

物联网开发平台

实践教程





【版权声明】

©2013-2025 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许可,任何主体 不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责 任。

【商标声明】

🔗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利人所有。未 经腾讯云及有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权 利人商标权的侵犯,腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则,腾讯云对本文档内容不做任何 明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或95716。



文档目录

实践教程

设备接入 RTOS设备通过AT+TCP模组上云

云喇叭接入平台

设备通过DTU接入平台

LoRa LR1110 快速接入平台

LoRa 温湿度传感器接入平台

Heltec HT-M00L 单通道网关接入

RAK LoRa环境监测产品接入平台

基于 TencentOS tiny 接入指引

蓝牙智能灯接入实践

设备 OTA 实践

实践教程 设备接入 RTOS设备通过AT+TCP模组上云

最近更新时间: 2024-11-01 09:43:32

场景简介

针对特定的几款模组,腾讯云物联网开发平台支持定制AT指令集接入,也提供了接入示例,请参考基于 腾讯云IoT定制模组接入。但是对于大部分 的WiFi/蜂窝模组来说,并没有集成定制的腾讯云AT指令集,但是基本所有的模组都支持AT+TCP的方式联网。本实践旨在帮助开发者完成 MCU+AT模组(TCP方式)来连接腾讯云物联网开发平台,完成相关的上云功能开发。

针对市面上MCU和WiFi/蜂窝模组千差万别的情况,本实践使用RT-Thread操作系统来屏蔽MCU的差别,使用 AT Device 软件包来适配不同 的WiFi/蜂窝模组,目前AT Device软件包已经适配了ESP8266、ESP32、M26、MC20、RW007、MW31、SIM800C、W60X、 SIM76XX、A9/A9G、BC26、AIR720、ME3616、M6315、BC28、EC200X、M5311、L610系列等市面上常见的WiFi/蜂窝模组。

开发环境搭建

示例是基于RT-Thread搭建,RT-Thread 与其他很多 RTOS 如 FreeRTOS、uC/OS 的主要区别之一是,它不仅是一个实时内核,还具备 丰富的中间层组件,如下图所示。



腾讯云物联网开发平台也提供了基于RT-Thread的软件包,请访问 这里 获取。由上图可见,客户适配了RT-Thread,并且使用了AT Device 软件包和腾讯云物联网开发平台软件包,则可以屏蔽掉不同MCU和蜂窝/WiFi模组的差异,可以快速开发应用程序并在不同的硬件设备之间移植。

硬件清单

- 1. STM32F103ZET6开发板
- 2. ESP8266 WiFi模组
- 3. Air780E 4G Cat.1模组
- 4. USB转串口
- 5. DAP Link

软件开发环境搭建



完成硬件准备工作后,开始软件适配,关于RT-Thread的相关知识,请访问 官网文档。

1. 首先根据自己项目需要下载rt-thread,推荐下载长期支持的稳定版本。

git clone -b lts-v4.1.x https://github.com/RT-Thread/rt-thread.git

2. 下载env配置环境。

git clone https://github.com/RT-Thread/env-windows

- 3. rt-thread提供了很多硬件的板级支持包,根据自己的硬件,选择合适的bsp,并构建rt-thread工程。
 - 1. 打开env开发环境,找到rt-thread bsp目录
 - \$ cd .\rt-thread\bsp\stm32\stm32f103-atk-warshipv3
 - 2. 使用scons命令来生成rt-thread工程
 - \$ scons --target=mdk5 --dist
 - 3. 将dist目录下的工程拷贝到工作环境中
- 4. 配置工程,使用腾讯云物联网开发平台软件包。

.config - RT-Thread Configuration → RT-Thread online packages → IoT - internet of things → IoT Cloud → tencent-iot-sdk: Tencent Cloud IoT — tencent-iot-sdk: Tencent Cloud IoT Explorer Plat Arrow keys navigate the menu. <enter> selects submenus> (or empty submenus). Highlighted ? features. Press <esc><esc> to exit, <? > for Help, for Search. Legend: [*] built-in [] exclude</esc></esc></enter>	Explorer Platform SDK for RT-Thread — form SDK for RT-Thread — etters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes d <m> module < > module capable</m></m></n></y>
<pre>Intercent-iot-sdk: Tencent Cloud IoT Explorer (@WuKPUCOTC) Config Product Id (dev@01) Config Device Name (' IPB4YDTRCquyOp4w=2) Config Device Secret [, cnable dynamic register [] Enable dynamic register [] Enable multi thread function [] Enable TLS/OTLS Select Product Type (Data template protocc [] Enable Event [] Enable Attion [] Enable Attion [] Enable Smart_light Sample [*] Enable GateWay Version (latest)></pre>	Platform SDK for RT-Thread 1)> wnload by http)>
(Select) < Exit > < Help > < Sav	e > < Load >

5. 使用AT Device软件包,我们以手上的Air780E模组为例(Air720的AT指令集和Air780兼容)。



Contrage of the contragentiation
Ri-Inread online packages 9 101 - Internet of things 9 AF Device, Ri-Inread AF component porting of samples for different device 9 10at Air/20
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</m></esc></esc></m></n></y></enter>
<pre>Inat Air720 [Enable initialize by thread [*] Enable AIR720 at socket [*] Enable sample [*] Formable sample (-1) Power pin (-1) Power status pin (uart3) AT client device name (2048) The maximum length of receive line buffer</pre>
<pre></pre>

6. 更新软件包,更新工程,通过上面的menuconfig配置后,需要拉取软件包并重新配置工程。

1. 拉取软件包			
\$ pkgsupdate			
2. 更新工程			
\$ sconstarget==mdk5			

通过以上六步,我们完成了工程的搭建,接下来就以此为基础,来介绍设备如何接入腾讯云物联网开发平台,以及如何OTA。

△ 注意:

RT−Thread的console和AT都需要串口支持,需要开启相关串口并使能串口驱动。本示例USART1是console串口,USART3是 AT串口。

设备上云

1. 以Keil开发为例,打开刚才配置好的工程如下,里面包含了rt-thread、AT Device和腾讯云物联网开发平台的c-sdk。





2. 配置rtconfig.h,配置腾讯云物联网开发平台生成的三元组信息。

#define PKG_USING_TENCENT_IOT_EXPLORER_PRODUCT_ID "YOUR_PRODUCT_ID"
#define PKG_USING_TENCENT_IOT_EXPLORER_DEVICE_NAME "YOUR_DEVICE_NAME"
#define PKG_USING_TENCENT_IOT_EXPLORER_DEVICE_SECRET "YOUR_DEVICE_SECRET"

3. 编译工程。

```
Build started: Project: project
*** Using Compiler 'V5.06 update 7 (build 960)', folder: 'D:\Keil_v5\ARM\ARMCC\Bin'
Build target 'rt-thread'
compiling main.c...
linking...
Program Size: Code=189506 RO-data=23594 RW-data=1320 ZI-data=12048
FromELF: creating hex file...
After Build - User command #1: fromelf --bin .\build\keil\Obj\rt-thread.axf --output rtthread.bin
".\build\keil\Obj\rt-thread.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:05
```

4. 烧录&运行,如果使用ifconfig命令能查到设备的IP和DNS,则表明入网成功。



```
[I/at.dev] AirM2M_780E_V1157_LTE_AUAT
[I/at.dev] air720 device(air720) GSM network is registered(0,1),
FLAGS: UP LINK_UP INTERNET_UP DHCP_ENABLE
```

5. AT Device软件包支持Socket编程,腾讯云物联网开发平台软件包也做了rt-thread socket的适配,所以可以直接编译运行。

msh />tc_data_template_example start INF|144|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\utils\qcloud_iot_device.c|iot_device_info_set(55): SDK_Ver: 3.2.2e6ff3455c4c0219b9c8448b63e0ef4a8e90be740, Product_ID: 5QQL7PE06Z, Device_Name: dev002 INF|144|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\utils\qcloud_iot_device.c|iot_device_info_set(55): SDK_Ver: 3.2.2e6ff3455c4c0219b9c8448b63e0ef4a8e90be740, Product_ID: 5QQL7PE06Z, Device_Name: dev002 DBG|144|packages\tencent-iot-sdklatest\ports\rtthread\HAL_TCP_rtthread.c|HAL_TCP_Connect(68): establish tcp connection with server(host=5QQL7PE06Z.iotcloud.tencentdevices.com port=1883) msh />DBG|145|packages\tencent-iot-sdklatest\ports\rtthread\HAL_TCP_rtthread.c|HAL_TCP_Connect(101): success to establish tcp, fd=3



c\sdk_src\protocol\mqtt\mqtt_client.c|IOT_MQTT_Construct(132): mqtt connect with id: 502qv c\sdk_src\protocol\mqtt_dient_subscribe.c|qcloud_iot_mqtt_subscribe(147): c\sdk_src\services\data_template\data_template_client.c|_template_mqtt_event_handler(275): $INF | 145 | packages \tencent-iot-sdk-latest \gcloud-iot-explorer-sdk-embedded$ latest\samples\data_template\data_template_sample.c|_register_data_template_property(285): latest\samples\data_template\data_template_sample.c|_register_data_template_property(285):



DBG|146|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\services\data_template\data_template_client.c|_get_status_reply_ack_cb(215): Received Json Document={"method":"get_status_reply","clientToken":"5QQL7PE06Z-1","code":0,"status":"success","data":{}} DBG|146|packages\tencent-iot-sdklatest\samples\data_template\data_template_sample.c|data_template_thread(438): Get data status success

6. 控制台也可以看到设备已经成功上线。

← 返回项目列表 最佳实践	产品开发 / STM32_AT	_Socket				
开发中心	✓ 物模型	→ 设备开发	〉 😧 交互开发 〉	4 设备调试	5 批量投产	
🏟 产品开发	() 设备调试提	共真实、虚拟设备调试功能	,便于测试设备上报、接收数据是否正常,词	可创建测试设备后进行调试		
(<u>)</u> 应用开发						
₩ 数据开发	新建设备	虚拟设备调试			设备名称	▼ 输入设备名称搜索
基础服务	设备名称	状态	激活时间	最后上线时间	操作	绑定网关
◎ 固件升级	dev002	在线	2023-12-21 14:19:43	2023-12-26 15:20:19	调试 二维码 删除	查看
2000 资源管理	dev001	离线	2023-12-13 16:26:39	2023-12-25 18:01:43	调试 二维码 删除	查看
□ 设备管理						
侣 场景联动						

7. 控制台下发开灯指令。

← 返回项目列表 最佳实践	 ✓ 物模型 ◇ 设备开发 ◇ 交互开发 ✓ 设备调试 ✓ 5 批量投产
	← dev002
开发中心	 从2023-08-01开始,日志存储时长将由原来的7天改成3天。
口 产品开发	
(<u>)</u> 应用开发	
式 数据开发	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息
基础服务	下发指令
⑥ 固件升级 ⑧	属性调试 行为调用
2 资源管理	- 功能名称/标识符 期望值 实时数据
➡ 设备管理	✓ 电灯开关(power_switch)
侣 场景联动	
数据流转	亮度(brightness) - 1 + % -
중 运营分析 ∽	颜色(color) Red -
增值服务	
⊙ 位置服务	色温(color_temp) - 0 + % -
♦ 告警中心	0/64
白 网络管理	灯位置名称(name) - 支持英文字母、数字、常见半角符号组合 -
● 语音技能	•
	发送

8. 设备收到并执行相关硬件操作。



INF|368|packages\tencent-iot-sdklatest\samples\data_template\data_template_sample.c|OnControlMsgCallback(259); Property=power_switch changed DBG|368|packages\tencent-iot-sdklatest\samples\data_template\data_template_sample.c|deal_down_stream_user_logic(296); someting about your own product logic wait to be done DBG|368|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\services\data_template\data_template_client.c|IOT_Template_ControlReply(733); Report Document: {"code":0, "clientToken":"v2528561461qhou0::f22c348f-98e5-4501-a3cdd58d715f4a6"} DBG|368|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\protocol\mgtt\mgtt_client_publish.c!qcloud_iot_mgtt_publish(346): publish packetID=0!topicName=Sthing/up/property/50QL7PE06Z/dev002[payload={"method":"control_reply", "code":0, "clientToken":"v2528561461qhou0::f22c348f-98e5-4501-a3cd-d58d7f15f4a6"} DBG|368|packages\tencent-iot-sdklatest\samples\data_template\data_template_sample.c|data_template_thread(473): Contol msg reply success DBG|368|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\protocol\mgtt\mgtt_client_publish.c!qcloud_iot_mgtt_publish(346): publish packetID=0!topicName=Sthing/up/property/50QL7PE06Z/dev002[payload={"method":"control_reply", "coleentToken":"50QL7PE06Z-2", "params":{"power_switch":1} INF]368|packages\tencent-iot-sdklatest\samples\data_template\data_template_sample.c|data_template_thread(473): Contol msg reply success

设备OTA

以STM32F103ZET6开发板为例,该芯片有512KBytes Flash, Flash的划分如下:

FLASH地址	大小	作用
0x08000000~0x080027ff	10 KBytes	bootloader 区,用于检测是否有固件, 以及固件搬运
0x08002800~0x08002fff	2 KBytes	OTA Info 区,存储OTA信息,包括文件 大小等
0x08003000~0x080417ff	250 KBytes	APP 区,代码运行区
0x08041800~0x0807ffff	250 KBytes	OTA Data 区,存储云端下载的固件

设备上电后,首先进入Boot loader来判断是否有新固件,如果有新固件则搬移固件到代码区,搬移完成后跳转到APP区执行,如果没有则跳转 到APP区执行代码。Boot loader流程如下图所示:





而APP代码则是先接入腾讯云物联网开发平台,然后等待平台推送新固件,如果收到新固件则存到OTA Data区,收完后校验固件的完整性,校验 通过则跳转到Bootloader进行升级。流程图如下所示:





() 说明:

关于Bootloader和APP的Flash划分以及详细的开发请参考本文提供的示例工程。

一个完整的OTA升级流程如下:

1. 首先生成一个版本号是mcu_v1.0.1的固件,通过更改宏即可。

#define FW_RUNNING_MCU_VERSION "mcu_v1.0.1"

2. 将编译生成的bin文件上传到腾讯云物联网开发平台。



 ・	固件升级	添加新固件	×
开发中心	添加固件	若固件大小限制不能满足您的业务需求,可提交工单区申请增加文件大小限制	
() 产品开发	固件名称	固件名称 * mcu_v1.0.1升级测试	
(ē) 应用开发	mcu_v1.0.8升级测试 MCU		
₩ 数据开发	ATSocket小固件升级测试 MCU	所属产品 * STM32_AT_Socket ·	
基础服务		□/// MCU ▼	
③ 固件升级			
		回开机体中等。	
		选择固件• 重新选择固件	
数据流转		文件名称: rtthread.bin 文件大小: 208.40KB MD5: 63aa4d0ce2e4c37dbfc623617719bb50	
		仅支持 .bin .dav .tar .gz .zip .gzip .apk .xz .pack 格式的文件,文件大小不能超过 2048MB	
增值服务 ○ 位置服务		固件描述 对本次上传的固件进行描述和记录,请输入0-100个字符	
	共 2 条		
🗋 网络管理			
· 语音技能		对本次上传的固件进行描述和记录,请输入0-100个字符 提交 重置	

3. 更改开发板固件版本为mcu_v1.0.0,编译并运行固件。



[I/at.dev] air720 device(air720) GPRS network is registered(0,1). [I/at.dev] air720 device(air720) attach GPRS tc_ota_example INF|28|packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c|ota_thread_entry(519): c\sdk_src\protocol\mqtt\mqtt_client.c|IOT_MQTT_Construct(132): mqtt connect with id: 07vZ8 INF|29|packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c|ota_thread_entry(538): Cloud Device Construct Success topicName=\$ota/update/5QQL7PE06Z/dev002|packet_id=3981 DBG 29 packages tencent-iot-sdk-latest qcloud-iot-explorer-sdk-embeddedc\sdk_src\services\ota\ota_client.c|IOT_OTA_ResetStatus(138): reset OTA state! c\sdk_src\protocol\mqtt\mqtt_client_publish.c|qcloud_iot_mqtt_publish(339): publish topic



```
c\sdk_src\services\ota\ota_client.c|IOT_OTA_ResetStatus(138): reset OTA state!
DBG 29 packages \tencent-iot-sdk-latest \qcloud-iot-explorer-sdk-embedded-
c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(111):
topic=$ota/update/5QQL7PE06Z/dev002
c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(112): len=91, topic_msg=
c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(111):
INF|32|packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c|process_ota(357): wait
INF |33|packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c|process_ota(357): wait
for ota upgrade command...
```

```
4. 在控制台创建升级任务。
```



添加固件					全部产品	▼ 请输入固件名称	Q,
	固件类型	固件升级		×			
	MCU	固件名称	mcu_v1.0.1升级测试				
	мси	所属产品	STM32_AT_Socket				
		固件类型	MCU				
		固件版本号	mcu_v1.0.1				
		批量升级方式 🕄	按固件版本 按设备名称				
		待升级版本号	mcu_v1.0.0 ▼				
		升级范围	全部设备 ▼				
		升级确认	静默升级				
		超时时长配置 🛈	- 15 + 分钟				
			保存				

5. 设备收到升级请求并开始升级。

同时控制台也显示下载进度:



← 返回项目列表 最佳实践	固件法	升级 / 固件详情	/ 任务详情			腾讯云1073						
开发中心 道: 产品开发		任务信息							任务统计			
··· () 应用开发		任务ID 1 产品名称 5	1817114 STM32_AT_Socket									
試 数据开发 基础服务		源版本号 r	ncu_v1.0.0									
		日标版本号 「 升级范围 含	hcu_v1.0.1 全部设备									
■ 设备管理		升级时间 2 升级方式 4	023-12-26 16:12:59 北景升级							升级中: 100% ● 升级中		
記 场景联动 数据流转 ~		100200										
⊙ 运营分析 ∨		设备详情										
增值服务 ① 位置服务		全部设备(1)	升级成功(0)	设备离线(0)	待推送(0)	已推送(0)	升级中(1)	升级失败(0)	71 60 42-ir	17-1-14.00	刷新	请输入设备名
 		设窗名称 dev002		当即版本号 mcu_v1.0.0		最后更新 2023-12	和印刷		并现 示 载中	不动丰肯 下载中,12%	操作	
 												

6. 升级完成后,设备重启,运行ota sample,固件版本号已经变为mcu_v1.0.1;同时,控制台也显示升级成功。

DBG 137 packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embedded-
c\sdk_src\services\ota\ota_client.c IOT_OTA_Ioct1(663):
origin=63aa4d0ce2e*************23617719bb50, now=63aa4d0ce2e********17719bb50
INF 137 packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c process_ota(438): The
firmware is valid
DBG 137 packages\tencent-iot-sdk-
<pre>latest\samples\ota\ota_mqtt_sample.c _save_fw_data_to_file(285): write index : 412 write len</pre>
DBG 13/ packages\tencent-iot-sdk-
latest\samples\ota\ota_mqtt_sample.c _save_fw_data_to_file(288): save offset [213404] 0
bytes to flash.
DBG 137 packages\tencent-iot-sdk-
latest\samples\ota\ota_mqtt_sample.c _save_fw_data_to_file(290): save the last to flash
0x08075800 412 bytes
DBG 137 packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embedded-
<pre>c\sdk_src\protocol\mqtt\mqtt_client_publish.c qcloud_iot_mqtt_publish(339): publish topic</pre>
<pre>seq=41406 topicName=\$ota/report/5QQL7PE06Z/dev002 payload={"type": "report_progress",</pre>
${\tt INF 137 packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embedded-interval}$
<pre>c\sdk_src\services\ota\ota_client.c IOT_OTA_ResetStatus(138): reset OTA state!</pre>
<pre>INF 138 packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c _event_handler(117):</pre>
publish success, packet-id=41406
<pre>INF 138 packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c ota_thread_entry(586):</pre>
OTA success! jump to bootloader
enter bootloader
get app data. update
try enter app
- RT - Thread Operating System



[I/at.dev] AirM2M_780E_V1157_LTE_AUAT [I/at.dev] air720 device(air720) GPRS network is registered(0,1). [I/at.dev] AirM2M_780E_V1157_LTE_AUAT [I/at.dev] air720 device(air720) GPRS network is registered(0,1). INF |23 | packages \tencent-iot-sdk-latest \samples \ota \ota_mqtt_sample.c | ota_thread_entry (519): c\sdk_src\utils\qcloud_iot_device.c|iot_device_info_set(55): SDK_Ver: 3.2.2-INF|23|packages\tencent-iot-sdk-latest\samples\ota\ota_mqtt_sample.c|ota_thread_entry(538):

腾讯云

4.1.1 build Dec 26 2023 15:15:56





DBG|24|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embedded-INF |24 | packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\services\ota\ota_client.c|IOT_OTA_ResetStatus(138): reset OTA state! DBG|24|packages\tencent-iot-sdk-latest\qcloud-iot-explorer-sdk-embeddedc\sdk_src\services\ota\ota_client.c|IOT_OTA_ResetStatus(138): reset OTA state! DBG 24 packages tencent-iot-sdk-latest gcloud-iot-explorer-sdk-embedded- $\label{eq:intermediate} INF \end{tabular} intermation \end{tabular}$ c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(111): topic=\$ota/update/5QQL7PE06Z/dev002 c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(112): len=91, topic_msg= INF 24 packages tencent-iot-sdk-latest qcloud-iot-explorer-sdk-embedded-INF 25 packages tencent-iot-sdk-latest samples ota ota_mqtt_sample.c process_ota (357): wait for ota upgrade command... c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(111): topic=\$ota/update/5QQL7PE06Z/dev002 c\sdk_src\services\ota\ota_mqtt.c|_otamqtt_upgrage_cb(112): len=94, topic_msg= INF 26 packages tencent-iot-sdk-latest qcloud-iot-explorer-sdk-embeddedfor ota upgrade command...



