

# ECx00U&EGx00U 系列

## QuecOpen FTP(S)

### API 参考手册

**LTE Standard 模块系列**

版本：1.1

日期：2022-04-29

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。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软硬件、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
-	2021-02-01	Larson LI	文档创建
1.0	2021-09-06	Larson LI/ Kruskal ZHU	受控版本
1.1	2022-04-29	Larson LI/ Kruskal ZHU	<ol style="list-style-type: none"><li>1. 新增错误码枚举成员 QL_FTP_CLIENT_ERR_INVALID_PARAMATER 和 QL_FTP_CLIENT_ERR_REQUESTED_FILENAME_INVALID（第 3.3.1.1 章）。</li><li>2. 新增函数 ql_ftp_client_rename()（第 3.3.18 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 FTP(S) API 调用流程 .....</b>	<b>7</b>
<b>3 FTP(S) API 介绍 .....</b>	<b>9</b>
3.1. 头文件 .....	9
3.2. 函数概览 .....	9
3.3. API 详解 .....	10
3.3.1. 错误码枚举 .....	10
3.3.1.1. QL_FTP_CLIENT_ERR_E .....	10
3.3.2. ql_ftp_client_new .....	12
3.3.3. ql_ftp_client_release .....	12
3.3.4. ql_ftp_client_setopt .....	12
3.3.4.1. QL_FTP_CLIENT_OPT_E .....	13
3.3.5. ql_ftp_client_open .....	14
3.3.6. ql_ftp_client_close .....	15
3.3.7. ql_ftp_client_get_ex .....	15
3.3.7.1. QL_FTP_CLIENT_WRITE_CB_EX .....	16
3.3.8. ql_ftp_client_put_ex .....	17
3.3.8.1. QL_FTP_CLIENT_READ_CB_EX .....	18
3.3.9. ql_ftp_client_delete .....	18
3.3.10. ql_ftp_client_cwd .....	19
3.3.11. ql_ftp_client_lcwd .....	19
3.3.12. ql_ftp_client_pwd .....	20
3.3.13. ql_ftp_client_lpwd .....	20
3.3.14. ql_ftp_client_mkdir .....	21
3.3.15. ql_ftp_client_rmdir .....	21
3.3.16. ql_ftp_client_list .....	22
3.3.16.1. QL_FTP_CLIENT_FILE_INFO_T .....	22
3.3.17. ql_ftp_client_size .....	23
3.3.18. ql_ftp_client_rename .....	24
<b>4 示例 .....</b>	<b>25</b>
<b>5 附录 参考文档及术语缩写 .....</b>	<b>26</b>

## 表格索引

表 1: 适用模块 .....	6
表 2: 函数概览 .....	9
表 3: 参考文档 .....	26
表 4: 术语缩写 .....	26

# 1 引言

移远通信 LTE Standard ECx00U 系列和 EGx00U 模块支持 QuecOpen<sup>®</sup>方案；QuecOpen<sup>®</sup>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<sup>®</sup>的详细信息，请参考文档 [1]。

