

# QSTM32

## SDK API Design

版本：2.0

日期：2025-09-24

状态：受控/临时文件

文件保密级别：（请勾选 ☒ ）

绝密 ☐

保密 ☐

公开 ☒

文件管控表

文件变更记录			
修订日期	最新版本	修订内容	编制
2025-09-24	2	首次发行	Wells Li

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，请随时登录网址：  
<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 2024，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2024.**

## 目录

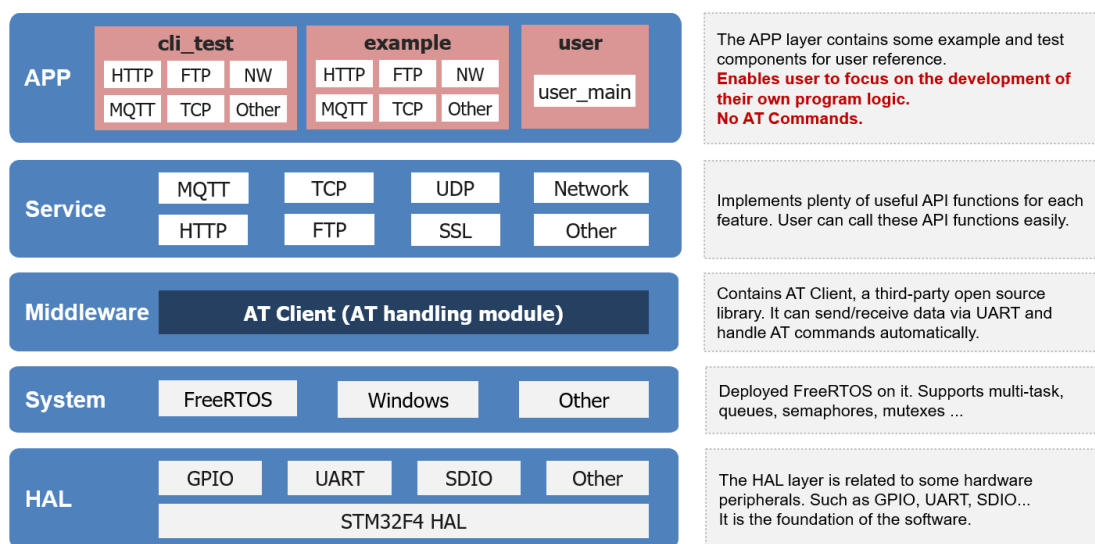
目录 .....	3
1 软件框架 .....	6
2 代码结构 .....	7
3 模块接口介绍 .....	8
3.1. 驻网模块 .....	8
3.1.1. 模块概述 .....	8
3.1.2. 接口概述 .....	8
3.1.3. 接口详细说明 .....	8
3.1.3.1. ql_net_init .....	8
3.1.3.2. ql_usim_get .....	8
3.1.3.3. ql_net_setopt .....	9
3.1.3.4. ql_net_attach .....	9
3.1.3.5. ql_net_get_signal_info .....	9
3.1.3.6. ql_net_is_ok .....	10
3.1.3.7. ql_net_reconnect .....	10
3.1.3.8. ql_module_reboot .....	10
3.1.3.9. ql_net_deinit .....	11
3.2. HTTP 模块 .....	11
3.2.1. 模块概述 .....	11
3.2.2. 接口概述 .....	11
3.2.3. 接口详细说明 .....	12
3.2.3.1. ql_http_init .....	12
3.2.3.2. ql_http_setopt .....	12
3.2.3.3. ql_http_set_ssl .....	12
3.2.3.4. ql_http_request .....	13
3.2.3.5. ql_http_recv .....	14
3.2.3.6. ql_http_deinit .....	14
3.3. FTP 模块 .....	14
3.3.1. 模块概述 .....	14
3.3.2. 接口概述 .....	14
3.3.3. 接口详细说明 .....	15
3.3.3.1. ql_ftp_init .....	15
3.3.3.2. ql_ftp_setopt .....	15
3.3.3.3. ql_ftp_set_ssl .....	16

