

ECx00U&EGx00U 系列

QuecOpen PING

API 参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-08-25

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

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。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	变更表述
-	2020-11-12	Chris PENG	文档创建
1.0	2021-08-25	Herry GENG	受控版本

目录

文档历史	3
目录	4
表格索引	5
1 引言	6
1.1. 适用模块	6
2 调用流程	7
3 PING API	8
3.1. 头文件	8
3.2. API 详解	8
3.2.1. ql_ping_start	8
3.2.1.1. ql_ping_config_type	9
3.2.1.2. ql_ping_event_cb	9
4 示例	13
5 附录 参考文档及术语缩写	14

表格索引

表 1: 适用模块	6
表 2: 参考文档	14
表 3: 术语缩写	14

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 PING 相关 API 函数、调用流程以及示例。

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 调用流程

在调用 PING 接口之前，首先需要注册网络并进行数据拨号以建立数据通道。若在其他任务中已经建立了数据通道，可跳过“等待注册网络”及“进行数据拨号”的步骤，调用流程如下图所示。

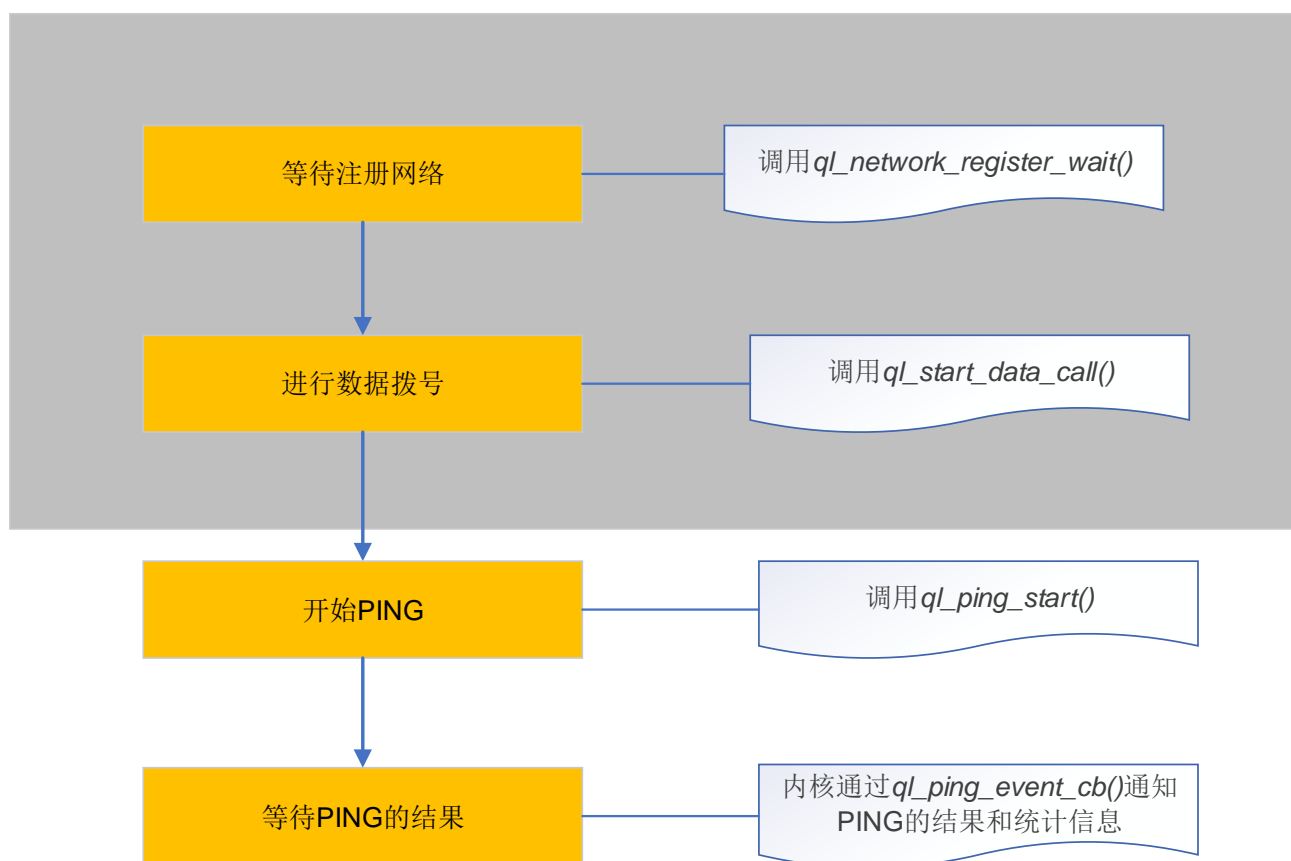


图 1：PING 接口调用流程

备注

有关等待注册网络和数据拨号相关的 API 调用信息，请参考文档 [2]。

3 PING API

3.1. 头文件

PING API 的头文件为 `ql_ping_app.h`，位于 SDK 包的 `\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. API 详解

3.2.1. ql_ping_start

该函数用于启用 PING 功能并根据配置信息设置发送的数据包。

● 函数原型

```
ping_session_id ql_ping_start(int cid, int sim_id, const char *host, ql_ping_config_type
*ping_options, ql_ping_event_cb cb_fcn)
```

● 参数

cid:

[In] 数据通道号。

sim_id:

[In] (U)SIM 卡编号。如模块不支持双卡功能，一般设置为 0。

0 (U)SIM 卡 1

1 (U)SIM 卡 2

host:

[In] PING 的远端服务器地址。

ping_options:

[In] PING 的配置信息，详见第 3.2.1.1 章。

