

腾讯云 IOT-SDK 嵌入式平台移植指导

Tencent 腾讯

腾讯机密，严禁外传。
未经授权，请勿扩散。

修订记录

修订日期	修订版本	修改描述	作者
20180620	V1.0.0	创建	yougaliu

修订流程

对于本文档中任何内容的增删改以及相关其它文档的创建，都应该知会部门同事。

接口人

本文档中的任何信息都应该被仔细的阅读。如果有任何疑虑，意见或问题，请直接联系下表中的接口人。

姓名	邮箱	电话	组织
yougaliu	yougaliu@tencent.com	86013388-87706	腾讯云产品部

缩略语清单：

缩略语	英文全名	中文解释
[DDL]	[data definition language]	[数据定义语言]

目 录

修订记录	II
1. 概述	1
2. 代码框架	1
2.1. 集成了腾讯云 SDK 的软件架构	1
2.1.1. 软件框图	1
2.1.2. 目录结构	2
3. 移植开展	2
3.1. 代码放置	2
3.2. 驱动实现	2
3.3. 接口挂接	3
3.4. 设备创建、配置、鉴权及通信	3
3.4.1. 设备创建	3
3.4.2. 设备端配置	3
3.4.3. 设备和平台通信	3
3.4.4. 云平台操作步骤截图	4

腾讯云 IOT-SDK 嵌入式平台移植指导

1. 概述

腾讯云官网设备端 SDK:

<https://github.com/tencentyun/tencent-cloud-iotsuite-embedded-c.git>

该 SDK 基于 linux 环境实现了 demo 程序, demo 程序包含 MQTT、COAP、通用数据模板等演示程序, 具体的嵌入式平台移植可以参考 demo 进行移植。

SDK 接口说明:

<https://github.com/tencentyun/tencent-cloud-iotsuite-embedded-c/blob/master/README.md>

本文基于 STM32L433 + SIM900 2G 模组示例说明移植过程。移植示例下载链接:

<http://git.tencent.com/yougaliu/tencent-cloud-iotsuite-for-STM32.git>

后续我们会逐渐将我们的 SDK 作为一个类似 lwip 的中间件集成到各 MCU 或者模组的 SDK 或者工具中, 敬请关注留意。

2. 代码框架

2.1. 集成了腾讯云 SDK 的软件架构

2.1.1. 软件框图

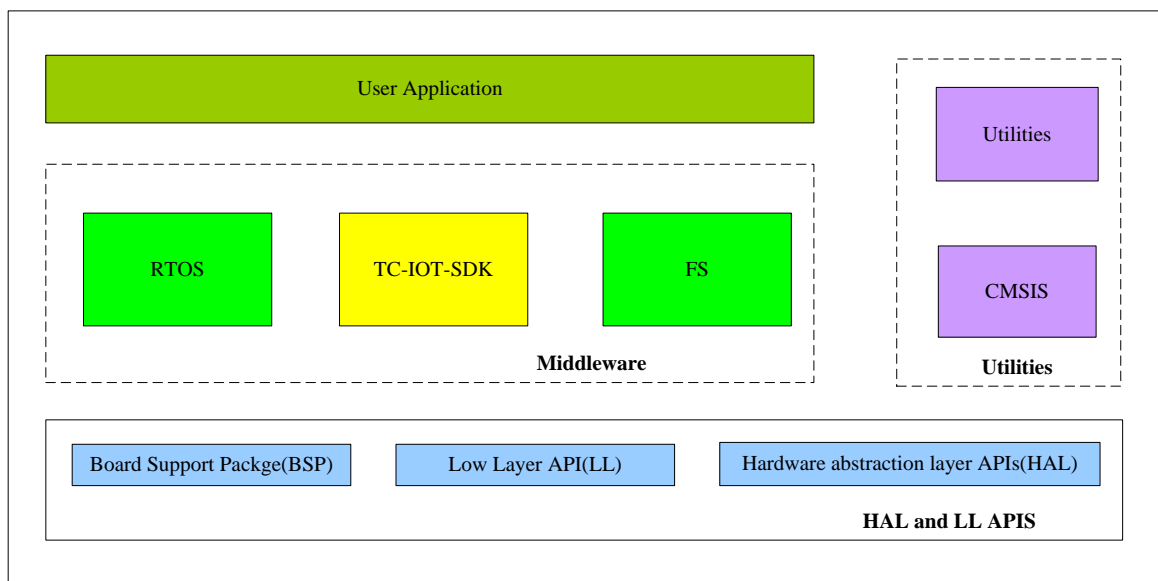


图1 Firmware Framework of Embedded system integrated Tencent IOT_SDK

2.1.2. 目录结构

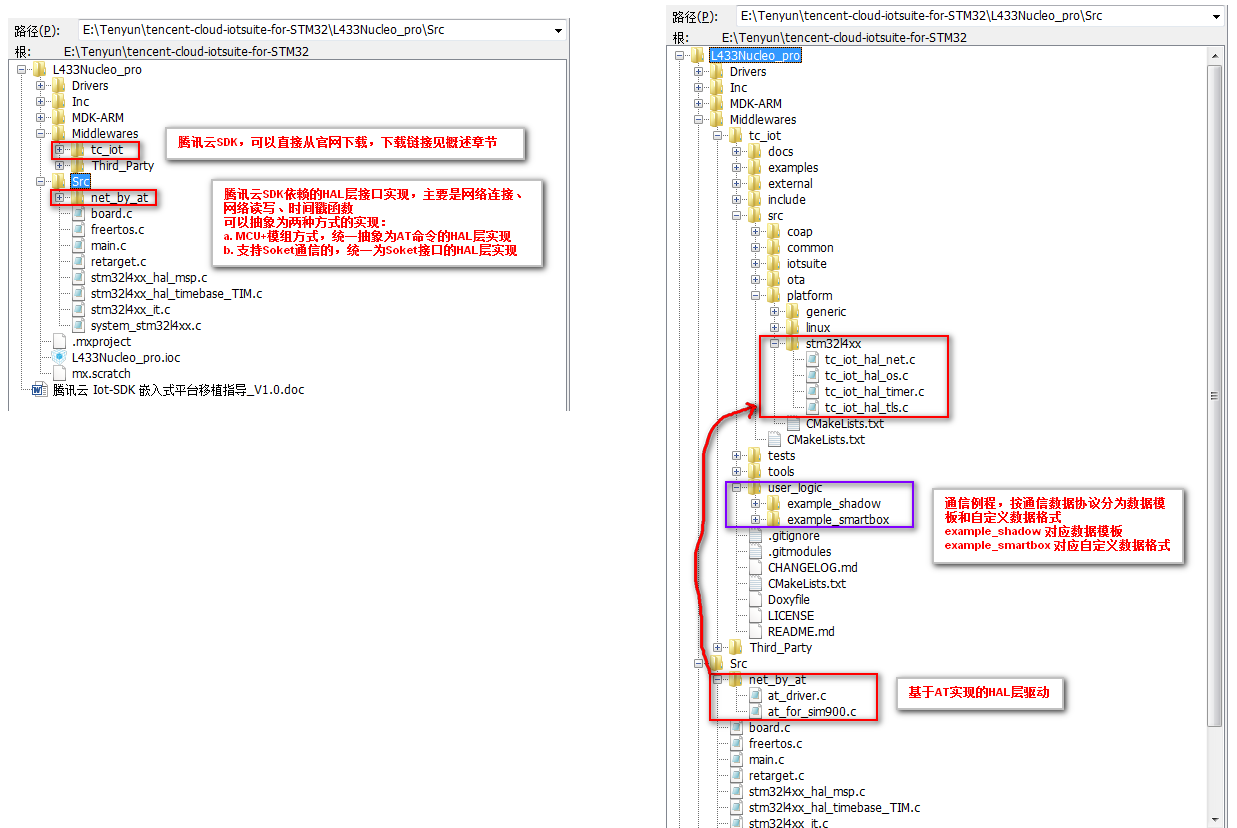


图2 目录结构

3. 移植开展

3.1. 代码放置

从官网下载 SDK 最新代码，链接见概述章节。将整个代码目录作为通信组件放在类似 LWIP、FATFS 的第三方组件的同级目录。如示例放置在 工程目录/Middlewares/tc_iot 下。

