

# ECx00U&EGx00U 系列

## QuecOpen 开关机开发指导

**LTE Standard 模块系列**

版本：1.1

日期：2022-02-25

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）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 5108 6236 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，请随时登陆网址：  
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，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。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。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，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2022，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2022.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2020-12-02	JoJo YAN	文档创建
1.0	2021-07-12	JoJo YAN	受控版本
1.1	2022-02-25	JoJo YAN	<ol style="list-style-type: none"> <li>1. 更新错误码 QL_POWER_KEY_STATUS_GET_ERR 的描述（第 2.3.1.2 章）。</li> <li>2. 更新 PWRKEY 按键状态参数解释（第 2.3.4.1 章）。</li> <li>3. 新增回调函数 ql_pwrkey_callback（第 2.3.5.1 章）。</li> <li>4. 新增参数 shutdown_time 的最小值（第 2.3.6 章）。</li> <li>5. 更新 USB AP Log 端口图（第 3.2.1 章）。</li> </ol>

# 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	5
<b>1 引言 .....</b>	<b>6</b>
1.1. 适用模块 .....	6
<b>2 开关机 API .....</b>	<b>7</b>
2.1. 头文件 .....	7
2.2. 函数概览 .....	7
2.3. 函数详解 .....	8
2.3.1. ql_power_down .....	8
2.3.1.1. ql_PowdMode .....	8
2.3.1.2. ql_errcode_power .....	9
2.3.2. ql_power_reset .....	10
2.3.2.1. ql_ResetMode .....	11
2.3.3. ql_get_powerup_reason .....	11
2.3.3.1. ql_pwrup_reason .....	11
2.3.4. ql_get_pwrkey_status .....	12
2.3.4.1. QL_PWRKEY_STATUS_E .....	13
2.3.5. ql_pwrkey_callback_register .....	13
2.3.5.1. ql_pwrkey_callback .....	14
2.3.6. ql_pwrkey_shutdown_time_set .....	14
<b>3 开关机开发流程 .....</b>	<b>15</b>
3.1. App 侧的开关机开发示例 .....	15
3.1.1. 开关机 Demo 说明 .....	15
3.1.2. Demo 自启动 .....	15
3.2. 开关机相关调试 .....	16
3.2.1. 调试准备 .....	16
3.2.2. 开关机功能调试 .....	17
3.2.3. 通过 PWRKEY 按键回调函数调试 .....	18
<b>4 附录 参考文档和术语缩写 .....</b>	<b>19</b>

表格索引

表 1: 适用模块 ..... 6

表 2: 函数概览 ..... 7

表 3: 参考文档 ..... 19

表 4: 术语缩写 ..... 19

# 1 引言

移远通信 LTE Standard ECx00U 系列和 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 QuecOpen®方案下 ECx00U 系列和 EGx00U 模块在 App 侧的开关机开发指导，包括相关 API 和开发示例等。

## 1.1. 适用模块

表 1：适用模块

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

## 2 开关机 API

### 2.1. 头文件

在 App 侧，开关机 API 的头文件为 `ql_power.h`，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_power_down()</code>	模块关机
<code>ql_power_reset()</code>	重启模块
<code>ql_get_powerup_reason()</code>	获取开机原因
<code>ql_get_pwrkey_status()</code>	获取模块的 PWRKEY 按键状态
<code>ql_pwrkey_callback_register()</code>	注册 PWRKEY 按键关机的回调函数
<code>ql_pwrkey_shutdown_time_set()</code>	设置长按 PWRKEY 按键关机的按键时长



## 2.3. 函数详解

### 2.3.1. ql\_power\_down

该函数用于模块关机。

- 函数原型

```
ql_errcode_power ql_power_down(ql_PowdMode powd_mode)
```

- 参数

*powd\_mode*:

[In] 关机方式；详见第 2.3.1.1 章。

- 返回值

错误码详见第 2.3.1.2 章。

#### 2.3.1.1.ql\_PowdMode

关机方式，枚举信息定义如下：

```
typedef enum
{
    POWD_IMMDLY,
    POWD_NORMAL
}ql_PowdMode;
```

