

ECx00U&EGx00U 系列

QuecOpen 网卡拨号

API 参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-10-15

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

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。

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软件、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。

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 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 | 变更描述 |
|-----|------------|---------------------------|------|
| - | 2021-07-08 | Baron ZHENG | 文档创建 |
| 1.0 | 2021-10-15 | Baron ZHENG/ Braden HE | 受控版本 |

目录

| | |
|------------------------------------|-----------|
| 文档历史 | 3 |
| 目录 | 4 |
| 表格索引 | 5 |
| 图片索引 | 6 |
| 1 引言 | 7 |
| 1.1. 适用模块 | 7 |
| 2 网卡拨号 API 调用流程 | 8 |
| 3 网卡拨号 API | 10 |
| 3.1. 头文件 | 10 |
| 3.2. 函数概览 | 10 |
| 3.3. 函数详解 | 11 |
| 3.3.1. ql_usbnet_set_type | 11 |
| 3.3.1.1. ql_usbnet_type_e | 11 |
| 3.3.1.2. ql_usbnet_errcode_e | 12 |
| 3.3.2. ql_usbnet_get_type | 13 |
| 3.3.3. ql_usbnet_start | 13 |
| 3.3.3.1. ql_data_call_conf_s | 14 |
| 3.3.4. ql_usbnet_stop | 14 |
| 3.3.5. ql_usbnet_get_status | 15 |
| 3.3.5.1. ql_usbnet_state_e | 15 |
| 3.3.6. ql_usbnet_register_cb | 16 |
| 3.3.6.1. ql_usbnet_callback | 16 |
| 4 示例 | 18 |
| 4.1. 网卡拨号示例 Demo | 18 |
| 4.2. 示例 Demo 自启动 | 18 |
| 5 附录 参考文档及术语缩写 | 19 |

表格索引

| | |
|-----------------|----|
| 表 1: 参考文档 | 19 |
| 表 2: 术语缩写 | 19 |

图片索引

| | |
|----------------------|----|
| 图 1：网卡拨号流程..... | 8 |
| 图 2：示例 Demo 自启动..... | 18 |