备详情									
全部设备(1)	升级成功(1)	设备离线(0)	待推送(0)	已推送(0)	升级中(0)	升级失败(0)			刷新
设备名称	当前版本号		最后更新时间 升级状态			状态详情	操作		
dev002		mcu_v1.0.1		2023-12	2-26 16:17:16	升約	级成功	-	

至此,完成升级。

常见问题

使用 RT-Thread 的 AT Device 软件包和腾讯云物联网开发平台软件包对硬件有要求吗?

原则上对硬件没有要求,但是由于 RT−Thread 的组件层导致需要更多的设备Flash空间,请注意选择Flash比较大的芯片。

如果 AT Device 没有支持我的模组怎么办?

AT Device支持了大部分市面上的模组,如果您的硬件不在支持列表里,

- 1. 找相近的模组,例如本实例中 AT Device 列表并没有支持 Air780E 模组,但是支持了 Air720,一般情况下同一厂家的模组AT指令集都是 相同的,仅有个别差别,
- 2. 请求 RT-Thread 官方支持,
- 3. 自己适配,为开源做贡献。

升级过程中出错怎么办?

升级出错可能是模组断开了tcp链路,本示例演示了断点续传功能,如果中间失败会从失败处接着拉取固件,直到升级完成。

固件可以存放到外部flash中吗?

可以的,如果有外部 flash 且适配了文件系统接口,则可以参考腾讯云物联网开发平台 c-sdk 的 ota sample,直接将固件存储到文件系统中。



云喇叭接入平台

最近更新时间: 2025-05-08 11:02:42

场景简介

随着大量的中小商户使用移动支付进行收款的普及,中小商户面临着收款逃单漏单的困扰。云喇叭(收款音箱)作为一款实用的收款播报工具,通 过收款的实时语音播报有效提升商户的收款体验,已成为各支付平台必备的商户服务智能硬件产品。本文档以云喇叭(收款音箱)应用场景为例, 介绍如何通过合宙 Cat.1模组 Air780E 接入腾讯云物联网开发平台以实现语音播报场景物联应用。

硬件设备

需要的硬件全家福如下,合宙官方有相关的开发套餐可以直接购买体验。



1. Air780E 开发板(需要一张可以上网的 SIM 卡)。



2. 音频扩展板。





3. 喇叭。



创建产品和设备

如果还没有开通腾讯云物联网开发平台,请先了解并开通 腾讯云物联网开发平台。

- 1. 登录 腾讯云物联网开发平台控制台。
- 2. 单击公共实例 > 新建项目 > 保存。



_{实例设备数上限} 11164012 _个	新建项目		×	个
	项目名称 *	云喇叭		
		支持中文、英文、数字、下划线的组合,最多不超过20个字符		
	项目描述	腾讯云物联网开发平台,云喇叭示例		日泊重山江山
loT_Test prj-9				2022-08-05 14
shichen_test prj-3		最多不超过80个字符		2022-04-22 10
teenhe_自动化测试勿删 prj-0		保存取消		2022-04-15 14
ning_test prj-8zc	cededy I	50770 4		2021-03-31 22

3. 进入上一步新建的项目,单击**新建产品**,进入新建产品界面。

÷		产品开发 / 新建产品
	最佳头践	
:E	开发中心	新建产品
88	🔅 产品开发	产品名称 * 云喇叭
	(<u>6</u>) 应用开发	支持中文、英文、数字、下划线、空格(非首尾字符)、中英文括号、-、@、\、/的组合,最多不超过4
	式 数据开发	产品品类 标准品类 自定义品类
	基础服务	请选择产品品类
	🐼 固件升级	设备类型 设备 网关 子设备
	20 资源管理	通信方式 * 2G/3G/4G ▼
	🔁 设备管理	请根据业务场景正确选择产品的通信方式,否则会影响后续产品开发
	侣 场景联动	认证方式 密钥认证 证书认证
	数据流转	数据协议 物模型 自定义透传 ①
	⊗ 运营分析 ~	
	增值服务	THKE 选填
	⊙ 位置服务	
	♦ 告警中心	最多不超过80个字符
=	白 网络管理	
—	、8, 语音技能	确定

4. 进入上一步创建的产品界面,在第一步**物模型界面**,单击**导入物模型**,然后将以下物模型复制粘贴到文本框。





```
"desc": "地区区域码",
"name": "基站码",
"desc": "基站码",
"name": "移动网号",
"desc": "移动网号",
"name": "移动国家号码",
"desc": "移动国家号码",
```



```
"name": "网络制式",
"desc": "网络制式",
"name": "心跳间隔",
"name": "信号强度",
"name": "剩余电量",
```



```
"name": "OTA<mark>类型</mark>",
"desc": "指定OTA类型,是否播报提示等\"",
       "0": "不播报系统升级提示",
       "1": "播报系统升级提示"
```



```
"name": "电池电压",
"desc": "电池电压 单位mv",
"name": "充电状态",
      "0": "未充电",
      "1": "正在充电"
"name": "播报结果反馈",
      "name": "幂等ID",
       "name": "收到播报消息时间戳",
```



```
"name": "播报时间戳",
      "name": "播报结果",
             "0": "播报成功",
             "1": "播报失败"
"name": "收款播报",
      "name": "播报内容",
       "name": "收款金额",
       "name": "收款来源",
```



"type": "enum",	
"mapping": {	
"0": "微信",	
"1": " 支付宝 ",	
"2": "云闪付"	
}	
}	
},	
{	
"id": "idempotent",	
"name": "幂等ID",	
"define": {	
"type": "string",	
"mın": "0",	
"max": "32"	
}	
}	
J, "output".	
0012put . [{	
"id": "plav rsp".	
"name": "播报结果",	
"define": {	
"type": "enum",	
"mapping": {	
"0": "加入播报队列成功",	
"1": "幂等ID 重复",	
"2": " 播报队列满",	
"3": " 播报参数错误 "	
}	
}	
}	
1,	
"required": false	
}	
],	
"profile": {	
"ProductId": "xxxxx",	
"CategoryId": "1"	
}	
}	

至此,我们导入了云喇叭专用的物模型。

5. 使能动态注册功能,然后记录这里的生成的产品密钥,后期要放到设备端 SDK 里。关于动态注册的更多内容,请访问 产品级密钥认证 。





至此,控制台的操作就已全部完成。

Air780E 设备端开发

1. 首先拿到合宙的 Air780E 二次开发的 C-SDK 。

t clone https://gitee.com/openLuat/luatos-soc-2022.git

- 2. 根据 Air780E 的二次开发 C−SDK 教程,搭建开发&编译环境,可以正常编译 project 目录下的项目,如果编译完成,可以在 out 目录下生 成相应的*.binpkg文件。如果编译过程中遇到问题,可以根据 C−SDK软件开发资料 排查,或者联系合宙提供技术支持。
- 3. 联系对应腾讯云物联网开发平台的销售或产品经理,拿到适配腾讯云物联网开发平台的云喇叭 SDK,导入到当前工程。
- 4. 修改 qcloud_speaker 工程中的三元组信息,填入上一步在控制台创建的产品的产品 ID、产品密钥,SDK 会在第一次连接腾讯云物联网开 发平台的时候,通过动态注册的方式生成一个设备名为当前模组的 IMEI 的设备,之后会通过这个三元组信息接入腾讯云物联网开发平台。

#define YOUR_PRODUCT_ID	"填入你控制台生成的产品ID"
#define YOUR_DEVICE_NAME	"设备名会获取当前模组的IMEI号,可以不填"
#define YOUR_DEVICE_SECRET	"IOT_PSK" // 无需修改
#define YOUR_PRODUCT_SECRET	"填入你控制台生成的产品密钥"

← 返回项目列表 最佳实践	产品开发 / 云喇叭					
开发中心	→ 物模型 〉 (🕑 设备开发	交互开发	4 设备调试	〉 (5) 批	是投产
🏟 产品开发	 设备调试提供真实、虚 	割以设备调试功能,便于测试	代设备上报、接收数据是否正常	常,可创建测试设备后进行调	式	
() 应用开发						
数据开发	新建设备虚拟设备	调试			设备名称 ▼	输入设备名称搜索
基础服务	设备名称	犬态	激活时间	最后上线时间	操作	绑定网关
🐼 固件升级	8690	王线	2023-11-28 11:09:54	2023-12-01 11:33:51	调试 二维码 删除	查看
日 资源管理						

5. 编译 qcloud_speaker 工程,生成相应的 qcloud_speaker.binpkg 烧录文件。

text data bss dec hex filename 2545256 28536 613166 3186958 30a10e build/qcloud_speaker/qcloud_speaker.elf -outfile ./out/qcloud_speaker/qcloud_speaker.binpkg



[100%]: build ok!

烧录及演示

- 1. 下载合宙 Luatools 最新版本的烧录调试工具。
- 2. Air780E 确认插好能正常上网的 SIM 卡,然后使用 USB-C 线连接 PC。



3. 打开 LuatTools 工具, Air780E 开发板按 POW 按键开机,正常开机后, PC 端设备管理器会枚举出三个 USB 转串口设备,同时 Luatools 也可以打印模组相关日志。





☎ Luatools_2.2.8 账户 圖件相关 选项及工具 合审特色服务 资源目录 帮助	模组相关	长信息		
 ✓ 46機块USB打的 通用串口打印 26機块自动选择串口 COM65 USB 串行设备 (COM65) ● 连接串口 打开串口 重启串口 重启機块 停止打印 清除打印 串口日志波特率 115200 √ 	启动原因: powerkey开机 系统状态: 已取网 固件版本: CSDK_V0003_KC618	信号强度: 26 当前网络: 46 软件类型: esak	「 下 截 薦 件 「 日曽理測试	▲ ● francisgan(甘智伟)的屏幕共享 ▲ ● francisgan(甘智伟)的屏幕共享 メ 推動打印
[2023-12-01 11:02:25.948][00000007.527] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.098][00000007.754] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.183][000000007.784] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.481][00000007.987] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.481][000000007.987] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.481][000000008.114] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.5481][00000008.356] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.978][00000008.403] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.948][00000008.503] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.948][00000008.503] audio_event_cb 72:cloud_speaker_aud [2023-12-01 11:02:25.948][00000008.505] audio_event_cb 72:cloud_speaker_aud	io_task event_cb 6 io_task event_cb 2 io_task event_cb 3 isult=0l	⊼ ⊠		*
QQ支持群(627625461) 淘宝店铺 Air780E资料直达 Air780E的CSDK已发布				有问题doc搜索一下

4. 单击下载固件打开烧录界面,然后找到上一步生成的 /out/qcloud_speaker/qcloud_speaker.binpkg 文件,单击下载。

☜ 下载固件-2.2.8	_		×
固件路径 GitHub\qcloud-iot-air780e\out\qcloud_speaker\qcloud_speaker.binpkg 选择	¥文件	下责	ż
进度	00:00	退出	1
解析固件中			
□ 清除kv分区,AT固件无需关注(仅针对EC618平台) □ 清除文件系统分区,AT固件无需关注(仅针双	寸EC618平台	í ≘)	
固件信息-支持拖拽固件文件到本界面			
平台: EC618			
⁴ 适配的模块: Air780E Air780EG Air600E Air700E			Ē
固件MD5: 10BC4168A6780E659C81FE3048F8D02B			
资料URL: https://air780e.cn			
<u>固件/模块资料链接,点击打开</u>			

5. 先按 BOOT 按键,然后按 RST 按键,让 Alr780E 进入升级模式。





6. 设备进入升级模式,开始烧录固件。

☞ 下载固件-2.2.8	_		\times	
固件路径 GitHub\qcloud-iot-air780e\out\qcloud_speaker\qcloud_speaker.binpkg 选择文	件	下载		
进度 00):21	退出		
连接模块完成,写入固件				
□清除kv分区,AT固件无需关注(仅针对EC618平台) □清除文件系统分区,AT固件无需关注(仅针对EC	618平台	â)		
固件信息-支持拖拽固件文件到本界面				
平台: EC618				
适配的模块: Air780E Air780EG Air600E Air700E				
固件MD5: 10BC4168A6780E659C81FE3048F8D02B				
资料URL: https://air780e.cn				
<u>固件/模块资料链接。点击打开</u>				

7. 烧录完成后,就可以在控制台模拟下发一笔收款。



← 返回项目列表	产品开发 / 云喇叭
最佳实践	
开发中心	
🏟 产品开发	← 8
(<u>0</u>) 应用开发	① 从2023-08-01开始,日志存储时长将由原来的7天改成3天。
₩ 数据开发	
基础服务	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息
🛞 固件升级	下发指令
□ 资源管理	居性调试。 行为调用
📄 设备管理	
铝 场景联动	izsi≉r17.3 •QaAtstip(payment_pasy) ▼
数据流转	参数设置
중 运营分析 √	_{播服内容(content)} test content 12/2048 需要播报的内容
增值服务	支持英文字母、数字、常见半角符号组合
◎ 位置服务	w _{款金额(money)} - 300 + 播报金额,单位:分
● 告警中心	
📋 网络管理	wäxxää wäi 收款来源: 微信、云闪付等
. 语音技能	冪等ID(idempotent) 11223344556677889900aabbccd 零等ID, 防重复播报 支持英文字母、数字、常见半角符号组合

8. 第三方平台可以对接 腾讯云物联网开发平台的云 API 来完成对接。

视频演示

观看视频



设备通过DTU接入平台

最近更新时间: 2024-10-08 18:37:11

场景简介

在工业、农业、商业、园区、楼宇存在着大量通过DTU采集南向设备接入物联网平台的应用场景,这些场景的设备一般需要通过DTU透传上云。 本示例主要展示通过有人G771 Cat.1 DTU如何接入腾讯云物联网开发平台实现数据的收发,然后会进阶接入温湿度传感器完成温湿度上报到物联 网开发平台。本实践整体网络拓扑图如下图所示:



环境准备

1. 所需设备/软件

设备/软件	获取地址	备注
有人DTU一台	https://www.usr.cn/Product/296.h tml	其他品牌的 DTU 也可以
DTU上位机配置软件	https://www.usr.cn/Download/826 .html	其他平台的 DTU 请对应找相关的配置上 位机
USB转RS485一个	自备	-
12V/1A电源适配器一个	自备	给 DTU 供电

2. 设备联网

2.1 正确安装 4G 天线,将 USB 转 RS485 和电源线接入对应端子,确保接线无误,给设备上电。硬件接线示意图如下:




- 2.2 设备正常上电后,红色的POW指示灯会亮起,G771型DTU内置的eSIM卡,上电后可以扫描DTU上的二维码,使用微信小程序配置,这 里先选择默认配置即可。
- 3. 注册有人云账号,查看设备基础信息。
 - 3.1 通过上一步扫描设备上的二维码,这时就可以登录有人云的控制台,查看当前这台设备的基础信息:

*	有人云控制台 www.usr.cn	Ĵ	loT	DM	SI	M 官方前	臹						О Ю	务支持	♥ 用户权限	English	150
83	概览			关管理 〉 M	网关列表	◎ 〉 网关详情											
ē	大屏管理	~	p	网关详情		网络调试	定位轨迹	0									
Ŷ	场景管理	~		网关信息	1											6	1 操作手册
୯	设备管理	~		参数锁已关	闭			状态: 信号强度	在线 强		名称: SN:	腾讯云DTU网关 00	网关型号: IMEI:	US 86	R-G771-CT		
æ	网关管理	^				CXCentery Annual Annual Annual Annual Annual	P				出厂固件: 网络制式:	 4G	固件版本: ICCID:	V2 89			
	网关列表				l		b				标签:						
	组态管理	~															
щ	数据中心	× <		重启网	关	更多、											

3.2 点击更多->编辑->关闭数据透传功能。



编辑		×
* 名称	腾讯云DTU网关	
* 所属组织	腾讯	
标签 ③	添加标签	
网络监测 ②		
数据透传 2		
	取消 确认	

创建产品与设备

如果还没有开通腾讯云物联网开发平台,请先了解并开通 腾讯云物联网开发平台。

- 1. 登录腾讯云物联网开发平台控制台。
- 2. 点击**公共实例->新建项目->保**存。

DTU上云
支持中文、英文、数字、下划线的组合,最多不超过20个字符
Cat.1 DTU上云实践
最多不超过80个字符
保存 取消

3. 点击进入上一步新建的项目,点击**新建产品**,进入新建产品界面。



← 返回项目列表 最佳实践	产品开发 / 新建产品
开发中心	新建产品
🔅 产品开发	产品名称 * DTU上云测试产品
(<u>0</u>) 应用开发	
式 数据开发	产品品类 标准品类 自定义品类
基础服务	设备类型 设备 网关 子设备
🛞 固件升级	·通信士···································
■ 资源管理	
🖃 设备管理	1911日の日本の1915年1月1日の1915年1月1日の1915年1日日の1915年1月月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日の1915年1月1日日日の1915年1月1日日日の1915年1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日
铝 场景联动	
数据流转	数据协议 物模型 自定义透传 ①
⊙ 运营分析 ~	描述 DTU上云测试
增值服务	
⊙ 位置服务	最多不超过80个字符
● 告警中心	
📋 网络管理	确定 取消

▲ 注意:

这里的数据协议需要选择自定义透传。

4. 点击设备调试−−>新建设备,设备名称可以自定义,这里建议使用当前DTU的IMEI作为设备名,方便后期对设备管理。

← 返回项目列表 最佳实践	产品开发 / 4GDTU	
开发中心	✓ 物模型 〉 ✓ 设备开发 〉 ✓ 交互开发 〉 ④ 设备调试 〉 5 批	
☆ 产品开发 (6) 应用开发		
11 数据开发	新建设备 虚拟设备调试 新建设备	S称射 ×
基础服务	设备名称 状态	刘关
@ 固件升级	所属产品 4GDTU 868892072284111 在线	
日 资源管理	设备名称*	
■ 设备管理	支持英文、数字、下划线的组合,最多不超过48个字符	
吕 场景联动	「 の の の の の の の の の の の の の	
数据流转		

5. 至此,我们拿到了设备的三元组信息,可以据此生成MQTT连接参数。



设备连接参	参数		
① 以	下MQTT三元组仅用于调试,将于2024-03-01 00:00:00后失效,正式设备请按 <u>算法生成</u> 🕻		
ClientId	V2ZNBTF 392072284111	MQTT服务器地址	V2ZNBTP3VY.iotcloud.tencentdevices.com
Username	V2ZNBTP3\^^^2284111;12010126;b9d64;1709222400	端口号	1883 🖻
Password	fc55e3f56a434efb793; 300c2b31dfa3484223252f33a8d0f64a6;hmacsha25 6 🕞		

▲ 注意:

控制台生成的MQTT设备连接参数只供测试使用,量产产品不可使用该连接参数,而是需要自行生成过期时间更久的连接参数。参考 算法 生成 。

连接平台

PC连接USB转RS485工具,在设备管理器会枚举出一个新的COM口,打开有人物联网上位机配置工具,连接此COM口,默认串口连接参数:115200,8,N,1;连接成功后先点击进入配置状态,然后点击查询IMEI,如果能正确查询到IMEI码,则表明硬件连接正常可以进入下一步的配置,如果查不到IMEI码则可能是硬件连接故障,请自行排查接线、供电、串口驱动等问题。

😸 USR-CAT1 V	1.1.4		- 0
文件 Language	8		
[PC串口参数]:8	串口号 COM8 - > 波特室 115200 - > 检验/数据/停止 NONI - 8 - > 1	→ 美闲串口	
选择工作模式		执行命令及提示	
○ 网络透传模	式 O HTTPD模式 O 短信透传模式 ● MQTT模式	🗋 获取当前参数	保存所有参数
		进入配置状态 进入道	通讯状态
PC	网络 M2M 设备 串口设备	编译时间查询IMEI查询本机号码	查询版本
MQTT模式相关参	受数	查信号强度 保存参数 恢复出厂设置	设备重启
MQTTClient			
连接	● 通用MQTT ○ 阿里云	☑ 时间戳 □ Hex 接收:38525	复位计数
	服务器地址和減口 dtdoud.tencentdevices.com 1883 用户名 V2ZNBTP3VY868892072284 密码 1bf54e1f61c7;hmacsha256 客户端D NBTP3VY868892072284111 NQTT版本 3.1.1 v	执行完毕 >[Tx->][11:07:16][asc] AT+IMEI >[Rx<-][11:07:16][asc] AT+IMEI	
		+IMEI:868892072284111	
TLS加密参数		ОК	
🗌 SSL加密		执行完毕	
全局参数			
串口参数	串ロ波特率 11520(〜 流控 NONE 〜 检验/数据(停止 NONE 〜 8 〜 1 〜	61 62 63 64 31 32 33 34 35 36 37 38 39 30	
- trán	打包时间(ms) 50 打包长度(Bytes) 1024	通过串口发送 ▼	❷ 发送 ▼

2. 配置MQTT连接参数,测试阶段可以使用腾讯云物联网开发平台控制台生成的MQTT连接参数。



选择工作模式								
○ 网络透传机	夏式 〇	HTTPD模式	() 短信透传樽	試	● MQT	「T模式]
PC	MQTT	网络	MQTT	► M2M 设备	串口≸	数据	串口设备	
MQTT模式相关	参数							
	服务	S器地址和端口 用户名 密码 客户端II MQTT版Z	iotclou	d.tencentde TP3VY86889 e1f61cc7;hm VY86889207	vices.com 92072284 acsha256 72284111	1883		^
	重 MQTT串I MQTT	连时间间隔(秒 口传输模式设置 心跳包时间(秒) 10 透传档) 120 	夏式	~	 		~

⚠ 注意:

工作模式选择**MQTT模式,**MQTT相关参数**请直接复制粘贴**腾讯云物联网开发平台的MQTT连接参数。

3. 订阅/发布腾讯云物联网开发平台的raw topic。

品开发 / 4GDTU		
物模型 2 设备开发 2	(3) 交互开发 〉 (4) 设备调试	t > (5) 批量投产
o备开发 Topic列表 云端解析		
透传Topic 系统级Topic 自定义Topic	;	
透传Topic 系统级Topic 自定义Topic Topic权限	操作权限	备注
透传Topic 系统级Topic 自定义Topic Topic权限 \$thing/up/raw/V2ZNBTP3VY/\${deviceName} \$thing/up/raw/V2ZNBTP3VY/\${deviceName}	· 操作权限 发布	备注 透传协议上行,并支持云端解析成JSO

MQTT模式相关参	数					
订阅主题	☑ 主题1	\$thing/down/raw/V2ZNBTP3	0	~		^
	□ 主题2	/SubTopic2	0	~		
	□ 主题3	/SubTopic3	0	\sim		
	🗌 主题4	/SubTopic4	0	\sim		
推送主题	☑ 主题1	\$thing/up/raw/V2ZNBTP3VY	0	\sim	🗌 消息保留	1.0
	□ 主题2	/PubTopic2	0	\sim	🗌 消息保留	
	🗌 主题3	/PubTopic3	0	\sim	🗌 消息保留	
	🗌 主题4	/PubTopic4	0	~	□ 消息保留	

△注意:

这里的\${deviceName}请替换为之前在腾讯云物联网开发平台新建的设备名称。

4. 点击配置上位机的 **设置并保存所有参数**按钮,完成配置。



执行命令及提示						
。	前参数	🔋 设置并保有	存所有参数			
进入配置	状态	进入通讯状态				
编译时间	查询IMEI	查询本机号码	查询版本			
查信号强度	保存参数	恢复出厂设置	设备重启			
 ☑ 时间戳 □ Hex 接收	ू र:42798		复位计数			
>[Tx->][11:17:17][asc] AT+MQTTCFG=120,0 >[Rx<-][11:17:17][asc] AT+MQTTCFG=120,0			^			
ок						
>[Tx->][11:17:17][asc] AT+MQTTWILL=0						
61 62 62 64 21 22 22	24.25.26.27.29.2	7	~			
01 02 03 04 31 32 33	34 33 30 37 38 3	JA 20				
通过串口发送 🔸 🕢 🗹	Hex 发送:225		🔍 发送 🕞			

5. 设置完成后,DTU的LINK1黄灯常亮且已经成功连接到腾讯云物联网开发平台,可以在控制台查看设备的在线状态。

← 返回项目列表 最佳实践	产品开发 / 4GDTU		
开发中心	・ 物模型 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	交互开发 交互开发 设备调	试 〉 (5) 批量投产
🏟 产品开发	 设备调试提供真实、虚拟设备调试功能,便于 	则试设备上报、接收数据是否正常,可创建测试设备后进行	 行调试
(<u>6</u>) 应用开发			
🕄 数据开发	新建设备虚拟设备调试		设备名称 ▼ 輸入设备名称搜索
基础服务	设备名称	激活时间 最后上线时间	操作
🐼 固件升级	868892072284111 在线	2023-12-01 16:07:30 2023-12-02 11:17:29	调试 二维码 删除 查看
日 资源管理			

6. 通过配置上位机发送数据测试。



>[Tx->][11:20:10][asc] abcd1234567890		~
abcd1234567890		
通过串口发送 🔸 🗌 🖽	ex 发送:239	❷ 发送 ▼

7. 腾讯云物联网开发平台收到 DTU 的上行数据。

← 返回项目列表 最佳实践	产品开发 / 4GDTU
开发中心	 ◇ 物模型 > ◇ 设备开发 > ◇ 交互开发 > ④ 设备调试 > 5 批量投产 ◆ 86i 11
(ē) 应用开发	 从2023-08-01开始,日志存储时长将由原来的7天改成3天。
💥 数据开发	
基础服务	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息
🐼 固件升级	透传日志 内容日志 上下线日志 • 口 自动刷新
■ 资源管理	
🔁 设备管理	日志类型 二进制透传 ▼ topic \$thing/up/raw/V2ZNBTP3VY/868892072284111, \$thing/do ▼
铝 场景联动	30分钟 1小时 今天 昨天 近3天 2023-12-02 10:50 ~ 2023-12-02 11:20 首
数据流转	时间 通讯类型 Topic 通信内容
⊗ 运营分析 ✓	\$thing/up/raw/\/22NBTP3\/V/86889207228411
增值服务	2023-12-02 11:20:15.412 上行 1 YWJjZDEyMzQ1Njc4OTA=

8. 控制台的上行数据是base64编码后展示的,如需要源数据则需要base64解码后查看。

YWJjZDEyMzQ1Njc40TA= base64 decode ---> abcd1234567890

9. 当需要从云端控制使用自定义透传协议上云的设备时,可参考 设备透传指令控制 云 API 直接使用 在线调试 工具进行调试。

温湿度传感器通过DTU接入平台

控制台创建温湿度传感器产品

1. 和上面的步骤一样,不过这次我们在当前项目下,创建一个Modbus温湿度传感器的产品,产品的物模型如下,主要是创建了一个温度属性, 一个湿度属性。

```
{
"version": "1.0",
"properties": [
{
"id": "lac",
"name": "地区区域码",
```



"desc": "地区区域码",

```
"name": "基站码",
"desc": "基站码",
"name": "温度",
"name": "湿度",
```



"required": false		
}		
],		
"events": [],		
"actions": [],		
"profile": {		
"ProductId": "J6XJ01DT63",		
"CategoryId": "1"		
}		
}		

2. 同样新建一个设备,拿到设备的MQTT连接参数。

设备连接参数		
③ 以下MQTT三元组仅用于调试,将于2024-03-03 00:00:00后失效,正式设备请按 <u>算法生成</u>		
Clientid J6XJ0 dev001 T	MQTT服务器地址	J6XJ0 33.iotcloud.tencentdevices.com
Username J6XJ01D1	端口号	1883 🖬
Password 83e462668a6d6208f73550a3e0a24aef7c9c2		

3. 重新配置DTU的MQTT参数,让新建这台设备上线。

⊦友 / Modbus温湿度传感器		
✓ 物模型 〉 ✓ 设备开发 〉 ✓ 交互开发	> 4 设备调试 > 5 批量投产	
- dev001		
() 从2023-08-01开始,日志存储时长将由原来的7天改成3天。		
设备信息 在线调试 云端诊断日志 设备云端日志	设备本地日志 扩展信息	
设备信息		
设备名称 dev001 	产品ID J6XJ01DT63 屆	所属产品 Modbus温湿度传感器
设备密钥 Zt puw== 值	设备创建时间 2023-12-03 13:49:47	最后上线时间 2023-12-03 17:08:21
激活时间 2023-12-03 15:16:47	设备状态 在线	固件版本 -
	_	

<u>小 注意</u>:

通过上位机配置DTU的MQTT连接参数时,订阅和发布的topic也需要同步更改为新设备的。

温湿度传感器的配置

 首先通过USB转RS485用PC的串口助手来调试温湿度传感器,请求温湿度和响应都正常后,再接入DTU。以本次使用的ZTS-3008-WS-N01型温湿度传感器为例。首先查阅其规格书,默认通讯波特率是4800,需要更改为115200来适配DTU,根据规格书更改波特率的指令直接 修改。

•4.4.4 修改当前波特率↔

腾讯云

问询帧: (假如设备当前地址是1波特率为4800,要修改为115200) ↔

地址码↩	功能码↩	起始地址↩	修改数值↩	校验码低位。	校验码高位
0x01←	0x06←	0x07 0xD1←	0x00 0x06←	0x58<⊐	0x85↩
地址码↩	功能码↩	起始地址↩	修改数值↩	校验码低位。	校验码高位
0x01←	0x06↩⊐	0x07 0xD1←	0x00 0x06↩□	0x58<⊐	0x85<⊐

▲ 注意:

应答帧: ↩

这里修改波特率前使用的还是其默认波特率4800,发送完该指令且温湿度传感器应答完成后,再切换串口助手的波特率为115200。

2. 查询温湿度传感数据

根据其规格书描述发送查询指令来查询。

查询传感器(地址为1)的数据(湿度,温度),主机→从机

地址	功能码	起始寄存 器地址高	起始寄存 器地址低	寄存器 长度高	寄存器 长度低	CRC16 低	CRC16 高
0X01	0X03	0X00	0X00	0X00	0X02	0XC4	0X0B

若传感器接收正确,返回以下数据,从机→主机

地址码	功能码	返回有效 字节数	湿度值	温度值	校验码 低位	校验码 高位
0x01	0x03	0x04	0x01 0xE6	0xFF 0x9F	0x1B	0xA0

温度计算:

当温度低于 0 ℃ 时温度数据以补码的形式上传。 温度: FF9F H(十六进制)= -97 => 温度 = -9.7℃

湿度计算:

湿度: 1E6 H (十六进制)= 486 => 湿度 = 48.6%RH

串口助手可以正确返回温湿度:

TX:01 03 00 00 00 02 C4 0B RX:01 03 04 00 E3 00 FA 8B 86

这样我们确认温湿度传感器工作正常,然后接入DTU。

温湿度数据上报云平台

通过前面的两个步骤,我们的准备工作完成,现在可以通过控制台来查询温湿度传感器的数据。整个查询上报的流程如下所示:





硬件接线图如下图所示:



由于温湿度传感器和DTU是Modbus通讯,DTU上报给腾讯云物联网开发平台的数据也是Modbus的hex数据,所以,需要将这些数据转换成平 台识别的物模型数据,这样我们也能更直观的观测数据,也方便后台的进一步处理和前端的展示。这里就需要用到我们的云端数据解析功能。

🔗 腾讯云

1. 开启云端数据解析功能。

产品开发 / Modk	us温湿度传感器				
✓ 物模型	2 设备开发	3 交互开发	4 设备调试	5 批量投产	
设备开发	Topic列表 云端解析				
是否使用证	设备数据解析功能 ①				

2. 编写云端数据解析函数,解析上下行数据。

```
// 上行数据解析函数, fPort为LoraWAN的端口号,这里可以不用关注。bytes: raw topic上报的裸数据
```



// 5. 返回构造好的数据到云平台

return da

3. 调试上行解析函数:将上面的上行解析函数粘贴到上行解析函数框中,然后填入一条测试上行数据,点击运行。如果解析函数编写正确,则右侧 运行结果框中会打印正确的report数据模板消息。

11.2.3冊一句32.1店牌					
:行数据解析	0	语法: JavaScript	下行数据解析	0	语法: JavaScript
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35	<pre>console.log(bytes); // 1. check crc16 if (!checkCrc16(bytes)) { return {}; } // 2. check is temp&RH if (bytes[0] != 0x01 bytes[1] != 0x03) { return {}; } let data = { "method": "report", "clientToken": new Date().getTime().toSt "params": {} }; </pre>	:ring(),	1 func 2 3 4 5 6 7 8 9 }	<pre>tion ProtocolToRaw(obj) { console.log(obj) if (obj.method == "action" && obj console.log(obj.params.addr) console.log(obj.params.functi return [0x01, 0x03, 0x00, 0x0 } return [];</pre>	.actionId == "get_temp") on) 0, 0x00, 0x02, 0xc4, 0x€
36 37 38	<pre>data.params.temperature = (bytes[5] << 8 b data.params.humidity = (bytes[3] << 8 byte</pre>	oytes[6]) / 10 es[4]) / 10.0;	-		
36 37 38 以调试 设备上行数: 1 [1,	data.params.temperature = (bytes[5] << 8 b data.params.humidity = (bytes[3] << 8 byte 屬① 设备下行数据 ③ 此处 3, 4, 0, 227, 0, 250, 139, 134] 2	bytes[6]) / 16 es[4]) / 10.0; 输入(Port的值	运行结果 1 { 2 "" 3 "" 4 "p 5 " 5 7 } 8 〗	<pre>method": "report", .lientToken": "1701657675938", aarams": { "temperature": 25, "humidity": 22.7</pre>	查看数据模板JSOI 4

- 4. 让DTU定时上报温湿度,如果您的DTU有边缘计算和二次开发能力,则编写一个定时任务,定期给温湿度传感器发送轮询温湿度的指令,然后 将传感器的上报数据发送到腾讯云物联网开发平台。本文使用的USR-G771并没有提供二次开发能力,但是它有一个心跳功能我们可以加以利 用,达到一样的定期上报功能。
 - 4.1 首先通过DTU配置上位机,打开DTU的心跳功能,然后心跳发送方式选择向串口终端发送心跳包。类型选择自定义,数据就是轮询温湿度 传感器的数据。勾选为hex数据。配置完成后点击设置并保存所有参数。



选择工作模式			
◉ 网络透传模式	○ HTTPD模式	○ 短信透传模式	○ MQTT模式
一定要被	选择网络透传博	試, 才能配	置心跳包
PC	网络	M2M 设备	串口设备
网络透传模式相关参	数		
☑ 启用心跳包	心跳时间(秒) 心跳发送方式 心跳数据类型 自定义数据	60 向串口终端发送心跳包 自定义数据 01030000002C40B	 ✓ ✓ ✓ Hex
🗌 启用注册包			

4.2 重新通过上位机将DTU配置为MQTT模式,然后点击设置并保存所有参数。

≙	注意:			
	有人DTU的心跳包只能在网络透传模式里配置,	所以需要先配置为网络透传模式,	配置好心跳包后,	再切换为MQTT模式。

5. 通过以上步骤,就可以定期将温湿度数据上报到腾讯云物联网开发平台了。

← 返回项目列表	产品开发 / Modbus温湿度传感器
取住大阪	 ・ 物模型
₽ 产品开发	← dev001
(0) 应用开发	从2023-08-01开始,日志存储时长将由原未的7天改成3天。
式 数据开发	
基础服务	设备信息 在线调试 云端诊断日志 设备去端日志 扩展信息
⑥ 固件升级 ⑧	透传日志 内容日志 上下线日志 🗘 🌑 自动刷新
1. 资源管理	
🖃 设备管理	
铝 场景联动	上行 下行 30分钟 1小时 今天 昨天 近3天 2023-12-04 10.34 ~ 2023-12-04 11.04 📋
数据流转 🗸	时间 通讯类型 Topic 数据
ᢙ 运营分析 √	2023-12-04 11.04.18.157 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method":"report","clientToken":"1701659058145","params", "temperature".25.6,"humidity".27.9))
增值服务	2023-12-04 11 03:18:078 上行 \$thing/up/raw/J6XJ01DT63/dev001 {"method":"report";"clientToken":"1701658998067";"params"; "temperature" 25.6,"humidity" 27.9)}
◎ 位置服务	2023-12-04 11 02-18 230 E/行 \$thino/up/raw/J6XJ01DT63/dev001 ("method" "recort" "clientToken" "1701658938217" "params" ("temperature" 25.6 "humidity" 27.5)
◎ 告警中心	
白 网络管理	2023-12-04 11:01:18:157 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method":"report","clientToken":"1701658878142","params": ["temperature":25.5,"humidity":27.9])
. 语音技能	2023-12-04 11.00.18.116 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method":"report","clientToken":"1701658818099","params","Temperature" 25.5,"humidity" 27.6))
	2023-12-04 10 59:18:053 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method": "report", "clientToken": "1701658758038", "params", "temperature" 25.5, "humidity" 27.6))
	2023-12-04 10 58:18.154 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method": "report", "clientToken": "17016586898138", "params", "temperature" 25.4, "humidity" 27.6))
	2023-12-04 10 57:18.088 上行 Sthing/up/raw/J6XJ01DT63/dev001 ("method":"report","clientToken":"1701658638074","params","temperature":25.4,"humidity":27.5))

步进电机通过DTU接入平台

控制台创建步进电机产品

和之前的步骤一样,首先在腾讯云物联网开发平台控制台创建一个步进电机产品:



← 返回项目列表 最佳实践	产品开发 / 新建产品
2000, 1-2-2, 2, 2-2, U	
开发中心	新建产品
🛱 产品开发	产品名称 * Modbus步进电机
(<u>6</u>) 应用开发	支持中文、英文、数字、下划线、空格(非首尾字符)、中英文括号、-、@、\、
式 数据开发	产品品类 标准品类 自定义品类
基础服务	设备类型 设备 网关 子设备
🐼 固件升级	
2 资源管理	通信方式 * 2G/3G/4G ▼ 请根据业务场景正确选择产品的通信方式,否则会影响后续产品开发
🔁 设备管理	认证方式 廖钼认证 证书认证
侣 场景联动	
数据流转 ~	数据协议 物模型 自定义透传 ①
⊗ 运营分析 ~~~	描述选填
增值服务	
◎ 位置服务	最多不超过80个字符
♦ 告警中心	
白 网络管理	确定取消

然后导入如下物模型:







```
"name": "电机运动方向",
"name": "加速度",
"desc": "范围0 - 255,0为直接启动,1 - 255设置曲线加减速的快慢",
"name": "转速",
```





步进电机有很多参数需要配置,这里只是取有代表性的三个属性来做示例,具体请根据不同应用来做相应调整。

步进电机配置

笔者手上是一个ZDT-X42_V1.2步进电机,带控制器和Modbus通讯,关于电机的详细资料,请点击这里获取。硬件调试接线方式如下:



- 1. 电机初次上电需要校准,校准时电机不要带负载。
- 2. 电机使用Modbus RTU协议,默认通讯115200,8N1,请在电机控制菜单中选择Checksum为Modbus。
- 3. 使用USB转RS485模块发送读取固件版本和硬件版本的指令,如果接线和供电都正常,则会收到相应的回复。



5. 读取固件版本和硬件版本

主机请求RTU帧(主机 → 驱动板)							
		寄存器地址		寄存器数量		CRC16校验码	
	シノ月ビロラ	Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	1FH	00H	01H	00H	0CH
		从机响	立RTU帧 (驱动板	→ 主机)			
				寄存器1	存器1 CRC16校验码		
从机地址	功能码	功能码 字节数	Hi		Lo	Hi	Lo



云平台控制步进电机

通过上面的步骤,电机已经可以通过USB转485成功控制,将电机接入DTU,硬件连接如下:



🔗 腾讯云

1. 通过腾讯云物联网开发平台控制台创建电机设备。

← 返回项目列表 最佳实践	产品开发 / Modbus步	进电机				
开发中心	→ 物模型	→ 设备开发	〉 交互开发	→ 设备调试	〉 5 批量投产	
🔅 产品开发	 设备调试提供 	共真实、虚拟设备调试功能	,便于测试设备上报、接收数据是否正	正常, 可创建测试设备后进行调试		
(<u>©</u>) 应用开发						
₩ 数据开发	新建设备	虚拟设备调试			设备名称	▼ 输入设备名称搜索
基础服务	设备名称	状态	激活时间	最后上线时间	操作	绑定网关
🐼 固件升级	motor001	未激活	-	-	调试 二维码 删除	查看
■ 次活答Ⅲ						

2. 打开云端解析功能。

✓ 返回项目列表 局佳实践	产品开发 / Modbus步进电机	
开发中心	✓ 物模型 〉 2 设备开发 〉 3 交互开发 〉 4 设备调试 〉 5 批量投产	
🛱 产品开发	设备开发 Topic列表 云端解析	
 应用开发 数据开发 	是否使用设备数据解析功能 ③	

3. 将步骤1新建的设备的MQTT连接参数通过DTU配置工具填入DTU,使得这台设备成功上线。

← 返回项目列表 最佳实践	产品开发 / Modbus步	进电机				
开发中心	→ 物模型	→ 设备开发	〉 😧 交互开发	4 设备调试	> (5) 批量投产	
☆ 产品开发 ○ 応田开发	() 设备调试提(共真实、虚拟设备调试功能,	便于测试设备上报、接收数据是否正常,	可创建测试设备后进行调试		
₩ 数据开发	新建设备	虚拟设备调试			设备名称	▼ 輸入设备名称搜索
基础服务	设备名称	状态	激活时间	最后上线时间	操作	绑定网关
⑥ 固件升级	motor001	在线	2023-12-20 15:34:35	2023-12-20 15:34:35	调试 二维码 删除	查看

4. 编写下行数据解析函数,调试通过后提交。

// 生成 CRC16 校验码
function createCrc16(buffer) {
<pre>let crc = 0xFFFF;</pre>
let odd;
for (let $i = 0$; $i < buffer.length - 2$; $i++$) {
<pre>crc ^= buffer[i];</pre>



```
// 下行测试数据,用于调试
   // 通过DTU发送的电机控制数据
     sendBytes[8] = obj.params.a; // 加速度
```



行数据解析	f 🕐	语法: JavaScript 下行	う数据解析 ②		语法: Java
1 fund	ction RawToProtocol(fPort, bytes) {		12	}	
2	<pre>return {};</pre>		13	}	
3 }			14 }		
			15 buf	fer[buffer.length - 2] = crc	& 0xff;
			16 buf	ffer[buffer.length - 1] = crc	>> 8;
			17 ret	urn buffer;	
			18 }		
			19	1	
			20		
			21 functio	on ProtocolloRaw(obj) {	
			22 let	sendBytes = [0x01, 0x10, 0x0	0, 0x+6, 0x00, 0x03, 0x
			23 11	(obj.method == control) {	
			24	sendBytes[8] = obj.params.di	r'j
			26	sendBytes[9] = obj.params.sn	eed & Øxff.
			27	<pre>sendBytes[10] = (obj.params.</pre>	speed $>> 8$ & 0xff:
				cneate(nc16(sendBytes)	
			28	CLEACECLCTO(SELIGDYCES)	
以调试	握 ① 设备下行数据	17	28		奇藝教授描編
以调试 设备上行数:	据 ① 设备下行数据	ĬZ	28 27 27 28 28 28 28 28 28 28 28 28 28 28 28 28		查看数据模板
以调试 译备上行数: 1 【	据 ① 设备下行数据	Ĭ	28 行结果		查看数据模板
以调试 设备上行数 1 { 2	据① <u>设备下行数据</u> "method": "control",	jz	行结果 1 2 1,		查看数据模板
以调试 设备上行数:	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5	jZ 2254008a4f10",	7 行结果 1 2 1, 3 16,		查看数据模板
以调试 运备上行数:	据① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	254008a4f10", }	28 行结果 1 [2 1, 3 16, 4 0, 5 246		查看数据模板
以调试 全备上行数: 1 { 2 3 4 5 }	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	254008a4f10", }	28 行结果 1 [2 1, 3 16, 4 0, 5 246, 6 0		直看 数据模板
以调试 全备上行数: 1 { 2 3 4 5 }	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2-9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	254008a4f10", }	28 行结果 1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3		查看数据模板
以调试 全备上行数: 1 [2 3 4 5]	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	1254008a4f10", }	28 行结果 1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6.		查看数据模板
以调试 全备上行数: 1 { 2 3 4 5 }	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	1254008a4f10", }	7 5 4 6 7 3 6 6 6 9 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	4	查看数据模板
以调试 全备上行数: 1 【 2 3 4 5 】	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	1254008a4f10", }	28 行结果 1 【 2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6, 9 0, 10 3,	4	查看数据模板
以调试 全备上行数: 2 3 4 5 }	据 ①	i254008a4f10", }	おけて 活结果 1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6, 9 0, 10 3, 11 1,	4	查看数据模板
以调试 2备上行数: 3 4 5 }	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2-89-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	i254008a4f10", }	おけた 新生産 新生産 新生産 新生産 新生産 新生産 新生産 新生産	4	查看数据模板
以调试 2备上行数: 1 { 2 3 4 5 }	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	i254008a4f10", }	お 行结果 1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6, 9 0, 10 3, 11 1, 12 0, 13 0,	4	查看数据模板
以调试 足备上行数: 1 【 2 3 4 5 】	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	5254008a4f10", }	28 行结果 1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6, 9 0, 10 3, 11 1, 12 0, 13 0, 14 0,	4	查看数据模板
以调试 全备上行数: 1 [2 3 4 5]	据 ① 设备下行数据 "method": "control", "clientToken": "20a4ccfd-d308-2e9-86c6-5 "params": { "speed": 1, "a": 3, "dir": 0	3254008a4f10", }	1 [2 1, 3 16, 4 0, 5 246, 6 0, 7 3, 8 6, 9 0, 10 3, 11 1, 12 0, 13 0, 14 0, 15 70,	4	查看数据模板

5. 编写上行数据解析函数,调试通过后提交。

// 校验 crc16 是否正确
<pre>function checkCrc16(buffer) {</pre>
<pre>let crc = 0xFFFF;</pre>
let odd;
for (let $i = 0$; $i < buffer.length - 2$; $i++$) {
<pre>crc ^= buffer[i];</pre>
for (let j = 0; j < 8; j++) {
odd = crc & 0x0001;
crc >>= 1;
if (odd) {
crc ^= 0xA001;
}
}
}
<pre>return (buffer[buffer.length - 1] << 8 buffer[buffer.length - 2]) == crc;</pre>
}
// 测试数据



```
const testBytes = [1, 16, 0, 246, 0, 3, 96, 58];
// 上行数据解析
function RawToProtocol(fPort, bytes) {
    console.log(bytes);
    // 1. check cro16
    if (!checkCrc16(bytes)) {
        console.log("crc16 error");
        return {};
    }
    // 2. 这里只是对速度控制的上行数据做解析,如果有其他的上行数据 都可以在这里处理,如电机主动上报的一些数
据,可以用report来解析上报
    if (bytes[0] != 0x01 || bytes[1] != 0x10) {
        return {};
    }
    let data = {
        "method": "control_reply", // 这里只是解析了速度控制的上行数据,构造了一条物模型数据
        "clientToken": new Date().getTime().toString(),
        "code": 0,
        "status": "success",
    };
    console.log(data)
    return data;
}
```



<pre>20 function RawToProtocol(fPort, bytes) { 21 console.log(bytes);</pre>	
<pre>22 // 1. check crc16 23 if (!checkCrc16(bytes)) { 24</pre>	<pre>1 function createCrc16(buffer) { 2 let crc = 0xFFFF; 3 let odd; 4 5 for (let i = 0; i < buffer.length - 2; i++) { 6 crc ^= buffer[i]; 7 for (let j = 0; j < 8; j++) { 8 dod = crc & 0x0001; 9 crc >>= 1; 10 if (odd) { 11 crc ^= 0xA001; 12 } 13 } 14 } 15 buffer[buffer.length - 2] = crc & 0xff; 16 buffer[buffer.length - 1] = crc >> 8; 17 return buffer; 18 1 运行结果 运行结果 查看数据模板JS 1 1 2 1 2 1 3 2 1 3 2 3 3 3 4 3 5 5 3 5 5 5 5 5 5 5 5 5 5 5 5</pre>

6. 上下行数据解析函数编写完成后,就可以通过在线调试来控制电机。



	产品开发 / Modbus 步进电机		腾讯云IoT技术交流群 💬 操作指南
 	 ◇ 物模型 〉 ◇ 设备开发 〉 ◇ 交互开发 〉 ④ 设备调试 〉 ⑤ 批量投产 ← motor001 ● N2023.08.01TH会 日志存続時に協由商主的7天改成3天 		
() 単成のの			
基础服务 ② 固件升级	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息 下发指令	调试日志	清空日志 💟 显示明应报文 💟 自动削新
10 页原目理 	周性调试 行为调用	时间 上下行	日志类型
I 场景联动	■ 功能名称ψ示识符 期望值 实时数据	▶ 2023-12-20 16:24:38.756 ↑ 上行	设备属性控制回复
数据流转	□ 地区区域码(lac) (只读) 数值范围: 0-32,初始值: 0,步长: 1,单位: -	▼ 2023-12-20 16:24:38.051 ↓ 下行	设备属性控制
♂ 运营分析 ∨	↓ 基站码(old) (只读) 数值范围: 0-32,初始值: 0,步长: 1,单位: -		
增值服务 位置服务 () 位置服务 () 告警中心 	 ✓ 电机运动方向(dir) CW CW<	3回前 2023-12-20 16:24:38.051 日志央型 没名属性控制 method "control" clientToken "v2528561461Wbnsa::28634ae params ("dir":0,"a*:10,"speed":500)	a-b93c-44d8-ac15-2667398484bb*
📋 网络管理	✔ 转递(speed) - 500 + rpm 500	▶ 2023-12-20 16:24:19.530 ↑ 上行	设备属性控制回复
④ 语音技能	22	▶ 2023-12-20 16:24:18.350 ↓ 下行	设备属性控制

每一条控制指令下行后,电机执行完成都会返回一条上行指令,通过上行解析函数后的日志如下:

← 返回项目列表	产品开发 / Modbus步进电机 腾讯云loT技
Re 主 Sel D8 开发中心 (注) 产品开发	 ◇ 物模型 > ◇ 设备开发 > ◇ 交互开发 > ◆ motor001
(Ô) 应用开发	0 从2023-08-01开始,日志存储时长将由原来的7天改成3天。
🔛 数据开发	
基础服务	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息
@ 固件升级	
日 资源管理	
🖃 设备管理	<i>順</i> 004日志 新竹日志
侣 场景联动	上行 下行 30分钟 1小时 今天 昨天 近3天 2023-12-20 16.04 ~ 2023-12-20 16.34 首
数据流转 🗸	时间 通讯类型 Topic 数据
♂ 运营分析 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	2023-12-20 16 24 38 710 上行 Sthing/up/raw/JPEKX9WWY4/motor001 ["method":"control_reply","clientToken":"1703060678686","code".0,"status":"success")
増値服务 ① 位置服务	2023-12-20 16 24 37 984 Trifs Sthing/down/raw/JPEKX9WWY4/motor001 ("method":"control":"clientToken:"\2528561461Wbnsa: 28634aea-b93c-44d8-ac15-2667398484bb";"params": ("dirt", "a" 10, "speed": 500))
告警中心	2023-12-20 16 24 19 482 上行 Sthingfup/raw/JPEKX9WWY4/motor001 ("method":"control_reply";"clientToken":"1703060659461";"code" 0;"status":"success")
📋 网络管理	2023-12-20 16 24:18 286 TF(7 Sthing/down/raw/JPEKX9WWY4/motor/01 ("method":"control","clientToken":"v2530389123qsjem::1896236-9c9a-4c99-82e5-8d13l3291021","params", ("speed" 0,"dir".0,"a" 0))

Q&A

1. 我使用的DTU无法定时发送Modbus轮询指令,该怎么操作?

答:可以使用云端解析脚本的下行解析函数,由云端来发送一个轮询的物模型协议指令,然后解析成Modbus协议的轮询指令下发给DTU,再 由DTU发送给温湿度传感器,但是这样就不能做到定时上报了。

2. 我的DTU无法获取传感器数据

- 答: 1. 先使用USB转RS485串口工具接传感器,模拟DTU来发送轮询指令,看传感器是否正常上报;如异常:
 - a. 检查硬件连接是否正常;
 - b. 根据传感器规格书检查串口波特率是否匹配(包括和DTU的串口波特率是否匹配)



- 2.再使用USB转RS485串口工具接DTU,看是否正确收到DTU的轮询指令
- 3. DTU数据正常上报,但是无法解析出正确的物模型数据
 - 答: 1. 首先在腾讯云物联网开发平台控制台查询上报的裸数据是否正确。

产品积	开发/ Modbus	温湿度传感	器							
(✔ 物模型 ← dev001		✔ 设备开	rg >)交互开发		4 设备调试	> 5	批量投产
	() 从2023-0	8-01开始,	日志存储时长	将由原来的7天	改成3天。					
	设备信息	在线调	周试 云	端诊断日志	设备	云端日志	设备本	地日志 扩展信	急	
	透传日志	内容日	志 上	下线日志	φ	自动刷	新			
	原始日志	解析日	志							
	上行	下行	30分钟	1小时	今天	昨天	近3天	2023-12-04 11:03	~ 2023-12-04 1	1:33
	时间		ĩ	通讯类型	Торіс			数据		
	2023-12-04 11:33:18.040 上行 2023-12-04 11:32:18.243 上行		上行	\$thing/u	up/raw/J6XJ0	1DT63/dev001	AQM	EARUBBmuZ		
			上行	\$thing/u	\$thing/up/raw/J6XJ01DT63/dev <mark>切aSe64</mark> 解释		ase64解研	兩查看渡	数据	
	2023-12-04 11	1:31:18.131	-	L行	\$thing/u	up/raw/J6XJ0	1DT63/dev001	I AQM	EARgBBbpb	
	2023-12-04 1	1:30:18.113	-	L行	\$thing/u	up/raw/J6XJ0	1DT63/dev001	I AQM	EARYBBduY	

2. 检查解析函数是否编写正确,可以将上面的裸数据填入模拟调试框中,来调试上行解析函数。



LoRa LR1110 快速接入平台

最近更新时间: 2024-11-27 17:34:33

背景

LoRa Edge 是 Semtech 在 2020 年推出的面向资产管理应用的产品系列,支持 GPS 及北斗卫星扫描、无源 Wi−Fi 扫描等多重定位技术,同 时集成了远距离 LoRa 收发器。其中 LR1110 是该系列的第一款芯片,主打特色是超低的定位功耗以及极具吸引力的 BOM 成本。



LR1110 在 GNSS 定位的功耗优势主要在于将传统的卫星解算流程进行优化。芯片本地只扫描空口的卫星导航电文,将导航电文上报到云平台进 行详细位置解算,从而节省了芯片本地的工作时长和运算需求。因此,LR1110 需要配套云平台进行使用。

腾讯云与 Semtech 双方达成协议,LoRa Edge 的 GNSS 及 Wi−Fi 扫描数据将在腾讯云物联网开发平台直接进行解算获得位置信息,在提升 访问速度的同时,也进一步满足相关应用的合规性要求。

目前针对中国用户,腾讯云物联网开发平台已经支持包括 LR1110 在内的 LoRa Edge 系列产品定位功能,用户还可以利用腾讯连连小程序、loT Enable 等功能快速开发特色应用。

🕛 说明:

这篇文章先总体介绍 LoRa Edge 接入腾讯云物联网开发平台的系统架构,接着分别介绍各组件的具体开发细节,包含节点、网关、物联 网平台及腾讯连连小程序的操作。

系统架构

🔗 腾讯云

系统架构图参考。



LR1110 节点操作

1. 节点硬件准备



Semtech LR1110 EVK, 采用 STM32L476RG Nucleo 底板 + LR1110 shield 的形式,两个 shield 分别支持不同的天线:

- PCB_E592V01B does not have a LNA and connects to the long antenna.
- PCB_E516V02B includes one LNA and can only use the short antenna.

