

河北骑士智能科技有限公司 QSDK 代码简介

QSDK 是河北骑士智能科技有限公司基于 RT-THREAD RTOS 编写的 NB-IOT 模块驱动代码，目前支持的模块型号有 M5310、M5310A、M5311。

源码地址：[\[https://github.com/hbqs/qsdk\]](https://github.com/hbqs/qsdk) 作者：[\[longmain\]](#)

由于目前各方都在大力推动 NB-IOT 产业发展，我们骑士智能科技也代理了 NB-IOT 模组，为了让大家在使用我们代理的模组时更加方便，我们编写了 QSDK（NB-IOT 驱动）代码，目前该代码在使用过程中简单方便，目前适配了 RTTHREAD 组件，可以图像化配置，减轻了大量的工作，可以促进产品快速开发。

在这里感谢 RT-THREAD 这款好用的物联网系统，同时更感谢广大开源的支持者。让我从中学到了很多，网络也是一个好平台，希望所有的开发者能形成良性循环，从网络中学知识，回馈到网络中去。

qsdk 简介：

qsdk 是一个灵活的 NB-IOT 驱动，目前支持的模块型号有 M5310、M5310A、M5311，后期将会增加更多模组支持。该代码根据不同的模块启用不同的功能，目前代码支持电信 IOT 平台、中移 ONENET 平台（LWM2M 协议和 MQTT 协议）和 UDP\TCP 协议。在 qsdk_callback.c 代码中是关于 IOT、onenet、NET 等下发处理回调函数，用户可以在相应函数中处理数据。

特点：

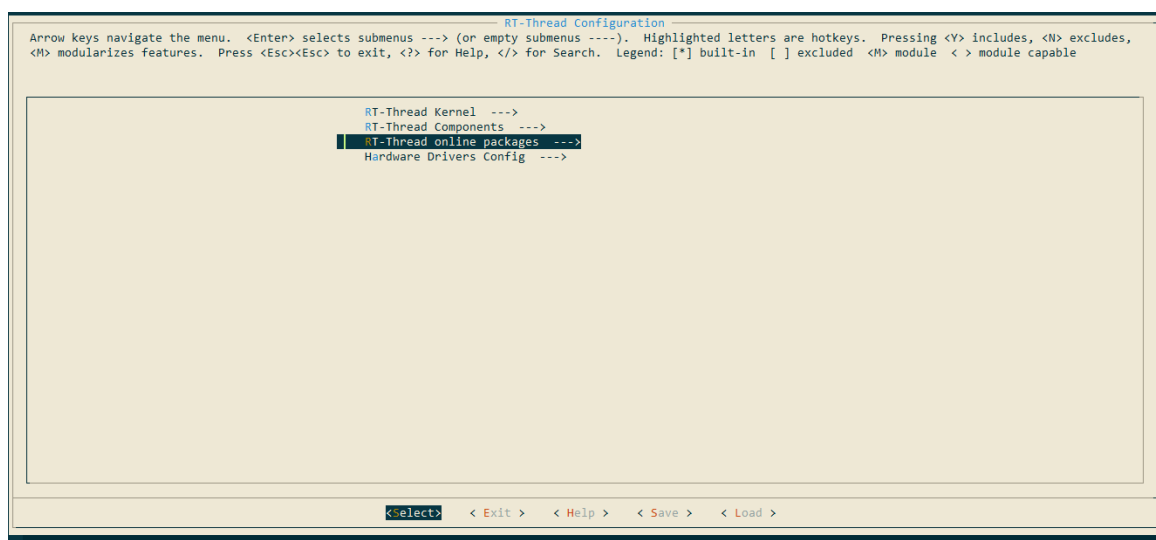
qsdk 开放源码，nb 控制块、协议处理块均采用数据结构方式，网络数据下发采用回调函数机制，单独的 qsdk_callback.c 可以完整的提示网络下发数据，用户可以在函数里面编写对网络下发数据的处理逻辑代码。同时代码支持快速初始化 NB 模块联网，快速连接 onenet 平台等功能，大大简化了客户的使用方式。

