ECx00U&EGx00U 系列

QuecOpen 数据拨号 API 参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-09-16

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
-	2020-10-28	Braden HE	文档创建
1.0	2021-09-16	Braden HE	受控版本

目录

表格索引

图片索引

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]***。

本文档主要介绍在 QuecOpen®方案下，ECx00U 系列和 EGx00U 模块的数据拨号功能，包括函数调用流程和函数详解。

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 API 调用流程

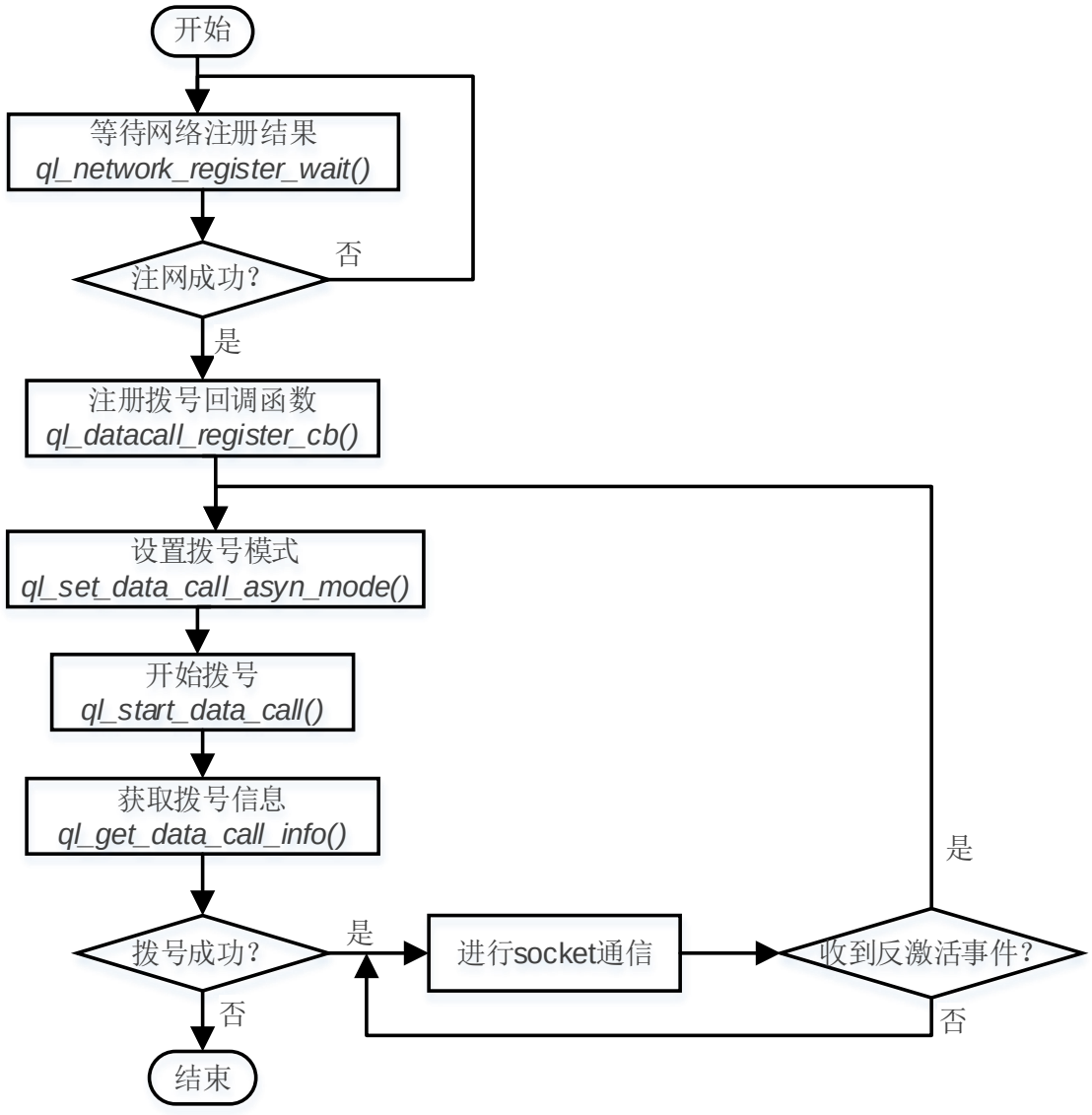

图 1：数据拨号操作流程

数据拨号操作流程主要分为 3 个过程：网络注册、数据拨号和 socket 创建。

网络注册是自动完成的，无需代码干预，只需调用 *ql_network_register_wait()*等待注册完成。若当前未注册网络，将阻塞当前调用该函数的线程，直至网络注册成功或超时后退出。用户亦可通过调用 *ql_nw_register_cb()*注册网络事件的回调函数，用于接收 *QUEC_NW_DATA_REG_STATUS_IND* 事件。有