本文档主要介绍 QuecOpen<sup>®</sup>方案下 ECx00U 系列和 EGx00U 模块 FTP(S)相关 API 函数、调用流程和示例。

## 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 FTP(S) API 调用流程

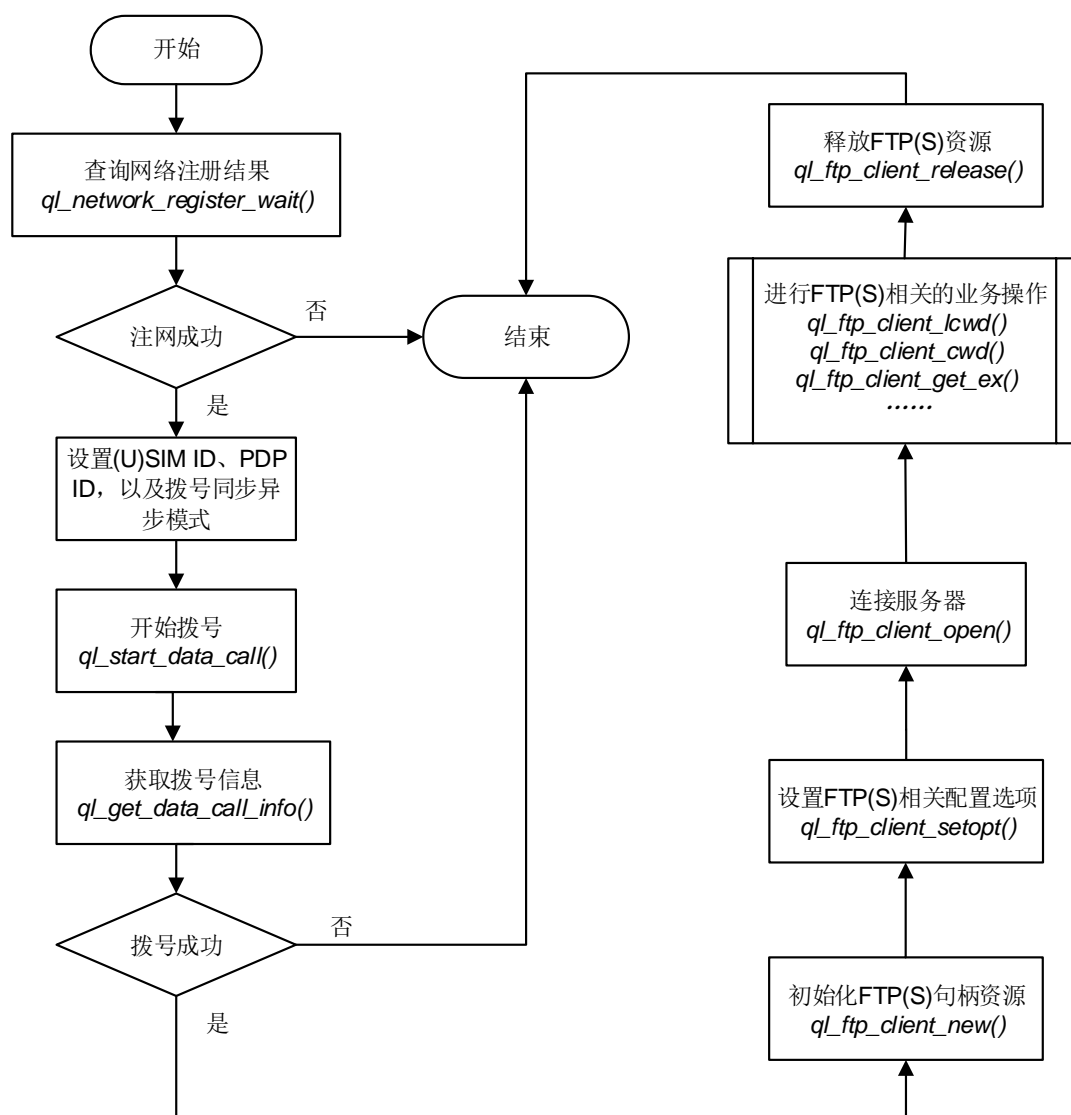


图 1: FTP(S)接口调用流程



发送 FTP(S)请求之前，需建立数据通道，并进行网络相关配置，FTP(S)基于数据通道进行数据收发。若在其他任务中已经建立了数据通道，则可省略这一步骤，但需获取数据通道的 SIM\_CID，因为 FTP(S)客户端通过 SIM\_CID 来使用此数据通道。

数据通道 SIM\_CID 的配置方法如下：

通过 `ql_ftp_client_new()`函数创建一个新的 FTP(S)客户端句柄之后，可以通过 `ql_ftp_client_setopt()`函数配置 FTP(S)客户端所使用数据通道的 SIM\_CID 以及是否需要 SSL 加密。

## 备注

1. FTP(S)的数据通道支持主动（PORT）和被动（PASV）两种模式，目前默认为被动模式。
2. 使用 `ql_rtos_task_create()`函数创建 FTP(S)任务时，若需要 FTP(S)函数完成基本功能，请至少分配 16 KB 大小的堆栈空间。有关 `ql_rtos_task_create()`函数详情，请参考文档 [2]。
3. `ql_network_register_wait()`、`ql_start_data_call()`、`ql_get_data_call_info()`函数详情，请参考文档 [3]。

## 3 FTP(S) API 介绍

### 3.1. 头文件

FTP(S) API 头文件为 `ql_ftp_client.h`，位于 `components\ql-kernel\inc` 目录下。如无特别说明，本文档所述头文件均在该目录下。