cb_fcn:

[In] PING 事件的回调函数，详见第 3.2.1.2 章。

- 返回值

PING 会话 ID。

整型；取值范围为 1~10。返回值为 0 时，表示创建会话失败。

3.2.1.1. ql_ping_config_type

PING 的配置信息定义如下：

```
typedef struct
{
    uint32_t  num_data_bytes;           /* Data byte size of the ping packet */
    uint32_t  ping_response_time_out; /* Wait time for ping response, ms */
    uint32_t  num_pings;                /* Number of times to ping */
    uint32_t  ttl;                      /* Time To Live for the ping packet */
} ql_ping_config_type;
```

- 参数

类型	参数	描述
uint32_t	<i>num_data_bytes</i>	PING 数据包的字节大小
uint32_t	<i>ping_response_time_out</i>	等待 PING 响应的超时时间。单位：毫秒
uint32_t	<i>num_pings</i>	PING 的次数
uint32_t	<i>ttl</i>	PING 数据包的 TTL

3.2.1.2. ql_ping_event_cb

该函数为 PING 事件的回调函数。

- 函数原型

```
typedef void(*ql_ping_event_cb)(ping_session_id ping_sess_id, uint16_t event_id, int evt_code, void *evt_param)
```

- 参数

ping_sess_id:

[In] PING 会话 ID，详见第 3.2.1.2.1 章。

event_id:

[In] PING 的事件类型，详见第 3.2.1.2.2 章。

evt_code:

[In] PING 的结果码，详见第 3.2.1.2.3 章。

evt_param:

[In] PING 的事件参数。

当 *event_id* 为 *QL_PING_STATS* 时，该参数的类型结构定义详见第 3.2.1.2.4 章。

当 *event_id* 为 *QL_PING_SUMMARY* 时，该参数的类型结构定义详见第 3.2.1.2.5 章。

● 返回值

无

3.2.1.2.1. ping_session_id

PING 会话 ID 定义如下：

```
typedef int ping_session_id
```

3.2.1.2.2. ql_ping_event_type

PING 的事件类型定义如下：

```
typedef enum{
    QL_PING_STATS    = 1,
    QL_PING_SUMMARY  = 2,
}ql_ping_event_type;
```