3.3.3.4.	ql_ftp_login .....	16
3.3.3.5.	ql_ftp_cwd .....	16
3.3.3.6.	ql_ftp_pwd .....	17
3.3.3.7.	ql_ftp_mkdir .....	17
3.3.3.8.	ql_ftp_rmdir .....	18
3.3.3.9.	ql_ftp_rename .....	18
3.3.3.10.	ql_ftp_list .....	18
3.3.3.11.	ql_ftp_nlist .....	19
3.3.3.12.	ql_ftp_list_free .....	19
3.3.3.13.	ql_ftp_file_size .....	19
3.3.3.14.	ql_ftp_upload .....	20
3.3.3.15.	ql_ftp_download .....	20
3.3.3.16.	ql_ftp_file_delete .....	21
3.3.3.17.	ql_ftp_file_get_modify_time .....	21
3.3.3.18.	ql_ftp_logout .....	22
3.3.3.19.	ql_ftp_uninit .....	22
3.4.	MQTT 模块 .....	22
3.4.1.	模块概述 .....	22
3.4.2.	接口概述 .....	22
3.4.3.	接口详细说明 .....	23
3.4.3.1.	ql_mqtt_create .....	23
3.4.3.2.	ql_mqtt_setopt .....	23
3.4.3.3.	ql_mqtt_set_ssl .....	23
3.4.3.4.	ql_mqtt_connect .....	24
3.4.3.5.	ql_mqtt_disconnect .....	24
3.4.3.6.	ql_mqtt_pub .....	25
3.4.3.7.	ql_mqtt_sub .....	25
3.4.3.8.	ql_mqtt_unsub .....	26
3.4.3.9.	ql_mqtt_destroy .....	26
3.5.	SOCKET 模块 .....	26
3.5.1.	模块概述 .....	26
3.5.2.	接口概述 .....	26
3.5.3.	接口详细说明 .....	27
3.5.3.1.	socket .....	27
3.5.3.2.	close .....	27
3.5.3.3.	shutdown .....	28
3.5.3.4.	bind .....	28
3.5.3.5.	connect .....	28
3.5.3.6.	sendto .....	29
3.5.3.7.	send .....	29
3.5.3.8.	recvfrom .....	30
3.5.3.9.	recv .....	31
3.5.3.10.	setsockopt .....	31
3.5.3.11.	listen .....	32

3.5.3.12.	accept.....	32
3.5.3.13.	select.....	32
3.6.	PSM 模块 .....	33
3.6.1.	模块概述 .....	33
3.6.2.	接口概述 .....	33
3.6.3.	接口详细说明 .....	33
3.6.3.1.	ql_psm_settings_write.....	33
3.6.3.2.	ql_psm_settings_read .....	34

# 1 软件框架

Quectel 设计了一套软件代码框架，提供了操作 AT 指令的接口，使开发者专注于自己业务的设计，而不需要关心底层指令，大大降低了开发者对模组开发的难度。



如上图，代码框架分了 5 层：

- 1 层 (Hardware Peripheral)：使用 STM32CubeIDE 工具配置自动生成的驱动代码；
- 2 层 (System)：提供 OS 接口, 目前基于 Freertos 操作系统开发；
- 3 层 (Middleware)：移植第三方 AT Client 组件；
- 4 层 (Device) 移远基于 AT 指令封装的各个模块的 api 接口，包含 http、ftp、mqtt、socket 等 c 代码接口；
- 5 层 (APP)：提供的各个模块的示例代码。



## 2 代码结构

代码目录由四部分部分组成：

### ■ app 目录

**example** 提供各功能模块的示例代码，方便用户参考和使用，包含 ftp、http、mqtt、socket、等模块功能；

**test** 移远内部使用的自测代码，用于功能验证和测试；

**user** 用户应用开发的主要目录， 其中 user\_main： 代码入口文件， 用户可基于此进行二次开发。

### ■ quectel 目录

该目录主要用于封装 AT 指令， 包含移远编写的核心代码。

**common** 封装内部通用的公共接口， 供其他模块调用；

**modules** 对 AT 指令封装， 包括 http、ftp、mqtt、socket 等功能；

**third\_party**： 存放第三方组件， 目前包含 at\_client 和 at\_socket；

### ■ system 目录

该目录主要是为了实现跨平台移植， 封装了一些依赖于系统的 API 接口, 包含串口、信号量、任务、互斥锁等相关接口。

### ■ tools 目录

该目录存放自动化编译脚本。

# 3 模块接口介绍

## 3.1. 驻网模块

### 3.1.1. 模块概述

本模块提供蜂窝网络驻网能力，包括初始化、参数配置、状态管理等功能，支持多模组兼容设计。

### 3.1.2. 接口概述

类型	文件路径	说明
头文件	ql_net.h	接口声明、结构体定义
源文件	ql_net.c	接口实现代码
示例	example_net.c	快速集成参考
依赖库	at_client.c	需提前初始化 AT 指令框架

### 3.1.3. 接口详细说明

#### 3.1.3.1. ql\_net\_init

函数功能：驻网初始化

函数原型：ql\_net\_t ql\_net\_init(at\_client\_t client)

