

ECx00U&EGx00U 系列

QuecOpen 虚拟 AT 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。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
-	2021-08-22	Neo KONG	文档创建
1.0	2021-08-31	Neo KONG	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 虚拟 AT 相关 API	7
2.1. 头文件	7
2.2. 函数概览	7
2.3. 虚拟 AT 相关 API	7
2.3.1. ql_virt_at_open	7
2.3.1.1. ql_virt_at_port_number_e	8
2.3.1.2. ql_errcode_virt_at_e	9
2.3.2. ql_virt_at_close	9
2.3.3. ql_virt_at_write	10
2.3.4. ql_virt_at_read	10
3 虚拟 AT 开发示例	12
3.1. APP 的虚拟 AT 开发示例	12
3.1.1. 虚拟 AT Demo 说明	12
3.1.2. Demo 自启动	13
3.2. 虚拟 AT 相关调试	14
3.2.1. 调试准备	14
3.2.2. 虚拟 AT 功能调试	14
4 附录 参考文档及术语缩写	15

表格索引

表 1: 适用模块 6

表 2: 函数概览 7

表 3: 参考文档 15

表 4: 术语缩写 15

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 模块的在 APP 侧虚拟 AT 的应用指导，包括相关 API、开发示例及调试方法。

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 虚拟 AT 相关 API

2.1. 头文件

虚拟 AT 的 API 接口的头文件为 `ql_api.virt.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_virt_at_open()</code>	启用虚拟 AT 通道
<code>ql_virt_at_close()</code>	禁用虚拟 AT 通道
<code>ql_virt_at_write()</code>	通过虚拟 AT 通道写入 AT 命令
<code>ql_virt_at_read()</code>	通过虚拟 AT 通道读取 AT 命令响应

2.3. 虚拟 AT 相关 API

2.3.1. ql_virt_at_open

该函数用于启用虚拟 AT 通道。

- 函数原型

```
ql_errcode_virt_at_e ql_virt_at_open(ql_virt_at_port_number_e port, ql_virt_at_callback virt_at_cb)
```


● 参数

port:

[In] 虚拟 AT 通道。请参考第2.3.1.1章枚举 ql_virt_at_port_number_e。

virt_at_cb:

[In] 回调函数。用于接收虚拟 AT 通道返回 AT 命令响应的通知。

● 返回值

返回虚拟 AT 错误码。请参考第2.3.1.2章2.3.1.2 枚举 ql_errcode_virt_at_e。

2.3.1.1. ql_virt_at_port_number_e

虚拟 AT 通道号，枚举信息定义如下：

```
typedef enum
{
    QL_VIRT_AT_PORT_0,
    QL_VIRT_AT_PORT_1,
    QL_VIRT_AT_PORT_2,
    QL_VIRT_AT_PORT_3,
    QL_VIRT_AT_PORT_4,
    QL_VIRT_AT_PORT_5,
    QL_VIRT_AT_PORT_6,
    QL_VIRT_AT_PORT_7,
    QL_VIRT_AT_PORT_8,
    QL_VIRT_AT_PORT_9,
    QL_VIRT_AT_PORT_MAX,
}ql_virt_at_port_number_e;
```

● 枚举

枚举	描述
QL_VIRT_AT_PORT_0	虚拟 AT 通道号为 0
QL_VIRT_AT_PORT_1	虚拟 AT 通道号为 1
QL_VIRT_AT_PORT_2	虚拟 AT 通道号为 2
QL_VIRT_AT_PORT_3	虚拟 AT 通道号为 3
QL_VIRT_AT_PORT_4	虚拟 AT 通道号为 4
QL_VIRT_AT_PORT_5	虚拟 AT 通道号为 5

QL_VIRT_AT_PORT_6	虚拟 AT 通道号为 6
QL_VIRT_AT_PORT_7	虚拟 AT 通道号为 7
QL_VIRT_AT_PORT_8	虚拟 AT 通道号为 8
QL_VIRT_AT_PORT_9	虚拟 AT 通道号为 9