我们采用 PCB_E592V01B,配套一根长的无源天线,方便测试。

2. 节点软件操作

节点软件推荐参考 LoRa Basics Modem Geolocation SDK。



腾讯云 IoT Explorer 控制台操作

腾讯云物联网开发平台完整的 使用手册说明。 LoRaWAN 产品完整的 使用手册说明。

1. LoRaWAN 网关接入

说明:
 用户的 LoRaWAN 网关需支持 Packet Forwarder 协议。

LoRaWAN 网关上的配置需做如下调整:

配置接入域名: loragw.things.qcloud.com 接入端口: 1700

详细的网关接入操作及腾讯云物联网开发平台的操作,可以查看文档 LoRaWAN 网关管理 。

2. 腾讯 LoRa 社区网络

除了自建网关之外,值得一提的是还可以借助腾讯 LoRa 社区网络实现更广的网络覆盖,可以极大方便 LR1110 的测试。 尤其是在深圳,可以看到周围有腾讯开放的一些社区网关,采用 80~87 上行信道,也就是 486.3 ~ 487.7 MHz。



3. 创建产品及设备

- 1. 创建产品。
- 产品品类:智慧生活-安防报警-定位器。您可按需选择产品品类,举例产品品类的好处是在腾讯连连中有一个地图免开发面板,方便查看位置。
- 设备类型:设备。
- 认证方式:密钥认证。
- 通信方式:LoRaWAN。



2. 物模型添加标准功能。

菜单操作为:物模型 > 添加标准功能 > 通用类型 > 勾选"wifi定位"、"GNSS导航电文"。

- "wifi 定位"为扫描的 AP 信息,包含 MAC 地址和 RSSI。
- "GNSS 导航电文"为视野卫星的导航电文。
- 3. 使能 LoRa Edge 定位。

在菜单设备开发 > LoRaWAN 参数配置中使能相关操作。

推荐软件 LoRa Basics Modem Geolocation SDK 默认使用的是 GNSS Middleware FPort 及 Wi-Fi Middleware FPort。

LoRa Edge 定位	
Modem FPort	- 199 +
	仅支持整数, 1-223之间
GNSS FPort	- 198 +
	仅支持整数, 1-223之间
WiFi FPort	- 197 +
	仅支持整数, 1-223之间
GNSS Middleware FPort	- 194 +
	仅支持整数, 1-223之间
WiFi Middleware FPort	- 196 +
	仅支持整数, 1-223之间

4. 新建设备。

在设备调试页面,单击新建设备。按照之前源码中配置的 DevEUI 和 AppKey 来创建设备。

4. 查看设备日志

LR1110 节点如果上电工作,且 LoRaWAN 网关保持上线,那么控制台中就会看到设备数据更新。 设备属性页面中可以看到数据更新:

Qa field Qa field Qa field Qa field Qa field $f = 0$ $f = 0$ Qa field Qa field $f = 0$ Qa field Qa field $f = 0$ $f =$)物模型 > 🕢 设备开发 > 🕢 交互开发 > 🚺 设备调试 > 5 批量投产 tr 24							
标识符功能名称历史数据数据 类型最新值更新问GPS_ExtGPS盘位富者字符幕・・・・・・・・・・・・battery_state电池情况重着整数型・・・・・・・・・・・・・・・positioning_mode世様式直者牧学型・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	设备信息 设备属性	设备日志 设备事件	设备行为 设备上下线日志	在线调试 扩展信息	设备调试日志			
GPS_ExtGPS定位重着字符串· · · · · · · · · · · · · · · · · · ·	标识符	功能名称	历史数据	数据类型	最新值	更新时间		
battery_state电池博况重者整改型・・positioning_mode定位模式重者枚举型型・・LBS_BS器法定位童者字符串・・Wifi_Infowifi定位童者数组【(Mac'1/58/7156/7.65/78/84'92), (Mac'1/705/944/066/78/84'94)]2021-08-2417/1507Wifi_Unfowifi定位童者数组【(Mac'1/68/94/066/78/84'94), (Mac'1/241/68/36/86)/78/84'94)]2021-08-2417/1507	GPS_Ext	GPS定位	查看	字符串				
positioning_mode 定位模式 重着 枚萃整型 · · · · · · · · · · · · · · · · · · ·	battery_state	电池情况	查看	整数型	-	-		
LBS_BS 器站定位 重音 字符串 · · · · · · · · · · · · · · · · · · ·	positioning_mode	定位模式	查看	枚举整型		•		
Wifi_Info WifiE位 算過 数组 [('Mac''58b715ca7c6';Rssi'-92), ('Mac''7cb59b4d7068';Rssi'-61), ('Mac''7cb59b4d7068';Rssi'-61), ('Mac''17cb59b4d7068';Rssi'-61), ('Mac''17cb59b4d7068';Rssi'-61), 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:	LBS_BS	基站定位	查看	字符串		-		
	Wifi_Info	wifi定位	查看	数组	[('Mac':'b8b7f15ca7c6';'Rssi''-92), ('Mac':'7cb59b4d7066';'Rssi''-61), ('Mac':'089b4b0065c9';'Rssi'-31), ('Mac':'04f02190dc6a';'Rssi'-50), ('Mac':'f241c8e3b168';'Rssi''-46)]	2021-08-24 17:15:07		
142-2412 小田 142-2412 142-2412 142-2412	GNSS_NAV	GNSS导航电文	查看	字符串				

设备日志页面中可以看到设备原始数据:



tr <mark>4</mark>				
设备信息 设备属	性设备日志	设备事件 设备	行为 设备上下线日	日志 在线调试 扩展信息 设备调试日志
上行 下行	30分钟 1小时	今天 昨天	近7天 近30天	2021-08-24 16:47 ~ 2021-08-24 17:17
时间	通讯类型	Topic		数据
2021-08-24 17:17:02	上行	\$thing/up/raw/H	!4	FCcxBMYII758tZtNcGjcYMWod19JygTwlZDcatkIm0sAZcnb8kHI47Fool2dOf02HcyWDBWgUjFrLF6ZSCIdXK4fXhJfjqldwJZ/kQ+a9nZQvL mTmPA=
2021-08-24 17:16:32	上行	\$thing/up/raw/H	!4	FCcKBMYII6a4t/FctTDbYMWod19J3AlbSwBlycgE8CGQ3GrN8kHI47FooOMD5Qc0gRRG+cQLesyeQnS0bB+/W9TEAXEdhDYJd31521L3Bi GNSqsx3xk=
2021-08-24 17:16:04	上行	\$thing/up/raw/H	14	FCfjA8YII9tgxah3X0mmuLfxXLUw4AibSwBlyc0E8CGQ3GrS8kHI47Foo0YPCJxL1bx0qRakg0c0FmahsRgV34WM3pLAUDE=
2021-08-24 17:15:36	上行	Sthing/up/raw/H	34	FBnKA7glFdxgxah3X0m2BPAhkNxqzvJByOOxaKC+6csgX89wCCTAarErFTMoJa4BL23H8Rb02UJ52ksgnF7J4raXPHS4Vw92
2021-08-24 17:15:07	上行	\$thing/up/raw/H	:4	FCejA8YII6S4t/Fcp8bDfLWbTXBo4QibSwBlyc4E8CGQ3GrS8kHI47FooEEeeEpUCDPts7EUQwpObBpYFmOunDPPruWCb91rwduV6svlW60r b1KAhLQ=

5. 位置空间操作

如果想要在控制台看到设备的实时位置,您需要在**增值服务 > 位置服务**新建位置空间。

新建空间				×
空间名称 *	myspace			
	支持中文、英文、数字、下	划线的组合, 最多	不超过20个字符	
关联产品 *	搜索产品	Q,	mytracker	×
	✓ mytracker			
	实物OTAA			
		4	→	



打开位置空间,将会看到最近的实时位置。

全部设备		请选择地理團栏 ▼	ぴ 历史轨迹 (热力图 (点) 合 热力图 (面)
	o 方正证券		2021-08-24 17:00:0	▶ ↓ 10 ~ 2021-08-24 17:59:59
	深圳市国兽房地产土地 估价颜问有限公司	○上上诉书房		2049(深圳湾 科技园区)
	A	。 深圳湾万丽酒店		道义虾子张 六羊肉粉
	高新南环路			5
				Ľ
10 米				+

位置空间还有更丰富的功能,可以查看历史轨迹、热力图,以及操作围栏等。

全部设备		请选择地理围栏 ▼	𝔅 历史轨迹 ○ 热力图 (点) ○ 热力图 (面)
	○ 方正证券		O ► ☆ 2021-08-24 17:00:00 ~ 2021-08-24 17:59:59
	深圳市国普房地产土地 估价顾问有限公司	○上上茶书房	2049(深圳湾 科技园区)
		深圳湾万丽酒店	◎ 道义\$F子张 六羊肉粉
	高新南环路		۲
10 * _			+
✓ 展讯地图 ©2021 Tencent - GS(2020)1720号 - Data© Navinfo			

6. 通过腾讯云 API 获取设备历史位置

腾讯云 API 提供位置服务相关接口,用户可以通过 API 的方式来获取位置服务相关信息,参考 位置服务相关接口 。

7. 第三方应用

第三方应用如果想要获得设备原始数据,可以使用**基础服务**中的**数据同步**功能,将数据通过 HTTP 或者 CKafka 推送到第三方服务器。

使用腾讯连连小程序

1. 扫码绑定设备

设备调试页面中单击目标设备的二维码,打开"腾讯连连"小程序扫码绑定设备。



✓ 物模型	〉 → 设备开发	> 😧 交互开发 >	4 设备调试	〉 5 批量投产		
 设备调试损 	a 設設 設 設 設 設 設 設 設 設 設 設 設 設 設 設 設 設 設	于测试设备上报、接收数据是否正常,可	可创建测试设备后进行调试			
新建设备	虚拟设备调试			设备名称	▼ 设备名称	Q
设备名称	状态	激活时间	最后上线时间	操作	绑定网关	
tr		2021-08-24 17:13:17	2021-08-25 09:41:28	调试 二维码 删除	查看	

2. 打开面板

交互开发页面默认是"标准面板",展现产品 json 物模型。如果要在小程序中看到地图,需要调整为"免开发面板"。

 ◇ 物模型 ◇ 没备 ← 编辑面板 	π <u></u>	3 交互开发	> (4 设备调试	> (5)批量投产
国板类型 免开发面板	Ŧ					
 温馨提示 1. 購訊云免开发面板为您提供 2. 若免开发面板提供的功能不 	;了产品常用的标准功能, 满足您的产品需求,您可	可帮助您快速完成 J从 <u>"开源免开发面标</u>	产品开发。 <u>反" 12</u> 中获取(代码进行修改,再通过	1H5自定义面板开	F发
10.19						
保存						

3. 小程序界面



打开腾讯连连小程序,即可看到设备的实时位置。





LoRa 温湿度传感器接入平台

最近更新时间: 2024-11-27 17:34:33

操作场景

假设一款 LoRa 温湿度传感器接入到物联网开发平台,通过物联网开发平台可以远程查看传感器的温度、湿度,并可远程配置传感器的上报周期。 本文档主要指导您如何在物联网开发平台控制台,接入 LoRa 温湿度传感器。

前提条件

为了通过下面的步骤快速理解该业务场景,需要做好以下准备工作:

- 申请物联网开发平台服务。
- 意法半导体 (ST) P-Nucleo-LRWAN3 开发套件。
- 拥有一台物理或虚拟的 Windows 环境,并根据 基于 TencentOS tiny 的 LoRaWAN 开发入门指南 的介绍完成开发环境搭建,包括 MDK 软件的安装及配置、ST-Link 驱动安装、串口软件的安装。

控制台操作 LoRa 节点

创建项目

- 1. 登录物联网开发平台控制台,选择新建项目。
- 2. 在新建项目页面,填写项目基本信息后,单击保存。
 - 项目名称:输入"LoRa 温湿度传感器演示"或其他名称。
 - 项目描述:按照实际需求填写项目描述。
- 3. 项目新建成功后,即可新建产品。

新建产品

- 1. 进入该项目的产品列表页面,单击新建产品。
- 2. 在新建产品页面,填写产品基本信息后,单击保存。
 - 产品名称: 输入"温湿度传感器"或其他产品名称。
 - 产品类型:选择"用户自定义"。
 - 设备类型:选择"设备"。
 - 认证方式:选择"密钥认证"。
 - 通信方式:选择"LoRaWAN"。



新建产品		×
产品名称★	支持中文、英文、数字、下划线的组合,最多不超过20个字符	
产品品类	用户自定义 🔹	
设备类型	设备 网关 子设备	
认证方式	证书认证 密钥认证	
通信方式	Wi-Fi 2G/3G/4G 5G BLE LoRaWAN 其它 ③	
数据协议	数据模板 自定义透传	
描述	选填	
	最多不超过80个字符	
	保存取消	

3. 产品新建成功后,您可在产品列表页查看"LoRa 温湿度传感器"。

创建数据模板(可选步骤,透传设备可不操作)

单击产品名称,进入产品配置页,在**数据模板**的"自定义功能"配置项下,单击**新建功能**,自定义产品功能。

自定义功能						
新建功能	您可以通过自定义功能按	需定义功能				
功能类型	功能名称	标识符	数据类型	读写类型	数据定义	

配置 LoRaWAN 参数

在设备开发页面中,按需调整 LoRaWAN 参数配置。本示例中使用默认的 OTAA 配置。

LoRaWAN参数配置	
协议版本	V1.0.2
加网方式	ΟΤΑΑ
设备类型	CLASS A
RX1 Delay	1秒
RX2 DR	0
RX2 Frequency	505.3 MHz

设备数据解析(可选步骤,透传设备可不操作)

在设备开发页面中,按需开启或关闭设备数据解析功能。

• 由于 LoRa 类资源有限设备不适合直接传输 JSON 格式数据,使能"设备数据解析"可以将设备原始数据转化为产品 JSON 数据。
\bigcirc

• 若未开启设备数据解析功能,则设备可进行透明数据传输。

是否使用设备数据解析功能 🛈

设备数据协议

在本示例中,设备上行数据共4字节:

○ 第1字节:温度。

腾讯云

- 第2字节:相对湿度。
- 第3、4字节:表示上报周期(单位秒)。

设备下行数据为2字节:上报周期(单位秒)。

数据解析脚本

上行数据解析的脚本主函数为 RawToProtocol,其带有 fPort、bytes 两个入参:

- fPort:设备上报的 LoRaWAN 协议数据的 FPort 字段。
- bytes:设备上报的 LoRaWAN 协议数据的 FRMPayload 字段。

脚本主函数的出参为产品数据模板协议格式的对象。

在上行数据解析部分,javascript 示例代码如下:

```
function RawToProtocol(fPort, bytes) {
   var data = {
        "method": "report",
        "clientToken" : new Date(),
        "params" : {}
   };
   data.params.temperature = bytes[0];
   data.params.humidity = bytes[1];
   data.params.period = bytes[2] | (bytes[3] << 8);
   return data;
}</pre>
```

下行数据解析的脚本主函数为 ProtocolToRaw,其入参为产品数据模板协议格式的对象,其出参为至少3个字节的数组:

- 第1字节:下发给设备的 LoRaWAN 协议数据的 FPort 字段。
- 第2字节: bytes 为下发给设备的 LoRaWAN 协议数据的 MType(0表示 Unconfirmed Data Down,1表示 Confirmed Data Down)。
- 第3字节:开始为下发给设备的 LoRaWAN 协议数据的 FRMPayload 字段。

在下行数据解析部分,javascript 示例代码如下:

```
function ProtocolToRaw(obj) {
    var data = new Array();
    data[0] = 5;// fport=5
    data[1] = 0;// unconfirmed mode
    data[2] = obj.params.period & 0x00FF;
    data[3] = (obj.params.period >> 8) & 0x00FF;
    return data;
}
```



设备数据解析 通过编写数据解析脚本,将 LoRa 产品的上下行原始数据转化成数据模版协议中的产品 JSON 数据。	您可以对脚本进行模拟测试	武,运行正常后可发布	
上行数据解析 ②	语法: JavaScript	下行数据解析 ⑦	语法: JavaScrip
<pre>1 function RawToProtocol(fPort, bytes) { 2 var data = { 3 "method": "report", 4 "clientToken" : new Date(), 5 "params" : {} 6 }; 7 data.params.temperature = bytes[0]; 8 data.params.humidity = bytes[1]; 9 data.params.period = bytes[2] (bytes[3] << 8); 10 return data; 11 } </pre>		<pre>1 function ProtocolToRaw(obj) { 2 var data = new Array(); 3 data[0] = 5;// fport=5 4 data[1] = 0;// unconfirmed mode 5 data[2] = obj.params.period & 0x00FF; 6 data[3] = (obj.params.period >> 8) & 0x00FF; 7 return data; 8 }</pre>	1

脚本模拟测试

您也可使用数据解析页面下方的模拟调试工具,如需开发更多的功能,请使用以下模拟脚本。

• 上行消息

设备原始数据为 0x11451E00,我们将其转化为数组,即上行模拟数据为:[17,69,30,0],填入设备上行数据的编辑框中。单击**运行**,即可在 模拟调试界面右侧查看结果。



• 下行消息

模拟测试数据如下,将其填入设备下行数据的编辑框中:

"params": {			
"period": 15			



模拟调试		
设备上行数据 设备下行数据	运行结果	看数据模板JSON
1 { 2 "params": { 3 "period": 15 4 } 5 }	1 [2 5, 3 0, 4 15, 5 0 6]	I
运行 提交		

创建测试设备

在设备调试页面中,单击新建设备,设备名为 dev001。

新建设备		×
所属产品	ble子设备	
设备名称 🗙		
	支持英文、数字、下划线的组合,最多不超过48个字符	
	保存取消	

控制台操作 LoRa 网关

- 1. 登录 物联网开发平台控制台,选择上文 "控制台操作 LoRa 节点"中对应的项目。
- 2. 在左侧工具列表中,选择网络管理 > LoRa 网关管理。
- 3. 进入 LoRa 网关管理页面,选择社区网络 > 添加网关。

LoRa网关管理			
社区网络	用户网关	用户自定义频点	
添加网关	腾讯LoRa社	区网络为开发者提供免费接入服务,	目前共有 923 个在运行的网关。

- 4. 在新建网关页面,填写网关基本信息。
 - 网关名称:本示例中填写 GW1。
 - GwEUI:为网关唯一 ID。本例中根据 ST NUCLEO LoRa GW 背部的 MAC 地址,将6字节 MAC 地址的中间补足 0xffff。





○ 是否公开:选择"是",表示社区开发者可在社区网络查看该网关,并可通过这个网关进行 LoRa 节点接入;选择"否",则仅用户自己 能查看该网关。

基本信息		
网关名称★	GW1	\odot
GwEUI *		
是否公开	●是 ○ 否	
网关描述		
频点信息		
用户自定义	频点 LoRaWAN _) 🔻	
位置信息		
⊠域★	北京市 ▼ 东城区 ▼ 东华门街道 ▼	${\boldsymbol{\oslash}}$
详情	东城区东华	
经度		\odot
纬度		\odot
确定添加	返回列表	

5. 网关新建成功后,您可在网关列表页查看"GW1"。

LoRa 网关实物操作

硬件连接

整个系统搭建需要由 LRWAN_GS_LF1 网关(网关模组和 STM32F746 Nucleo 核心板)、5V 适配器和电脑组成。

- 1. 先使用 5V 适配器通过 USB 线连接到 LRWAN_GS_LF1 网关的网关模组上的 Micro USB 接口,给整个网关供电。
- 2. Nucleo 核心板上的 Micro USB 口(非以太网口那边的 Micro USB 口),通过 USB 线连接到 PC 端,可以实现虚拟串口的功能。



3. 网关开发板通过网线连接到上一级路由器的 LAN 口,从而可以实现 DHCP 的方式连接以太网。



串口准备

1. 硬件连接成功后,打开 PC 上的设备管理器,即可查看网关所对应的串口(请确保已安装 stlink 驱动)。

- ✓ 算 端□ (COM 和 LPT)
 - 🛱 Intel(R) Active Management Technology SOL (COM3)
 - STMicroelectronics STLink Virtual COM Port (COM5)
 - 🚆 通信端口 (COM1)

2. 打开串口工具,做好相应配置后,打开串口。

- 端口号。本例中为 COM5。
- 波特率。本例中为 115200。

隆 SSCOM V5.13.1 串口/网络数据调试器	_		\times
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助			
UPLINK UDP PORT: 1780 DOWNLINK UDP PORT: 1780 CHANNEL1: 471500000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL1: 471700000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL2: 471900000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL3: 472100000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL4: 472300000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 472500000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 472500000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 472700000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 472700000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 472900000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF) CHANNEL5: 0FF (LORA_STANDARD) CHANNEL9: 0FF (LORA_STANDARD)			^
Concentrator starting Concentrator Radio A type SX1255 Concentrator Radio B type SX1255 Concentrator started (2926ms) ST LoRa GW V2 Ethernet starting Ethernet starting Ethernet started DHCP IF: 192.168.3.249 Downlink UDP Connected Vplink VDP Connected			~
	刻—		
端口号 COME STMicroelectronics ST. □ HEX显示 保存数据 「接收数据到文件」 HEX发送 定时发送: 100 ms/次 ms/次 小回车: ● 关闭串口 ● 理多串口设置 □ 加时间戳和分包显示, 超时时间: 20 ms 第1 字节 至 末尾 ▼加校验 None ▼ ■ RTS □ DTR 波特率 115200 ▼ AT+Reset ● 发送 次迎使用专业串口调试工具SSCOM ! ●	<mark>换行</mark> 2		
S:10 R:2100 COM5 已打开 115200bps,8,1,None,None	СТ	S=0 DS	R=0



配置修改

请按照如下步骤完成相关配置:

- 1. 按照上图完成硬件连接和系统搭建。
- **2. 配置服务器地址。本示例中设置的是腾讯云物联网开发平台的 LoRa 服务器地址(接入域名:** loragw.things.qcloud.com , 接入端口: 1700)。

AT+PKTFWD=loragw.things.qcloud.com,1700,1700

3. 配置频率计划。调整频点信息到486.3MHz - 487.7 MHz,指令修改如下(需要逐条发送):

示例截图如下所示:

Downlink VDP Conne	cte	d		
Uplink UDP Connect	ed			
AT+PKTFWD=loragw.t	hin	gs. qcloud.	com, 1700, 1	1700
+PKTFWD: loragw.th	ing	s. geloud. c	om, 1700,	1700
AT+CH=0, 486.3, A				
+CH: 0, 486300000,	А,	SF7/SF12,	BW125KHz	(LORA_MULTI_SF)
AT+CH=1, 486.5, A				
+CH: 1, 486500000,	A,	SF7/SF12,	BW125KHz	(LORA_MULTI_SF)
AT+CH=2, 486. 7, A				
+CH: 2, 486700000,	А,	SF7/SF12,	BW125KHz	(LORA_MULTI_SF)
AT+CH=3, 486.9, A				(
+CH: 3, 486900000,	А,	SF7/SF12,	BW125KHz	(LORA_MULTI_SF)
AT+CH=4, 487.1, B	-			(
+CH: 4, 487100000,	В,	SF//SF12,	BW125KHz	(LUKA_MOLTI_SF)
AT+CH=5, 487. 3, B		082 (0840	DULLOCIAL	(TOD) MOTOT OR
HCH: 5, 487300000,	В,	SF7/SF12,	BW125KHz	(LUKA_MOLII_SF)
AI +U, H=0, 487.5, B		082 (0840	DULLOCIAL	(TOD) MODET OR
+C.H.: 6, 487500000,	В,	SF7/SF12,	BW125KHz	(LUKA_MOLII_SF)
AI+CH=7,487.7,8		CR7 (CR10	DULLOCIAL	(TODA HOUTT CR)
TCR: (, 40((00000),	Д,	51(/5112,	DWIZOVUZ	(TORY_WOFIT_2L)
AI TONTO, UFF				(LODA CTANDARD)
AT ACK-O OFF				(DORK_SINNDARD)
HCH 9 OFF				(RSR)
AT+log=op				(ron)
HUG ON				
AT +RESET				
+RESET: OK				
Restarting				
line in the second s				

4. 其他指令。

○ 通过 "AT+log=on" 打开网关日志。

○ 通过 "AT+EUI" 查询网关的 ID。

运行

通过 AT+Reset 即可复位网关,开始服务器连接。 从串口日志查看:

LORAWAN SERVER: loragw.things.qcloud.com



表明服务器地址修改成功。

Ethernet started DHCP IP: 192.168.3.249 Downlink UDP Connected Uplink UDP Connected

表明网关 DHCP 入网成功,网络连接正常。

LoRa 节点实物操作

编译及下载

Step 1. 下载 LoRaWAN 例程

- 1. 请下载 TencentOS tiny 官方开源仓 下载源码 。
- 2. 进入 <TencentOS-tiny\board\NUCLEO_STM32L073RZ\KEIL\lorawan> 目录,打开 TencentOS_tiny.uvprojx 工程。
- 3. 示例工程包含 STM32L073 外设驱动、TencentOS tiny 内核、AT 框架、RHF76 LoRaWAN 模组驱动、LoRaWAN 示例案例。

Step 2. 代码修改

1. 请先修改 \examples\LoRaWAN\lora_demo.c.。

cos_lora_module_join_otaa("8cf957200000f806", "8cf957200000f8061b39a****d204a72");

填入节点相应的 DevEUI 和 AppKEY,可从 LoRa 节点开发板背面贴纸上获取。

2. 修改 \devices\rhf76_lora\RHF76.h。

#define RHF76_ATCMD_SET_CHANNEL

"at+ch=num, $0-7\r\n$ "

由于本示例中计划使用80-87信道,因此调整为:

#define RHF76_ATCMD_SET_CHANNEL

"at+ch=num,80-87\r\n

Step 3. 编译

单击 MDK 工具栏Rebuild All,编译整个工程。

Step 4. 下载

单击 MDK 工具栏Download,下载编译好的固件。

查看运行结果



1. 节点下载好固件后,会自动重启运行,从串口即可查看设备的运行日志。



2. 当您看到串口打印如下日志,即说明 LoRa 节点已经通过网关成功入网。



查看设备状态

- 1. 保持 LoRa 节点和 LoRa 网关 为运行状态。
- 2. 进入控制台 > 产品开发 > 设备调试,可查看到设备 "dev001"。
- 3. 单击调试,可进入设备详情页。

新建设备	虚拟设备调试		设备名称 ▼	设备名称 Q
设备名称	状态	激活时间	最后上线时间	操作
test	未激活	-	-	调试 二维码 删除

- 4. 单击设备属性,可查询设备上报到开发平台的最新数据及历史数据。
 - 设备属性的最新值:会显示设备上报的最新数据。
 - 设备属性的更新时间:显示数据的更新时间。

5. 单击查看,可查看某个属性的历史上报数据。

查看设备通信日志

单击**设备云端日志−透传日志**,可查询该设备某段时间范围的所有上下行数据。**原始日志**页面是设备数据的base64编码格式,**解析日志**页面在使 能**设备数据解析功能**时才可使用。

- 上行: 上行指设备端上报到开发平台的数据。
- 下行:下行指从开发平台下发到设备的数据。

在线调试

- 1. 当 LoRa 节点 成功连接到物联网开发平台后,您可在控制台**设备调试**列表,单击调试,进入在线调试。
- 2. 将"上报周期"设置为15秒,单击发送。
- 3. 查看 LoRa 节点的串口日志,可查看已成功接收到下发的数据。

() 说明:



- 由于本示例中 LoRa 节点是 LoRaWAN Class A 类设备,这类设备不会立即下发数据,需要在有数据上行后,服务器才会向该设备 下行数据。
- 因此在 LoRa 节点上报数据之后,才能查看下发的周期调整命令。

LoRa 节点的串口会显示如下日志,表示成功下发了指令到设备端。

rhf76_incoming_data_process 4: 0F00
len: 2
data[0]: 15
data[1]: 0
report_period: 15



Heltec HT-M00L 单通道网关接入

最近更新时间: 2024-12-13 21:36:22

操作场景

HT-M00L 是 Heltec Automation 与"腾讯连连"小程序联名推出的一款低成本的单通道网关。能实现 SF7 - SF12 的全速率监听,支持 LoRaWAN 协议,大幅的降低 LoRa 网关的成本。主要面向智能家居、通信方案评估等应用场景,本文档将实现以下操作:

- 配置 HT-M00L 单通道网关,使其连接到腾讯云平台。
- 配置 LoRa 节点,与 HT-M00L 网关通过 LoRaWAN 协议进行通信。
- 在腾讯云平台上完成 数据调试。
- 将节点发送的 数据显示 到"腾讯连连"小程序中。
- 配置 LoRaWAN 用户 自定义频点。

前提条件

为了通过下面的步骤快速理解该业务场景,需要做好以下准备工作:

- 拥有一台 HT-M00L 单通道网关。
- 拥有支持任意 LoRaWAN 协议并可以修改前导码长度的节点,本例采用 CubeCell HTCC-AB01 节点。

() 说明:

节点的前导码长度需修改为16。如果前导码长度为8,则需将最小 SF 与最大 SF 设为相等,否则将只能收到最小 SF。例如节点前导码长 度为8,网关设置最小 SF 为7,最大 SF 为12,那么将只能收到 SF7。

控制台创建设备

创建 HT-M00L 网关

- 1. 登录物联网开发平台控制台 创建项目。
- 2. 进入已创建的项目,单击左侧导航栏网络管理 > LoRa 网关管理,进入 LoRa 网关管理页面。



3. 单击用户网关 > 添加网关并填写网关相关信息。

基本信息		
网关名称★		
GwEUI *		
是否公开	●是○否	
网关描述		
频点信息		
用户自定义。	顺点 LoRaWAN Regional Parameters 1.0 ▼	
位置信息		
⊠域★	北京市 ▼ 东城区 ▼ 东华门街道 ▼	0
详情	东城区东华门东北100米(北池子大街西)	
经度	116.4028	0
纬度	39.91583	0
确定添加	返回列表	

○ 网关名称:根据项目需要填写,本示例中填写 HT_M00L。

- GwEUI: 即网关 ID,可以在 网关配置页 上查看。
- 是否公开:选择"是",表示社区开发者可在社区网络查看该网关,并可通过这个网关进行 LoRa 节点接入;选择"否",则仅用户自己 能查看该网关。
- 频点信息:根据具体需求选择。
- 位置信息: 在地图上选择网关位置。
- 4. 单击确定添加即可完成网关创建。

创建节点

- 1. 登录物联网开发平台控制台 创建产品。
- 2. 根据实际情况填写"产品名称"。按需可选择自定义品类,通信方式选择"LoRaWAN",其它保持默认即可。



产品开发 / 新	健产品
新建产品	
产品名称 *	请输入产品名称
	支持中文、英文、数字、下划线、空格(非首尾字符)、中英文括号、-、@、\、/的组合,最多不超过40个字符
产品品类	标准品类 自定义品类
设备类型	设备 网关 子设备
通信方式 *	LoRaWAN 👻
	请根据业务场景正确选择产品的通信方式,否则会影响后续产品开发
认证方式	密钥认证 证书认证
数据协议	自定义透传
描述	选填
	最多不超过80个字符
确定	取当
WEAE	

3. 单击已创建的产品名称,进入物模型页,单击新建自定义功能创建本文需要的"温度","湿度","电池电压"三个功能,功能中的"数据类型"应与解析出的数据类型相匹配。详情请参见物模型。

功能类型	功能名称	标识符	数据类型	读写类型	数据定义	操作
属性	Temp	Temp	浮点型	读写	数值范围: 0-150 初始值: 0 步长: 1 单位:	编辑删除
属性	RH	RH	浮点型	读写	数值范围: 0-100 初始值: 0 步长: 1 单位:	编辑删除
属性	Battery	Battery	整数型	读写	数値范囲: 0-5000 初始値: 0 步长: 1 单位:	编辑删除

4. 单击**下一步**,进入**设备开发 > 云端解析**页面,开启"是否使用设备数据解析功能"开关后,可以配置相应数据解析脚本。



物模型 2 设备开发 3 含备开发 Topic列表 乙端解析	交互开发 〉 ④ 设备调试 〉 ⑤ 批量投产
 一 云端函数解析功能升级中,暂停云端函数在线调试、修改及提出 	交,恢复时间待另行通知
是否使用设备数据解析功能①	
设备数据解析 通过编写数据解析题本、将设备的上下行原始数据链化成数据模版协	议中的数据模板 JSON 数据。 您可以对脚本进行模拟测试,运行正常后可发布。
上行数据解析 ⑦	语法: JavaScript 下行数据解析 ⑦ 语法: JavaScr
<pre>runclion RawigProtocol(TPort, bytes) { return {}; }</pre>	2 return []; 3 }
模拟调试 设备上行数据 ③ 设备下行数据 ① 此	处输入(Portb)值 运行结果 查看数爆横板JSC

5. 单击设备调试 > 新建设备创建设备节点。

新建设备	×
所属产品 Heltec HT-M00	1
设备名称 *	
支持英文、数字	、下划线的组合,最多不超过48个字符
DevEUI *	
仅支持16进制与	蒋,长度16位
АррКеу *	
仅支持16进制字	·符,长度32位
	保子

○ 设备名称:支持英文、数字、下划线的组合,最多不超过48个字符。

○ DevEUI: 仅支持16进制字符,长度16位。必须与 LoRaWAN 节点相对应。

○ AppKey:仅支持16进制字符,长度32位。必须与 LoRaWAN 节点相对应。

6. 单击保存即可完成设备节点创建。

配置设备(硬件操作)

配置网关



1. 保持 "USR" 罐处于按下状态,单次按下 "RST"并松开,待 RGB 灯变成黄绿色后,松开 "USR" 键。此时网关将创建一个名 为"M00L_****"的Wi-Fi。



2. 将电脑或手机接入此 Wi-Fi, 密码为 heltec.org ,在浏览器中输入 "192.168.4.1",进入网关配置界面。

HT-M00L
注:使用本网关LoraWan节点发射状态前导码长度(preamble length)需更改为16(默认为8)
WiFi名称
TP-LINK_B8BC
WiFi密码
heltec_test
信道频率(Hz)
470300000
最小扩频因子(MIN SF:7~12)
7
最大扩频因子(MAX SF:MIN SF~12)
12
网关ID
C44
服务器地址
lora
端口号
1700
时区
(UTC+8 🗸
坦立
徒 交
固件升级
firmware version : V1.0

- Wi-Fi 名称:HT-M00L 网关将接入 Wi-Fi 名称进行配置并提交后,即可通过此 Wi-Fi 连接腾讯云。
- Wi-Fi密码: HT-M00L 网关将接入 Wi-Fi 密码。
- 信道频率 (Hz): 该网关 LoRa 将要监听的频率,该频率必须与服务器、节点发射频率相对应。
- 最小扩频因子 (MIN SF: 7 12): LoRa通信的最小扩频因子。
- 最大扩频因子(MAX SF: MIN SF 12): LoRa 通信的最大扩频因子,该值必须大于等于"最小扩频因子"。
- 网关 ID: 网关的唯一序列号,根据硬件 Mac 地址自动产生并已绑定。
- 服务器地址:已绑定腾讯云。
- 端口号:HT-M00L 网关与 LoRa 服务器通信的端口,一般默认1700。
- 时区: 网关所在位置的时区。
- () 说明:



HT-MOOL 已绑定腾讯云,服务器地址不可修改。

配置节点

节点部分的通信实验在基于 CubeCell HTCC-AB01,外接 HDC1080 温湿度传感器,同时通过开发板上的 ADC 读取电池电压,并将读到的 数据通过 LoRaWAN 协议发送到 HT-M00L 网关上。

CubeCell 支持 Arduino 开发环境,首先安装 CubeCell Arduino 开发环境。

修改节点代码

将传感器与主板连接好后,使用 HDC1080 传感器 + LoRaWAN 例程。在 Arduino IDE 菜单中,单击工具按下图进行配置:

LoRaWan_HDC1	080 Arduino 1.8.13		
文件编辑项目工具	具 帮助		
	自动格式化	Ctrl+T	
	项目存档		
LoRaWan_HD	修正编码并重新加载		
<pre>#include "LoR</pre>	管理库	Ctrl+Shift+I	
<pre>#include "Ard</pre>	串口监视器	Ctrl+Shift+M	
<pre>#include <wir "hdc<="" #include="" pre=""></wir></pre>	串口绘图器	Ctrl+Shift+L	
<pre>#include "LoR</pre>			-
<pre>#include "Ard</pre>	WIFITUT / WIFINING FIRMware Updater		_
18	开发板: "CubeCell-Board(HTCC-AB01)"		>
* set LoraWa	LORAWAN_REGION: "REGION_CN470"		>
* RGB red me	LORAWAN_CLASS: "CLASS_A"		>
* RGB purple	LORAWAN_DEVEUI: "CUSTOM"		>
* RGB blue m * RGB vellow	LORAWAN_NETMODE: "OTAA"		>
* RGB green	LORAWAN_ADR: "ON"		>
*/	LORAWAN_UPLINKMODE: "CONFIRMED"		>
/* OTAA para*	LORAWAN_Net_Reservation: "OFF"		>
uint8_t devEu	LORAWAN_AT_SUPPORT: "ON"		> 2
uint8_t appEu	LORAWAN_RGB: "ACTIVE"		> D
uint8_t appKe	LoRaWan Debug Level: "无"		> ^B
/* ABP para*/	端口		>
uint8_t nwkSK	取得开发板信息		þ
uint8_t appSK	<i>住</i> 招嬰		्वः
uint32_t devA	海住時		1
/*LoraWan chamme	施設しまれた		

以下三个地方需要根据实际情况进行修改:

• Arduino 菜单中,工作频段设置为 CN470。

开发板: "CubeCell-Board (HTCC-AB01) "	>
LORAWAN_REGION: "REGION_CN470"	REGION_AS923(AS1)
LORAWAN_CLASS: "CLASS_A"	REGION_AS923(AS2)
LORAWAN_DEVEUI: "CUSTOM"	REGION_AU915
LORAWAN_NETMODE: "OTAA"	 REGION_CN470
LORAWAN_ADR: "ON"	REGION_CN779
LORAWAN_UPLINKMODE: "CONFIRMED"	REGION_EU433
LORAWAN_Net_Reservation: "OFF"	REGION_EU868
LORAWAN AT SUPPORT ON	REGION KR920

• DevEUI 和 AppKey,必须与腾讯云上节点的信息一致。

uint8_t devEui[] = { 0x22, uint8_t appEui[] = { 0x00, uint8_t appKey[] = { 0x88, uint8_t appKey[]

• 因为 HT-M00 是单通道网关,信道掩码必须配置为网关对应的通道。

/*LoraWan channelsmask, default channels 0-7*/
uintl6_t userChannelsMask[6]={ 0x0001,0x0000,0x0000,0x0000,0x0000 };



数据调试

• 网关正确配置完成后,可在物联网开发平台控制台 > 网络管理 > LoRa 网关管理 > 用户网关中看到网关处于"在线"状态。

LoRa网关管理						
社区网络	用户网关 用户	自定义频点				
	添加网关					
	网关名称	GwEUI	描述	区域	运行状态	最后上报时间
	HT_M00L	C4	HT_M00L	北京市 北京市 东城区	在线	2021-01-26 14:17:12

● 节点正确配置完成后,可在产品详情页 > **设备开发** > 设备名称 > 在线调试中看到节点的上行数据,可在"属性调试"中看到解析完成的数据。

✓ 数据模板← HDC1080	> 🤄)设备开发	〉 👽 交互开发	4 设备调试	> 5 批量投						
设备信息	设备属性	设备日志	设备日志(透传数据)	设备事件 设备行为	设备上下线日志	在线调试	扩展信息	设备调试日志	5		
下发指令						通信日志		清空日志	✔ 深色背景	打开响应报文	🖌 自动刷新
属性调试	行为调用					设备上报数据	≣: 2021-01-26	14:18:30			复制
✓ 功能	名称/标识符	期望值		实时数据		{ "method":	: "report",				
🔽 Tem	p(Temp)	-	0.000 +	25.16815185546875		"clientTo "params": "Rottor	oken": "2021-0 : { : {	1-26T06:18:30.24	22",		
🔽 RH(I	RH)	-	0.000 +	39.923095703125		"RH": 3 "Temp":	9.92309570312 25.168151855	5, 46875			
✓ Batte	ery(Battery)	-	0 +	4146		}, "metaLoRa 0,\"dr\":5, }	a": "{\"frameT ,\"rssi\":-64,	ype\":4,\"fPort\' \"snr\":25,\"pay	":2,\"fCnt\" loadSize\":1	':7,\"frequency\" 10}"	:48410000

数据显示

配置面板

在产品详情页 > 交互开发 > 面板配置 > 配置页面中设置"腾讯连连"小程序显示面板。

关联设备

 节点和网关正确配置完成后,在设备调试进入设备列表页,单击设备右侧的二维码可查看该设备的"二维码",打开"腾讯连连"小程序,通过 小程序扫描设备二维码可快速添加设备。

✓ 物模型	> 🕑 设备开发	> 交互开发	4 设备调试	> (5)	批量投产		
 设备调试提 	供真实、虚拟设备调试功能,	便于测试设备上报、接收数据是否正常	1, 可创建测试设备后进行调试				
新建设备	虚拟设备调试				设备名称	▼ 輸入设备名称搜索	Q,
设备名称	状态	激活时间	最后上线时间	操作		绑定网关	
				调试 二维	語 删除	查著	



• 设备添加完成后,可在"腾讯连连"小程序点击对应设备查看设备数据。



配置自定义频点

下文以单信道频点计划为例进行说明。

创建自定义频点

- 1. 登录 物联网开发平台控制台,选择公共实例或您购买的标准企业实例进入项目列表页。
- 2. 选择某一个已新建产品的项目进入,选择左侧导航栏网络管理 > LoRa 网关管理进入网关管理页面。



3. 单击用户自定义频点 > 添加频点进入"添加用户自定义频点"页面,填写相关信息。

添加用户自定义频点	
基本信息	
频点配置名称★	
支持英文、数字、下划线,不超过48个字符	
描述 *	
数据信道配置	
仅支持数字, 范围为47000000-51000000	
数据上行信道(Hz)* 数据下行RX1信道(Hz)* 数据下行RX2信道(Hz)*	
€	
	7
1、又引致子, 2日内47000000-510000000, 日本配直へ附信道, 则仅振致商目進入所 入网ト行信道(山-) 入网下行py1信道(山-) 入网下行py2信道(山-)	y
保存取消	

4. 单击保存即可创建自定义频点。

网关关联频点计划

您可 创建网关 或直接单击编辑修改"用户自定义频点"为您创建的 自定义频点。

设备关联频点计划

- 1. 选择左侧导航菜单**产品开发**进入产品列表页面,单击目标产品名称进入产品详情页。
- 2. 单击**设备开发 > 编辑**进入 LoRaWAN 参数配置页面,将**用户自定义频点**选择为您创建的 自定义频点,关联与设备对应的频点计划。