参数	类型	说明
----	----	----

client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例。
--------	-------------	--

返回值：初始化成功返回驻网句柄，失败返回 NULL

#### 3.1.3.2. ql\_usim\_get

函数功能：查询 SIM 卡状态

函数原型: `QL_NET_ERR_CODE_E ql_usim_get(ql_net_t handle)`

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值

返回值: SIM 卡状态正常返回 `QT_NET_OK`, 否则返回其他

### 3.1.3.3. ql\_net\_setopt

函数功能: 配置驻网参数

函数原型: `bool ql_net_setopt(ql_net_t handle, QL_NET_OPTION_E type, ...)`

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值
type	QL_NET_OPTION_E	参考 SDK 头文件中的完整枚举定义
...	可变参数	参数实际值

返回值: 配置成功返回 `true`, 失败返回 `false`

### 3.1.3.4. ql\_net\_attach

函数功能: 驻网

函数原型: `QL_NET_ERR_CODE_E ql_net_attach(ql_net_t handle)`

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值

返回值: 驻网成功返回 `QT_NET_OK`, 否则返回其他

### 3.1.3.5. ql\_net\_get\_signal\_info

函数功能: 查询网络信号强度 和质量

函数原型: `int ql_net_get_signal_info(ql_net_t handle, int *strength, int *quality);`

参数	类型	说明
----	----	----

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值
strength	int *	输出参数，用于接收网络信号强度值
quality	int *	输出参数，用于接收网络信号质量值

返回值：查询成功返回 0，否则返回-1。

### 3.1.3.6. ql\_net\_is\_ok

函数功能：查询网络是否正常

函数原型：int ql\_net\_is\_ok(ql\_net\_t handle)

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值

返回值：true 网络正常，false 网络异常

### 3.1.3.7. ql\_net\_reconnect

函数功能：重新驻网

函数原型：QL\_NET\_ERR\_CODE\_E ql\_net\_reconnect(ql\_net\_t handle)

参数	类型	说明
handle	ql_net_t	ql_net_init 返回值

返回值：驻网成功返回 QT\_NET\_OK，否则返回其他

### 3.1.3.8. ql\_module\_reboot

函数功能：重启模组

函数原型：void ql\_module\_reboot(ql\_net\_t handle)

参数	类型	说明
----	----	----

参数	类型	说明
----	----	----

handle	ql_net_t	ql_net_init 返回值
--------	----------	-----------------

返回值：无

### 3.1.3.9. ql\_net\_deinit

函数功能：驻网反初始化

函数原型：void ql\_net\_deinit(ql\_net\_t handle)

参数	类型	说明
----	----	----

handle	ql_net_t	ql_net_init 返回值
--------	----------	-----------------

返回值：无

## 3.2. HTTP 模块

### 3.2.1. 模块概述

本模块提供基于蜂窝网络的 HTTP/HTTPS 协议通信能力，支持设备与云端服务的双向数据交互，适配主流证书体系 (CA 认证、双向认证)。

### 3.2.2. 接口概述

类型	文件路径	说明
----	------	----

头文件	ql_http.h	接口声明、结构体定义
-----	-----------	------------

源文件	ql_http.c	接口实现代码
-----	-----------	--------

示例	example_http.c	快速集成参考
----	----------------	--------

依赖库	at_client.c	需提前初始化 AT 指令框架
-----	-------------	----------------

### 3.2.3. 接口详细说明

#### 3.2.3.1. ql\_http\_init

函数功能：http 接口初始化

函数原型：ql\_http\_t ql\_http\_init(at\_client\_t client)

参数	类型	说明
client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例。

返回值：初始化成功返回 http 句柄，失败返回 NULL

注：虽然接口支持多实例，但同一时刻模组只支持一个 HTTP 请求，外部调用时需保证不要同时调用多个 http 请求。

#### 3.2.3.2. ql\_http\_setopt

函数功能：http 参数配置

函数原型：bool ql\_http\_setopt(ql\_http\_t handle, QL\_HTTP\_OPTION\_E type, ...)