3.2. 驱动实现

SDK 移植到具体平台硬件，需要做的移植工作是实现 工程目录/Middlewares/tc_iot/src/platform/ 下的平台相关的 hal 层接口，拷贝平台目录 generic 目录到同级目录，并重命名为自定义平台硬件目录，譬如本例的目录 stm3214xx，需要实现使用到的 hal 层接口，为了便于开发者移植，我们在 hal 层和底层驱动层中间抽象了一个中间层。

中间层根据通常的接入网络的硬件设计方式分为两种 hal 层实现方式。

一种硬件设计方式为 **MCU+通信模组** 的方式，我们将这种 hal 层的实现统一抽象为 AT 接口的实现方式，譬如示例的 STM32+SIM900 模组的方式即为此方式。开发者参考此 hal 中间层的实现，修改各 AT 驱动接口中调用的 AT 指令为对应模组的 AT 指令。

另一种方式为硬件平台支持基于 TCP/IP 的 **Socket 通信方式**，一般支持了 LWIP 协议栈的都支持 Socket

通信。我们将这种 hal 层的实现统一抽象为 Socket 接口的实现方式，Socket 接口比较统一，譬如示例工程目录/Middlewares/tc_iot/src/platform/linux 下的 hal 层即是基于 Socket 的接口实现的。

中间层接口再挂接到工程目录/Middlewares/tc_iot/src/platform/stm3214xx(自定义平台目录)下的平台相关的 hal 层接口目录下，当然开发者也可以根据个人喜好，直接实现平台目录下的 hal 层接口也是可以的。

3.3. 接口挂接

如果驱动层参考了中间层的方式，需要将中间层接口挂接到对应的 hal 层接口上去。参考示例头文件 at_for_sim900.h 相关接口的挂接位置。涉及的接口主要为网络 hal 层的连接、网络读接口（要实现按字节读取）、网络写接口、网络连接状态接口、延时接口、时间戳接口（UTC，鉴权需要时间戳）。注意处理好各接口返回值。

3.4. 设备创建、配置、鉴权及通信

3.4.1. 设备创建

设备完成 SDK 移植后，下一步即是接入腾讯云平台，并进行数据通信。

首先需要登录腾讯云官网 <https://cloud.tencent.com/> 申请账号，审批通过后登录并进入控制台，如步骤 1。按 3.4.4 截图步骤依次完成产品创建、设备创建。

3.4.2. 设备端配置

设备创建完毕后，有 5 个信息请拷贝对应修改到配置头文件 tc_iot_device_config.h 中的宏定义。

这 5 个信息如下，如步骤 5、步骤 6 所示信息：

产品 ID	对应宏	TC_IOT_CONFIG_DEVICE_PRODUCT_ID
产品 KEY	对应宏	TC_IOT_CONFIG_DEVICE_PRODUCT_KEY
MQTT 链接	对应宏	TC_IOT_CONFIG_SERVER_HOST
设备名	对应宏	TC_IOT_CONFIG_DEVICE_NAME
设备密钥	对应宏	TC_IOT_CONFIG_DEVICE_SECRET

设备和云平台的连接和鉴权过程需要这个 5 个信息，鉴权接口参考 http_refresh_auth_token_with_expire。这 5 个信息需要在生产过程写入设备，需要做好驱动上的支持。