- 成员

成员	描述
<i>POWD_IMMDLY</i>	立即关机
<i>POWD_NORMAL</i>	正常关机

### 2.3.1.2. ql\_errcode\_power

开关机错误码表示函数是否执行成功，若失败则返回错误原因，枚举信息定义如下：

```
typedef enum
{
    QL_POWER_POWD_SUCCESS = QL_SUCCESS,
    QL_POWER_RESET_SUCCESS = QL_POWER_POWD_SUCCESS,

    QL_POWER_CFW_CTRL_ERR = 1|QL_POWER_ERRCODE_BASE,
    QL_POWER_CFW_CTRL_RSP_ERR,
    QL_POWER_CFW_RESET_BUSY,

    QL_POWER_SEMAPHORE_CREATE_ERR = 5|QL_POWER_ERRCODE_BASE,
    QL_POWER_SEMAPHORE_TIMEOUT_ERR,

    QL_POWER_POWD_EXECUTE_ERR = 11|QL_POWER_ERRCODE_BASE,
    QL_POWER_POWD_INVALID_PARAM_ERR,

    QL_POWER_RESET_EXECUTE_ERR = 21|QL_POWER_ERRCODE_BASE,
    QL_POWER_RESET_INVALID_PARAM_ERR,

    QL_POWER_UP_REASON_GET_ERR = 31|QL_POWER_ERRCODE_BASE,
    QL_POWER_UP_REASON_MEM_NULL_ERR,

    QL_POWER_KEY_CB_NULL_ERR = 41|QL_POWER_ERRCODE_BASE,
    QL_POWER_KEY_STATUS_GET_ERR,
    QL_POWER_KEY_MEM_NULL_ERR,
    QL_POWER_KEY_SHUTDOWN_TIME_SET_ERR,

    QL_POWER_USB_DETECT_INVALID_PARAM = 51|QL_POWER_ERRCODE_BASE,
    QL_POWER_USB_DETECT_SAVE_NV_ERR,

}ql_errcode_power;
```

#### ● 成员

成员	描述
QL_POWER_POWD_SUCCESS	关机执行成功
QL_POWER_RESET_SUCCESS	重启执行成功
QL_POWER_CFW_CTRL_ERR	CFW 执行失败

QL_POWER_CFW_CTRL_RSP_ERR	CFW 响应错误
QL_POWER_CFW_RESET_BUSY	当前不处于开机状态
QL_POWER_SEMAPHORE_CREATE_ERR	信号量创建失败
QL_POWER_SEMAPHORE_TIMEOUT_ERR	信号量等待超时
QL_POWER_POWD_EXECUTE_ERR	关机执行失败
QL_POWER_POWD_INVALID_PARAM_ERR	关机的参数为错误值
QL_POWER_RESET_EXECUTE_ERR	重启执行失败
QL_POWER_RESET_INVALID_PARAM_ERR	重启的参数为错误值
QL_POWER_UP_REASON_GET_ERR	开机原因获取失败
QL_POWER_UP_REASON_MEM_NULL_ERR	开机原因参数为空指针
QL_POWER_KEY_CB_NULL_ERR	PWRKEY 按键的回调函数为空指针
QL_POWER_KEY_STATUS_GET_ERR	获取到错误的 PWRKEY 按键状态
QL_POWER_KEY_MEM_NULL_ERR	PWRKEY 按键状态参数为空指针
QL_POWER_KEY_SHUTDOWN_TIME_SET_ERR	PWRKEY 按键长按关机时间设置错误
QL_POWER_USB_DETECT_INVALID_PARAM	设置 USB 检查时间参数为错误值
QL_POWER_USB_DETECT_SAVE_NV_ERR	保存设置 USB 检查时间值到 NVRAM 失败

### 2.3.2. ql\_power\_reset

该函数用于重启模块。

- 函数原型

```
ql_errcode_power ql_power_reset(ql_ResetMode reset_mode)
```

- 参数

*reset\_mode*:

[In] 重启方式；详见第 2.3.2.1 章。

- 返回值

错误码详见第 2.3.1.2 章。

### 2.3.2.1. ql\_ResetMode

重启方式，枚举信息定义如下：

```
typedef enum
{
    RESET_QUICK,
    RESET_NORMAL
}ql_ResetMode;
```

