

ECx00U&EGx00U 系列

QuecOpen Wi-Fi Scan API

参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-08-31

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	Jensen FANG	文档创建
1.0	2021-08-31	Jensen FANG/ Terry TAO	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 Wi-Fi Scan 相关 API.....	7
2.1. 头文件	7
2.2. 函数概览	7
2.3. 函数详解	8
2.3.1. ql_wifiscan_open.....	8
2.3.1.1. ql_errcode_wifi_scan_e.....	8
2.3.2. ql_wifiscan_close	9
2.3.3. ql_wifiscan_option_set.....	9
2.3.3.1. ql_wifiscan_channel_e	10
2.3.4. ql_wifiscan_do.....	11
2.3.4.1. ql_wifi_ap_info_s	12
2.3.4.2. ql_wifiscan_result_s	12
2.3.5. ql_wifiscan_async	13
2.3.6. ql_wifiscan_register_cb.....	13
2.3.6.1. ql_wifiscan_callback	14
2.3.6.2. ql_wifiscan_ind_msg_s.....	14
3 Wi-Fi Scan 开发示例	15
3.1. Wi-Fi Scan 示例说明.....	15
3.2. Wi-Fi Scan 示例自启动	16
4 附录 参考文档及术语缩写	17

表格索引

表 1: 适用模块 6

表 2: 函数概览 7

表 3: 参考文档 17

表 4: 术语缩写 17

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]。

本文档主要介绍移远通信 ECx00U 系列和 EGx00U QuecOpen®模块 Wi-Fi Scan 相关 API 及开发示例。

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 Wi-Fi Scan 相关 API

2.1. 头文件

Wi-Fi Scan API 接口的头文件为 `ql_wifi_scan.h`，位于 SDK 包的 `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_wifiscan_open()</code>	启用 Wi-Fi Scan
<code>ql_wifiscan_close()</code>	禁用 Wi-Fi Scan
<code>ql_wifiscan_option_set()</code>	配置 Wi-Fi Scan 扫描参数
<code>ql_wifiscan_do()</code>	进行 Wi-Fi Scan 同步模式扫描
<code>ql_wifiscan_register_cb()</code>	注册回调函数
<code>ql_wifiscan_async()</code>	开始 Wi-Fi Scan 异步模式扫描

备注

若无特别说明，本文档所述的 Wi-Fi Scan API 接口函数均不支持并发调用，并且不可在任何回调函数中调用所述函数，否则会对后续消息的处理造成影响。

2.3. 函数详解

2.3.1. ql_wifiscan_open

该函数用于启用 Wi-Fi Scan。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_open (void);
```

- 参数

无

- 返回值

错误码详见第 2.3.1.1 章。

2.3.1.1. ql_errcode_wifi_scan_e

Wi-Fi Scan 错误码定义枚举如下：

```
typedef enum
{
    QL_WIFISCAN_SUCCESS                = 0,
    QL_WIFISCAN_EXECUTE_ERR            = 1 | QL_DEV_ERRCODE_BASE,
    QL_WIFISCAN_MEM_ADDR_NULL_ERR,
    QL_WIFISCAN_INVALID_PARAM_ERR,
    QL_WIFISCAN_SEMAPHORE_WAIT_ERR,
    QL_WIFISCAN_MUTEX_TIMEOUT_ERR,
    QL_WIFISCAN_OPEN_FAIL,
    QL_WIFISCAN_BUSY_ERR,
    QL_WIFISCAN_ALREADY_OPEN_ERR,
    QL_WIFISCAN_NOT_OPEN_ERR,
    QL_WIFISCAN_HW_OCCUPIED_ERR,
}ql_errcode_wifi_scan_e;
```

- 参数

参数	描述
QL_WIFISCAN_SUCCESS	执行成功
QL_WIFISCAN_EXECUTE_ERR	执行失败

QL_WIFISCAN_MEM_ADDR_NULL_ERR	内存申请失败
QL_WIFISCAN_INVALID_PARAM_ERR	无效参数
QL_WIFISCAN_SEMAPHORE_WAIT_ERR	等待信号量异常
QL_WIFISCAN_MUTEX_TIMEOUT_ERR	获取 Mutex 异常
QL_WIFISCAN_OPEN_FAIL	Wi-Fi Scan 启用异常
QL_WIFISCAN_ALREADY_OPEN_ERR	Wi-Fi Scan 尝试重复启用错误
QL_WIFISCAN_NOT_OPEN_ERR	Wi-Fi Scan 未启用
QL_WIFISCAN_BUSY_ERR	Wi-Fi Scan 忙碌，如正在进行扫描
QL_WIFISCAN_HW_OCCUPIED_ERR	硬件被占用

2.3.2. ql_wifiscan_close

该函数用于禁用 Wi-Fi Scan。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_close (void);
```