qsdk 代码在 env 使用：

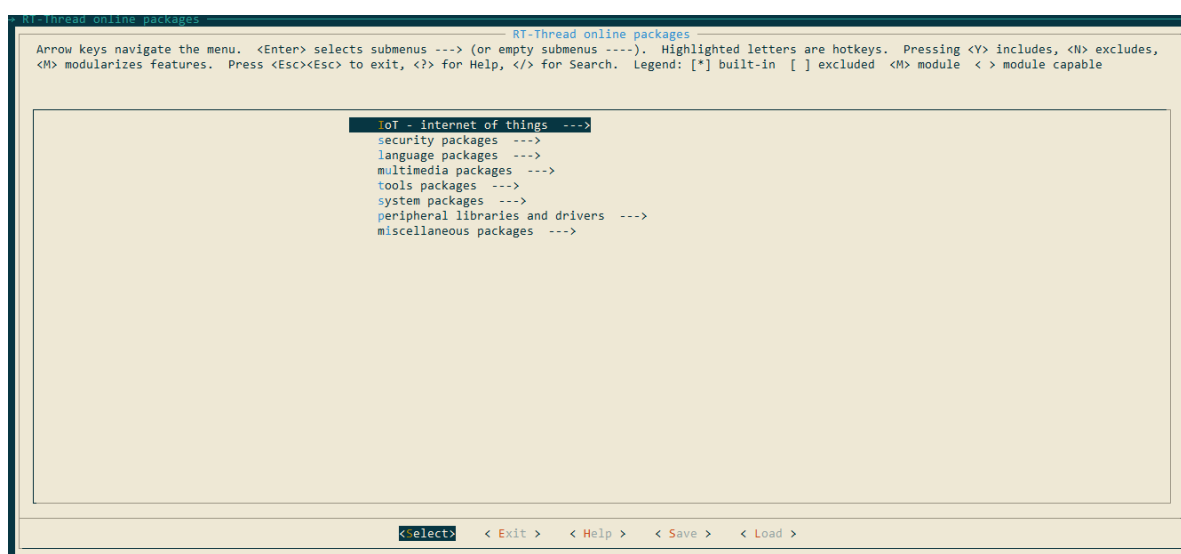
目前我们将 NB-IOT 模块驱动代码做成软件包 (packages)，如果使用 RT-Thread 操作系统的话，可以在 env 中直接配置使用，使用前请开启 AT 组件支持！