### 3.2. 函数概览

表 2：函数概览

函数	说明
<code>ql_ftp_client_new()</code>	创建新的 FTP(S)客户端句柄，初始化 FTP(S)客户端资源
<code>ql_ftp_client_release()</code>	释放 FTP(S)客户端资源
<code>ql_ftp_client_setopt()</code>	配置 FTP(S)客户端相关属性
<code>ql_ftp_client_open()</code>	连接 FTP(S)服务器
<code>ql_ftp_client_close()</code>	断开与 FTP(S)服务器的连接
<code>ql_ftp_client_get_ex()</code>	从 FTP(S)服务器下载文件
<code>ql_ftp_client_put_ex()</code>	上传本地文件或用户数据到 FTP(S)服务器
<code>ql_ftp_client_delete()</code>	删除 FTP(S)服务器文件
<code>ql_ftp_client_cwd()</code>	更改 FTP(S)服务器当前使用的目录
<code>ql_ftp_client_lcwd()</code>	更改本地当前使用的目录
<code>ql_ftp_client_pwd()</code>	获取 FTP(S)服务器当前使用的目录
<code>ql_ftp_client_lpwd()</code>	获取本地当前使用的目录

<code>ql_ftp_client_mkdir()</code>	在 FTP(S)服务器上新建目录
<code>ql_ftp_client_rmdir()</code>	删除 FTP(S)服务器上的目录
<code>ql_ftp_client_list()</code>	获取 FTP(S)服务器上指定文件或目录的列表
<code>ql_ftp_client_size()</code>	获取 FTP(S)服务器上指定路径下指定文件的大小
<code>ql_ftp_client_rename()</code>	对 FTP(S)服务器上指定文件或目录进行重命名

### 3.3. API 详解

#### 3.3.1. 错误码枚举

##### 3.3.1.1. QL\_FTP\_CLIENT\_ERR\_E

FTP(S) API 错误码枚举定义如下：

```
typedef enum
{
    QL_FTP_CLIENT_ERR_SUCCESS                = 0,
    QL_FTP_CLIENT_ERR_INVALID_CLIENT         = 1 | QL_FTP_ERRCODE_BASE,
    QL_FTP_CLIENT_ERR_INVALID_HOST,
    QL_FTP_CLIENT_ERR_DNS_FAIL,
    QL_FTP_CLIENT_ERR SOCK_CREATE_FAIL,
    QL_FTP_CLIENT_ERR SOCK_BIND_FAIL        = 5 | QL_FTP_ERRCODE_BASE,
    QL_FTP_CLIENT_ERR SOCK_CONN_FAIL,
    QL_FTP_CLIENT_ERR SOCK_SEND_FAIL,
    QL_FTP_CLIENT_ERR SOCK_RECV_FAIL,
    QL_FTP_CLIENT_ERR SOCK_CLOSE_FAIL,
    QL_FTP_CLIENT_ERR SOCK_DISCONNECTED     = 10 | QL_FTP_ERRCODE_BASE,
    QL_FTP_CLIENT_ERR_SSL_CONN_FAIL,
    QL_FTP_CLIENT_ERR_RESP_TIMEOUT,
    QL_FTP_CLIENT_ERR_CREATE_FILE_FAIL,
    QL_FTP_CLIENT_ERR_NO_FILE,
    QL_FTP_CLIENT_ERR_NO_DIR                = 15 | QL_FTP_ERRCODE_BASE,
    QL_FTP_CLIENT_ERR_INVALID_PARAMATER,
    QL_FTP_CLIENT_ERR_REQUESTED_FILENAME_INVALID,
    QL_FTP_CLIENT_ERR_UNKNOWN,
} QL_FTP_CLIENT_ERR_E
```

