ECx00U&EGx00U 系列 QuecOpen 语音通话 API 参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-09-07

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。

文档历史

修订记录

版本	日期	作者	变更表述
-	2020-11-12	Marvin NING	文档创建
1.0	2021-09-07	Marvin NING	受控版本

目录

表格索引

1 引言

移远通信 LTE Standard ECx00U 系列和 EGx00U 模块支持 QuecOpen®方案；QuecOpen®是开源的基 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®模块的语音通话相关 API 及语音通话示例。

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 语音通话 API 介绍

2.1. 头文件

ql_api_voice_call.h 文件即为语音通话 API 的头文件，位于 SDK 包的*\components\ql-kernel\inc* 目录下；若无特别说明，本文档所提及的头文件均位于该目录下。

2.2. 函数概览

表 2：函数概览

函数	说明
ql_voice_auto_answer()	设置自动接听语音电话
ql_voice_call_start()	拨打语音电话
ql_voice_call_answer()	接听语音电话
ql_voice_call_end()	挂断语音电话
ql_voice_call_start_dtmf()	发送双音多频信号
ql_voice_call_wait_set()	设置呼叫等待操作
ql_voice_call_wait_get()	获取呼叫等待状态
ql_voice_call_fw()	设置或查询呼叫转移补充业务
ql_voice_call_hold()	处理呼叫保持、多方通话等功能
ql_voice_call_clcc()	获取当前所有通话的详细信息
ql_voice_call_callback_register()	设置用户回调函数

2.3. 函数描述

2.3.1. ql_voice_auto_answer

该函数用于设置自动接听语音电话。

- **函数原型**

```
ql_vc_errcode_e ql_voice_auto_answer(uint8_t nSim, uint8_t times);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

times：
[In] 设置自动接听前的响铃次数。0 表示关闭自动接听功能。默认值：0。

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.1.1. ql_vc_errcode_e

语音通话错误码枚举定义如下：

```
typedef enum
{
    QL_VC_SUCCESS                   = 0,
    QL_VC_ERROR                     = 1 | (QL_COMPONENT_VOICE_CALL << 16),
    QL_VC_NOT_INIT_ERR,
    QL_VC_PARA_ERR,
    QL_VC_NO_MEMORY_ERR,
    QL_VC_NO_CALL_ERR,
    QL_VC_SEM_CREATE_ERR,
    QL_VC_SEM_TIMEOUT_ERR,
}ql_vc_errcode_e;
```

- **参数**

参数	描述
QL_VC_SUCCESS	函数执行成功
QL_VC_ERROR	函数执行失败
QL_VC_NOT_INIT_ERR	语音拨号业务未初始化
QL_VC_PARA_ERR	参数错误
QL_VC_NO_MEMORY_ERR	申请内存未成功
QL_VC_NO_CALL_ERR	当前无任何通话
QL_VC_SEM_CREATE_ERR	信号量创建失败
QL_VC_SEM_TIMEOUT_ERR	信号量超时

2.3.2. ql_voice_call_start

该函数用于拨打语音电话。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_start(uint8_t nSim, char* dial_num);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

dial_num：
[In] 电话号码。

- **返回值**

错误码详见***第 2.3.1.1 章***。

备注

该接口为异步接口，返回 *QL_VC_SUCCESS* 仅表示调用该接口成功，最终将通过 *ql_voice_call_callback_register()*注册的回调函数通知拨打语音电话是否成功。

2.3.3. ql_voice_call_answer

该函数用于接听语音电话。该函数不支持自动接听语音电话。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_answer(uint8_t nSim);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.4. ql_voice_call_end

该函数用于挂断语音电话。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_end(uint8_t nSim);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.5. ql_voice_call_start_dtmf

该函数用于发送双音多频信号。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_start_dtmf(uint8_t nSim, char *dtmf, uint16_t duration);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

dtmf：
[In] DTMF 字符串。最大字符数：32 个。有效字符数包括：0~9、A、B、C、D、*、#。

duration：
[In] 持续时间。范围：1~10；默认值：1；单位：100 毫秒。

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.6. ql_voice_call_wait_set

该函数用于设置呼叫等待操作。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_wait_set(uint8_t nSim, uint8_t mode);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

mode：
[In] 呼叫等待操作。
0 关闭呼叫等待
1 打开呼叫等待

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.7. ql_voice_call_wait_get

该函数用于获取呼叫等待状态。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_wait_get(uint8_t nSim, uint8_t   *mode);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

mode：
[Out] 获取呼叫等待操作。
0 呼叫等待已关闭
1 呼叫等待已打开

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.8. ql_voice_call_fw

该函数用于设置或查询呼叫转移补充业务。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_fw(uint8_t nSim, int reason, int fwmode, char* phone_num);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

reason：
[In] 呼叫转移的条件或原因。
0 无条件转移
1 占线通话转移
2 无应答转移
3 无法接通转移

4 所有呼叫转移
5 所有条件呼叫转移

fwmode：
[In] 对呼叫转移的操作。
0 禁用
1 启用
2 查询状态
3 注册
4 擦除

phone_num：
[In] 呼叫转移的目标电话。

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.9. ql_voice_call_hold

该函数用于处理呼叫保持、多方通话等功能。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_hold(uint8_t nSim, uint8_t n);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

n：
[In] 对多方通话的控制。
0 释放所有保持或等待的电话
1 释放正在通话的电话，接听保持或等待的电话
1X 释放 X 序号的电话，可以是正在通话的、保持的、等待的电话
2 将所有正在通话的电话设置成保持，并接受保持或者等待的电话
2X 请求接受 X 序号的电话，保持其他电话

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.10. ql_voice_call_clcc

该函数用于获取当前所有通话的详细信息。

- **函数原型**