RAK LoRa环境监测产品接入平台

最近更新时间: 2024-11-27 17:34:33

操作场景

RAK LoRa 环境监测套件接入到物联网开发平台,通过物联网开发平台可以远程查看传感器的温度、湿度、GPS 等数据。本文档主要指导您如何 在物联网开发平台控制台,接入 RAK LoRa 环境监测套件。

前提条件

为了通过下面的步骤快速理解该业务场景,需要做好以下准备工作:

- 申请物联网开发平台服务。
- RAK LoRa 环境监测套件(RAK7243 网关和 RAK5205 节点)。
- 拥有一台物理或虚拟的 Windows 环境, RAK5205 串口驱动安装、串口软件的安装。

控制台操作 LoRa 网关

- 1. 登录 物联网开发平台控制台,选择上文"RAK 环境监测演示"中对应的项目。
- 2. 在左侧工具列表中,选择网络管理 > LoRa 网关管理。
- 3. 进入 LoRa 网关管理页面,选择社区网络 > 添加网关。
- 4. 在新建网关页面,填写网关基本信息。
 - 网关名称:本示例中填写 RAK7243。
 - GwEUI: 网关唯一 ID,可以在网关标签上查看到。此处填写时字母用小写。
 - 是否公开:选择"是",表示社区开发者可在社区网络查看该网关,并可通过这个网关进行 LoRa 节点接入;选择"否",则仅用户自己 能查看该网关。
 - 位置信息:在地图上选择网关位置。

网关名称 *	RAK7243网关	Ø
GwEUI *	b	${\boldsymbol{\oslash}}$
是否公开	● 是 ○ 否	
网关描述	RAK 树莓派LoRa网关	
位置信息		
⊠域 *	陕西省 ▼ 西安市 ▼ 雁塔区 ▼	${\it O}$
详情	雁塔区	
经度	108.8	${\boldsymbol{\oslash}}$

网关新建成功后,您可在网关列表页查看"RAK7243"。

LoRa 网关实物操作



1. 网关联网配置。

配置联网方式请参见 文档。

- 2. 修改网关目的 server 和端口 sudo gateway-config。
 - 2.1 修改选择 server 类型,选择 TTN,频段为 CN470,如下图所示:

 Set pi password Setup RAK Gateway LoRa concentrator Restart packet-forwarder Edit packet-forwarder config Configure WIFI Configure APN name Configure LTE Module Configure LAN 	
L < 0X > < Quit >	
Server-plan configuration Select the Server-plan:	TTN Channel-plan configuration Select the Channel-plan:
2 Server is TTN 2 Server is ChirpStack	1 AS_923 2 AU_915_928 3 CN 470 510 4 EU_863_870 5 IN_865_867 6 KR_920_923 7 RU_864_870 8 US_902_928
< <mark>OK ></mark> <cancel></cancel>	< <mark>OX ></mark> <cancel></cancel>
?修改 packet-forwarder 配置,如下图所示:	
RAK7243 (Gateway ID:B827EBFFFE758CA0 Version: 4.1.0R Configuration options:	3)
Set pi password Setup RAK Gateway LoRa concentrator Restart packet-forwarder Edit packet-forwarder config Configure WIFI Configure APN name Configure LTE Module Configure LAN	



2.3 gateway_conf.server_address 改为 loragw.things.qcloud.com,如下图所示:



3. 网关上线。

重启网关,刷新网关管理界面,即可看到网关已经在线。

控制台操作 LoRa 节点

创建项目

- 1. 登录物联网开发平台控制台,选择新建项目。
- 2. 在新建项目页面,填写项目基本信息后,单击**保存。**
 - 项目名称: 输入 "RAK 环境监测演示" 或其他名称。
 - 项目描述:按照实际需求填写项目描述。

新建项目	×
项目名称*	RAK环境监测演示
	支持中文、英文、数字、下划线的组合,最多不超过20个字符
项目描述	选填
	最多不超过80个字符
	保存取消

项目新建成功后,即可新建产品。

新建产品

- 1. 进入该项目的产品列表页面,单击新建产品。
- 2. 在新建产品页面,填写产品基本信息后,单击**保存**。



- 产品名称: "RAK5205" 或其他产品名称。
- 产品类型:选择"用户自定义"。
- 设备类型:选择"设备"。
- 认证方式:选择"密钥认证"。
- 通信方式:选择"LoRaWAN"。

新建产品							×		
产品名称 *	RAK5205	RAK5205							
	支持中文、英文、数字、下划线的组合,最多不超过20个字符								
产品品类	用户自定义	v							
设备类型	设备	网关 子	设备						
认证方式	证书认证	密钥认证							
通信方式	Wi-Fi	2G/3G/4G	5G	BLE	LoRaWAN	其它	(i)		
数据协议	自定义透传								
描述	选填								
	最多不超过80个字符								
			保存	取消					

产品新建成功后,您可在产品列表页查看"RAK5205"。

创建数据模板

- 1. 单击产品名称,进入产品配置页,在**自定义功能**配置项下,可以单击**新建功能**,自定义产品功能。
- 2. 您也可在标准功能配置项下,单击右上角导入 JSON,导入数据模板。



打开 文件,并复制文件里的内容至上图对应的本文框,单击导入。
 导入成功后,结果如下图:



功能类型	功能名称	标识符	数据类型	读写类型	数据定义	操作
扈性	温度	temperature	浮点型	读写	数值范围:-100-100 初始值:0 步长:0.01 单位:℃	编辑 删除
雇性	湿度	relative_humidity	整数型	读写	数值范围:0-100 初始值:0 步长:1 单位:%RH	编辑 删除
属性	大气压	barometric_pressur e	浮点型	读写	数值范围:-10000-10000 初始值:0 步长:0.01 单位:hPa	编辑 删除

配置 LoRaWAN 参数

- 1. 单击下一步进入数据开发。
- 2. 在设备开发页面中,按需调整 LoRaWAN 参数配置。本示例中使用默认的 OTAA 配置。

LoRaWAN参数配置					
协议版本	V1.0.2				
加网方式	OTAA				
设备类型	CLASS A				
RX1 Delay	1秒				
RX2 DR	0				
RX2 Frequency	505.3 MHz				

设备数据解析

在设备开发页面中,按需调整设备数据解析。由于 LoRa 类资源有限设备不适合直接传输JSON 格式数据,使用"设备数据解析"可以将设备原 始数据转化为产品 JSON 数据。

数据解析脚本

- 1. 上行数据解析的脚本主函数为 RawToProtocol,其带有 fPort、bytes 两个入参:
 - fPort: 设备上报的 LoRaWAN 协议数据的 FPort 字段。
 - bytes:设备上报的 LoRaWAN协议数据的 FRMPayload 字段。

脚本主函数的出参为产品数据模板协议格式的对象。

2. 打开 文件,复制文件内容至"上行数据解析"文本框,覆盖原有内容,并提交。

脚本模拟测试

您也可使用数据解析页面下方的模拟调试工具,如需开发更多的功能,请使用以下模拟脚本。

上行消息设备原始数据为 0x0768580673256D0267011D,我们将其转化为数组,即上行模拟数据为:
 [7,104,88,6,115,37,109,2,103,1,29]。



• 填入设备上行数据的编辑框中。单击运行,即可在模拟调试界面右侧查看结果。



创建测试设备

在设备调试页面中,单击新建设备,设备名为 RAK7205_1。

• DevEUI:每一个设备有一个唯一的身份识别地址,DevEUI为 60c5a8fff****efe,如下图所示:

MODEL: RAK7205	IN DAL
POWER: 5.0V	
SN: 111111111	
DevEUI: WWWWWW	1111
AppEUI: LIVE BOCSABFF	RAKwireless Technology Co., Ltd MADE IN SHENZHEN

• AppKey:设备的密钥。本例中填写 6fbdbb37b8c0dbd82af4e93f****4177,用户可以填写为其他值。

新建设备		×
所属产品	RAK	
设备名称*	RAFTON	
	支持英文、数字、下划线的组合,最多不超过48个字符	
DevEUI *	60c5a	
	仅支持英文、数字,长度16位	
AppKey *	6fbdbb37b8c0dbd82at	
	仅支持英文、数字,长度32位	
	保存取消	

△ 注意:

DevEUI 和 AppKey 统一使用小写字母。

LoRa 节点实物操作

配置设备三元组

1. 设备通过 USB 口连接到 PC,打开 com 工具设置设备三元组并启动 join:



at+set_config=lora:dev_eui:60c***********
at+set_config=lora:app_eui:800************
at+set_config=lora:app_key:6fb**********************4177
at+join
[12:13:37.650]发 + ◇at+set_config=lora:dev_eui:60c5a8:
[12:13:37.709]版←◆at+set_config=lora:dev_eui:60c5a8fffe75fefe
[12:13:37.854]收←◆LoRa dev_eui configure success OK
[12:13:44.001]发◇at+set_config=lora:app_key:6fbdbb37b8c0dbd82af4
[12:13:44.063]版←◆at+set_config=lora:app_key:6fbdbb37b8c0dbd82af4e93f74c64177
[12:13:44.208]收←◆LoRa app_key configure success OK
[12:13:48.153]发→◆at+set_config=lora:app_eui:800
[12:13:48.212]版←◆at+set_config=lora:app_eui:8000000000000000000000000000000000000
[12:13:48.359]版←◆LoRa app_eui configure success OK
[12:14:D1.225]发→◇at+join
[12:14:01.278]收+ ◆at+join
DevEui:60C5A8FFFE75FEFE
AppEu1:8000000000000000000000000000000000000
OTAA Join Start
[12:14:06.454]收★ ◆ [LoRa]: Join Success OK
[12:14:06.561]收←◆Start Search Satellite(about 100 seconds)
端口号 COM3 Silicon Labs CP210x U. ▼ □ HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送: 5000 ms/次
● 关闭串口 き 更多串口设置 ▼ 加时间戳和分包显示, 超时时间: 20 ms 第1 字节 至 末尾 ▼ 加校验 None
▼ RTS ▼ DTR 波特率: 115200 ▼ at+join
发送

2. 当您看到"Join Success 字样",代表设备已经连接成功。

查看设备状态

- 1. 保持 LoRa 节点和 LoRa 网关为运行状态。
- 2. 进入控制台 > 产品开发 > 设备调试,可查看到设备 "RAK****"。
- 3. 单击调试,可进入设备详情页。
- 4. 单击**设备属性**,可查询设备上报到开发平台的最新数据及历史数据。
 - 设备属性的最新值:会显示设备上报的最新数据。
 - 设备属性的更新时间:显示数据的更新时间。

设备属性	设备日志	设备事件	设备行为	设备上下线日志	在线调试	扩展信息		
标识符		功能名称		历史数据	数据类型		最新值	更新时间
power_switch	ı	电灯开关		查看	布尔型		-	-
color		颜色		查看	枚举型		-	-
brightness		亮度		查看	整数型		-	-
name		灯位置名称		查看	字符串		-	-



在线调试

当 LoRa 节点成功连接至物联网开发平台后,您可在控制台设备调试列表,单击调试,进入在线调试。



基于 TencentOS tiny 接入指引

最近更新时间: 2024-11-26 11:09:51

操作场景

假设一款智能灯接入到物联网开发平台,通过物联网开发平台可以远程控制灯的亮度、颜色、开关,并实时获取智能灯上报到物联网开发平台的数 据。本文档主要指导您如何基于 TencentOS-tiny 物联网操作系统,使用 MQTT 协议在物联网开发平台控制台接入智能灯。

前提条件

为了通过下面的步骤快速理解该业务场景,需要做好以下准备工作:

- 申请 物联网开发平台服务,详情请参见 购买指南。
- 准备 TencentOS-tiny 官方 EVB_MX_Plus 开发板以及一个 ESP8266 Wi-Fi 模组。

控制台操作

创建项目

- 1. 登录物联网开发平台控制台。单击新建项目,填写"项目名称"和"项目描述"。
 - 项目名称: 输入"智能灯演示"或其他名称。
 - 项目描述:按照实际需求填写项目描述。

新建项目	
项目名称 *	
	支持中文、英文、数字、下划线的组合,最多不超过20个字符
项目描述	选填
	最多不超过80个字符
	保存取消

- 2. 项目基本信息填写完成后,单击保存,即可完成项目创建。
- 3. 项目创建成功后,即可新建产品。

新建产品

- 1. 进入该项目的产品列表页面,单击新建产品。
- 2. 在新建产品页面,填写产品基本信息。
 - 产品名称: 输入"智能灯"或其他产品名称。
 - 产品品类:选择智慧城市 > 公共事业 > 路灯照明。
 - 设备类型:选择"设备"。
 - 认证方式:选择"密钥认证"。
 - 通信方式:按需选择。
 - 数据协议:默认为"数据模板"。
 - 描述: 按照实际需求填写项目描述。



新建产品		×
产品名称 *	智能灯	
	支持中文、英文、数字、下划线的组合,最多不超过20个字符	
产品品类	智能城市 🔻 公共事业 🔻 路灯照明	*
设备类型	设备 网关 子设备	
认证方式	证书认证 密钥认证	
通信方式	Wi-Fi 2G/3G/4G 5G BLE LoRaWAN 其它 (i)	
数据协议	数据模板 自定义透传	
描述	选填	
	最多不超过80个字符	
	保存取消	

3. 产品信息填写完成后,单击保存,即可完成产品创建。

4. 产品创建成功后,可在产品列表页查看到"智能灯"。

创建数据模板

当新建产品将"产品类型"选择为**路灯照明**时,此时单击该产品进入产品详情页,系统将会自动生成标准功能。

於 購訊云 总览	云产品 ~ │ 网站窗龛 +	# © E ²⁹
← 物联网开发平台 智能灯演示	产品开发 / 智能灯	
开发中心	① 数据模版 2 设备开发 3 交互开发 3 交互开发 4 设备调试	> 5 批量投产
🏟 产品开发	标准功能	导入JSON 查看JSON
(ē) 应用开发 ~	标准功能为系统推荐,您可按需选择	
戦 数据开发	功能类型 功能名称 标识符 数据类型 读写类型 数据定义	操作
服务中心		编辑
🐼 固件升级	屬性 "电灯开天 power_switch 布尔型 读号 1-开	删除
🖃 量产管理	0 - Red	仲培
□ 网络管理 🚽 🗸	属性 颜色 color 枚举型 读写 1 - Green 2 - Blue	調調
④ 数据同步	数值范围: 0-100	
⊗ 运营分析 ✓	初始值: 1 居性 完度 brightness 整数型 读写 初始值: 1 步长: 1 单位: %	编辑 删除
	属性 灯位置名称 name 字符串 读写 字符串长度:0-64个字符	编辑删除

创建测试设备

在设备调试页面中,单击新建设备,设备名为 dev001。



於 購讯云 总范	云产品 🗸 📗 网站备驾	\$ +			© #	2 <mark>9</mark> ‡
← 物联网开发平台 智能灯演示	产品开发 / 智能)	1				
开发中心	✓ 数据模板	> 🕑 设备开发	交互开发	👌 🥑 设备调试	> (5) 批編	設产
🏟 产品开发	 设备调试损 	供真实、虚拟设备调试功能,	便于测试设备上报、接收数据是否]	王常, 可创建测试设备后进行调试,	开发中产品最多创建50个	设
(6) 应用开发 · · · · · · · · · · · · · · · · · · ·	备					
弐 数据开发	新建设备	虚拟设备调试			设备名称	Q
服务中心	设备名称	状态	激活时间	最后上线时间	操作	
🛞 固件升级		als Million C				
🗈 量产管理	dev001	未激活	-	-	调试 二維始 劃除	
□ 网络管理 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~						
② 数据同步						
⊘ 运营分析 ✓						

开发板实物操作

硬件连接

开发板实物如下:



- 1. 使用 USB 线连接到开发板顶端的 USB 接口,在为开发板供电的同时,可以在 PC 端通过串口调试助手查看输出日志。
- 2. 使用 ST-Link 下载器连接开发板右侧的接口,为开发板下载程序。
- 3. 将开发板配套的 ESP8266 模组插入到开发板右侧的通信模组接口。

串口准备

1. 硬件连接成功后,打开 PC 端上的设备管理器,即可查看开发板所对应的串口(请确保已安装 CH340 驱动)。



- 2. 打开串口工具,做好相应配置后,打开串口。
 - 端口号:本例中为 COM20。

🔗 腾讯云

○ 波特率:本例中为 115200。

▲ SSCOM V5.13.1 串口/网络数据调试器(×
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助			
			^
			\sim
清除窗口 打开文件 发送文件 停止 清发送区 □ 最前 □ English 保存参数 扩展			
端口号 COM20 USB-SERIAL CH340 ▼	ŧ Ţ		
● 美闭串口 ●<	-		
RTS DTR 波神華: 15200 _			
发送	~		
S:0 R:0 COM20 已打开 115200bps,8,1,None,None	CTS=0 DSR=0 RL	SD=0	11

属性上报和控制命令下发

步骤1: 下载官方例程

1. 使用 git 工具下载 TencentOS-tiny 源码,也可以访问 Github 官方仓库下载。

git clone https://github.com/Tencent/TencentOS-tiny.git

2.下载之后,进入 TencentOS-tiny\board\TencentOS_tiny_EVB_MX_Plus\KEIL\qcloud_iot_explorer_sdk_data_template 目录。

TencentOS-tiny > board > TencentOS_tiny_EVB_MX_Plus > KEIL > qcloud_iot_explorer_sdk_data_template

名称 ^	修改日期	类型	大小
startup_stm32l431xx.s	2020/7/24 13:05	S 文件	19 KB
TencentOS_tiny.uvoptx	2020/8/7 11:20	UVOPTX 文件	100 KB
TencentOS_tiny.uvprojx	2020/8/7 11:20	礦ision5 Project	64 KB

3. 示例工程中包含 STM32L431 外设驱动、TencentOS-tiny 内核、AT 框架、SAL 框架、腾讯云 C-SDK,以及示例程序。

步骤2:配置修改

- 1. 双击 TencentOS_tiny.uvprojx 打开工程(请确保已经安装好 Keil-MDK 开发环境)。
- 2.在 TencentOS-tiny\components\connectivity\qcloud-iot-explorer-sdk\port\TencentOS_tiny 目录:
 - 2.1 修改 HAL_Device_tencentos_tiny.c 文件。
 - 2.2 登录 物联网开发控制台,单击**项目 > 产品开发 > 设备调试**,单击**调试**进入设备详情页,将下图红色线框中的数据分别替换为"设备详情 页"中的参数并保存。
 - 产品 ID: 将控制台的产品 ID ,复制到下图 sg_product_id。
 - 设备名称: 将控制台的设备名称,复制到下图 sg_device_name。
 - 设备密钥:将控制台的设备密钥,复制到下图 sg_device_secret。



В	V Device tensentos tinys
	L DERCE CHCHOS any A
19	
20	<pre>#include "gcloud_iot_import.h"</pre>
21	#include "gcloud_lot_export.h"
22	
23	#include "utlis_param_check.n"
24	
25	/* Enable this macro (also control by cmake) to use static string buller to store device into */
26	/* To use specific storing methods like files/flash, disable this macro and implement dedicated methods */
27	Faerine DEBUG_DEV_INFO_USED
28	CALCARE DEPUG DEL INEO HEED
29	
30	/* product id "/
	Boddoo Onar By_produce_ra(mat_brid_or_rate + r) 修改为您的产品ID
33	/* device name */
34	static char sg device_name[MAX_SIZE_OF_DEVICE_NAME + 1] = "{1;1;
35	
36	Hifdef DEV_DYN_REG_ENABLED
37	/* product secret for device dynamic Registration */
38	<pre>static char sg_product_secret[MAX_SIZE_OF_PRODUCT_SECRET + 1] = "YOUR_PRODUCT_SECRET";</pre>
39	#endif
40	
41	Fifdef AUTH_MODE_CERT
42	/* public cert file name of certificate device */
43	<pre>static char sg_device_cert_file_name[MAX_SIZE_OF_DEVICE_CERT_FILE_NAME + 1] = "YOUR_DEVICE_NAME_cert.crt";</pre>
44	/* private key file name of certificate device */
45	<pre>static char sg_device_privatekey_file_name[MAX_SIZE_OF_DEVICE_SECRET_FILE_NAME + 1] = "YOUR_DEVICE_NAME_private.key";</pre>
46	felse
47	//* device secret of PSK device */
48	static char sg_device_secret[MAX_SIZE_OF_DEVICE_SECRET + 1] = "X1
49	fendir