步骤如下：

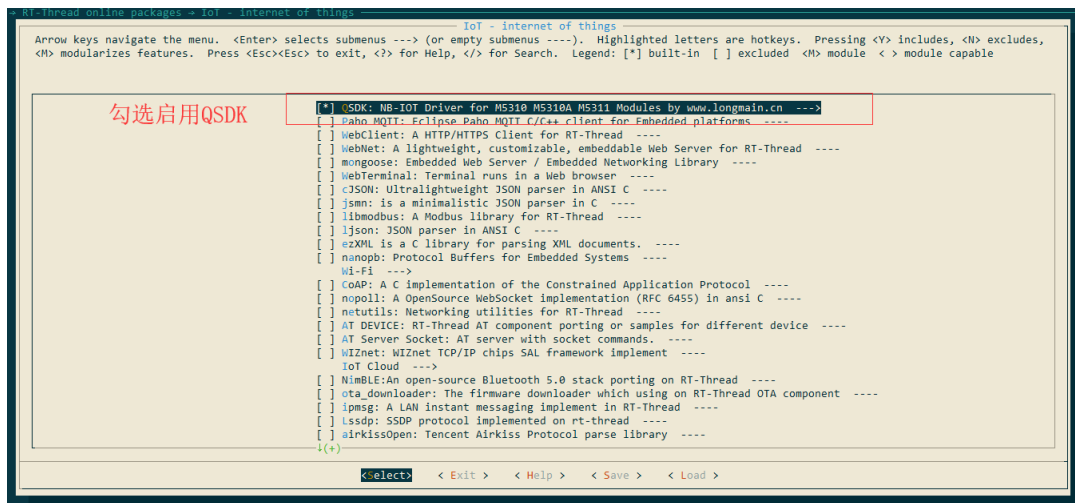
1. **选择在线软件包**



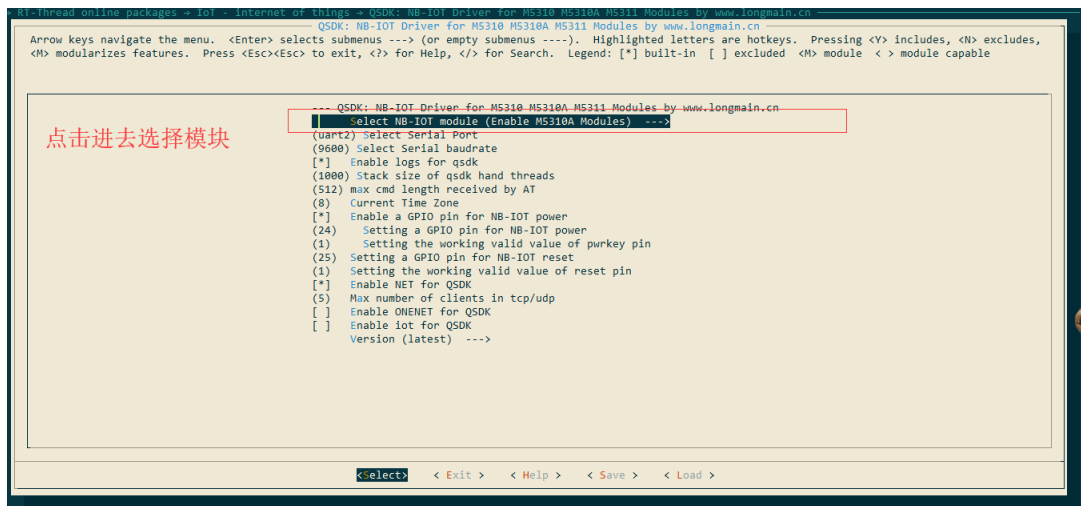
2. **选择软件包属性为物联网相关**



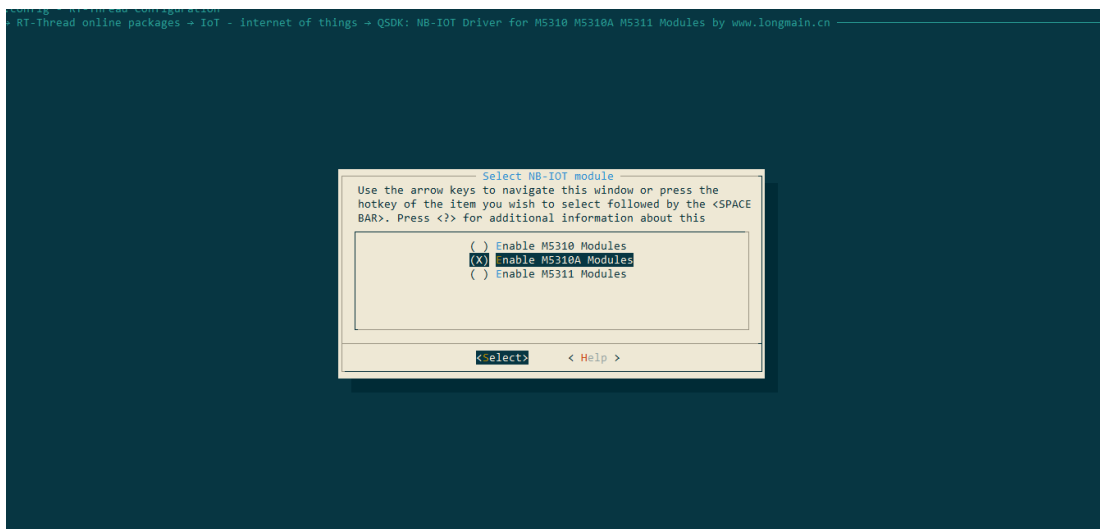
3. **选择 qsdk 组件**



4. **点击进去选择支持的 nb-iot 模块（目前支持 M5310、M5310A、M5311）**



5. **选择需要支持的 nb-iot 模块**



6. **设置需要支持的引脚信息和协议**

```
RT-Thread online packages -> IoT -> Internet of things -> QSDK: NB-IOT Driver for M5310 M5310A M5311 Modules by www.longmain.cn
QSDK: NB-IOT Driver for M5310 M5310A M5311 Modules by www.longmain.cn
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

--- QSDK: NB-IOT Driver for M5310 M5310A M5311 Modules by www.longmain.cn
[*] Select NB-IOT module (Enable M5310A Modules) --->
(uart2) Select Serial Port
(9600) Select Serial baudrate
[*] Enable logs for qsdk
(1000) Stack size of qsdk hand threads
(512) max cmd length received by AT
(8) Current Time Zone
[*] Enable a GPIO pin for NB-IOT power
(24) Setting a GPIO pin for NB-IOT power
(1) Setting the working valid value of pwrkey pin
(25) Setting a GPIO pin for NB-IOT reset
(1) Setting the working valid value of reset pin
[*] Enable NET for QSDK
(5) Max number of clients in tcp/udp
[ ] Enable ONENET for QSDK
[ ] Enable iot for QSDK
Version (latest) --->

<Select> <Exit> <Help> <Save> <Load>
```