- 成员

成员	描述
QL_FTP_CLIENT_ERR_SUCCESS	函数执行成功
QL_FTP_CLIENT_ERR_INVALID_CLIENT	不可用的 FTP(S)客户端句柄
QL_FTP_CLIENT_ERR_INVALID_HOST	请求的主机地址不可用
QL_FTP_CLIENT_ERR_DNS_FAIL	对服务器进行 DNS 解析失败。指定的代理主机无法解析,或指定的远程主机无法解析。
QL_FTP_CLIENT_ERR_SOCKET_CREATE_FAIL	创建 socket 失败
QL_FTP_CLIENT_ERR_SOCKET_BIND_FAIL	绑定 socket 失败
QL_FTP_CLIENT_ERR_SOCKET_CONN_FAIL	执行 socket 连接失败
QL_FTP_CLIENT_ERR_SOCKET_SEND_FAIL	发送数据失败。FTP(S)服务器拒绝存储客户端发送的数据,并且缓冲区内会返回对此错误的说明。
QL_FTP_CLIENT_ERR_SOCKET_RECV_FAIL	接收数据失败。将接收到的数据写入本地文件时发生错误,或接收网络数据失败。
QL_FTP_CLIENT_ERR_SOCKET_CLOSE_FAIL	与 FTP(S)服务器断开连接失败
QL_FTP_CLIENT_ERR_SOCKET_DISCONNECTED	Socket 连接断开
QL_FTP_CLIENT_ERR_SSL_CONN_FAIL	SSL 连接失败。SSL/TLS 握手过程发生错误。
QL_FTP_CLIENT_ERR_RESP_TIMEOUT	请求超时
QL_FTP_CLIENT_ERR_CREATE_FILE_FAIL	创建文件失败
QL_FTP_CLIENT_ERR_NO_FILE	文件不存在。无法通过给定的 <b>FILE://</b> 路径打开文件。
QL_FTP_CLIENT_ERR_NO_DIR	目录不存在
QL_FTP_CLIENT_ERR_INVALID_PARAMATER	无效参数
QL_FTP_CLIENT_ERR_REQUESTED_FILENAME_INVALID	请求的文件名无效
QL_FTP_CLIENT_ERR_UNKNOWN	其他错误

### 3.3.2. ql\_ftp\_client\_new

该函数用于创建新的 FTP(S)客户端句柄，初始化 FTP(S)客户端资源。

- 函数原型

```
void *ql_ftp_client_new(void)
```

- 参数

无

- 返回值

FTP(S)客户端句柄	函数执行成功
NULL	函数执行失败

### 3.3.3. ql\_ftp\_client\_release

该函数用于释放 FTP(S)客户端资源。

- 函数原型

```
void ql_ftp_client_release(void *client)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。即 *ql\_ftp\_client\_new()*函数执行成功后的非空返回值。

- 返回值

无

### 3.3.4. ql\_ftp\_client\_setopt

该函数用于配置 FTP(S)客户端相关属性。

- 函数原型

```
int ql_ftp_client_setopt(void *client, QL_FTP_CLIENT_OPT_E tag, ...)
```

### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*tag*:

[In] 属性标签；详见第 3.3.4.1 章。

...:

[In] 属性参数；详见第 3.3.4.1 章。

### ● 返回值

错误码；详见第 3.3.1.1 章。

返回值为 `QL_FTP_CLIENT_ERR_UNKNOWN` 时，请检查参数 *tag* 与 *tag* 后的属性参数是否为第 3.3.4.1 章 `QL_FTP_CLIENT_OPT_E` 中指定的参数。

#### 3.3.4.1. QL\_FTP\_CLIENT\_OPT\_E

属性标签/属性参数枚举定义如下：

```
typedef enum {
    QL_FTP_CLIENT_SIM_CID,
    QL_FTP_CLIENT_SSL_CTX,
    QL_FTP_CLIENT_SSL_ENABLE,
    QL_FTP_CLIENT_OPT_PDP_CID,
    QL_FTP_CLIENT_OPT_SSL_CTX,
    QL_FTP_CLIENT_OPT_START_POS,
    QL_FTP_CLIENT_SSL_VERIFY_LEVEL,
    QL_FTP_CLIENT_SSL_CACERT_PATH,
    QL_FTP_CLIENT_SSL_OWCERT_PATH,
    QL_FTP_CLIENT_SSL_CIPHERSUITE,
    QL_FTP_CLIENT_SSL_VERSION,
    QL_FTP_CLIENT_SSL_TRANSPORT
}QL_FTP_CLIENT_OPT_E
```

### ● 成员

成员	描述
<code>QL_FTP_CLIENT_SIM_CID</code>	(U)SIM 卡的 CID。由所用的(U)SIM 卡（nSim）和 PDP 上下文 ID（profile_idx）通过 <code>ql_bind_sim_and_profile()</code> 绑定后得到。详情请参考文档 [3]。

QL_FTP_CLIENT_SSL_CTX	FTP(S)客户端使用的 SSL 上下文
QL_FTP_CLIENT_SSL_ENABLE	FTP(S)客户端是否使用 SSL 连接。 0 连接 FTP 服务器 1 隐式加密连接 FTPS 服务器 2 显式加密连接 FTPS 服务器 3 连接 SFTP 服务器
QL_FTP_CLIENT_OPT_PDP_CID	FTP(S)客户端使用的 PDP 上下文 ID，即执行数据拨号操作时使用的 PDP_CID。详情请参考文档 [3]。
QL_FTP_CLIENT_OPT_SSL_CTX	SSL 上下文
QL_FTP_CLIENT_OPT_START_POS	下载的起始偏移位置
QL_FTP_CLIENT_SSL_VERIFY_LEVEL	SSL 校验方式。详情请参考文档 [4]。
QL_FTP_CLIENT_SSL_CACERT_PATH	SSL 证书
QL_FTP_CLIENT_SSL_OWNCERT_PATH	SSL 本地证书
QL_FTP_CLIENT_SSL_CIPHERSUITE	SSL 加密套件
QL_FTP_CLIENT_SSL_VERSION	SSL 版本信息。详情请参考文档 [4]。
QL_FTP_CLIENT_SSL_TRANSPORT	SSL 协议类型。详情请参考文档 [4]。

### 3.3.5. ql\_ftp\_client\_open

该函数用于连接 FTP(S)服务器。

#### ● 函数原型