参数	类型	说明
handle	ql_http_t	ql_http_init 返回值
type	QL_HTTP_OPTION_E	参考 SDK 头文件中的完整枚举定义
...	可变参数	参数实际值

返回值：配置成功返回 true, 失败返回 false

#### 3.2.3.3. ql\_http\_set\_ssl

函数功能：ssl 参数配置

函数原型：void ql\_http\_set\_ssl(ql\_http\_t handle, ql\_SSL\_Config config)

参数	类型	说明
handle	ql_http_t	ql_http_init 返回值

参数	类型	说明
----	----	----

config	ql_SSL_Config	ssl 参数信息，参考 ssl 头文件
--------	---------------	---------------------

返回值：无

#### 3.2.3.4. ql\_http\_request

函数功能：发送 http 请求

函数原型：QL\_HTTP\_ERR\_CODE\_E ql\_http\_request(ql\_http\_t handle, const char\* url, QL\_HTTP\_METHOD\_E method, const char\* data, size\_t data\_len)

参数	类型	说明
----	----	----

handle	ql_http_t	ql_http_init 返回值
--------	-----------	------------------

url	const char*	URL 地址，支持 HTTP(S)
-----	-------------	-------------------

method	QL_HTTP_METHOD_E	HTTP 请求方法
--------	------------------	-----------

data	const char*	请求体数据指针
------	-------------	---------

data_len	size_t	请求体数据的长度
----------	--------	----------

返回值：请求成功返回 QT\_HTTP\_OK，否则返回其他

注：

1. 该接口为同步接口，因为模组通过串口通信，http 请求过程中，不允许有其他操作；
2. 模组不支持带 body 的 get 请求，所以 get 请求时 ql\_http\_request 函数中的 data 只能为 NULL。

读取 http(s) 请求响应的数据方式有两种：

1. 直接调用 ql\_http\_recv 函数读取，当返回值 <= 0，说明读取完毕。请注意，这种读取方式，不能调用 ql\_http\_setopt 接口设置写回调函数；
2. 通过 ql\_http\_setopt 接口设置写回调函数接收数据，请注意此时不能调用 ql\_http\_recv 读取数据。

post 请求支持 2 种方式

1. 直接将 body 数据放到请求体数据指针 data 中，适用与内存较充足的情况；
2. 当内存不足以满足 body 的大小时，通过 ql\_http\_setopt 接口设置读回调函数，分批发送 body 数据。

### 3.2.3.5. ql\_http\_recv

函数功能：读取 http 请求响应的信息

函数原型：int ql\_http\_recv(ql\_http\_t handle, char\* buf, size\_t size)

参数	类型	说明
handle	ql_http_t	ql_http_init 返回值
buf	char*	存储响应数据的指针，需提前分配
size	size_t	要读取的字节数

返回值：读取的实际长度，返回 0 代表已读取完毕

### 3.2.3.6. ql\_http\_deinit

函数功能：http 接口反初始化

函数原型：void ql\_http\_deinit(ql\_http\_t handle)

参数	类型	说明
handle	ql_http_t	ql_http_init 返回值

返回值：无

## 3.3. FTP 模块

### 3.3.1. 模块概述

本模块提供基于蜂窝网络的 ftp(s)通信能力，支持设备与服务器之间的文件上传、下载及文件(夹)管理等功能，支持 CA 认证。

### 3.3.2. 接口概述

类型	文件路径	说明
----	------	----



类型	文件路径	说明
头文件	ql_ftp.h	接口声明、结构体定义
源文件	ql_ftp.c	接口实现代码
示例	example_ftp.c	快速集成参考
依赖库	at_client.c	需提前初始化 AT 指令框架

### 3.3.3. 接口详细说明

#### 3.3.3.1. ql\_ftp\_init

函数功能：ftp 接口初始化

函数原型：ql\_ftp\_t ql\_ftp\_init(at\_client\_t client)

参数	类型	说明
client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例。

返回值：初始化成功返回 ftp 句柄，失败返回 NULL

**注：**虽然接口支持多实例，但同一时刻模组只支持一个 HTTP 请求，外部调用时需保证不要同时登录多个 ftp

#### 3.3.3.2. ql\_ftp\_setopt

函数功能：ftp 参数配置

函数原型：bool ql\_ftp\_setopt(ql\_ftp\_t handle, QL\_FTP\_OPTION\_E type, ...)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
type	QL_FTP_OPTION_E	参考 SDK 头文件中的完整枚举定义

参数	类型	说明
----	----	----

... 可变参数 参数实际值

返回值：配置成功返回 **true**,失败返回 **false**

### 3.3.3.3. ql\_ftp\_set\_ssl

函数功能：ssl 参数配置

函数原型：bool ql\_ftp\_set\_ssl(ql\_ftp\_t handle, ql\_SSL\_Config config)

参数	类型	说明
----	----	----

handle ql\_ftp\_t ql\_ftp\_init 返回值

config ql\_SSL\_Config ssl 参数信息，参考 ssl 头文件

返回值：配置成功返回 **true**，否则返回 **false**

### 3.3.3.4. ql\_ftp\_login

函数功能：客户端登录

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_login(ql\_ftp\_t handle, const char\* username, const char\* password)