选择支持PWRKEY
设置引脚和有效电平

设置reset引脚和有效电平

选择支持的协议，如net（TCP/UDP）

7. **更新软件包**

```
*** End of the configuration.
*** You can execute 'scons' to start the build or try 'scons -h'
*** If you want to generate the IDE's project file, you can use command:
*** 'scons --target=mdk/mdk4/mdk5/iar/cb -s'.
*** If you want to install rt-thread component online, try 'pkgs'.

longmain@QISHI E:\RTTOS\qsd_k_rtos\stm32l151\7_M5310A_UDP
> pkgs --update
cmd.exe*[64]:8404
```

下载软件包

8. **编译生成 mdk/iar 工程**

```
*** End of the configuration.
*** You can execute 'scons' to start the build or try 'scons -h'
*** If you want to generate the IDE's project file, you can use command:
*** 'scons --target=mdk/mdk4/mdk5/iar/cb -s'.
*** If you want to install rt-thread component online, try 'pkgs'.

longmain@QISHI E:\RTTOS\qsd_k_rtos\stm32l151\7_M5310A_UDP
> scons --target=mdk5
```

生成MDK5工程

1、QSDK 在 ENV 配置教程

设置串口和波特率

设置hand 线程栈大小

选择是否开启PWRKEY

设置引脚

设置有效电平

选择需要支持的网络协议

```
--- QSDK: NB-IOT Driver for M5310 M5310A M5311 Modules by www.longmain.cn
Select NB-IOT module (Enable M5310A Modules) ---> 选择NB-IOT模块
(uart2) Select Serial Port
(9600) Select Serial baudrate
[*] Enable logs for qsdkLOG开启选项
(1000) Stack size of qsdk hand threads
(512) max cmd length received by AT 设置AT命令最大接收长度
(8) Current Time Zone 设置时区差
[*] Enable a GPIO pin for NB-IOT power
(24) Setting a GPIO pin for NB-IOT power
(1) Setting the working valid value of purkey pin
(25) Setting a GPIO pin for NB-IOT reset 选择reset引脚
(1) Setting the working valid value of reset pin 设置有效电平
[*] Enable NET for QSDK
(5) Max number of clients in tcp/udp
[ ] Enable ONENET for QSDK
[ ] Enable iot for QSDK
Version (latest) --->
```

2、QSDK 之 NB 模组控制函数

2.01、清空 QSDK 中 NB 模组当前状态

Function	void qsdk_nb_clear_environment(void)
函数功能	清空 QSDK 中 NB 模组当前状态
参数	无
返回值	无

2.02、QSDK 一键初始化并且附着网络

Function	int qsdk_nb_quick_connect(void)
函数功能	NB-IOT 模组快速联网函数
参数	无
返回值	0：联网成功 1：联网失败

2.03、重启 NB-IOT 模组

Function	int qsdk_nb_reboot(void)
函数功能	控制 reset 引脚复位模块
参数	无
返回值	0：成功 1：失败

2.04、检测模块是否正常

Function	int qsdk_nb_wait_connect(void)
函数功能	发送 AT, 判断是否收到 OK
参数	无
返回值	0: 正常 1: 模块异常

2.05、检测模块是否为正常模式

Function	int qsdk_nb_sim_check(void)
函数功能	发送 AT+CFUN, 判断返回值
参数	无
返回值	0: 成功 1: 最小功能模式

2.06、设置 PSM 休眠时间

Function	int qsdk_nb_set_psm_mode(char *tau_time, char *active_time)		
函数功能	模块 PSM 模式设置		
参数	类型	参数	说明
	char *	tau_time	TAU 更新时间
	char *	active_time	PSM 时间
返回值	0: 成功 1: 失败		