1 引言

移远通信 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]。

本文档主要介绍在 QuecOpen[®]方案下，ECx00U 系列和 EGx00U 模块的网卡拨号功能，包括 API 调用流程、网卡拨号 API 及相关示例。

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 网卡拨号 API 调用流程

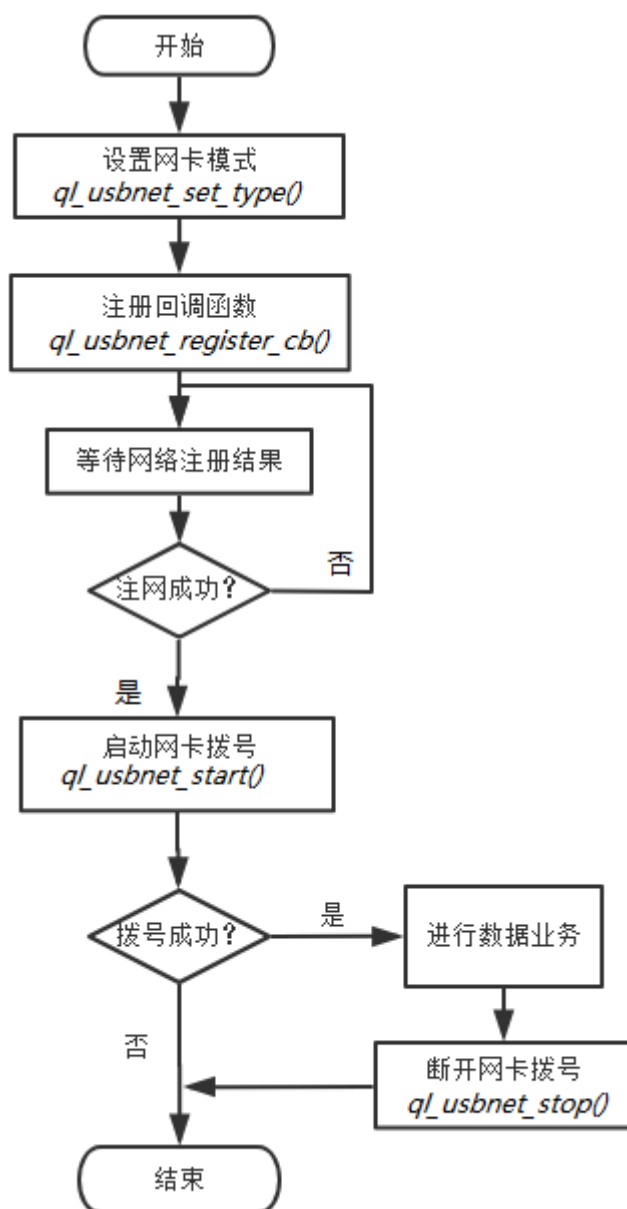


图 1：网卡拨号流程

备注

网络注册是自动完成的，无需代码干预，只需调用 `ql_network_register_wait()` 等待网络注册完成。用户亦可通过调用 `ql_nw_register_cb()` 注册网络事件的回调函数，判断网络注册是否成功。
`ql_network_register_wait()` 和 `ql_nw_register_cb()` 函数详情，请参考文档 [2]。

3 网卡拨号 API

3.1. 头文件

网卡拨号 API 的头文件为 `ql_api_usbnet.h`，位于 `components/ql-kernel/inc` 目录下。若无特别说明，本文档所述头文件均位于该目录下。

3.2. 函数概览

表 2：函数概览

| 函数 | 说明 |
|--------------------------------------|--------------|
| <code>ql_usbnet_set_type()</code> | 设置网卡模式 |
| <code>ql_usbnet_get_type()</code> | 获取当前网卡模式 |
| <code>ql_usbnet_start()</code> | 启动网卡拨号 |
| <code>ql_usbnet_stop()</code> | 终止网卡拨号 |
| <code>ql_usbnet_get_status()</code> | 获取网卡拨号连接状态信息 |
| <code>ql_usbnet_register_cb()</code> | 注册网卡拨号的回调函数 |

3.3. 函数详解

3.3.1. ql_usbnet_set_type

该函数用于设置网卡模式，目前仅支持 RNDIS 和 ECM 模式。调用该函数后，需重启模块后生效。

- 函数原型

```
ql_usbnet_errcode_e ql_usbnet_set_type(ql_usbnet_type_e usbnet_type)
```

- 参数

usbnet_type:

[In] 需设置的网卡模式。详见第 3.3.1.1 章。

- 返回值

详见第 3.3.1.2 章。

3.3.1.1. ql_usbnet_type_e

网卡模式枚举信息定义如下：

```
typedef enum
{
    QL_USBNET_NONE = 0,
    QL_USBNET_ECM,
    QL_USBNET_MBIM,
    QL_USBNET_RNDIS,
    QL_USBNET_MAX
}ql_usbnet_type_e;
```

- 参数

| 参数 | 描述 |
|-----------------|--------------|
| QL_USBNET_NONE | 无实际意义，客户无需使用 |
| QL_USBNET_ECM | ECM 模式 |
| QL_USBNET_MBIM | MBIM 模式（不支持） |
| QL_USBNET_RNDIS | RNDIS 模式 |
| QL_USBNET_MAX | 无实际意义，客户无需使用 |

3.3.1.2. ql_usbnet_errcode_e

网卡拨号函数的结果码，表示函数执行是否成功。枚举信息定义如下：

```
typedef enum
{
    QL_USBNET_SUCCESS                = 0,
    QL_USBNET_EXECUTE_ERR            = 1 | QL_USBNET_ERRCODE_BASE,
    QL_USBNET_MEM_ADDR_NULL_ERR,
    QL_USBNET_INVALID_PARAM_ERR,
    QL_USBNET_USB_NOT_CONNECT_ERR,
    QL_USBNET_PDP_ACTIVE_ERR         = 5 | QL_USBNET_ERRCODE_BASE,
    QL_USBNET_REPEAT_CONNECT_ERR,
    QL_USBNET_REPEAT_DISCONNECT_ERR,
}ql_usbnet_errcode_e;
```