关 *ql_nw_register_cb()*函数，详情请参考***文档 [2]***。

注网成功后方可进行 PDP 上下文激活，即本文所说的数据拨号。当前数据拨号同时支持同步模式和异步模式，可通过 *ql_set_data_call_asyn_mode()*设置。详情请参考***第 3.2.2 章***。

拨号成功后方可进行 socket 网络通信。为了便于客户了解拨号流程，QuecOpen SDK 中提供了 *datacall_demo.c* 示例程序，位于 *components\ql-application\nw*目录下。

如需自启动该示例程序，在 *components\ql-application\init\ql_init.c* 文件中的 *ql_init_demo_thread()*下取消 *ql_datacall_app_init()*前的注释，然后编译并烧录至模块。模块启动时，该示例程序将启动并自动进行数据拨号。

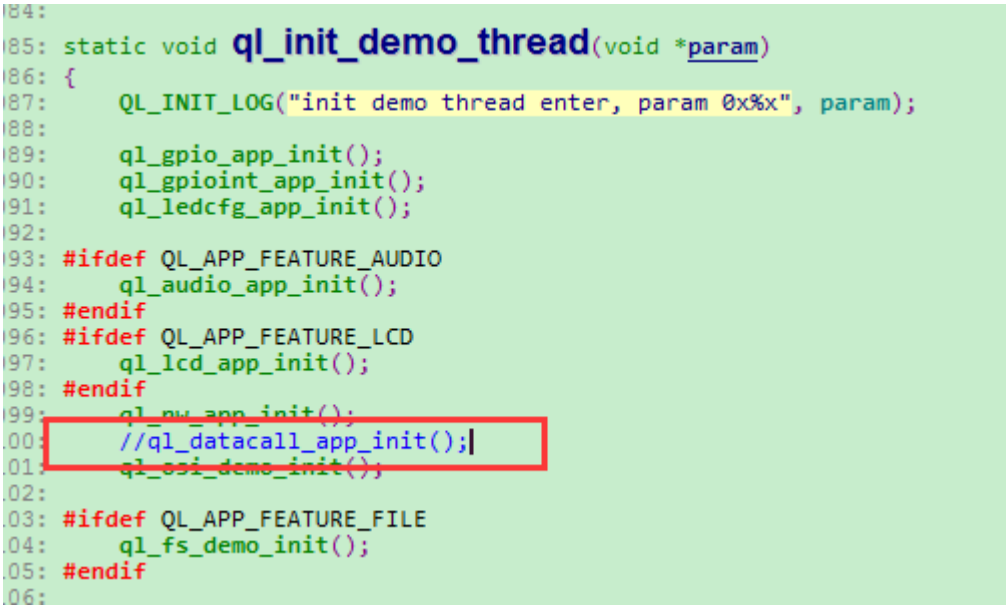

图 2：数据拨号示例自启动方法

3 数据拨号 API

本章节介绍数据拨号 API 函数、枚举和结构体等信息。

3.1. 头文件

数据拨号 API 的头文件为 *ql_api_datacall.h*，位于 *components\ql-kernel\inc* 目录下。若无特别说明，本文档所述头文件均位于该目录下。