2.07、读取 NB 卡 IMSI

Function	char *qsdk_nb_get_imsi(void);
函数功能	读取 NB 卡的 IMSI 号码
参数	无
返回值	IMSI 指针: 成功 RT_NULL: 失败

2.08、读取 NB 模组的 IMEI 编号

Function	char *qsdk_nb_get_imei(void);
函数功能	读取 NB 模组的 IMEI
参数	无
返回值	IMEI 指针: 成功 RT_NULL: 失败

2.09、获取 CCLK 时间

Function	int qsdk_nb_get_time(void);
函数功能	获取 CCLK 时间，获取成功后会进入 RTC 回调函数
参数	无
返回值	0:成功 1:失败

2.10、获取信号强度值

Function	int qsdk_nb_get_csq(void);
函数功能	获取信号强度，发送 AT+CSQ
参数	无
返回值	0-99:成功 -1:失败

2.11、手动附着网络

Function	int qsdk_nb_set_net_start(void)
函数功能	发送 AT+CGATT=1
参数	无
返回值	0 设置成功 1 设置失败

2.12、获取当前网络连接状态

Function	int qsdk_nb_get_net_connect(void)
函数功能	发送 AT+CEREG?, 判断返回值
参数	无
返回值	0 连接成功 1 连接失败

2.13、查询已获取的网络状态

Function	int qsdk_nb_get_net_connect_status(void)
函数功能	查询已获取的网络连接状态
参数	无
返回值	0 连接成功 1 连接失败

2.14、获取 nb-iot 模块重启原因

Function	int qsdk_nb_get_reboot_event(void)
函数功能	获取 nb-iot 模块重启原因

参数	无
返回值	1:引脚复位 2:AT 复位 3:FOTA 复位 4: 异常复位

2.15、nb-iot 模块进入 PSM 模式

Function	void qsdk_nb_enter_psm(void)
函数功能	切断 AT 通讯功能，进入此函数后竟无法进行其他 AT 操作
参数	无
返回值	无

2.16、nb-iot 模块退出 PSM 模式

Function	int qsdk_nb_exit_psm(void)
函数功能	执行前，请先获取模块当前是否进入 PSM 模式
参数	无
返回值	0:成功 1:失败

2.17、查询 nb-iot 模块是否进入 psm 模式

Function	int qsdk_nb_get_psm_status(void)
函数功能	查询 nb-iot 模块是否进入 psm 模式
参数	无
返回值	0: 当前在 psm 模式 1: 当前未进入 psm 模式

2.18、模块 ping 服务器 IP

Function	int qsdk_nb_ping_ip(char *ip)
函数功能	ping 服务器 IP
参数	无
返回值	0:成功 1:失败

2.19、打开网络指示灯

Function	int qsdk_nb_open_net_light(void)
函数功能	打开网络指示灯
参数	无
返回值	0:成功 1:失败

2. 20、关闭网络指示灯

Function	int qsdk_nb_close_net_light(void)
函数功能	关闭网络指示灯
参数	无
返回值	0:成功 1:失败

2. 21、模块启用自动进入 PSM 模式功能

Function	int qsdk_nb_open_auto_psm(void)
函数功能	当 10S 内与模块没有交互，模块自动进入休眠
参数	无
返回值	0:成功 1:失败

2. 22、模块关闭自动进入 PSM 模式功能

Function	int qsdk_nb_close_auto_psm(void)
函数功能	关闭模块在 10S 内无 AT 交互进入休眠。
参数	无
返回值	0:成功 1:失败

2. 23、查询模块在核心网的 IP 地址

Function	char *qsdk_nb_query_ip(void)
函数功能	查询模组在核心网的 IP 地址
参数	无
返回值	char * IP 地址

2. 24、字符串转换为 16 进制字符串

Function	int string_to_hex(const char *pString, int len, char *pHex)		
函数功能	字符串转 HEX 类型的字符串		
参数	类型	参数	说明
	char *	pString	待转换字符串
	int	Len	待转换字符串长度
	char *	pHex	转换后的 HEX 字符串
返回值	0: 成功 1: 失败		

2.25、16 进制字符串转换为字符串

Function	void hexstring_to_string(char * pHex, int len, char * pString)		
函数功能	16 进制字符串转字符串		
参数	类型	参数	说明
	char *	pHex	待转换 16 进制字符串
	Int	len	pHex 字符串长度
	char *	pString	转换好的字符串
返回值	0: 成功 1: 失败		