3.4.3. 设备和平台通信

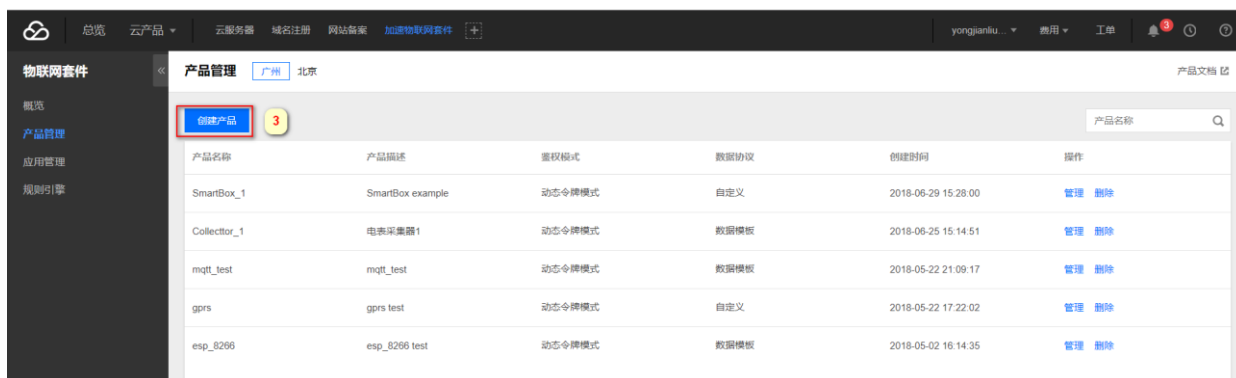
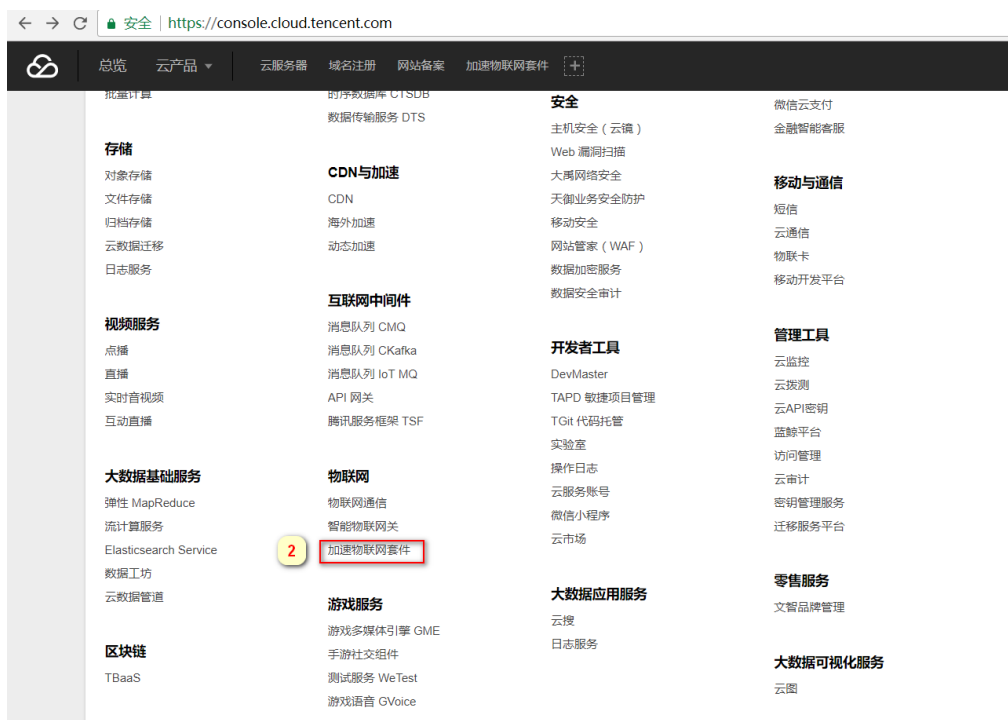
产品创建过程数据协议有数据模板和自定义两个选择，如步骤 4，即平台支持开发者按数据模板或者自定义数据格式两种方式和云平台进行数据交互。