```
ql_vc_errcode_e ql_voice_call_clcc(uint8_t nSim, uint8_t *total, ql_vc_info_s vc_info[QL_VC_MAX_NUM]);
```

- **参数**

nSim：
[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

total：
[Out] 当前电话总数。

vc_info：
[Out] 所有通话的详细信息，包括序号、呼叫状态、号码等。详请请参考***第 2.3.10.1 章***。

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.10.1.ql_vc_info_s

通话的详细信息结构体定义如下：

```
typedef struct
 {
    uint8_t idx;
    uint8_t direction;
    uint8_t status;
    uint8_t multiparty;
    char number[23];
 }ql_vc_info_s;
```

- **参数**

参数	参数	描述
uint8_t	*idx*	序号，范围：1~7

uint8_t	*direction*	呼叫方 0 主叫 1 被叫
uint8_t	*status*	呼叫状态 0 通话中 1 呼叫保持 2 拨号中 3 响铃中 4 来电 5 呼叫等待中 7 通话断开 8 握手
uint8_t	*multiparty*	0 未在多方通话里 1 在多方通话里
char	*number*	电话号码

2.3.11. ql_voice_call_callback_register

该函数用于设置用户回调函数，以处理语音通话相关事件。

- **函数原型**

```
void ql_voice_call_callback_register(ql_vc_event_handler_t cb);
```

- **参数**

cb：
[In] 用户回调函数。详见***第 2.3.11.1 章***。

- **返回值**

错误码详见***第 2.3.1.1 章***。

2.3.11.1. ql_vc_event_handler_t

回调函数定义如下：

```
typedef void (*ql_vc_event_handler_t)(uint8_t sim, ql_vc_event_id_e event_id, void *ctx);
```

- **参数**

sim：

[In] (U)SIM 卡索引。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

event_id：
[In] 语音通话事件。详见***第 2.3.11.2 章***。

ctx：
[In] 回调函数的传参指针。

- **返回值**

无

2.3.11.2. ql_vc_event_id_e

语音通话事件枚举定义如下：