3、 QSDK 之电信 IOT 平台连接

3.01 打开 IOT 平台上行消息提示

Function	int qsdk_iot_open_update_status(void)		
函数功能	开启 IOT 平台上行信息提示		
参数	无		
返回值	0: 成功 1: 失败		

3.02 开启 IOT 平台下行消息提示

Function	qsdk_iot_open_down_date_status		
函数功能	开启 IOT 平台下行信息提示		
参数	无		
返回值	0: 成功 1: 失败		

3.03 上报信息到 IOT 平台

Function	int qsdk_iot_update(char *str)		
函数功能	向 IOT 平台上报消息		
参数	类型	参数	说明
	char *	str	上行消息指针
返回值	0: 成功 1: 失败		

4、QSDK 之 NET 代码（UDP/TCP）

4.01 net client 客户端初始化

Function	qsdk_net_client_t qsdk_net_client_init(int type, int port, char *server_ip, unsigned short server_port)		
函数功能	net client 客户端初始化		
参数	类型	参数	说明
	int	type	网络类型
	int	port	本地端口号
	char *	server_ip	服务器 IP
	unsigned short	server_port	服务器端口
	注：type 类型为： 1: QSDK_NET_TYPE_TCP TCP 连接 2: QSDK_NET_TYPE_UDP UDP 连接		
返回值	成功返回：client 地址 失败返回：RT_NULL		

4.02 创建网络套子号 socket

Function	int qsdk_net_create_socket(qsdk_net_client_t client)		
函数功能	创建网络套子号		
参数	类型	参数	说明
	qsdk_net_client_t	Client	Client 客户端地址
返回值	0: 创建成功 1: 创建失败		

4.03 连接到 TCP 服务器

Function	int qsdk_net_connect_to_server(qsdk_net_client_t client)		
函数功能	连接到服务器		
参数	类型	参数	说明
	qsdk_net_client_t	Client	Client 客户端地址
返回值	0: 成功 1: 失败		

4.04 向服务器发送数据

Function	int qsdk_net_send_data(qsdk_net_client_t client, char *str, unsigned int len)		
函数功能	向服务器发送数据		
参数	类型	参数	说明

	qsdnet_client_t	client	Client 客户端地址
	char *	str	待发送数据首地址
返回值	0: 成功 1: 失败		

4.05 查看 net client 客户端是否受到数据

Function	int qsdnet_get_client_revive(qsdnet_client_t client)		
函数功能	查看 client 是否收到消息		
参数	类型	参数	说明
	qsdnet_client_t	client	Client 客户端地址
返回值	0: 收到消息 1: 未收到消息		

4.06 查看 NET 客户端连接状态

Function	int qsdnet_get_client_connect(qsdnet_client_t client)		
函数功能	查看 client 网络连接状态		
参数	类型	参数	说明
	qsdnet_client_t	client	Client 客户端地址
返回值	0: 已连接 1: 未连接		

4.07 关闭 NET 网络套子号

Function	int qsdnet_close_socket(qsdnet_client_t client)		
函数功能	查看 client 网络连接状态		
参数	类型	参数	说明
	qsdnet_client_t	client	Client 客户端地址
返回值	0: 成功关闭 1: 关闭失败		

4.08 清理 net 运行环境参数

Function	int qsdnet_clear_environment(void)		
函数功能	查看 client 网络连接状态		
参数	无		
返回值	0: 成功		

5、QSDK 之 ONENET（LWM2M） 连接

5.01 初始化一个 onenet 平台 object 数据流

Function	qsdk_onenet_stream_t qsdk_onenet_object_init(int objid,int insid,int resid,int inscount,char *bitmap, int atts,int acts,int type)		
函数功能	初始化一个 onenet 平台 object 数据流		
参数	类型	参数	说明
	int	objid	object 编号
	int	insid	Instance 编号
	int	resid	resource 编号
	int	inscount	相同 object 数据流个数
	char *	bitmap	instance bitmap
	int	atts	是否可 write/read
	int	acts	是否可 exec 下发命令
	int	type	数据类型
	type 类型: 1: qsdk_onenet_value_String 字符串 2: qsdk_onenet_value_Opaque 自定义 3: qsdk_onenet_value_Integer int 4: qsdk_onenet_value_Float float 5: qsdk_onenet_value_Bool bool 6: qsdk_onenet_value_HexStr HEX 形式字符串		
返回值	object 指针: 成功 RT_NULL:失败		