3.2. API 详解

3.2.1. ql_network_register_wait

该函数用于等待网络注册结果。网络注册是开机后自动完成的，只有注册成功后，才可进行数据拨号。若当前未注册网络，将阻塞当前调用该函数的线程，直至在网络注册成功或超时后退出。

- **函数原型**

```
ql_datacall_errcode_e ql_network_register_wait(uint8_t nSim, unsigned int timeout_s)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

timeout_s：
[In] 等待网络注册的超时时间。单位：秒。

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.1.1. ql_datacall_errcode_e

数据拨号结果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_DATACALL_SUCCESS                         = 0,
    QL_DATACALL_EXECUTE_ERR                     = 1 | QL_DATACALL_ERRCODE_BASE,
    QL_DATACALL_MEM_ADDR_NULL_ERR,
    QL_DATACALL_INVALID_PARAM_ERR,
    QL_DATACALL_NW_REGISTER_TIMEOUT_ERR,
    QL_DATACALL_CFW_ACT_STATE_GET_ERR    = 5 | QL_DATACALL_ERRCODE_BASE,
    QL_DATACALL_REPEAT_ACTIVE_ERR,
    QL_DATACALL_REPEAT_DEACTIVE_ERR,
    QL_DATACALL_CFW_PDP_CTX_SET_ERR,
    QL_DATACALL_CFW_PDP_CTX_GET_ERR,
    QL_DATACALL_CS_CALL_ERR                    = 10 | QL_DATACALL_ERRCODE_BASE,
    QL_DATACALL_CFW_CFUN_GET_ERR,
    QL_DATACALL_CFUN_DISABLE_ERR,
    QL_DATACALL_NW_STATUS_GET_ERR,
    QL_DATACALL_NOT_REGISTERED_ERR,
    QL_DATACALL_NO_MEM_ERR                     = 15 | QL_DATACALL_ERRCODE_BASE,
    QL_DATACALL_CFW_ATTACH_STATUS_GET_ERR,
    QL_DATACALL_SEMAPHORE_CREATE_ERR,
    QL_DATACALL_SEMAPHORE_TIMEOUT_ERR,
    QL_DATACALL_CFW_ATTACH_REQUEST_ERR,
    QL_DATACALL_CFW_ACTIVE_REQUEST_ERR = 20 | QL_DATACALL_ERRCODE_BASE,
    QL_DATACALL_ACTIVE_FAIL_ERR,
    QL_DATACALL_CFW_DEACTIVE_REQUEST_ERR,
}ql_datacall_errcode_e;
```

- **参数**

参数	描述
QL_DATACALL_SUCCESS	函数执行成功
QL_DATACALL_EXECUTE_ERR	函数执行失败
QL_DATACALL_MEM_ADDR_NULL_ERR	参数地址为 NULL
QL_DATACALL_INVALID_PARAM_ERR	参数无效
QL_DATACALL_NW_REGISTER_TIMEOUT_ERR	网络注册超时
QL_DATACALL_CFW_ACT_STATE_GET_ERR	获取 PDP 上下文激活状态失败

QL_DATACALL_REPEAT_ACTIVE_ERR	重复激活 PDP 上下文
QL_DATACALL_REPEAT_DEACTIVE_ERR	重复反激活 PDP 上下文
QL_DATACALL_CFW_PDP_CTX_SET_ERR	设置 PDP 上下文错误
QL_DATACALL_CFW_PDP_CTX_GET_ERR	获取 PDP 上下文错误
QL_DATACALL_CS_CALL_ERR	正在通话导致数据业务操作失败
QL_DATACALL_CFW_CFUN_GET_ERR	功能模式获取失败
QL_DATACALL_CFUN_DISABLE_ERR	非全功能模式导致数据业务操作失败
QL_DATACALL_NW_STATUS_GET_ERR	网络注册状态获取失败
QL_DATACALL_NOT_REGISTERED_ERR	网络未注册
QL_DATACALL_NO_MEM_ERR	内存申请失败
QL_DATACALL_CFW_ATTACH_STATUS_GET_ERR	获取网络附着状态失败
QL_DATACALL_SEMAPHORE_CREATE_ERR	创建信号量失败
QL_DATACALL_SEMAPHORE_TIMEOUT_ERR	等待信号量超时
QL_DATACALL_CFW_ATTACH_REQUEST_ERR	网络附着被拒
QL_DATACALL_CFW_ACTIVE_REQUEST_ERR	PDP 上下文激活被拒
QL_DATACALL_ACTIVE_FAIL_ERR	PDP 上下文激活失败
QL_DATACALL_CFW_DEACTIVE_REQUEST_ERR	PDP 上下文反激活被拒