● 参数

| 参数 | 描述 |
|---------------------------------|------------|
| QL_USBNET_SUCCESS | 函数执行成功 |
| QL_USBNET_EXECUTE_ERR | 函数执行失败 |
| QL_USBNET_MEM_ADDR_NULL_ERR | 参数地址为 NULL |
| QL_USBNET_INVALID_PARAM_ERR | 参数无效 |
| QL_USBNET_USB_NOT_CONNECT_ERR | USB 未连接 |
| QL_USBNET_PDP_ACTIVE_ERR | PDP 激活失败 |
| QL_USBNET_REPEAT_CONNECT_ERR | 重复连接 |
| QL_USBNET_REPEAT_DISCONNECT_ERR | 重复断开连接 |

3.3.2. ql_usbnet_get_type

该函数用于获取当前网卡模式。

- 函数原型

```
ql_usbnet_errcode_e ql_usbnet_get_type(ql_usbnet_type_e *usbnet_type)
```

- 参数

usbnet_type:

[Out] 网卡模式。详见第 3.3.1.1 章。

- 返回值

详见第 3.3.1.2 章。

3.3.3. ql_usbnet_start

该函数用于启动网卡拨号。函数执行结束后，返回 *QL_USBNET_SUCCESS* 并不表示拨号成功，仅仅表示函数执行成功。注册的回调函数会通过 *QUEC_USBNET_START_RSP_IND* 事件通知上层是否拨号成功。详见第 3.3.6.1 章。

- 函数原型

```
ql_usbnet_errcode_e ql_usbnet_start(uint8_t nSim, int profile_idx, ql_data_call_conf_s *config)
```

- 参数

nSim:

[In] 所使用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。

0 (U)SIM 卡 1

1 (U)SIM 卡 2

profile_idx:

[In] PDP 上下文 ID。范围：1~7。

config:

[In] 数据拨号配置信息。详见第 3.3.3.1 章。若此参数为 NULL，则使用默认参数。

- 返回值

详见第 3.3.1.2 章。

3.3.3.1. ql_data_call_conf_s

数据拨号配置信息，结构体定义如下：

```
typedef struct
{
    int ip_version;
    char apn_name[APN_LEN_MAX];
    char username[USERNAME_LEN_MAX];
    char password[PASSWORD_LEN_MAX];
    int auth_type;
}ql_data_call_conf_s;
```

● 参数

| 类型 | 参数 | 描述 |
|------|-------------------|------------------------------------------------|
| int | <i>ip_version</i> | IP 类型 1 IPv4 2 IPv6 3 IPv4v6 |
| char | <i>apn_name</i> | APN 接入点名称。最大长度为 64 字节。 |
| char | <i>username</i> | 用户名。最大长度为 64 字节。 |
| char | <i>password</i> | 密码。最大长度为 64 字节。 |
| int | <i>auth_type</i> | 认证方式类型 0 无认证协议 1 PAP 认证协议 2 CHAP 认证协议 |

3.3.4. ql_usbnet_stop

该函数用于终止网卡拨号。终止拨号后，网卡状态为初始状态。

● 函数原型

```
ql_usbnet_errcode_e ql_usbnet_stop(void)
```

● 参数

无

- 返回值

详见第 3.3.1.2 章。

3.3.5. ql_usbnet_get_status

该函数用于获取网卡拨号连接状态信息。

- 函数原型

```
ql_usbnet_errcode_e ql_usbnet_get_status(ql_usbnet_state_e *status)
```

- 参数

status:

[Out] 网卡拨号连接状态信息。详见第 3.3.5.1 章。

- 返回值

详见第 3.3.1.2 章。

3.3.5.1. ql_usbnet_state_e

网卡拨号连接状态枚举信息定义如下：

```
typedef enum
{
    QL_USBNET_STATE_NONE = 0,
    QL_USBNET_STATE_START,
    QL_USBNET_STATE_CONNECT,
    QL_USBNET_STATE_PORT_DISCONNECT,
    QL_USBNET_STATE_MAX,
}ql_usbnet_state_e;
```