- 参数

无

- 返回值

错误码详见第 2.3.1.1 章。

2.3.3. ql_wifiscan_option_set

该函数用于配置 Wi-Fi Scan 扫描参数。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_option_set (uint16_t expect_ap_cnt, uint8_t round, uint16_t scan_channel, uint32_t channel_time);
```

- 参数

expect_ap_cnt:

[In] 可扫描的最大 AP 数量。范围：1~300。

round:

[In] 扫描循环次数。

scan_channel:

[In] 扫描的通道。bit0 到 bit12 的每个 bit 代表一个通道（bit0 代表通道 1，bit12 代表通道 13）；某个 bit 值为 1 表示需要扫描这个通道。若 bit0 到 bit12 全部设置为 0 则表示扫描所有通道。通道信息详见第 2.3.3.1 章。

channel_time:

[In] 每个通道每个循环最长扫描时间。

- 返回值

错误码详见第 2.3.1.1 章。

2.3.3.1. ql_wifiscan_channel_e

Wi-Fi Scan 通道相关枚举类型定义如下：

```
typedef enum
{
    QL_WIFISCAN_CHANNEL_ALL_BIT    = 0x1FFF,
    QL_WIFISCAN_CHANNEL_ALL        = 0,
    QL_WIFISCAN_CHANNEL_ONE        = 0x0001,
    QL_WIFISCAN_CHANNEL_TWO        = 0x0002,
    QL_WIFISCAN_CHANNEL_THREE      = 0x0004,
    QL_WIFISCAN_CHANNEL_FOUR       = 0x0008,
    QL_WIFISCAN_CHANNEL_FIVE       = 0x0010,
    QL_WIFISCAN_CHANNEL_SIX        = 0x0020,
    QL_WIFISCAN_CHANNEL_SEVEN      = 0x0040,
    QL_WIFISCAN_CHANNEL_EIGHT      = 0x0080,
    QL_WIFISCAN_CHANNEL_NINE       = 0x0100,
    QL_WIFISCAN_CHANNEL_TEN        = 0x0200,
    QL_WIFISCAN_CHANNEL_ELEVEN     = 0x0400,
    QL_WIFISCAN_CHANNEL_TWELVE     = 0x0800,
    QL_WIFISCAN_CHANNEL_THIRTEEN   = 0x1000,
    QL_WIFISCAN_CHANNEL_TOTAL      = 13,
    QL_WIFISCAN_CHANNEL_TOTAL      = QL_WIFISCAN_CHANNEL_THIRTEEN,
    QL_WIFISCAN_CHANNEL_MAX,
```