```
int ql_ftp_client_open(void *client, char *hostname, char *username, char *password)
```

#### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*hostname*:

[In] FTP(S)服务器地址。最大长度为 200 字节。

*username*:

[In] 用户名。最大长度为 255 字节。

*password:*

[In] 密码。最大长度为 255 字节。

- 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.6. ql\_ftp\_client\_close

该函数用于断开与 FTP(S)服务器的连接。

- 函数原型

```
int ql_ftp_client_close(void *client)
```

- 参数

*client:*

[In] FTP(S)客户端句柄。

- 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.7. ql\_ftp\_client\_get\_ex

该函数用于从 FTP(S)服务器下载文件。

- 函数原型

```
int ql_ftp_client_get_ex(void *client, char *remotefile, char *localfile, QL_FTP_CLIENT_WRITE_CB_EX write_cb, void* user_data)
```

- 参数

*client:*

[In] FTP(S)客户端句柄。

*remotefile:*

[In] 需要获取的 FTP(S)服务器文件名。可以是相对路径或绝对路径，最大长度为 255 字节。

*localfile:*

[In] 存储到本地的文件名。可以是相对路径或绝对路径，最大长度为 255 字节。

如果 *write\_cb* 不为 NULL，此参数无效；

如果 *write\_cb* 为 NULL，且 *localfile* 值为 NULL，则本地文件名和 FTP(S)服务器上的文件名相同；



如果 *write\_cb* 为 NULL，且 *localfile* 值不为 NULL，则下载文件到本地文件名，该文件名由用户通过指针 *localfile* 指定。

本地默认使用的目录为 *UFS*。

*write\_cb*:

[In] 回调函数。用于接收从 FTP(S)服务器获取的文件数据。详见第 3.3.7.1 章。

如果 *write\_cb* 不为 NULL，则不再将从 FTP(S)服务器下载的文件写入 *localfile* 中，用户可通过自定义的 *write\_cb*，自行决定 *user\_data* 从 FTP(S)服务器获取的内容。

如果 *write\_cb* 为 NULL，则将从 FTP(S)服务器下载的文件写入 *localfile* 中。

*user\_data*:

[In] 用户数据。

#### ● 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.7.1. QL\_FTP\_CLIENT\_WRITE\_CB\_EX

该回调函数用于接收从 FTP(S)服务器获取的文件数据，并返回接收数据的大小。

#### ● 函数原型

```
typedef size_t (*QL_FTP_CLIENT_WRITE_CB_EX)(void *ptr, size_t size, size_t nmemb, void *stream)
```

#### ● 参数

*ptr*:

[In] 数据获取指针。可通过该指针将获取的数据写入用户所指定的存储位置。

*size*:

[In] 需写入的每个元素的大小。单位：字节。

*nmemb*:

[In] 需写入的元素个数。最终写入数据大小为 *size\*nmemb*。

*stream*:

[In] 用户传入的回调函数参数指针。可通过该指针指定参数 *ptr* 中数据存放的位置。

#### ● 返回值

写入数据大小      函数执行成功

小于 0 的数值      函数执行失败，FTP(S)异常结束

### 3.3.8. ql\_ftp\_client\_put\_ex

该函数用于上传本地文件或用户数据到 FTP(S)服务器。

#### ● 函数原型