参数	类型	说明
----	----	----

handle ql\_ftp\_t ql\_ftp\_init 返回值

username const char\* 鉴权用户名

password const char\* 鉴权密码

返回值：登录成功返回 **QT\_FTP\_OK**，否则返回其他

### 3.3.3.5. ql\_ftp\_cwd

函数功能：设置工作目录

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_cwd(ql\_ftp\_t handle, const char\* path)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
path	const char*	工作目录路径
返回值：设置成功返回 QT_FTP_OK，否则返回其他		

### 3.3.3.6. ql\_ftp\_pwd

函数功能：查询当前工作目录

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_pwd(ql\_ftp\_t handle, char\* path)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
path	const char*	存储当前工作目录路径的指针，需提前分配
返回值：查询成功返回 QT_FTP_OK，否则返回其他		

### 3.3.3.7. ql\_ftp\_mkdir

函数功能：创建远程文件夹

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_mkdir(ql\_ftp\_t handle, const char\* path)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
path	const char*	需要创建的远程目录路径
返回值：创建成功返回 QT_FTP_OK，否则返回其他		

### 3.3.3.8. ql\_ftp\_rmdir

函数功能：删除远程文件夹

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_rmdir(ql\_ftp\_t handle, const char\* path)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
path	const char*	需要删除的远程目录路径

返回值：删除成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.9. ql\_ftp\_rename

函数功能：重命名文件(夹)

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_rename(ql\_ftp\_t handle, const char\* old\_name, const char\* new\_name)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
old_name	const char*	原文件(夹)的路径
new_name	const char*	新文件(夹)的路径

返回值：删除成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.10. ql\_ftp\_list

函数功能：查询指定目录下的所有文件(夹)信息

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_list(ql\_ftp\_t handle, const char \*path, ql\_ftp\_file\_info\_s \*\*head)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
path	const char*	查询目录路径

参数	类型	说明
----	----	----

head      ql\_ftp\_file\_info\_s \*\*      存储文件信息链表头指针的地址，需外部调用  
ql\_ftp\_list\_free 释放，结构体信息参见头文件

返回值：查询成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.11.ql\_ftp\_nlist

函数功能：查询指定目录下的所有文件(夹)名称

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_nlist(ql\_ftp\_t handle, const char \*path, ql\_ftp\_file\_info\_s \*\*head)

参数	类型	说明
----	----	----

handle      ql\_ftp\_t      ql\_ftp\_init 返回值

path      const char\*      查询目录路径

head      ql\_ftp\_file\_info\_s \*\*      存储文件信息链表头指针的地址，需外部调用  
ql\_ftp\_list\_free 释放，结构体信息参见头文件

返回值：查询成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.12.ql\_ftp\_list\_free

函数功能：释放由 ql\_ftp\_list 或 ql\_ftp\_nlist 分配的内存

函数原型：void ql\_ftp\_list\_free(ql\_ftp\_file\_info\_s \*head)

参数	类型	说明
----	----	----

head      ql\_ftp\_file\_info\_s \*      ql\_ftp\_list 或 ql\_ftp\_nlist 分配的内存

返回值：无

### 3.3.3.13.ql\_ftp\_file\_size

函数功能：获取指定文件的大小

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_file\_size(ql\_ftp\_t handle, const char \*remote\_file\_name, size\_t \*file\_size)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
remote_file_name	const char*	文件名
file_size	size_t *	输出参数，函数成功时会填充文件大小

返回值：查询成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.14.ql\_ftp\_upload

函数功能：上传本地文件到远程服务器

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_upload(ql\_ftp\_t handle, const char \*local\_file\_name, const char \*remote\_file\_name)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
local_file_name	const char*	本地文件路径
remote_file_name	const char*	远程服务器文件路径

返回值：上传成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.15.ql\_ftp\_download

函数功能：下载远程服务器文件到本地

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_download(ql\_ftp\_t handle, const char \*remote\_file\_name, const char \*local\_file\_name)

参数	类型	说明
handle	ql_ftp_t	ql_ftp_init 返回值
remote_file_name	const char*	远程服务器文件路径

参数	类型	说明
----	----	----

local\_file\_name      const char\*      本地文件路径

返回值：下载成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.16.ql\_ftp\_file\_delete

函数功能：删除文件

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_file\_delete(ql\_ftp\_t handle, const char \*remote\_file\_name)

参数	类型	说明
----	----	----

handle      ql\_ftp\_t      ql\_ftp\_init 返回值

remote\_file\_name      const char\*      远程服务器文件路径

返回值：删除成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.17.ql\_ftp\_file\_get\_modify\_time