3.2.2. ql_set_data_call_asyn_mode

该函数用于设置启动和终止数据拨号函数（即 *ql_start_data_call()*和 *ql_stop_data_call()*）的执行模式。执行模式分为同步和异步两种模式。

- **函数原型**

```
ql_datacall_errcode_e ql_set_data_call_asyn_mode(uint8_t nSim, int profile_idx, bool enable)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM1
1 (U)SIM2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

enable：
[In] 函数执行模式。
0 同步模式
1 异步模式

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.3. ql_start_data_call

该函数用于启动数据拨号。默认为同步模式，如需异步模式请参考***第 3.2.2 章***进行设置。同步模式和异步模式区别如下所示：

同步模式：函数执行结束后返回数据拨号的结果码，若返回 *QL_DATACALL_SUCCESS* 则表示数据拨号成功，获取到 IP 地址，可进行 socket 通信。

异步模式：函数执行结束后返回函数执行结果码，若返回 *QL_DATACALL_SUCCESS* 并不表示数据拨号成功，仅表示函数执行成功，注册的回调函数将通过 *QUEC_DATACALL_ACT_RSP_IND* 事件通知上层是否拨号成功。

- **函数原型**

```
ql_datacall_errcode_e ql_start_data_call
(
    uint8_t nSim,
    int profile_idx,
    int ip_version,
    char *apn_name,
    char *username,
    char *password,
    int auth_type
)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM1
1 (U)SIM2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

ip_version：
[In] IP 类型。
1 IPv4
2 IPv6
3 IPv4v6

apn_name：
[In] APN 名称。

username：
[In] 用户名称。

password：
[In] 用户密码。

auth_type：
[In] 认证类型。详情请参考***第 3.2.3.1 章***。

- **返回值**

详情请参考***第 3.2.1.1 章***。

备注

若已开启 VoLTE 功能，执行数据拨号时仅可使用第 1 路至第 5 路 PDP 上下文。

3.2.3.1. ql_pdp_auth_type_e

认证类型枚举信息定义如下：

```
typedef enum
{
    QL_PDP_AUTH_TYPE_NONE = 0,
    QL_PDP_AUTH_TYPE_PAP,
    QL_PDP_AUTH_TYPE_CHAP,
}ql_pdp_auth_type_e;
```

- **参数**

参数	描述
QL_PDP_AUTH_TYPE_NONE	无认证协议
QL_PDP_AUTH_TYPE_PAP	PAP 认证协议
QL_PDP_AUTH_TYPE_CHAP	CHAP 认证协议

3.2.4. ql_stop_data_call

该函数用于终止数据拨号。默认为同步模式，如需异步模式请参考***第 3.2.2 章***进行设置。

- **函数原型**

```
ql_datacall_errcode_e ql_stop_data_call(uint8_t nSim, int profile_idx)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM1
1 (U)SIM2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.5. ql_get_data_call_info

该函数用于获取数据拨号信息。

- **函数原型**

```
ql_datacall_errcode_e ql_get_data_call_info(uint8_t nSim, int profile_idx, ql_data_call_info_s *info)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM1