```
int ql_ftp_client_put_ex(void *client, char *localfile, char *remotefile, QL_FTP_CLIENT_READ_CB_EX read_cb, void *user_data)
```

#### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*localfile*:

[In] 需上传的本地文件名。可以是相对路径或绝对路径，最大长度为 255 字节。

如果 *read\_cb* 不为 NULL，此参数无效；

如果 *read\_cb* 为 NULL，则 *localfile* 值不能为 NULL，否则函数返回错误。

本地默认使用的目录为 UFS:。

*remotefile*:

[In] 存储到 FTP(S)服务器的文件名。可以是相对路径或绝对路径，最大长度为 255 字节。

如果该值为 NULL，则本地文件名和 FTP(S)服务器上的文件名相同；

如果 *read\_cb* 不为 NULL，则必须设置 *remotefile* 参数，否则函数返回错误。

*read\_cb*:

[In] 回调函数。用于发送数据到 FTP(S)服务器。详见第 3.3.8.1 章。

如果 *read\_cb* 不为 NULL，则 *localfile* 参数无效，不再上传本地文件。用户可通过自定义的 *read\_cb* 自行决定上传 *user\_data* 所指向的数据内容。

*user\_data*:

[In] 用户数据。

#### ● 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.8.1. QL\_FTP\_CLIENT\_READ\_CB\_EX

该回调函数用于发送数据到 FTP(S)服务器。

#### ● 函数原型

```
typedef size_t (*QL_FTP_CLIENT_READ_CB_EX)(void *ptr, size_t size, size_t nmemb, void *stream)
```

#### ● 参数

*ptr*:

[Out] 读取数据的指针。用户传入的数据可被该指针读取，完成用户数据的记录。

*size*:

[In] 需读取的每个元素的大小。单位：字节。

*nmemb*:

[In] 需读取的元素个数。最终读取数据大小为 *size\*nmemb*。

*stream*:

[In] 用户传入的回调函数参数指针。可通过该指针向参数 *ptr* 传入数据。

#### ● 返回值

读取数据大小	函数执行成功
小于 0 的数值	函数执行失败，FTP(S)异常结束

### 3.3.9. ql\_ftp\_client\_delete

该函数用于删除 FTP(S)服务器文件。

#### ● 函数原型

```
int ql_ftp_client_delete(void *client, char *remotefile)
```

#### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*remotefile*:

[In] 需删除的 FTP(S)服务器文件名。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码：详见第 3.3.1.1 章。

### 3.3.10. ql\_ftp\_client\_cwd

该函数用于更改 FTP(S)服务器当前使用的目录。

- 函数原型

```
int ql_ftp_client_cwd(void *client, char *path)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[In] FTP(S)服务器目录。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码：详见第 3.3.1.1 章。

### 3.3.11. ql\_ftp\_client\_lcwd

该函数用于更改本地当前使用的目录。

- 函数原型

```
int ql_ftp_client_lcwd(void *client, char *path)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[In] 本地目录。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码：详见第 3.3.1.1 章。

### 3.3.12. ql\_ftp\_client\_pwd

该函数用于获取 FTP(S)服务器当前使用的目录。

- 函数原型

```
int ql_ftp_client_pwd(void *client, char *path, int path_len)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[Out] FTP(S)服务器目录。绝对路径，最大长度为 255 字节。

*path\_len*:

[In] *path* 的最大存储长度。

- 返回值

错误码；详见第3.3.1.1章。

### 3.3.13. ql\_ftp\_client\_lpwd

该函数用于获取本地当前使用的目录。

- 函数原型

```
int ql_ftp_client_lpwd(void *client, char *path, int path_len)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[Out] 本地目录。绝对路径，最大长度为 255 字节。

*path\_len*:

[In] *path* 的最大存储长度。

- 返回值

错误码；详见第3.3.1.1章。

### 3.3.14. ql\_ftp\_client\_mkdir

该函数用于在 FTP(S)服务器上新建目录。

- 函数原型

```
int ql_ftp_client_mkdir(void *client, char *path)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[In] 需新建的 FTP(S)服务器目录。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.15. ql\_ftp\_client\_rmdir