5.02、删除模组实例 instance

Function	int qsdk_onenet_delete_instance(void)
函数功能	删除模组实例 instance
参数	无
返回值	0: 成功 1: 失败

5.03、删除模组内已注册 object 数据流

Function	int qsdk_onenet_delete_object(qsdk_onenet_stream_t stream)
函数功能	删除模组内已注册 object 数据流
参数	stream: object 编号
返回值	0: 成功 1: 失败

5.04、设备登录到 onenet 平台

Function	int qsdk_onenet_open(void)
函数功能	一键连接到 onenet 平台
参数	无
返回值	0: 成功 1: 失败

5.05、在 onenet 平台注销设备

Function	int qsdk_onenet_close(void)
函数功能	在 onenet 平台注销设备
参数	无
返回值	0: 成功 1: 失败

5.06、更新 onenet 设备维持时间

Function	int qsdk_onenet_update_time(int flge)
函数功能	更新 onenet 设备维持时间
参数	flge :1 同时更新云端 object 信息 0: 不更新
返回值	0: 成功 1: 失败

5.07、一键连接 onenet 平台

Function	int qsdk_onenet_quick_start(void)
函数功能	一键连接到 onenet 平台
参数	无
返回值	0: 成功 1: 失败

5.08、notify 数据到 onenet 平台

Function	int qsdk_onenet_notify(qsdk_onenet_stream_t stream, int len, qsdk_onenet_value_t data, int flge)
函数功能	notify 数据到 onenet 平台
参数	stream: object 结构体
	len: 消息长度
	data: 消息内容
	flge: 消息标识
返回值	0: 成功 1: 失败

5.09、获取 onenet 平台连接状态

Function	int qsdk_onenet_get_connect(void)
函数功能	获取 onenet 平台连接状态
参数	无
返回值	0: 连接成功 1: 连接已断开

5.10、查询当前 object 是否收到平台 read 消息

Function	int qsdk_onenet_get_object_read(qsdk_onenet_stream_t stream)
函数功能	查询当前 object 是否收到平台 read 消息
参数	stream: object 结构体
返回值	0: 收到 1: 没收到

5.11、查询当前 object 是否收到平台 write 消息

Function	int qsdk_onenet_get_object_write(qsdk_onenet_stream_t stream)
函数功能	查询当前 object 是否收到平台 write 消息
参数	stream: object 结构体
返回值	0: 收到 1: 没收到

5.12、查询当前 object 是否收到平台 exec 消息

Function	int qsdk_onenet_get_object_exec(qsdk_onenet_stream_t stream)
函数功能	查询当前 object 是否收到平台 exec 消息
参数	stream: object 结构体
返回值	0: 收到 1: 没收到

5.13、响应 onenet read 操作

Function	int qsdk_onenet_read_rsp(int msgid, int result, qsdk_onenet_stream_t stream, int len, qsdk_onenet_value_t data, int index, int flge)		
函数功能	响应 onenet read 操作		
参数	类型	参数	说明
	Int	msgid	消息 ID
	int	result	读写状态
	qsdk_onenet_stream_t	stream	object 结构体
	int	len	上报的消息长度

	qsdk_onenet_value_t	data	上报的消息值
	int	index	指令序列号
	int	flge	消息标识
返回值	0: 成功 1: 失败		

5.14、清理 onenet 运行环境

Function	int qsdk_onenet_clear_environment(void)
函数功能	清理 onenet 运行环境
参数	无
返回值	0: 成功 1: 失败

5.15、notify 数据到 onenet 平台（带 ACK）

Function	int qsdk_onenet_notify_and_ack(qsdk_onenet_stream_t stream, int len, qsdk_onenet_value_t data, int flge)
函数功能	notify 数据到 onenet 平台（带 ACK）
参数	stream: object 结构体
	len: 消息长度
	data: 消息内容
	flge: 消息标识
返回值	0: 成功 1: 失败

联系人

* 技术支持: longmain QQ: 1239799850

* 邮箱: [longmain@longmain.cn]

* 主页: [qsdk](https://github.com/hbqs)

* 仓库: [Github](https://github.com/hbqs)