2.3.1.2. ql_errcode_virt_at_e

虚拟 AT 错误码表示函数是否执行成功，若失败则返回错误原因，枚举信息定义如下：

```
typedef enum
{
    QL_VIRT_AT_SUCCESS = QL_SUCCESS,
    QL_VIRT_AT_INVALID_PARAM_ERR    = 1|QL_VIRT_AT_ERRCODE_BASE,
    QL_VIRT_AT_EXECUTE_ERR,
    QL_VIRT_AT_OPEN_REPEAT_ERR
}ql_errcode_virt_at_e;
```

- 枚举

枚举	描述
QL_VIRT_AT_SUCCESS	函数执行成功
QL_VIRT_AT_INVALID_PARAM_ERR	无效参数
QL_VIRT_AT_EXECUTE_ERR	函数执行错误
QL_VIRT_AT_OPEN_REPEAT_ERR	重复启用虚拟 AT 通道错误

2.3.2. ql_virt_at_close

该函数用于禁用虚拟 AT 通道。

- 函数原型

```
ql_errcode_virt_at_e ql_virt_at_close(ql_virt_at_port_number_e port)
```

- 参数

port:

[In] 虚拟 AT 通道。请参考第 2.3.1.1 章枚举 ql_virt_at_port_number_e。

- 返回值

返回虚拟 AT 错误码。请参考第 2.3.1.2 章 2.3.1.2 枚举 `ql_errcode_virt_at_e`。

2.3.3. ql_virt_at_write

该函数用于通过虚拟 AT 通道写入 AT 命令。

- 函数原型

```
ql_errcode_virt_at_e ql_virt_at_write (ql_virt_at_port_number_e port, unsigned char *data, unsigned int data_len)
```

- 参数

port:

[In] 虚拟 AT 通道。请参考第 2.3.1.1 章枚举 `ql_virt_at_port_number_e`。

data:

[In] 待写入的 AT 命令字符串指针。

data_len:

[In] 待写入的 AT 命令字符串长度。单位：字节。

- 返回值

返回虚拟 AT 错误码。请参考第 2.3.1.2 章 2.3.1.2 枚举 `ql_errcode_virt_at_e`。

2.3.4. ql_virt_at_read

该函数用于通过虚拟 AT 通道读取 AT 命令响应。

- 函数原型

```
ql_errcode_virt_at_e ql_virt_at_read(ql_virt_at_port_number_e port, unsigned char *data, unsigned int data_len)
```

- 参数

port:

[In] 虚拟 AT 通道。请参考第 2.3.1.1 章枚举 `ql_virt_at_port_number_e`。

data:

[Out] 读取的 AT 命令响应字符串指针。

data_len:

[In] 读取的 AT 命令响应字符串长度。单位：字节。

- 返回值

返回虚拟 AT 错误码。请参考第 2.3.1.2 章 2.3.1.2 枚举 *ql_errcode_virt_at_e*。

3 虚拟 AT 开发示例

本章节主要介绍在 APP 侧如何使用上述 API 进行虚拟 AT 开发以及简单的调试。

3.1. APP 的虚拟 AT 开发示例

3.1.1. 虚拟 AT Demo 说明

移远通信 ECx00U 系列和 EGx00U QuecOpen SDK 代码中提供了虚拟 AT 的示例，即 *virt_at_demo.c* 文件。

virt_at_demo.c 文件中主要包含启用虚拟 AT 通道、通过虚拟 AT 通道写入 AT 命令和读取 AT 命令响应等设置。入口函数为 *ql_virt_at_app_init()*，如下图所示。

```
void ql_virt_at_app_init()
{
    QIOSStatus err = QL_OSI_SUCCESS;
    ql_task_t virt_at_task = NULL;

    err = ql_rtos_task_create(&virt_at_task, 1024, APP_PRIORITY_NORMAL, "ql_virtatdemo", ql_virt_at_demo_thread, NULL, 1);
    if( err != QL_OSI_SUCCESS )
    {
        QL_VIRTAT_DEMO_LOG("virt at demo task created failed");
        return;
    }
}
```