- 参数

| 参数 | 描述 |
|----------------------------------------|--------------|
| <i>QL_USBNET_STATE_NONE</i> | 网卡初始状态 |
| <i>QL_USBNET_STATE_START</i> | 网卡正在连接 |
| <i>QL_USBNET_STATE_CONNECT</i> | 网卡连接成功 |
| <i>QL_USBNET_STATE_PORT_DISCONNECT</i> | 网卡 USB 端口未连接 |

QL_USBNET_STATE_MAX

无实际意义，客户无需使用

3.3.6. ql_usbnet_register_cb

该函数用于注册网卡拨号事件的回调函数，通过注册的回调函数与上层进行事件通知。

● 函数原型

```
ql_usbnet_errcode_e ql_usbnet_register_cb(ql_usbnet_callback usbnet_cb, void *ctx)
```

● 参数

usbnet_cb:

[In] 网卡拨号事件回调函数。函数定义详见第 3.3.6.1 章。

ctx:

[In] 回调函数的传参指针。

● 返回值

详见第 3.3.1.2 章。

3.3.6.1. ql_usbnet_callback

该函数为网卡拨号事件回调函数。

● 函数原型

```
typedef void (*ql_usbnet_callback)(unsigned int ind_type, ql_usbnet_errcode_e errcode, void *ctx)
```

● 参数

ind_type:

[In] 网卡拨号事件类型。

QUEC_USBNET_START_RSP_IND

启动网卡拨号响应事件

QUEC_USBNET_DEACTIVE_IND

网卡断开连接事件，网络信号丢失或网络断开

QUEC_USBNET_PORT_CONNECT_IND

USB 端口连接事件

QUEC_USBNET_PORT_DISCONNECT_IND

USB 端口断开连接事件

errcode:

[In] 函数执行结果码。详见第 3.3.1.2 章。

ctx:

[In] 回调函数的传参指针。

4 示例

4.1. 网卡拨号示例 Demo

移远通信 ECx00U 系列和 EGx00U QuecOpen 模块的 SDK 中提供了 App 层网卡拨号示例 Demo 文件 `usbnet_demo.c`，位于 `components\ql-application\usbnet` 目录下。

4.2. 示例 Demo 自启动

如需启动示例 Demo 文件，可在 `ql_init_demo_thread()` 中取消 `ql_usbnet_app_init()` 的注释，将编译版本烧录至模块后，示例 Demo 便可实现开机自启动。

```

468:
469: #ifdef QL_APP_FEATURE_USBNET
470: » //ql_usbnet_app_init();
471: #endif
472:
473: #ifdef QL_APP_FEATURE_SFTP
474: » //ql_sftp_app_init();
475: #endif
476:
477:     ql_rtos_task_sleep_ms(1000); /*Chaos change: set to 1000 for the camera power on*/
478:     ql_rtos_task_delete(NULL);
479: } « end ql_init_demo_thread »
480:

```

图 2: 示例 Demo 自启动

5 附录 参考文档及术语缩写

表 1：参考文档

| 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_注网信息 API_参考手册 |

表 2：术语缩写

| 缩写 | 英文全称 | 中文全称 |
|-------|-----------------------------------------------|------------|
| API | Application Programming Interface | 应用程序接口 |
| APN | Access Point Name | 接入点名称 |
| App | Application | 应用 |
| CHAP | Challenge Handshake Authentication Protocol | 挑战握手认证协议 |
| ECM | Ethernet Networking Control Model | 以太网控制模型 |
| IPv4 | Internet Protocol Version 4 | 互联网通信协议第四版 |
| IPv6 | Internet Protocol Version 6 | 互联网通信协议第六版 |
| RNDIS | Remote Network Driver Interface Specification | 远程网络驱动接口规范 |
| PAP | Password Authentication Protocol | 密码认证协议 |
| PDP | Packet Data Protocol | 分组数据协议 |