函数功能：查询文件最后修改时间

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_file\_get\_modify\_time(ql\_ftp\_t handle, const char \*remote\_file\_name, char \*time)

参数	类型	说明
----	----	----

handle      ql\_ftp\_t      ql\_ftp\_init 返回值

remote\_file\_name      const char\*      文件名

time      char\*      存放文件最后修改时间地址，时间格式 YYYYMMDDHHMMSS  
或 YYYYMMDDHHMMSS.NNN

返回值：查询成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.18. ql\_ftp\_logout

函数功能：客户端登出

函数原型：QL\_FTP\_ERR\_CODE\_E ql\_ftp\_logout(ql\_ftp\_t handle)

参数	类型	说明
----	----	----

handle	ql_ftp_t	ql_ftp_init 返回值
--------	----------	-----------------

返回值：登出成功返回 QT\_FTP\_OK，否则返回其他

### 3.3.3.19. ql\_ftp\_uninit

函数功能：ftp 反初始化

函数原型：void ql\_ftp\_uninit(ql\_ftp\_t handle)

参数	类型	说明
----	----	----

handle	ql_ftp_t	ql_ftp_init 返回值
--------	----------	-----------------

返回值：无

## 3.4. MQTT 模块

### 3.4.1. 模块概述

本模块提供基于 MQTT 协议的物联网（IoT）通信能力，支持设备与 MQTT Broker（服务器）之间的安全连接、消息发布/订阅及 QoS 等级控制，并支持 TLS/SSL 加密 和 CA 证书认证，确保数据传输的安全性。

### 3.4.2. 接口概述

类型	文件路径	说明
头文件	ql_mqtt.h	接口声明、结构体定义
源文件	ql_mqtt.c	接口实现代码



类型	文件路径	说明
示例	example_mqtt.c	快速集成参考
依赖库	at_client.c	需提前初始化 AT 指令框架

### 3.4.3. 接口详细说明

#### 3.4.3.1. ql\_mqtt\_create

函数功能：创建 mqtt 客户端句柄

函数原型：ql\_mqtt\_t ql\_mqtt\_create (at\_client\_t client)

参数	类型	说明
client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例。

返回值：初始化成功返回 MQTT 句柄，失败返回 NULL

**注：最多支持 6 个 mqtt 实例**

#### 3.4.3.2. ql\_mqtt\_setopt

函数功能：ftp 参数配置

函数原型：bool ql\_ftp\_setopt(ql\_mqtt\_t handle, QL\_MQTT\_OPTION\_E type, ...)

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
type	QL_MQTT_OPTION_E	参考 SDK 头文件中的完整枚举定义
...	可变参数	参数实际值

返回值：配置成功返回 true, 失败返回 false

#### 3.4.3.3. ql\_mqtt\_set\_ssl

函数功能：ssl 参数配置

函数原型: `bool ql_mqtt_set_ssl(ql_mqtt_t handle, ql_SSL_Config config)`

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
config	ql_SSL_Config	ssl 参数信息, 参考 ssl 头文件

返回值: 配置成功返回 true, 否则返回 false

#### 3.4.3.4. ql\_mqtt\_connect

函数功能: 连接 MQTT 服务器

函数原型: `QL_MQTT_ERR_CODE_E ql_mqtt_connect(ql_mqtt_t handle, const char* server, int port, const char* username, const char* password)`

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
server	const char*	MQTT 服务器地址
port	int	MQTT 服务器端口
username	const char*	认证用户名, 如果没有传 NULL
password	const char*	认证密码, 如果没有传 NULL

返回值: 连接成功返回 QL\_MQTT\_OK, 否则返回其他

#### 3.4.3.5. ql\_mqtt\_disconnect

函数功能: 断开服务器连接

函数原型: `void ql_mqtt_disconnect(ql_mqtt_t handle)`

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值

返回值: 无

### 3.4.3.6. ql\_mqtt\_pub

函数功能：发布主题消息

函数原型：QL\_MQTT\_ERR\_CODE\_E ql\_mqtt\_pub(ql\_mqtt\_t handle, const char \*topic, const char \*message, QL\_MQTT\_QOS\_LEVEL\_E qos, bool retain)

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
topic	const char *	发布主题名称
message	const char *	消息内容
qos	QL_MQTT_QOS_LEVEL_E	消息服务质量等级
retain	bool	是否保留消息