该 Demo 首先启用 3 个虚拟 AT 通道，注册对应的回调函数，然后循环通过 3 个虚拟 AT 通道写入 AT 命令，通过回调函数中读取 AT 命令响应。

```
static void ql_virt_at_demo_thread(void *param)
{
    int ret = 0;
    QLOSStatus err = 0;

    char *cmd0 = "ATI\r\n";
    char *cmd1 = "AT^HEAPINFO\r\n";
    char *cmd2 = "at+qdbgcfg=\"oem\"\r\n";

    ret = ql_virt_at_open(QL_VIRT_AT_PORT_0, ql_virt_at0_notify_cb);
    if(QL_VIRT_AT_SUCCESS != ret)
    {
        QL_VIRTAT_DEMO_LOG("virt at0 open error,ret: 0x%x", ret);
        goto |exit;
    }
    QL_VIRTAT_DEMO_LOG("virt at0 open success");

    ret = ql_virt_at_open(QL_VIRT_AT_PORT_1, ql_virt_at1_notify_cb);
    if(QL_VIRT_AT_SUCCESS != ret)
    {
        QL_VIRTAT_DEMO_LOG("virt at1 open error,ret: 0x%x", ret);
        goto |exit;
    }
    QL_VIRTAT_DEMO_LOG("virt at1 open success");

    ret = ql_virt_at_open(QL_VIRT_AT_PORT_2, ql_virt_at2_notify_cb);
    if(QL_VIRT_AT_SUCCESS != ret)
    {
        QL_VIRTAT_DEMO_LOG("virt at2 open error,ret: 0x%x", ret);
        goto |exit;
    }
    QL_VIRTAT_DEMO_LOG("virt at2 open success");

    while(1)
    {
        QL_VIRTAT_DEMO_LOG("VAT0 -->: %s", cmd0);
        ql_virt_at_write(QL_VIRT_AT_PORT_0, (unsigned char*)cmd0, strlen((char *)cmd0));
        ql_rtos_task_sleep_ms(5000);

        QL_VIRTAT_DEMO_LOG("VAT1 -->: %s", cmd1);
        ql_virt_at_write(QL_VIRT_AT_PORT_1, (unsigned char*)cmd1, strlen((char *)cmd1));
        ql_rtos_task_sleep_ms(5000);

        QL_VIRTAT_DEMO_LOG("VAT2 -->: %s", cmd2);
        ql_virt_at_write(QL_VIRT_AT_PORT_2, (unsigned char*)cmd2, strlen((char *)cmd2));
        ql_rtos_task_sleep_ms(5000);
    }

    exit:
    err = ql_rtos_task_delete(NULL);
    if(err != QL_OSI_SUCCESS)
    {
        QL_VIRTAT_DEMO_LOG("task deleted failed");
    }
} « end ql_virt_at_demo_thread »
```

3.1.2. Demo 自启动

上述 Demo 已在 `ql_init_demo_thread` 线程中默认不启动，如有需要，取消 `ql_virt_at_app_init()` 的注释即可运行，如下图所示：

```
#ifndef QL_APP_FEATURE_HTTP_FOTA
>> //ql_fota_http_app_init();
#endif

#ifndef QL_APP_FEATURE_DECODER
//ql_decoder_app_init();
#endif

#ifndef QL_APP_FEATURE_KEYPAD
//ql_keypad_app_init();
#endif

#ifndef QL_APP_FEATURE_RTC
//ql_rtc_app_init();
#endif

#ifndef QL_APP_FEATURE_USB_CHARGE
//ql_charge_app_init();
#endif









#ifndef QL_APP_FEATURE_VIRT_AT
//ql_virt_at_app_init();
#endif
```

3.2. 虚拟 AT 相关调试

3.2.1. 调试准备

移远通信 ECx00U&EGx00U 系列 QuecOpen 模块可通过使用移远通信 LTE OPEN EVB 板进行虚拟 AT 功能调试。

编译版本烧录到模块中，使用 USB 线连接 LTE OPEN EVB 板的 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* 打印的 log 信息可以查看该 Demo 的相关信息，log 的抓取方法请参考文档 [2]。