- 成员

成员	描述
<i>RESET_QUICK</i>	立即重启
<i>RESET_NORMAL</i>	正常重启

### 2.3.3. ql\_get\_powerup\_reason

该函数用于获取开机原因。

- 函数原型

```
ql_errcode_power ql_get_powerup_reason(uint8_t *pwrup_reason)
```

- 参数

\*pwrup\_reason:

[Out] 开机原因；详见第2.3.3.1章。

- 返回值

错误码详见第2.3.1.2章。

#### 2.3.3.1. ql\_pwrup\_reason

开机原因，枚举信息定义如下：

```
typedef enum
{
    QL_PWRUP_UNKNOWN,           //未知开机原因
    QL_PWRUP_PWRKEY,           //通过 PWRKEY 按键开机
    QL_PWRUP_PIN_RESET,        //通过 RESET_N 按键开机
}
```

```

    QL_PWRUP_ALARM,           //通过警告开机
    QL_PWRUP_CHARGE,          //通过 USB 充电开机
    QL_PWRUP_WDG,             //通过看门狗开机
    QL_PWRUP_PSM_WAKEUP,      //通过 PSM 唤醒开机
    QL_PWRUP_PANIC            //通过 Dump 开机
}ql_pwrup_reason;

```

#### ● 成员

成员	描述
<code>QL_PWRUP_UNKNOWN</code>	未知开机原因
<code>QL_PWRUP_PWRKEY</code>	通过 PWRKEY 按键开机
<code>QL_PWRUP_PIN_RESET</code>	通过 RESET_N 按键开机
<code>QL_PWRUP_ALARM</code>	通过警告开机
<code>QL_PWRUP_CHARGE</code>	通过 USB 充电开机
<code>QL_PWRUP_WDG</code>	通过看门狗开机
<code>QL_PWRUP_PSM_WAKEUP</code>	通过 PSM 唤醒开机
<code>QL_PWRUP_PANIC</code>	通过 Dump 开机

### 2.3.4. ql\_get\_pwrkey\_status

该函数用于获取模块的 PWRKEY 按键状态。

#### ● 函数原型

```
ql_errcode_power ql_get_pwrkey_status(uint8_t *pwrkey_status)
```

#### ● 参数

*\*pwrkey\_status:*

[Out] PWRKEY 按键状态；详见第 2.3.4.1 章。

#### ● 返回值

错误码详见第 2.3.1.2 章。

### 2.3.4.1. QL\_PWRKEY\_STATUS\_E

PWRKEY 按键状态，枚举信息定义如下：

```
typedef enum
{
    PWRKEY_RELEASE = 0,
    PWRKEY_PRESSED
}QL_PWRKEY_STATUS_E;
```

- 成员

成员	描述
<i>PWRKEY_RELEASE</i>	PWRKEY 按键松开
<i>PWRKEY_PRESSED</i>	PWRKEY 按键按下

### 2.3.5. ql\_pwrkey\_callback\_register

该函数用于注册 PWRKEY 按键关机的回调函数。注册该回调函数后，当通过 PWRKEY 按键关机时，不再执行原本的关机流程，转而执行 PWRKEY 按键关机的回调函数，执行完客户的程序后需要手动执行 *ql\_power\_down()*函数来关机。

- 函数原型

```
ql_errcode_power ql_pwrkey_callback_register(ql_pwrkey_callback pwrkey_cb)
```

- 参数

*pwrkey\_cb*:

[In] PWRKEY 按键关机回调函数；详见第2.3.5.1章。

- 返回值

错误码详见第2.3.1.2章。

### 2.3.5.1. ql\_pwrkey\_callback

该回调函数用于 PWRKEY 按键关机。

- 函数原型

```
typedef void (*ql_pwrkey_callback)(void)
```

- 参数

无

- 返回值

无

### 2.3.6. ql\_pwrkey\_shutdown\_time\_set

该函数用于设置长按 PWRKEY 按键关机的按键时长。

- 函数原型

```
ql_errcode_power ql_pwrkey_shutdown_time_set(uint32_t shutdown_time)
```

- 参数

*shutdown\_time*:

[In] 长按 PWRKEY 按键关机的所需时长。最小值：420；单位：毫秒。

- 返回值

错误码详见第2.3.1.2章。

## 3 开关机开发流程

本章主要介绍在 App 侧如何使用上述 API 进行开关机开发以及简单的调试。

### 3.1. App 侧的开关机开发示例

#### 3.1.1. 开关机 Demo 说明

移远通信 ECx00U 系列和 EGx00U QuecOpen SDK 代码中提供了关机 and 重启的示例文件 *power\_demo.c*，路径为 *components\ql-application\power*。

*power\_demo.c* 文件中主要设置了两两种关机 and 重启的方式，该 Demo 根据关机 and 重启的参数，选择不同的关机 and 重启方式。入口函数为 *ql\_power\_app\_init()*，如下图所示：

```
void ql_power_app_init(void)
{
    ... QIOSStatus err = QL_OSI_SUCCESS;
    ... ql_task_t power_task = NULL;

    ... err = ql_rtos_task_create(&power_task, 1024, APP_PRIORITY_NORMAL, "ql_powerdemo", ql_power_demo_thread, NULL, 1);
    ... if (err != QL_OSI_SUCCESS)
    ... {
    ...     QL_POWERDEMO_LOG("power_demo_task_created_failed[%d]", err);
    ... }
}
```

*power\_demo.c* 文件中包含了注册 PWRKEY 按键关机回调函数的示例。入口函数为 *ql\_pwrkey\_app\_init()*，如下图所示。该 Demo 在通过 PWRKEY 按键关机时调用用户注册的回调函数，并发送关机开始标识事件 *QUEC\_PWRKEY\_SHUTDOWN\_START\_IND* 到 task。

```
void ql_pwrkey_app_init(void)
{
    ... QIOSStatus err = QL_OSI_SUCCESS;

    ... err = ql_rtos_task_create(&pwrkey_task, 1024, APP_PRIORITY_NORMAL, "ql_pwrkeydemo", ql_pwrkey_demo_thread, NULL, 3);
    ... require_action(err, return, "pwrkey_demo_task_created_failed");
}
```

#### 3.1.2. Demo 自启动

上述 Demo 已在 *ql\_init\_demo\_thread()* 线程中默认不启动，如下图所示：



```
static void ql_init_demo_thread(void *param)
{
    ... QL_INIT_LOG("init_demo_thread_enter, param:0x%x", param);

    /*Caution:If the macro of secure boot and the function are opened, download firmware
    ... the secret key cannot be changed forever*/...
    #ifdef QL_APP_FEATURE_SECURE_BOOT
        ... //ql_dev_enable_secure_boot();
    #endif

    #if 0
        ... ql_gpio_app_init();
        ... ql_gpioint_app_init();
    #endif

    #ifdef QL_APP_FEATURE_LEDCFG
        #if !defined(QL_APP_PROJECT_DEF_EC600U) || \
        ... (defined(QL_APP_PROJECT_DEF_EC600U) && !defined(QL_APP_FEATURE_DSSS) && !defined(
        ... ql_ledcfg_app_init();
    #endif
    #endif

    #ifdef QL_APP_FEATURE_LCD
        ... //ql_lcd_app_init();
    #endif

    ... //ql_sim_app_init();
    ... ql_power_app_init();

    #ifdef QL_APP_FEATURE_PMIC
        ... //ql_pwrkey_app_init();
    #endif
}
```

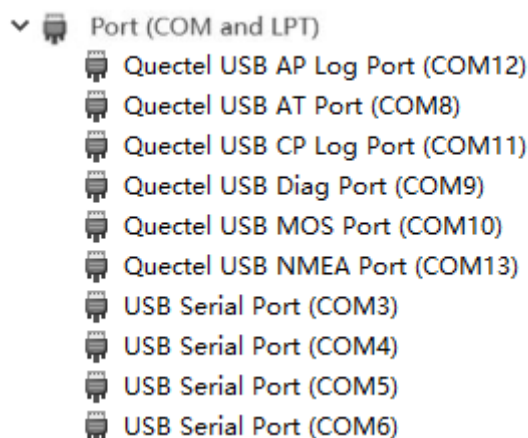
demo默认打开是为了休眠/唤醒部分，  
不会自动跑关机/重启部分

## 3.2. 开关机相关调试

### 3.2.1. 调试准备

移远通信 ECx00U 系列和 EGx00U QuecOpen 模块可通过使用移远通信 LTE OPEN EVB 板进行开关机功能调试。

编译版本烧录到模块中，使用 USB 线连接 LTE OPEN EVB 板的 USB 端口和 PC，USB 的 AP Log 端口主要用于显示系统调试信息，通过 *cooltools* 可以查看该 Demo 的相关信息，log 抓取方法请参考文档 [2]。