1 (U)SIM2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

info：
[Out] 数据拨号信息。详情请参考***第 3.2.5.1 章***。

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.5.1. ql_data_call_info_s

数据拨号信息，结构体定义如下：

```
typedef struct
{
    int profile_idx;
    int ip_version;
    struct v4_info v4;
    struct v6_info v6;
}ql_data_call_info_s;
```

- **参数**

类型	参数	描述
int	*profile_idx*	PDP 上下文 ID。范围：1~7。
int	*ip_version*	IP 类型。 *1* IPv4 *2* IPv6 *3* IPv4v6
struct v4_info	*v4*	IPv4 信息。详情请参考***第 3.2.5.2 章***。
struct v6_info	*v6*	IPv6 信息。详情请参考***第 3.2.5.4 章***。

3.2.5.2. v4_info

IPv4 信息，结构体定义如下：

```
struct v4_info
{
    int state;
    struct v4_address_status addr;
};
```

- **参数**

类型	参数	描述
int	*state*	拨号状态。 *0* 未拨号 *1* 拨号成功
struct v4_address_status	*addr*	IPv4 地址信息。详情请参考***第 3.2.5.3 章***。

3.2.5.3. v4_address_status

IPv4 地址信息，结构体定义如下：

```
struct v4_address_status
{
    ip4_addr_t ip;
    ip4_addr_t pri_dns;
    ip4_addr_t sec_dns;
};
```

- **参数**

类型	参数	描述
ip4_addr_t	*ip*	获取的 IPv4 地址
ip4_addr_t	*pri_dns*	主域名服务器 IPv4 地址
ip4_addr_t	*sec_dns*	辅助域名服务器 IPv4 地址

3.2.5.4. v6_info

IPv6 信息，结构体定义如下：

```
struct v6_info
{
    int state;
    struct v6_address_status addr;
};
```

- **参数**

类型	参数	描述
int	*state*	拨号状态。 *0* 未拨号 *1* 拨号成功
struct v6_address_status	*addr*	IPv6 地址信息。详情请参考***第 3.2.5.5 章***。

3.2.5.5. v6_address_status

IPv6 地址信息，结构体定义如下：

```
struct v6_address_status
{
    ip6_addr_t  ip;
    ip6_addr_t  pri_dns;
    ip6_addr_t  sec_dns;
} ;
```

- **参数**

类型	参数	描述
ip6_addr_t	*ip*	获取的 IPv6 地址
ip6_addr_t	*pri_dns*	主域名服务器 IPv6 地址
ip6_addr_t	*sec_dns*	辅助域名服务器 IPv6 地址

3.2.6. ql_datacall_register_cb

该函数用于注册数据拨号的回调函数。无论是异步模式还是同步模式，均需注册回调函数用于上报被网络去激活或者去附着的事件，当接收到此事件后可发起重新拨号流程。

- **函数原型**

```
ql_datacall_errcode_e ql_datacall_register_cb(uint8_t nSim, int profile_idx, ql_datacall_callback datacall_cb, void *ctx)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

datacall_cb：
[In] 待注册的回调函数。详情请参考***第 3.2.6.1 章***。

ctx：
[In] 回调函数的传参指针。

- **返回值**

请参考***第 3.2.1.1 章***。

3.2.6.1. ql_datacall_callback

该回调函数用于上报数据拨号事件。

- **函数原型**

```
typedef void (*ql_datacall_callback)(uint8_t nSim, unsigned int ind_type, int profile_idx, bool result, void *ctx)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

ind_type：
[In] 数据拨号事件类型。

QUEC_DATACALL_ACT_RSP_IND	异步模式下 PDP 激活的结果响应事件
QUEC_DATACALL_DEACT_RSP_IND	异步模式下 PDP 反激活的结果响应事件
QUEC_DATACALL_PDP_DEACTIVE_IND	PDP 被网络去激活或者去附着的事件