-  Quectel Modem (COM8)
-  Quectel USB AP Log Port (COM14)
-  Quectel USB AT Port (COM5)
-  Quectel USB CP Log Port (COM11)
-  Quectel USB Diag Port (COM12)
-  Quectel USB MOS Port (COM13)
-  Quectel USB Serial-1 Port (COM9)
-  Quectel USB Serial-2 Port (COM10)

3.2.2. 虚拟 AT 功能调试

开机后自动启动 *ql_virt_at_app_init*，通过 log 信息可以看到 AT 命令响应。

AP log 端口调试信息如下图所示：

Index	Received	Tick	Level	Description
728	11:14:39.134	4365	QOPM/I	[q1_VIRTATDEMO][q1_virt_at0_notify_cb, 68] VATH <--: Quectel
734	11:14:39.134	4367	QOPM/I	[q1_VIRTATDEMO][q1_virt_at0_notify_cb, 68] VATH <--: OK
1586	11:14:44.055	20710	QOPM/I	[q1_VIRTATDEMO][q1_virt_at_demo_thread, 146] VAT1 -->: AT+HEAPINFO
1512	11:14:44.055	20722	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: AT+HEAPINFO
1519	11:14:44.055	20728	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: "HEAPINFO: 8x809ef100,6030024,3003112,3795232
1525	11:14:44.059	20730	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: OK
1600	11:14:45.000	37095	QOPM/I	[q1_VIRTATDEMO][q1_virt_at_demo_thread, 149] VAT2 -->: <+qdbgcfg="oem"
1667	11:14:45.063	37105	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: <+qdbgcfg="oem"
1679	11:14:45.063	37108	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: +QDBCFG: "oem","EC200U-CHAA QCPU RELEASE","MOSEM"
1679	11:14:45.063	37110	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: OK
1726	11:14:54.062	62477	QOPM/I	[q1_VIRTATDEMO][q1_virt_at_demo_thread, 141] VATH -->: ATZ
1742	11:14:54.062	62486	QOPM/I	[q1_VIRTATDEMO][q1_virt_at0_notify_cb, 62] VATH <--: ATZ
1748	11:14:54.062	62490	QOPM/I	[q1_VIRTATDEMO][q1_virt_at0_notify_cb, 62] VATH <--: Quectel
1754	11:14:54.062	62491	QOPM/I	[q1_VIRTATDEMO][q1_virt_at0_notify_cb, 62] VATH <--: OK
1809	11:14:55.064	4325	QOPM/I	[q1_VIRTATDEMO][q1_virt_at_demo_thread, 145] VAT1 -->: AT+HEAPINFO
1830	11:14:55.064	4334	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: AT+HEAPINFO
1902	11:14:55.064	4339	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: "HEAPINFO: 8x809ef100,6030024,3003112,3795232
1908	11:14:55.064	4341	QOPM/I	[q1_VIRTATDEMO][q1_virt_at1_notify_cb, 81] VAT1 <--: OK
2051	11:15:04.075	20700	QOPM/I	[q1_VIRTATDEMO][q1_virt_at_demo_thread, 140] VAT2 -->: <+qdbgcfg="oem"
2058	11:15:04.076	20710	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: <+qdbgcfg="oem"
2094	11:15:04.076	20723	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: +QDBCFG: "oem","EC200U-CHAA QCPU RELEASE","MOSEM"
2100	11:15:04.076	20725	QOPM/I	[q1_VIRTATDEMO][q1_virt_at2_notify_cb, 99] VAT2 <--: OK

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_快速开发指导
[2] Quectel_ECx00U&EGx00U 系列_QuecOpen_Log_抓取操作指导

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	应用程序编程接口
APP	Application	应用程序
COM	Communication	通信
EVB	Evaluation Board	评估板
IoT	Internet of Things	物联网
RTOS	Real-Time Operating System	实时操作系统
PC	Personal Computer	个人计算机
SDK	Software Development Kit	软件开发工具包
USB	Universal Serial Bus	通用串行总线