```
}ql_wifiscan_channel_e;
```

- 参数

参数	描述
QL_WIFISCAN_CHANNEL_ALL_BIT	按 bit 设置所有通道
QL_WIFISCAN_CHANNEL_ALL	所有通道
QL_WIFISCAN_CHANNEL_ONE	通道 1
QL_WIFISCAN_CHANNEL_TWO	通道 2
QL_WIFISCAN_CHANNEL_THREE	通道 3
QL_WIFISCAN_CHANNEL_FOUR	通道 4
QL_WIFISCAN_CHANNEL_FIVE	通道 5
QL_WIFISCAN_CHANNEL_SIX	通道 6
QL_WIFISCAN_CHANNEL_SEVEN	通道 7
QL_WIFISCAN_CHANNEL_EIGHT	通道 8
QL_WIFISCAN_CHANNEL_NINE	通道 9
QL_WIFISCAN_CHANNEL_TEN	通道 10
QL_WIFISCAN_CHANNEL_ELEVEN	通道 11
QL_WIFISCAN_CHANNEL_TWELVE	通道 12
QL_WIFISCAN_CHANNEL_THIRTEEN	通道 13
QL_WIFISCAN_CHANNEL_TOTAL	总通道数
QL_WIFISCAN_CHANNEL_MAX	无效通道

2.3.4. ql_wifiscan_do

该函数用于进行 Wi-Fi Scan 同步模式扫描。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_do (uint16_t *p_ap_cnt, ql_wifi_ap_info_s *p_ap_infos);
```

- 参数

p_ap_cnt:
[Out] 扫描到的 AP 数量。

p_ap_infos:
[Out] 扫描到的所有 AP 的信息。详见第 2.3.4.1 章。

- 返回值

错误码详见第 2.3.1.1 章。

2.3.4.1. ql_wifi_ap_info_s

扫描的 Wi-Fi AP 具体信息结构体定义如下：

```
typedef struct
{
    uint32_t bssid_low;
    uint16_t bssid_high;
    uint8_t channel;
    int8_t rssival;
}ql_wifi_ap_info_s;
```

- 参数

类型	参数	描述
uint32_t	<i>bssid_low</i>	Wi-Fi AP MAC 地址的低 4 位
uint16_t	<i>bssid_high</i>	Wi-Fi AP MAC 地址的高 2 位
uint8_t	<i>channel</i>	所在通道
int8_t	<i>rssival</i>	信号强度

2.3.4.2. ql_wifiscan_result_s

存储每一次扫描到的所有 Wi-Fi AP 结果的枚举定义如下：

```
typedef struct
{
    uint32_t ap_cnt;
    ql_wifi_ap_info_s *ap_infos;
```

```
}ql_wifiscan_result_s;
```

- 参数

类型	参数	描述
uint32_t	ap_cnt	扫描到的 Wi-Fi AP 的数量
ql_wifi_ap_info_s	ap_infos	扫描到的 Wi-Fi AP 具体信息所在内存地址

2.3.5. ql_wifiscan_async

该函数用于开始 Wi-Fi Scan 异步模式扫描。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_async (void);
```

- 参数

无

- 返回值

错误码详见第 2.3.1.1 章。

2.3.6. ql_wifiscan_register_cb

该函数用于注册回调函数。

- 函数原型

```
ql_errcode_wifi_scan_e ql_wifiscan_register_cb (ql_wifiscan_callback wifiscan_cb);
```

- 参数

wifiscan_cb:

[In] 回调函数指针。

- 返回值

错误码详见第 2.3.1.1 章。

2.3.6.1. ql_wifiscan_callback

Wi-Fi Scan 回调函数指针定义如下：

```
typedef void (*ql_wifiscan_callback)(ql_wifiscan_ind_msg_s *msg_buf);
```

2.3.6.2. ql_wifiscan_ind_msg_s

内核通过回调函数通知 APP 侧的信息结构体定义如下：

```
typedef struct
{
    uint32_t msg_id;
    uint32_t msg_err_code;
    void *msg_data;
}ql_wifiscan_ind_msg_s;
```

● 参数

类型	变量成员	描述
uint32_t	<i>msg_id</i>	信息 ID
uint32_t	<i>msg_err_code</i>	返回的错误码
void	<i>msg_data</i>	通知的具体数据

3 Wi-Fi Scan 开发示例

3.1. Wi-Fi Scan 示例说明

ECx00U 系列和 EGx00U QuecOpen SDK 代码中提供了 Wi-Fi Scan 的示例，即 *wifi_scan_demo.c* 文件，位于 `\components\ql-application\wifi_scan` 目录下。入口函数为 *ql_wifiscan_app_init()*，如下图所示。