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

result：
[In] PDP 上下文激活和反激活的结果。
0 失败
1 成功

ctx：
[In] 回调函数的传参指针。

3.2.7. ql_datacall_unregister_cb

该函数用于取消通过 *ql_datacall_register_cb()*注册的回调函数。取消后，回调函数将不再接收数据拨号的相关事件。

- **函数原型**

```
ql_datacall_errcode_e ql_datacall_unregister_cb(uint8_t nSim, int profile_idx, ql_datacall_callback datacall_cb,  void *ctx)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

datacall_cb：
[In] 待取消的回调函数。详情请参考***第 3.2.6.1 章***。

ctx：
[In] 回调函数的传参指针。

- **返回值**

详情请参考*第 3.2.1.1 章*。

3.2.8. ql_datacall_get_sim_profile_is_active

该函数用于获取当前 PDP 上下文激活状态。

- **函数原型**

```
bool ql_datacall_get_sim_profile_is_active(uint8_t nSim, int profile_idx)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

- **返回值**

0 未激活
1 已激活

3.2.9. ql_bind_sim_and_profile

该函数用于绑定所用(U)SIM 卡和 PDP 上下文 ID，获取(U)SIM 卡和 PDP 上下文 ID 的组合，即 *sim_cid*。

- **函数原型**

```
ql_datacall_errcode_e ql_bind_sim_and_profile(uint8_t nSim, int profile_idx, uint16_t *sim_cid)
```

- **参数**

nSim：
[In] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

profile_idx：
[In] PDP 上下文 ID。范围：1~7。

sim_cid：
[Out] (U)SIM 卡和 PDP 上下文 ID 的组合。计算公式为：nSim << 5 | profile_idx。

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.10. ql_get_sim_and_profile

该函数用于从 *sim_cid* 中获取(U)SIM 卡和 PDP 上下文 ID。

- **函数原型**

```
ql_datacall_errcode_e ql_get_sim_and_profile(uint16_t sim_cid, uint8_t *nSim, int *profile_idx)
```

- **参数**

sim_cid：
[In] (U)SIM 卡和 PDP 上下文 ID 的组合。

nSim：
[Out] 所用的(U)SIM 卡。若模块只支持 1 个(U)SIM 接口，此参数可设置为 0。
0 (U)SIM 卡 1
1 (U)SIM 卡 2

profile_idx：
[Out] PDP 上下文 ID。范围：1~7。

- **返回值**

详情请参考***第 3.2.1.1 章***。

3.2.11. ql_datacall_set_nat

该函数用于启用 NAT 功能。模块重启后配置生效。

- **函数原型**

```
ql_datacall_errcode_e ql_datacall_set_nat(uint32_t sim_profile_list)
```

- **参数**

sim_profile_list：
[In] (U)SIM 卡和 PDP 上下文 ID 组合列表。低 16 位为(U)SIM 卡 1，高 16 位为(U)SIM 卡 2；每一位表示对应的 PDP 上下文 ID。

- **返回值**

详情请参考*第 3.2.1.1 章*。

3.2.12. ql_datacall_get_nat

该函数用于查询已启用 NAT 功能的(U)SIM 卡和 PDP 上下文 ID。

- **函数原型**

```
ql_datacall_errcode_e ql_datacall_get_nat(uint32_t *sim_profile_list)
```

- **参数**

sim_profile_list：
[Out] (U)SIM 卡和 PDP 上下文 ID 组合列表。低 16 位为(U)SIM 卡 1，高 16 位为(U)SIM 卡 2；每一位表示对应的 PDP 上下文 ID。

- **返回值**

详情请参考*第 3.2.1.1 章*。

4 附录 参考文档及术语缩写

表 2：参考文档

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_注网信息 API_参考手册

表 3：术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序接口
CHAP	Challenge Handshake Authentication Protocol	挑战握手认证协议
IPv4	Internet Protocol Version 4	互联网通信协议第四版
IPv6	Internet Protocol Version 6	互联网通信协议第六版
NAT	Network Address Translation	网络地址转换
PAP	Password Authentication Protocol	密码认证协议
PDP	Packet Data Protocol	分组数据协议
SDK	Software Development Kit	软件开发工具包
(U)SIM	(Universal) Subscriber Identification Module	（全球）用户识别模块