返回值：发布成功返回 QT\_MQTT\_OK，否则返回其他

### 3.4.3.7. ql\_mqtt\_sub

函数功能：订阅主题

函数原型：QL\_MQTT\_ERR\_CODE\_E ql\_mqtt\_sub(ql\_mqtt\_t handle, const char \*topic, QL\_MQTT\_QOS\_LEVEL\_E qos)

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
topic	const char *	订阅主题名称
qos	QL_MQTT_QOS_LEVEL_E	消息服务质量等级

返回值：订阅成功返回 QT\_MQTT\_OK，否则返回其他

### 3.4.3.8. ql\_mqtt\_unsub

函数功能：退订主题

函数原型：QL\_MQTT\_ERR\_CODE\_E ql\_mqtt\_unsub(ql\_mqtt\_t handle, const char \*topic)

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值
topic	const char *	订阅主题名称

返回值：退订成功返回 QT\_MQTT\_OK，否则返回其他

### 3.4.3.9. ql\_mqtt\_destroy

函数功能：销毁 mqtt 句柄

函数原型：void ql\_mqtt\_destroy(ql\_mqtt\_t handle)

参数	类型	说明
handle	ql_mqtt_t	ql_mqtt_init 返回值

返回值：无

## 3.5. SOCKET 模块

### 3.5.1. 模块概述

本模块提供基于 TCP/UDP 套接字（Socket）的网络通信能力，支持设备与服务器之间的稳定数据传输，适用于需要低延迟、高可靠性的通信场景。

### 3.5.2. 接口概述

类型	文件路径	说明
----	------	----

类型	文件路径	说明
头文件	ql_socket.h	接口声明、结构体定义
源文件	ql_socket.c	接口实现代码
示例	socket 目录	快速集成参考
依赖库	at_client.c	需提前初始化 AT 指令框架

### 3.5.3. 接口详细说明

#### 3.5.3.1. socket

函数功能：创建一个 socket 描述符

函数原型：int socket (int domain, int type, int protocol)

参数	类型	说明
domain	int	指定通信域/协议族，目前仅支持 AF_INET (IPv4)
type	int	指定 socket 类型，目前支持 SOCK_STREAM (TCP)、SOCK_DGRAM (UDP)

返回值：创建成功返回非负整数，失败返回-1

**注:**同一时刻最多同时存在 12 个 socket 描述符

#### 3.5.3.2. close

函数功能：关闭文件描述符

函数原型：int close (int sockfd)

参数	类型	说明
sockfd	int	socket 描述符

返回值：关闭成功返回 0，失败返回-1

### 3.5.3.3. shutdown

函数功能：关闭文件描述符

函数原型：int shutdown(int sockfd, int how)

参数	类型	说明
sockfd	int	socket 描述符
how	int	保持兼容，实际没用到

返回值：关闭成功返回 0，失败返回-1

### 3.5.3.4. bind

函数功能：将 socket 绑定到指定的 IP 地址和端口号

函数原型：int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen)

参数	类型	说明
sockfd	int	socket 描述符
addr	const struct sockaddr*	要绑定的 IP(任意，内部自动绑定到 127.0.0.1 上) 和端口信息
addrlen	socklen_t	addr 结构体的长度

返回值：绑定成功返回 0，失败返回-1

### 3.5.3.5. connect

函数功能：客户端通过 TCP/UDP 套接字主动连接服务器

函数原型：int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen)

参数	类型	说明
sockfd	int	socket 描述符

参数	类型	说明
addr	const struct sockaddr*	指向服务器地址的结构体, 包含服务器 IP 和端口信息
addrlen	socklen_t	addr 结构体的长度
返回值: 连接成功返回 0, 失败返回-1		

### 3.5.3.6. sendto

函数功能: 通过 UDP 发送数据到指定目标地址

函数原型: int sendto(int sockfd, const void \*buf, size\_t len, int flags, const struct sockaddr \*to, socklen\_t tolen)

参数	类型	说明
sockfd	int	socket 描述符
buf	const void*	待发送数据的缓冲区
len	size_t	数据长度
flags	int	控制发送行为, 默认 0
to	const struct sockaddr *	目标地址
tolen	socklen_t	to 实际的长度
返回值: 发送成功返回已发送的数据长度, 失败返回-1		

### 3.5.3.7. send

函数功能: 通过 TCP 发送数据到指定目标地址

函数原型: int send(int sockfd, const void \*buf, size\_t len, int flags)

参数	类型	说明
----	----	----

参数	类型	说明
sockfd	int	socket 描述符
buf	const void*	待发送数据的缓冲区
len	size_t	数据长度
flags	int	控制发送行为，默认 0
to	const struct sockaddr *	目标地址
tolen	socklen_t	to 实际的长度