● 参数

参数	描述
<i>QL_PING_STATS</i>	单次的 PING 结果
<i>QL_PING_SUMMARY</i>	PING 的统计信息

3.2.1.2.3. ql_ping_result_code

PING 的结果码定义如下：

```
typedef enum{
    QL_PING_OK                        = 0,
    QL_PING_ERR_INVALID_PARAM = (QL_COMPONENT_LWIP_SOCKET << 16) | 552 /*
Invalid parameters */,
    QL_PING_ERR_SOCKET_NEW_FAILURE = (QL_COMPONENT_LWIP_SOCKET << 16) |
554 /* Create socket failed */,
    QL_PING_ERR_SOCKET_BIND_FAILURE = (QL_COMPONENT_LWIP_SOCKET << 16) |
556 /* Socket bind failed */,
    QL_PING_ERR_DNS_FAILURE          = (QL_COMPONENT_LWIP_SOCKET << 16) |
565 /* DNS parse failed */,
    QL_PING_ERR_TIMEOUT              = (QL_COMPONENT_LWIP_SOCKET << 16) | 569
/* Operation timeout */,
}ql_ping_result_code;
```

● 参数

参数	描述
<i>QL_PING_OK</i>	PING 成功
<i>QL_PING_ERR_INVALID_PARAM</i>	无效参数
<i>QL_PING_ERR_SOCKET_NEW_FAILURE</i>	Socket 创建失败
<i>QL_PING_ERR_SOCKET_BIND_FAILURE</i>	Socket 绑定失败
<i>QL_PING_ERR_DNS_FAILURE</i>	域名解析失败
<i>QL_PING_ERR_TIMEOUT</i>	PING 超时

3.2.1.2.4. ql_ping_stats_type

单次的 PING 结果定义如下：

```
typedef struct{
    uint32_t    ping_rtt;
    uint32_t    ping_size;
    uint32_t    ping_ttl;
    char        resolved_ip_addr[256];
}ql_ping_stats_type;
```

● 参数

类型	参数	描述
uint32_t	<i>ping_rtt</i>	RTT 值
uint32_t	<i>ping_size</i>	PING 数据包的大小
uint32_t	<i>ping_ttl</i>	PING 数据包的 TTL 值
char	<i>resolved_ip_addr</i>	PING 目标地址

3.2.1.2.5. ql_ping_summary_type

PING 的统计信息定义如下：

```
typedef struct{
    uint32_t  min_rtt;          /* Minimum RTT so far, in millisecs */
    uint32_t  max_rtt;          /* Maximum RTT so far, in millisecs */
    uint32_t  avg_rtt;          /* Average RTT so far, in millisecs */
    uint32_t  num_pkts_sent;    /* Number of pings sent so far */
    uint32_t  num_pkts_recvd;   /* Number of responses recieved so far */
    uint32_t  num_pkts_lost;    /* Number of responses not received */
}ql_ping_summary_type;
```

● 参数

类型	参数	描述
uint32_t	<i>min_rtt</i>	RTT 的最小值。单位：毫秒
uint32_t	<i>max_rtt</i>	RTT 的最大值。单位：毫秒
uint32_t	<i>avg_rtt</i>	RTT 的平均值。单位：毫秒
uint32_t	<i>num_pkts_sent</i>	已发送数据包数量
uint32_t	<i>num_pkts_recvd</i>	已接收回应包数量
uint32_t	<i>num_pkts_lost</i>	丢失数据包数量

4 示例

有关 PING 功能相关示例,请参考位于 SDK 包的 `\components\ql-application\ping` 目录下的应用层 PING 客户端示例文件 `ping_demo.c`。

5 附录 参考文档及术语缩写

表 2：参考文档

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_数据拨号 API_参考手册

表 3：术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序编程接口
DNS	Domain Name Server	域名系统（服务）协议
IoT	Internet of Things	物联网
IP	Internet Protocol	网际互连协议
PING	Packet Internet Groper	因特网包探索器
RTOS	Real-Time Operating System	实时操作系统
RTT	Round-Trip Time	往返时间
SDK	Software Development Kit	软件开发工具包
TTL	Time to Live	生存时间