### 3.2.2. 开关机功能调试

开机后自动启动 `ql_power_app_init()`，通过 log 信息可以看到立即关机/重启与正常关机/重启前的执行程序不同。与此同时，可以观察到实际执行了关机或重启，**Powerdown EXE** 表示关机，**Reset EXE** 表示重启。

调试串口调试信息如下图所示：

```
AT get event: 505(EV_CFW_NW_DEREGISTER_RSP)/0x00000000/0x00000000/0x00ff0000
AT: unregistered event
AT get event: 5048(EV_CFW_EXIT_IND)/0x00000001/0x00000000/0x00000000
AT RESPONSE EVENT HANDLED BY ID!
NVM id/4 ops/4 paddr/80002D00 len/2128
ipc: ch4 write len/1 ret/16
IPC ISR status/0x00000008
NVM id/1 ops/4 paddr/80007100 len/2156
ipc: ch4 write len/1 ret/16
RTC unset alarm
RTC write day/7595 hour/8 min/30 sec/59
RTC store nv length/6
[ql_GPIODEMO][ql_gpio_demo_thread, 95] gpio[8] output low-level
[ql_GPIODEMO][ql_gpio_demo_thread, 96] gpio[8] set dir:[1], lvl:[0]
[ql_GPIODEMO][ql_gpio_demo_thread, 102] gpio[8] output low-level
[ql_GPIODEMO][ql_gpio_demo_thread, 103] gpio[8] get dir:[1], lvl:[0]
[ql_GPIODEMO][ql_gpio_demo_thread, 95] gpio[22] output low-level
[ql_GPIODEMO][ql_gpio_demo_thread, 96] gpio[22] set dir:[1], lvl:[0]
[ql_GPIODEMO][ql_gpio_demo_thread, 102] gpio[22] output low-level
[ql_GPIODEMO][ql_gpio_demo_thread, 103] gpio[22] get dir:[1], lvl:[0]
[ql_POWER][ql_power_down, 130] Powerdown EXE
```

```
monitor 0 battery 1/3609
[ql_AUDIO][ql_audio_demo_thread, 39] hello audio demo 4
[osi_event][dog_timer_cb, 758] [kevin]timer run
[ql_AUDIO][ql_audio_demo_thread, 39] hello audio demo 5
[osi_event][dog_timer_cb, 758] [kevin]timer run
tcpip: \n
[ql_AUDIO][ql_audio_demo_thread, 39] hello audio demo 6
[osi_event][dog_timer_cb, 758] [kevin]timer run
[ql_POWER][ql_power_down, 83] Powerdown EXE
```

#### 备注

因为运行该 Demo 会造成模块关机或重启，故其他 Demo 无法同时进行测试。

### 3.2.3. 通过 PWRKEY 按键回调函数调试

开机后自动启动 `ql_pwrkey_app_init()`，通过 log 信息可以看到通过 PWRKEY 按键关机触发了回调函数，执行客户的程序后再通过 `ql_power_down()` 手动关机。

```
switch(event.id)
{
...
case QUEC_PWRKEY_SHUTDOWN_START_TND:
...
QL_POWERDEMO_LOG("customer process");
...
/*do something*/
ql_power_down(POWD_NORMAL);
break;
...
default:
...
break;
...
}
```

## 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_CSDK_快速开发指导
[2] Quectel_ECx00U&EGx00U 系列_QuecOpen_Log 抓取指导

表 4: 术语缩写

缩写	英文全称	中文全称
AP	Application Processor	应用处理器
API	Application Programming Interface	应用程序接口
App	Application	应用
CFW	Communication Framework	通信框架
IoT	Internet of Things	物联网
NVRAM	Non-Volatile Random Access Memory	非易失性随机访问存储器
PC	Personal Computer	个人计算机
PSM	Power Saving Mode	省电模式
RTOS	Real-Time Operating System	实时操作系统
SDK	Software Development Kit	软件开发工具包
USB	Universal Serial Bus	通用串行总线