UNLOCK_FAIL       = (QL_COMPONENT_PM_SLEEP << 16) | 1004,
    /* failed to unlock the wake lock */
    QL_SLEEP_LOCK_AUTOSLEEP_FAIL    = (QL_COMPONENT_PM_SLEEP << 16) | 1005,
    /* failed to unlock the wake lock */
    QL_SLEEP_PARAM_SAVE_FAIL        = (QL_COMPONENT_PM_SLEEP << 16) | 1006,
    /* failed to save sleep param   */
}ql_errcode_sleep;
```



- 参数

参数	描述
QL_SLEEP_SUCCESS	操作成功
QL_SLEEP_INVALID_PARAM	参数无效
QL_SLEEP_LOCK_CREATE_FAIL	创建休眠锁失败
QL_SLEEP_LOCK_DELETE_FAIL	删除休眠锁失败
QL_SLEEP_LOCK_LOCK_FAIL	锁定休眠锁失败
QL_SLEEP_LOCK_UNLOCK_FAIL	释放休眠锁失败
QL_SLEEP_LOCK_AUTOSLEEP_FAIL	启动自动休眠失败
QL_SLEEP_PARAM_SAVE_FAIL	参数保存失败

### 2.3.1.2. QL\_SLEEP\_FLAG\_E

```
typedef enum
{
    QL_NOT_ALLOW_SLEEP = 0,
    QL_ALLOW_SLEEP,
}QL_SLEEP_FLAG_E;
```

- 参数

参数	描述
QL_NOT_ALLOW_SLEEP	禁止系统进入休眠
QL_ALLOW_SLEEP	允许系统进入休眠

### 2.3.2. ql\_autosleepex\_enable

该函数用于使能增强型休眠。

- 函数原型

```
ql_errcode_sleep ql_autosleepex_enable(QL_SLEEP_FLAG_E sleep_flag, uint8_t no_data_time,
uint16_t punish_time)
```

- 参数

*sleep\_flag:*

[In] 允许/禁止系统进入休眠的标志。详见第2.3.1.2章。

*no\_data\_time:*

[In] 无数据交互后释放 RRC 时间。

*punish\_time:*

[In] 异常注网后不可使用增强休眠时间。

- 返回值

错误码详见第2.3.1.1章。

### 2.3.3. ql\_lpm\_wakelock\_create

该函数用于创建唤醒锁。

- 函数原型

```
int ql_lpm_wakelock_create(char *lock_name, int name_size)
```

- 参数

*lock\_name:*

[In] 唤醒锁的名字。长度不超过 32 个字节。

*name\_size:*

[In] 此参数无效。

- 返回值

小于等于 0 函数执行失败

大于 0 函数执行成功，返回锁的句柄，后续对锁的操作会传入此句柄。

### 2.3.4. ql\_lpm\_wakelock\_delete

该函数用于删除创建的唤醒锁。

- 函数原型

```
ql_errcode_sleep ql_lpm_wakelock_delete(int wakelock_fd)
```

- 参数

*wakelock\_fd*:

[In] 锁的句柄。由 *ql\_lpm\_wakelock\_create()*接口返回。

- 返回值

错误码详见第2.3.1.1章。

### 2.3.5. ql\_lpm\_wakelock\_lock

该函数用于锁住唤醒锁。系统中存在一个或多个唤醒锁时，不允许进入休眠。

- 函数原型

```
ql_errcode_sleep ql_lpm_wakelock_lock(int wakelock_fd)
```

- 参数

*wakelock\_fd*:

[In] 锁的句柄。由 *ql\_lpm\_wakelock\_create()*接口返回。

- 返回值

错误码详见第2.3.1.1章。

### 2.3.6. ql\_lpm\_wakelock\_unlock

该函数用于释放唤醒锁。仅当系统中所有的休眠锁被释放时，才允许系统进入休眠。

- 函数原型

```
ql_errcode_sleep ql_lpm_wakelock_unlock(int wakelock_fd)
```

- 参数

*wakelock\_fd*:

[In] 锁的句柄。由 *ql\_lpm\_wakelock\_create()*接口返回。

- 返回值

错误码详见第2.3.1.1章。

### 2.3.7. ql\_sleep\_register\_cb

该函数用于注册休眠回调函数。

- 函数原型

```
ql_errcode_sleep ql_sleep_register_cb(ql_enter_sleep_callback cb)
```

- 参数

*cb*:

[In] 休眠回调函数指针。

- 返回值

错误码详见第2.3.1.1章。

### 2.3.8. ql\_wakeup\_register\_cb

该函数用于注册唤醒回调函数。

- 函数原型

```
ql_errcode_sleep ql_wakeup_register_cb(ql_exit_sleep_callback cb)
```

- 参数

*cb*:

[In] 唤醒回调函数指针。

- 返回值

错误码详见第2.3.1.1章。

## 3 应用程序示例操作及调试

本章节主要介绍如何使用上述 API 及进行简单的调试。

### 3.1. 应用程序示例介绍

应用程序示例 `power_demo.c` 位于 SDK 包中的 `\components\ql-application\power` 目录下。应用程序示例中，在创建 `power task` 时，创建了两个唤醒锁，以及一个定时器。定时器运行 1 秒后，释放这两个唤醒锁，并允许系统进入休眠。

### 3.2. 低功耗功能调试

#### 3.2.1. 休眠失问题分析

(1) 进入休眠的条件。必须满足以下所有条件，系统才能进入休眠：

- USB 拔出；
- 系统唤醒锁全部释放；
- 无外部引脚中断干扰；
- 使能系统自动进入休眠；
- **AT+CFUN=0** 或者为正常注网状态。

(2) 休眠失败 log 抓取。

若（1）中条件全部满足，系统仍长时间无法进入休眠，log 中将会每 3 秒打印一次无法进入休眠的原因，如下图。届时请保存调试串口的日志并提交至移远通信工程师分析。

时间戳	进程ID	进程名	日志内容
13:59:18.823	15880	QUEC/I	[quec_power_mg][quec_pmsource_auto_sleep, 274] start auto sleep mode
13:59:18.823	15882	QOPN/I	[ql_POWER][ql_lpm_wakelock_unlock, 337] unlock ok
13:59:18.823	15882	QOPN/I	[ql_POWER][ql_lpm_wakelock_unlock, 337] unlock ok
13:59:18.823	15882	QOPN/I	[ql_POWER][ql_power_demo_thread, 83] set auto sleep mode ok
13:59:21.810	65035	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source CP not allow sleep
13:59:21.831	65035	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source USB2 not allow sleep
13:59:24.799	48650	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source CP not allow sleep
13:59:24.819	48651	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source USB2 not allow sleep
13:59:27.798	32267	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source CP not allow sleep
13:59:27.860	32267	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source USB2 not allow sleep
13:59:30.813	15883	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source CP not allow sleep
13:59:30.813	15883	QUEC/I	[quec_power_mg][quec_print_os_pm_state, 791] source USB2 not allow sleep

### 3.2.2. 休眠注意事项

休眠模式下，若没有开启主串口流控，不可直接通过主串口发送大数据（超过 127 字节）给模块，需要退出休眠模式后，才可发送大数据，否则有可能出现丢数据情况。

## 4 附录 参考文档和术语缩写

表 3: 参考文档

文档名称

[1] Quectel\_ECx00U&EGx00U 系列\_QuecOpen\_快速开发指导

表 4: 术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序接口
RRC	Radio Resource Control	无线资源控制
RTOS	Real-Time Operating System	实时操作系统
SDK	Software Development Kit	软件开发工具包
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
USB	Universal Serial Bus	通用串行总线