```
typedef enum
 {
      QL_VC_INIT_OK_IND =          1| (QL_COMPONENT_VOICE_CALL << 16),
      QL_VC_RING_IND,
      QL_VC_CONNECT_IND,
      QL_VC_NOCARRIER_IND,
      QL_VC_ERROR_IND,
      QL_VC_CCFC_IND,
      QL_VC_CCWA_IND,
 }ql_vc_event_id_e;
```

- **参数**

参数	描述
QL_VC_INIT_OK_IND	暂未支持
QL_VC_RING_IND	有新来电
QL_VC_CONNECT_IND	建立语音通话连接
QL_VC_NOCARRIER_IND	断开语音通话连接
QL_VC_ERROR_IND	未知错误
QL_VC_CCFC_IND	呼叫转移通知
QL_VC_CCWA_IND	呼叫等待通知

3 语音通话开发示例

本章节主要介绍在 App 侧如何使用上述 API 进行语音通话开发以及简单的调试。

3.1. 开发示例

ECx00U 系列和 EGx00U QuecOpen SDK 代码中提供了语音通话的示例，即 *voice_call_demo.c* 文件，该文件位于*\components\ql-application\voice_call* 目录下。该文件包含了拨打电话、接听电话、呼叫等待、呼叫转移等功能相关示例。入口函数为 *ql_voice_call_app_init()*，该函数启动 *voice_call_demo_task* 任务，如下图所示。

```
QlOSStatus ql_voice_call_app_init(void)
{
    QlOSStatus err = QL_OSI_SUCCESS;

    err = ql_rtos_task_create(&vc_task, 4096, APP_PRIORITY_NORMAL, "vc_task", voice_call_demo_task, NULL, 10);
    if(err != QL_OSI_SUCCESS)
    {
        QL_VC_LOG("voice_call_demo_task created failed, ret = 0x%x", err);
    }

    return err;
}
```

*ql_voice_call_app_init()*默认不会自动调用。需要调用时，取消其注释即可，如下图所示。

```
#ifdef QL_APP_FEATURE_VOICE_CALL
    ql_voice_call_app_init();
#endif
```

3.2. 功能调试

ECx00U 系列和 EGx00U QuecOpen 模块可通过使用移远通信 LTE OPEN EVB 进行语音通话功能调试。

将编译版本烧录到模块中，使用 USB 线连接 EVB 的 USB 端口和 PC，USB 的 AP Log 端口主要用于显示系统调试信息，通过 cooltools 工具搜索“ql_vc”可以过滤出关键 log，log 抓取方法请参考***文档 [2]***。

voice_call_demo_task 任务启动后将注册回调函数，并等待电话呼入。收到电话呼入事件时，等待 10 秒后，模块会自动接听，并在 10 秒后自动挂断，如下图所示。

```
QL_VC_LOG("wait call");
while(1){
    ql_event_wait(&vc_event, QL_WAIT_FOREVER);
    switch(vc_event.id)
    {
        case QL_VC_RING_IND:
            ql_rtos_task_sleep_s(10);//
            err = ql_voice_call_answer(nSim);
            if(err != QL_VC_SUCCESS){
                QL_VC_LOG("ql_voice_call_answer FAIL");
            }else{
                QL_VC_LOG("ql_voice_call_answer OK");
            }
            break;
        case QL_VC_CONNECT_IND:
            QL_VC_LOG("CONNECT");
            ql_rtos_task_sleep_s(10);
            err = ql_voice_call_end(nSim);
            if(err != QL_VC_SUCCESS){
                QL_VC_LOG("ql_voice_call_end FAIL");
            }else{
                QL_VC_LOG("ql_voice_call_end OK");
            }
            break;
        case QL_VC_NOCARRIER_IND:
            QL_VC_LOG("NOCARRIER");
            break;
        default:
            QL_VC_LOG("event id = 0x%x",vc_event.id);
            break;
    } « end switch vc_event.id »
} « end while 1 »
```

备注

如需测试语音通话的其他功能，如主动拨打电话、设置呼叫等待、呼叫转移等，可在该示例中自行打开对应宏定义进行测试。

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
[2] Quectel_ECx00U&EGx00U 系列_QuecOpen_Log_抓取指导

表 4：术语缩写

缩写	英文全称	中文全称
AP	Application Processor	应用处理器
API	Application Programming Interface	应用程序接口
App	Application	应用程序
DTMF	Dual Tone Multi Frequency	双音多频
EVB	Evaluation Board	评估板
IoT	Internet of Things	物联网
PC	Personal Computer	个人电脑
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	（全球）用户识别模块
USB	Universal Serial Bus	通用串行总线