返回值：发送成功返回已发送的数据长度，失败返回-1

### 3.5.3.8. recvfrom

函数功能：通过 UDP 接收数据，并获取对方 IP

函数原型：int recvfrom(int sockfd, void \*buf, size\_t len, int flags, struct sockaddr \*from, socklen\_t \*fromlen)

参数	类型	说明
sockfd	int	socket 描述符
buf	void*	接收数据的缓冲区
len	size_t	缓冲区的最大长度
flags	int	控制接收行为，默认 0
from	struct sockaddr *	存储发送方地址
fromlen	socklen_t	输入时为 from 的缓冲区长度，输出时为实际地址长度



返回值：接收成功返回已接收的数据长度，失败返回-1

### 3.5.3.9. recv

函数功能：通过 TCP 接收数据

函数原型：int ql\_recv(int sockfd, void \*buf, size\_t len, int flags)

参数	类型	说明
sockfd	int	socket 描述符
buf	void*	接收数据的缓冲区
len	size_t	缓冲区的最大长度
flags	int	控制接收行为，默认 0

返回值：接收成功返回已接收的数据长度，失败返回-1

### 3.5.3.10. setsockopt

函数功能：设置套接字选项(目前仅支持设置超时时长)

函数原型：int setsockopt(int sockfd, int level, int optname, const void \*optval, socklen\_t optlen)

参数	类型	说明
sockfd	int	socket 描述符
level	int	选项定义的层次
optname	int	要设置的选项名称
optval	const void*	指向包含新选项值的缓冲区
optlen	socklen_t	optval 缓冲区的大小

返回值：设置成功返回 0，失败返回-1

### 3.5.3.11. listen

函数功能：将套接字设置为被动监听模式，等待客户端连接请求

函数原型：int listen(int sockfd, int backlog)

参数	类型	说明
sockfd	int	socket 描述符
backlog	int	等待连接队列的最大长度

返回值：监听成功返回 0，失败返回-1

### 3.5.3.12. accept

函数功能：接受一个客户端连接，并返回一个新的套接字描述符

函数原型：int accept(int sockfd, struct sockaddr \*name, socklen\_t \*namelen)

参数	类型	说明
sockfd	int	socket 描述符
name	struct sockaddr *	存储客户端地址信息
namelen	socklen_t*	输入时指定 name 缓冲区大小，返回时存储实际地址长度

返回值：成功返回非负整数值，失败返回-1

### 3.5.3.13. select

函数功能：监视多个文件描述符检测它们是否可读, 实现单线程处理多个 I/O 操作

函数原型：int select(int nfds, ql\_fd\_set \*readfds, ql\_fd\_set \*writefds, ql\_fd\_set \*exceptfds, struct timeval \*timeout)

参数	类型	说明
nfds	int	最大文件描述符值 +1

参数	类型	说明
readfds	ql_fd_set*	检查可读的文件描述符集合
writefds	ql_fd_set*	设为 NULL
exceptfds	ql_fd_set*	设为 NULL
timeout	struct timeval *	超时时间（NULL 表示阻塞，0 表示立即返回）

返回值：> 0 表示返回就绪的文件描述符数量， =0 表示超时， <0 表示出错

## 3.6. PSM 模块

### 3.6.1. 模块概述

本模块提供对通信模组电源节能模式 的管理接口，主要包括 PSM 模式的设置、休眠控制等功能。

### 3.6.2. 接口概述

类型	文件路径	说明
头文件	ql_psm.h	接口声明、结构体定义
源文件	ql_psm.c	接口实现代码
示例	example_psm.c	快速集成参考
依赖库	at_client.c	需提前初始化 AT 指令框架

### 3.6.3. 接口详细说明

#### 3.6.3.1. ql\_psm\_settings\_write

函数功能：psm 参数设置

函数原型：int ql\_psm\_settings\_write(at\_client\_t client, ql\_psm\_setting\_s settings)

参数	类型	说明
client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例
settings	ql_psm_setting_s	psm 要设置的参数内容, 结构参见头文件

返回值: 配置成功返回 0, 失败返回-1

### 3.6.3.2. ql\_psm\_settings\_read

函数功能: psm 参数读取

函数原型: int ql\_psm\_settings\_read(at\_client\_t client, ql\_psm\_setting\_s \*settings)

参数	类型	说明
client	at_client_t	AT 指令客户端句柄。单模组时传入 at_client_get_first(), 多模组需指定实例
settings	ql_psm_setting_s*	指向 ql_psm_setting_s 结构体的指针, 用于保存读取到的 PSM 配置参数

返回值: 读取成功返回 0, 失败返回-1