```
void ql_wifiscan_app_init(void)
{
    QlOsStatus err = Ql_OSI_SUCCESS;
    err = ql_rtos_task_create(&ql_wifiscan_app_task, Ql_WIFISCAN_TASK_STACK_SIZE, Ql_WIFISCAN_TASK_PRIO, "wifiscan DEMO", ql_wifiscan_app_thread, NULL, Ql_WIFISCAN_TASK_EVENT_CNT);
    if(err != Ql_OSI_SUCCESS)
    {
        Ql_WIFISCAN_DEMO_LOG("demo task created failed");
    }
}
```

图 1: ql_wifiscan_app_init()

Wi-Fi Scan 示例中，首先同步扫描 AP 热点信息并打印，然后再异步扫描 AP 热点信息并打印。最后进入循环扫描状态，扫描事件到来时会重复上述步骤。

```
static void ql_wifiscan_app_thread(void *param)
{
    ...ql_rtos_task_sleep_ms(15000); // sleep some time, then to do demo
    ...ql_wifiscan_synchro_complete_flow();
    ...ql_rtos_task_sleep_ms(5000);
    ...ql_wifiscan_async_start(); // start a async scan, and it should trigger one event sent at callback
    ...
    while(1)
    {
        ql_event_t event = {0};
        if(ql_event_try_wait(&event) != 0)
        {
            continue;
        }

        if(event.id == QUEC_WIFISCAN_EVENT_ASYNC_IND)
        {
            ql_wifiscan_close(); // close the wifiscan device for the async scan
            ql_rtos_task_sleep_ms(15000);

            // next cycle demo
            ql_wifiscan_synchro_complete_flow();
            ql_rtos_task_sleep_ms(5000);
            ql_wifiscan_async_start();
        }
    }
} // end ql_wifiscan_app_thread
```

图 2: ql_wifiscan_app_thread()

3.2. Wi-Fi Scan 示例自启动

Wi-Fi Scan 示例默认不启动，如下图所示，测试时请在 `ql_init_demo_thread` 线程中取消注释。

```
#endif
#ifdef QL_APP_FEATURE_VOLTE
    >> //ql_volte_app_init();
#endif

#ifdef QL_APP_FEATURE_GNSS
    ....//ql_gnss_app_init();
#endif

#ifdef QL_APP_FEATURE_SPI
    ....//ql_spi_demo_init();
#endif

#ifdef QL_APP_FEATURE_CAMERA
    ..../*因为camera被i2c用来演示功能，所以打开camera_demo时不可同时打
    ..../*because camera as a demonstration for i2c, so camera_demo.c:
    ....//ql_camera_app_init();
#endif

#ifdef QL_APP_FEATURE_SPI_NOR_FLASH
    ....//ql_spi_nor_flash_demo_init();
#endif

#ifdef QL_APP_FEATURE_SPI_NAND_FLASH
    ....//ql_spi_nand_flash_demo_init();
#endif

#ifdef QL_APP_FEATURE_WIFISCAN
    ....//ql_wifiscan_app_init();
#endif

#ifdef QL_APP_FEATURE_HTTP_FOTA
    >> //ql_fota_http_app_init();
#endif

#ifdef QL_APP_FEATURE_FTP_FOTA
    ....//ql_fota_ftp_app_init();
#endif

#ifdef QL_APP_FEATURE_DECODER
```

图 3: ql_init_demo_thread()

4 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

表 4: 术语缩写

术语	英文全称	中文全称
API	Application Programming Interface	应用程序编程接口
AP	Access Point	无线接入点
APP	Application	应用程序
MAC	Media Access Control Address	媒体访问控制地址（物理地址）