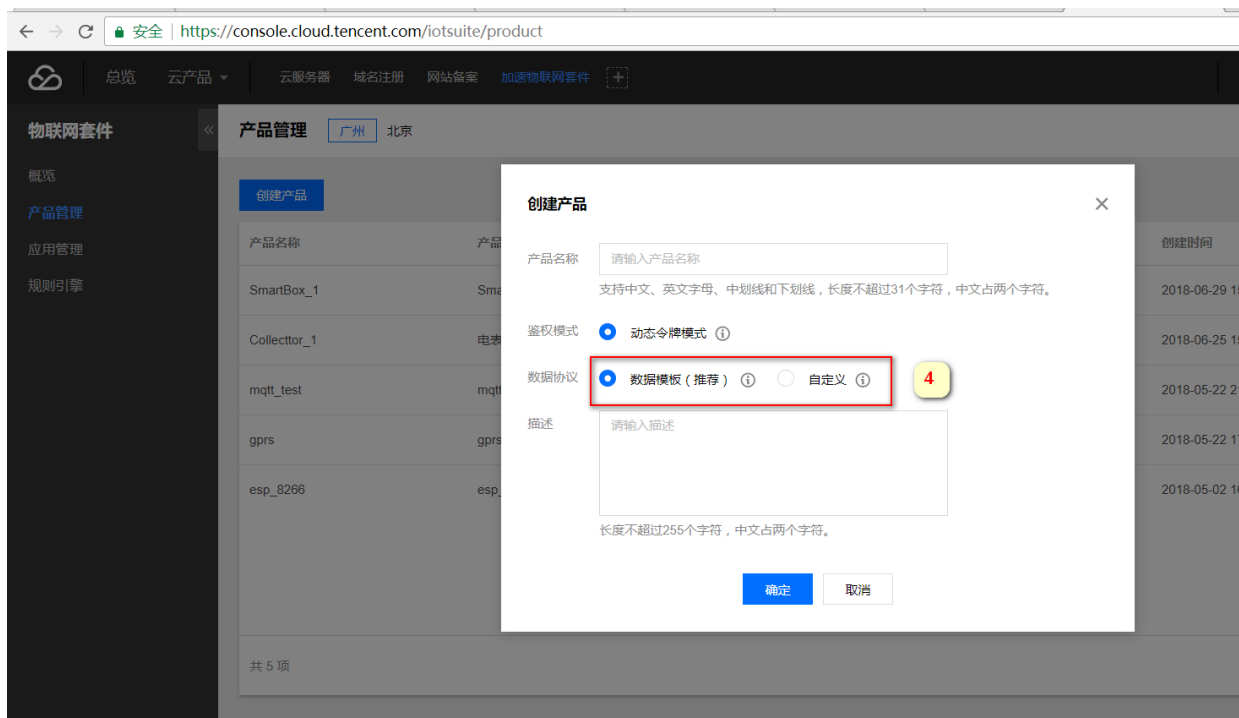
对应到设备端有两个例子。

一个例子是 工程目录/Middlewares/tc_iot/user_logic/example_shadow 这个是数据模板的示例，开发者可以基于这个例子修改自己的业务逻辑实现和云平台进行基于数据模板的通信，数据模板通信的方式下，可以在日志服务页面看到设备上传的数据，如步骤 8 示。

另一个例子是 工程目录/Middlewares/tc_iot/user_logic/example_smartbox 这个是自定义数据的示例，开发者可以基于这个例子修改自己的业务逻辑实现和云平台进行基于自定义数据格式的通信，自定义数据方式下日志服务页面看不到设备上传的数据，只能看到上下线信息。若要观察数据可以同归规则转发的方式查看，参考：http://tapd.oa.com/speed/markdown_wikis/#1010161851007113149

3.4.4. 云平台操作步骤截图





Collector_1

基本信息 数据模板 设备管理 日志服务

基本信息

产品名称	Collector_1
产品描述	电表采集器1
鉴权模式	动态令牌模式
数据协议	数据模板
产品 id ⓘ	iot-eib00wp2
产品 key ⓘ	mqtt-m1j432sw
mqtt 链接地址 ⓘ	mqtt-m1j432sw.ap-guangzhou.mqtt.tencentcloudmq.com
mqtt username	AKIDNgssgTw1pW2NahKR4oRt9D6ofNuGgSKG
mqtt password	fpLAP/AI02G/91KLp4ht/ApHBr1FHcvEndDpod6MZww=
产品创建时间	2018-06-25 15:14:51

5