该函数用于删除 FTP(S)服务器上的目录。

- 函数原型

```
int ql_ftp_client_rmdir(void *client, char *path)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*path*:

[In] 需删除的 FTP(S)服务器目录。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码；详见第 3.3.1.1 章。

### 3.3.16. ql\_ftp\_client\_list

该函数用于获取 FTP(S)服务器上指定文件或目录的列表。

#### ● 函数原型

```
int ql_ftp_client_list(void *client, char *path_or_file, char *match, QL_FTP_CLIENT_FILE_INFO_T
*file_info_list, int max_file_num)
```

#### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*path\_or\_file*:

[In] FTP(S)服务器上的目录或文件。可以是相对路径或绝对路径，最大长度为 255 字节。

*match*:

[In] 匹配字符。用于列出包含 *match* 字符的文件。

*file\_info\_list*:

[Out] 获取到的文件和目录列表；详见第 3.3.16.1 章。

*max\_file\_num*:

[In] *file\_info\_list* 可以存储文件的最大个数。

#### ● 返回值

大于 0 实际获取到的列表个数

-1 获取失败

#### 3.3.16.1. QL\_FTP\_CLIENT\_FILE\_INFO\_T

文件和目录列表结构体定义如下：

```
typedef struct{
    char filename[MAX_FILE_NAME_SIZE];
    int type;
    int permissions;
    int links;
    char owner[MAX_OWNER_SIZE];
    char group[MAX_OWNER_SIZE];
    struct tm timestamp;
    unsigned long size;
```

```
}QL_FTP_CLIENT_FILE_INFO_T
```

#### ● 参数

类型	参数	描述
char	<i>filename</i>	文件或目录
int	<i>type</i>	类型 0 未知 1 目录 2 文件
int	<i>permissions</i>	权限
int	<i>links</i>	链接数
char	<i>owner</i>	所属用户
char	<i>group</i>	所属用户组
<i>struct tm</i>	<i>timestamp</i>	创建时间
unsigned long	<i>size</i>	文件大小。单位：字节

### 3.3.17. ql\_ftp\_client\_size

该函数用于获取 FTP(S)服务器上指定路径下指定文件的大小。

#### ● 函数原型

```
int ql_ftp_client_size(void *client, const char *filename, double *nSize)
```

#### ● 参数

*client*:

[In] FTP(S)客户端句柄。

*filename*:

[In] FTP(S)服务器上的目录或文件。可以是相对路径或绝对路径，最大长度为 255 字节。

*nSize*:

[Out] 获取的 FTP(S)服务器上指定文件的大小。



- 返回值

错误码：详见第 3.3.1.1 章。

### 3.3.18. ql\_ftp\_client\_rename

该函数用于对 FTP(S)服务器上指定文件或目录进行重命名。

- 函数原型

```
int ql_ftp_client_rename(void *client, const char *oldname, const char *newname)
```

- 参数

*client*:

[In] FTP(S)客户端句柄。

*oldname*:

[In] FTP(S)服务器上指定文件或目录的当前名称。可以是相对路径或绝对路径，最大长度为 255 字节。

*newname*:

[In] FTP(S)服务器上指定文件或目录的新名称。可以是相对路径或绝对路径，最大长度为 255 字节。

- 返回值

错误码：详见第 3.3.1.1 章。

## 4 示例

有关 FTP(S)功能相关示例，请参考位于 SDK 的 `\components\ql-application\ftp` 目录下的应用层示例文件 `ftp_demo.c` 和 `ftp_demo2.c`，以及 `\components\ql-application\sftp` 目录下的示例文件 `sftp_demo.c`。

## 5 附录 参考文档及术语缩写

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_RTOS_开发指导
[3] Quectel_ECx00U&EGx00U 系列_QuecOpen_数据拨号 API_参考手册
[4] Quectel_ECx00U&EGx00U 系列_QuecOpen_SSL_API_参考手册

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
DNS	Domain Name Server	域名服务器
FTP	File Transfer Protocol	文件传输协议
FTPS	FTP over SSL	对常用的文件传输协议（FTP）添加传输层安全（TLS）和安全套接层（SSL）加密协议支持的扩展协议
IoT	Internet of Things	物联网
RTOS	Real-Time Operating System	实时操作系统
SDK	Software Development Kit	软件开发工具包
(U)SIM	(Universal) Subscriber Identity Module	（通用）用户识别模块
SSL	Secure Sockets Layer	安全套接层
TLS	Transport Layer Security	传输层安全协议