3. 修改 entry.c 中的 Wi-Fi 网络接入配置信息。



步骤3:编译

单击 MDK 工具栏Rebuild All,编译整个工程。

步骤4:下载

单击 MDK 工具栏Download,下载编译完成的固件。

步骤5: 查看运行结果

在串口助手中可以看到设备在上报之后处于等待平台下发控制指令的状态。

—>Recv 57 bytes	/
	tt_keep_alive(2017: PING request 1 has
90 [223]. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample ofdata_template_light_thread(530): 190 [2246]. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample ofdata_template_light_thread(530): 190 [2250]. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample ofdata_template\light_thread(530):	No control msg received No control msg received No control msg received
no (253), vexamples voloud_or_explorer_sdr_att_emplate vata_template_sample o data_template_lipht_thread(530) B0 (2556), vexamples voloud_or_explorer_sdr_att_emplate\data_template_sample o data_templatellipht_thread B0 (2559), vexamples voloud_or_explorer_sdr_data_template\data_template_sample o data_templatellipht_thread(530)	No control msg received No control msg received No control msg received No control msg received
B0/22069), \ vexmples\goloud_ot_explorer_sdk_data_template\data_template_smmple o data_template_light_thread(530) B0/22733), \ vexmples\goloud_ot_explorer_sdk_data_template\data_template_smmple o data_template_light_thread(530) B0/22733, \ \ vexmples\goloud_ot_explorer_sdk_data_template_smmple o data_template_light_thread(530)	No control msg received No control msg received No control msg received No control msg received
B0[2279], \ veramples\qcloud_iot_explorer_sdk_dat_template\data_template_smmple o dat_template_light_thread(530) B0[2283], \ veramples\qcloud_iot_explorer_sdk_dat_template\data_template_smmple o dat_template_light_thread(530) B0[2286], \ \ veramples\qcloud_iot_explorer_sdk_data_template_smmple o data_template_light_thread(530)	No control msg received No control msg received No control msg received No control msg received
B6[2293], veramples\qolund_iot_explorer_sik_data_template\data_template_sample, oldata_template_light_thread(530); B6[2296], veramples\qolund_iot_explorer_sik_data_template_sample, oldata_template_light_thread(530); B6[2299], veramples\qolund_iot_explorer_sik_data_template_sample, oldata_template_light_thread(530);	No control msg received No control msg received No control msg received No control msg received
BC 2306 . \. \. \. \. veramples\qloud_iot_explorer_ddt_data_template\data_template_smmple oldsta_template_light_thread(530): BC 2309 . \. \. \. \. veramples\qloud_iot_explorer_sdd_data_template\data_template_sample oldsta_template_light_thread(530): BC 2312 . \. \. \. \. veramples\qloud_iot_explorer_sdd_data_template\data_template_sample oldsta_template_light_thread(530):	No control msg received No control msg received No control msg received

物联网平台查看上报数据

1. 保持系统处于运行状态。

腾讯云

- 2. 登录物联网开发控制台,单击项目 > 产品开发 > 设备调试,可查看到设备 "dev001"。
- 3. 单击调试,可进入设备详情页。

新建设备	虚拟设备调试			设备名称	Q
设备名称	状态	激活时间	最后上线时间	操作	
dev001	未激活	-	-	调试删除	

- 4. 单击设备属性,可查询设备上报到开发平台的最新数据及历史数据。
 - 最新值:显示设备上报的最新数据。
 - 更新时间:显示数据的更新时间。
 - 历史数据: 单击**查看**, 可查看某个属性的历史上报数据。

✓ 数据模板	> 📀	设备开发	> 📀	交互开发	4 设备调试		5 批量投产				
- dev001											
设备信息	设备属性	设备日志	设备事件	设备行为	设备上下线日志	在线调试	扩展信息	设备调试日志			
标识符		功能名称		历史	数据	数据类	型	最新	值	更新时间	
power_switch		电灯开关		查看	ŀ	布尔型		-		2020-08-07 10:49:29	
color		颜色		查看	•	枚举型				2020-08-07 10:49:29	
brightness		完度		查看	ł	整数型		1		2020-08-07 10:49:29	
name		灯位置名和	<u>۳</u>	查看	ł	字符串		-		2020-08-07 10:49:29	

物联网平台在线调试

1. 在控制台**设备调试列**表,单击**在线调试**,进入在线调试。



2. 设置电灯开关为 on,颜色为 Green,亮度为3,单击发送。

2备信息 设备属性	设备日志 设备事件	设备行为	设备上下线日志	在线调试 扩展信息	3 设备调试日志			
发指令					通信日志	清空日志	✔ 深色背景 🔄 打开响应报	文 🔽 自动
居性调试 行为调用					下安坊創版会, 2020	-08-07 10.55.50		1
✓ 功能名称/标识符	期望值		实时数据		{ { "method": "cont	rol",		
✓ 电灯开关(power_switch)			开		"clientToken":			r,
✔ 颜色(color)	Green	v	Green		"power_switch "color": 1,			
✔ 完度(brightness)	- 3	+ %	3		"name": "" }			
 (丁)小栗/ブジャ(n ann a) 		0/64			3			
✓ X1业宣名标(name)	支持英文字母、数字、常见半角	符号组合	-					

3. 在串口上查看系统打印的串口日志,判断出系统成功收到并响应了下发的控制指令。

NSCOM V5.13.1 串口/网络数据调试器	-		×
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助			
IDEG 412 . \. \. \. \. \. \. \. \. \. \. \. \. \	rol msg received rol msg received rol msg received anager.c htness changed		^
IMF 425]\.\.\examples\qcloud_iot_explorer_sdw_data_template\data_template_sample.0DmControlMsgCallback(256): Property=cond IMF 425]\.\.\examples\qcloud_iot_explorer_sdw_data_template\data_template_sample.0DmControlMsgCallback(256): Property=cond DB6 425]\.\.\examples\qcloud_iot_explorer_sdw_data_template\data_template_sample.clducontolMsgCallback(256): OLED Clear OLED Clear	rr changed 设备収到 rr_switch changed ng about your own p	空制指令 oduct lo	; gic
[0LED SkweString: 3 DEG[425]. V \components\connectivity\qcloud=igt=explorer=sdk\3rdparty\sdk_src\services\data_template\data_template_client.c [6980]: Report Document: ["code":0, "clientToken-16800119-a800-4393-9b9C=764d-382469a } DEG[425]. V \components\connectivity\qcloud=ict=explorer=sdk\3rdparty\sdk_src\protocol_metVagt_client_publish.c[qcloud_ic packetD=0[topinNeme=Sthing/up/property/TWREPGACUS/dev001[payload=["method":"control_reply", "code":0, "clientToken":"clientTok Aft CMD: Aft CMD: Aft CMD: Aft CMD:	IOT_Template_Contr ot_mgtt_publish(340) en-bf880118-e8f0-43:	lReply publish 3-b9b5-	
>0K			
>> >Recv 185 bytes			
>SEND OK DBG[426]. \. \. \. \cwsmples\qcloud_iot_explorer_sdk_data_template_template_sample.c data_template_light_thread(524): Contol DBG[426]. \. \. \. \components\connectivity\qcloud-iot-explorer-sdk\Sydparty\sdk_src\protocol\mqt\mqt_client_publish.g qcloud_io packetDPO[topinName=Sthing/up/property/FMEDFOACUS/devO01 payload=[method]: report, clientTokan : FWBDFGACUS-3, params : [power_switch :1, color :1, brightness :3]} Af CDD: Af UND:	msg reply success vt_mqtt_publish(340)	publish	
>0K			
>SEND OK IMF 427 \. \. \examples\qoloud_iot_explorer_sdk_data_template\data_template_sample. o data_template_light thread(565): data te IMF 430 \. \. \examples\qoloud_iot_explorer_sdk_data_template\data_template_sample. o data_template_light thread(560): No cont DB6 430 \. \. \examples\qoloud_iot_explorer_sdk_data_template\data_template_sample. o data_template_light_thread(530) No cont DB6 434 \. \. \examples\qoloud_iot_explorer_sdk_data_template\data_template_sample. o data_template_light_thread(530) No cont DB6 437 \. \. \examples\qoloud_iot_explorer_sdk_data_template_sample. o data_template_light_thread(530) No cont DB6 437 \. \. \. \examples\qoloud_iot_explorer_sdk_data_template\data_template_sample. o data_template_light_thread(530) No cont	再次进入等待指令>> mplate reporte succ reply response, rep. rol msg received rol msg received rol msg received	犬态 ess y ack: 0	~
	-		
端口号 COM20 USB-SERIAL CH340 □ HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送: 50 ■s/欠 □ 加回车换行	i a		
● <u>美田田口</u> き <u>更多串口设置</u> □ 加时间戳和分包显示。 細时时间 20 ms 第1 字节 至 末尾 ▼加校验 None ▼ ■ RTS □ DTR 波特室: 115200 ▼	~		
发送			
		_	
S:0 R:29128 COM20 已打开 115200bps,8,1,None,None	CTS=0 DSR=0 RLSD	=0	



4. 同时,在开发板的OLED显示屏幕上,可以看到系统模拟出的智能灯状态。



设备行为调用

1. 在**物联网开发控制台 > 项目 > 产品开发 > 数据模板 > 新建功能**,手动新建一个 设备行为功能 。

自定义	功能								
类型	属性	事件	行为						
名称*									
	支持中文、引	英文、数字、	下划线的组合,最多不;	留过20个字符					
符*									
1	第一个字符7	不能是数字,	支持英文、数字、下划	践的组合,最多不超过32个字符	÷				
参数	参数名称		参数标识符	数据类型	数据	定义			
				布尔型 ▼	0	×	1	Я	
					支持	中文、英文、数字、	下划线	勤的组合,最多不超过12个字符	
						添加参数			
参数	参数名称		参数标识符	数据类型	数据	定义			
				布尔型 🔻	0	¥	1	Я	
					支持	中文、英文、数字、	下划約	助组合,最多不超过12个字符	
						添加参数			
	洗垣								
ł	最多不超过8	30个字符							
						He 21/			
					保存	取消			



2. 修改工程文件,在 data_template_sample.c 文件中使能 Action。



- 3. 重新编译下载,在开发板上按下复位键使系统运行。
- 4. 在物联网开发控制台上下发设备行为调用。

设备信息 设备属性 设备	日志 设备事件 设备行为	设备上下线日志	在线调试	广展信息 设备调试日	志		
发指令 屢性调试 行为调用 ^案 行为 行为动作(light_blink)	×			<mark>通信日志</mark> 调用设备行为: 2 {	清空日志 020-08-07 11:01:32	✓ 深色背景	打开响应报文 🔽
&数名称/示识符 ime(time)	参数设置 — 3 +			"Uin": "1000 "RequestId": "ProductId": "DeviceName" "ActionId":	01047321", "a4226932-2b35-4825-9; "FwR8PGACUS", : "dev001", "light_blink",	213-4fdef51d7ced",	
color(color)				"Flag": "1", "InputParam" }	: "{\"time\":3,\"color'	\":0,\"total_time\":	:1000)"
otal_time(lotal_time)	- 1000 +			设备上报款据: 2 { "method": "r "clientToken "params": { "power_swi "color": 0	920-98-07 11:01:22 eport", ": "FWR8PGACUS-2", tch": 0, ,		

5. 在串口助手中可查看到设备行为被调用。

腾讯云

Na SSCOM V5.13.1 串口/网络数据调试器	-		\times
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助			
<pre>replyAch=0 DEG[31], \components\connectivity\qcloud-iot=explorer=sdk\3rdparty\sdk_src\services\data_template\data_template_client o Received Ison Document=["method": getstatus_reply", clientToken: FWEBFGACUS=1", code":0, "status": "success", data '["reported ["power_switch"1", doild_servicess", data '["reported ["power_switch", doild_servicess", data '["reported ["power_switch", doild_servicess", data '["reported ""]] BG[32], \components\connectivity\qclud_iot=explorer_sdd_data_template_sample c data_template_light_thread(532): No cont BG[32], \. \components\connectivity\qclud_iot=explorer_sdd_data_template_sample c data_template_light_publick_log2Out_scate ""] BG[32], \. \components\connectivity\qclud_iot=explorer_sdd_data_template_light_report", clientToken '["] BG[32], \components\connectivity\connectivity\qclud_iot=explorer_sdd_data_template_light_report", clientToken '["] BG[32], \components\connectivity\connectity\connectivity\connectivi</pre>	∫_get_status_reply_ac ; a status success rol msg received t_mqtt_publish(340):	k_ob(189 publish): ^
>XOK			
>>			
>Reav 201 bytes			
>SEND OK INF 33 \. \. \examples\qcloud_iot_explorer_sdk_data_template_tamplate_sample.c data_template_light_thread(547): data te INF 33 \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 36 \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 36 \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 36 \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_light_samples\qcloud_ta_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): No cont DBG 46. \. \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.c data_template_light_thread(532): SD66020 "time '3, color '10, 'total_time'1000]} DBG 46. \. \. \. \. \. \. \. \. \. \. \. \. \.	mplate reporte succes reply response, reply rol msg received rol msg received rol msg received rol msg received 	s ack: 0 11back ŋ调用 publish ode ^{**} :0,	
X			
>			
>Recv 233 bytes			
->SEND OK DBG 477 . \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample.o data_template_light_thread(532): No cont IMF 50 . \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample.o data_template_light_thread(532): No cont DBG 50 . \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample.o data_template_light_thread(532): No cont DBG 55 . \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_sample.o data_template_light_thread(532): No cont DBG 55 . \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.o data_template_light_thread(532): No cont DBG 560 . \. \. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template_sample.o data_template_light_thread(532): No cont DBG 660 . \. \. \. \. \. \. \. \. \. \. \. \. \	rol msg received ket-id=64791 rol msg received rol msg received rol msg received rol msg received		*
	-		
端口号 COM20 USB-SERIAL CH340 ▼ □ HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送: 50 ms/次 □ 加回车换1	Ī ₂		
 ● 美词串口 2 更多串口设置 □ 加时间戳和分包显示。超时时间: 20 ms 第1 字节 至 末尾 ▼ 加校验 None ▼ 	_		
□ RTS □ DTR 波特率: 115200	^		
发送	~		
		0	
5:0 R:10944 COM20 C317T 1152006ps,8,1,None,None	ICIS=0 DSK=0 KLSD=	-0	11

设备事件上报

1. 修改工程文件中的 config.h 文件,将设备行为调用支持屏蔽,开启设备事件上报支持。




2. 修改工程文件中的 data_template_sample.c 文件,关闭设备行为调用示例,开启事件上报示例。

) main	data template sample.c							
1.0	/*							
2	* Tencent is pleased to support	the open source community by making IoT Hub available.						
3	* Copyright (C) 2016 THL A29 Li	mited, a Tencent company. All rights reserved.						
4								
5	* Licensed under the MIT Licens	e (the "License"); you may not use this file except in						
6	* compliance with the License. You may obtain a copy of the License at							
7	* http://opensource.org/license	s/MIT						
8								
9	* Unless required by applicable	law or agreed to in writing, software distributed under the License is						
10	* distributed on an "AS IS" bas	is, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,						
11	* either express or implied. Se * limitations under the License	e the License for the specific language governing permissions and						
13	*							
14	*/							
15								
16	<pre>#include <stdio.h></stdio.h></pre>							
17	<pre>#include <stdlib.h></stdlib.h></pre>							
18	<pre>#include <limits.h></limits.h></pre>							
19	<pre>#include <stdbool.h></stdbool.h></pre>							
20	<pre>#include <string.h></string.h></pre>							
21	<pre>#include <time.h></time.h></pre>							
22	Idofing FUENT DOST FNARIED	┓ 开启设备事件上报示例						
23	FOST_ENABLED							
25	<pre>#include "gcloud iot export.h"</pre>							
26	#include "gcloud iot import.h"							
27	<pre>#include "lite-utils.h"</pre>							
28	<pre>#include "data_config.c"</pre>							
29		1						
30	//#define ACTION_ENABLED	屏蔽设备行为调用示例						
31		-						

3. 编译程序,下载到开发板中,在开发板上按**复位**使系统运行。

4. 在串口助手中可查看到设备上报事件的日志。

1 SSCOM V5.13.1 串口/网络数据调试器	-		×
通讯满口 串口设置 显示 发送 多字符串 小工具 帮助			
DBG 31 \. \examples \u00ed int explorer_sdk_deta_template\data_template_sample o data_template_light thread(502): Get data st DBG 31 \. \. \examples \u00ed_oid_explorer_sdk_data_template\data_template_sample o data_template_light_thread(536): No control DBG 31 \. \. \examples \u00ed_oid_oid_explorer_sdk_data_template\data_template_sample o data_template_light_thread(536): No control DBG 31 \. \. \examples \u00ed_oid_oid_explorer_sdk_data_template\data_template_sample o data_template_light_thread(536): No control DBG 31 \. \. \examples \u00ed_oid_oid_explorer_sdk_data_template_sample o data_template_light_thread(536): No control DBG 31 \. \. \examples \u00ed_oid_oid_oid_oid_oid_oid_oid_oid_oid_oi	atus success msg received tt_publish(340):	publi sh	^
X			
>			
>Recv 201 bytes			
<pre>>SEND OK INF 33]. \. \. \. \examples\qcloud_iot_explorer_sdk_dsta_templste\data_template_mplste\data_templste person_templsteddata_templsteddata_templste\data_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_templsteddata_temp</pre>	te reporte succes tt_publish(334): ": lert", "timestamp	ss publish ":0,	
X			
>			
──>Recv 441 bytes			
>SEND OK INF [37]. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_o[OnReportReplyCallback(272); reov report rep1 DBG [37]. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_ample_o[data_template_licht_thread(536); No_control DBG [37]. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template_ample_oldata_template_licht_thread(536); No_control DBG [37]. \. \. \. \examples\qcloud_iot_explorer_sdk_data_template\data_template\data_template.licht_thread(536); No_control DBG [37]. \. \. \. \examples\template\template\template\data_template\template\template\template.licht_thread(536); No_control DBG [37]. \. \. \examples\template\t	y response, reply <u>mss received .</u> <u>tt_publish(334):</u> : lert", "timestamp	v ack: 0 publish o":0,]
AT HCLESEND=0,441 事件上报日志			
—>0K			
──>Recv 441 bytes			
>send ok			~
端口号 COM20 USB-SERIAL CH340 ▼ 「HEX最示 保存教报 「接收数据到文件 「HEX发送 「定时按送: 50 ms/次 「加回车执行」 ● 実用用□ ● 更多串口设置 「加时间戳和分包显示 超时时间 20 ms 第1 字节 至 末尾 ▼加校验 ^{None} ▼			
bergenerative be			
S:0 R:9561 COM20 已打开 115200bps,8,1,None,None CTS	S=0 DSR=0 RLSD	=0	



5. 在物联网开发控制台上可查看到设备上报事件的日志。

 ✓ 数据模板 ✓ dev001 	•	没备开发	✓ 交互开发 〉 ② 交互开发 〉 ③ 设备调试 〉 (5) 就量损产
设备信息 设备	备属性	设备日志	设备事件 设备行为 设备上下线日志 在线调试 扩展信息 设备调试日志
全部事件类型 ▼	30分钟	1小时	今天 能天 近7天 2020-08-07 10.56~2020-08-07 11.26
时间		日志类型	事件信息
2020-08-07 11:26:07		信息	{"status":0,"message":""}
2020-08-07 11:26:07		告警	{"voltage":1.000000}
2020-08-07 11:26:07	,	故障	{"name":"","error_code":1}
2020-08-07 11:26:03	i -	告誓	["voltage".1.000000}
2020-08-07 11:26:03		故障	{"error_code":1,"name":""}
2020-08-07 11:26:03		信息	{"status":0,"message":""}
2020-08-07 11:25:58	1	信息	("status":0,"message":"")



蓝牙智能灯接入实践

最近更新时间: 2024-10-28 11:45:12

操作场景

使用 BLE 蓝牙模拟一款智能灯,配合"腾讯连连"小程序实现设备接入,物联网开发平台可以远程控制灯的亮度、颜色、开关,并实时获取智能 灯上报到物联网开发平台的数据。

本文档主要指导您如何使用 NRF52832 模组在物联网开发平台创建智能灯产品并通过手机接入。

前提条件

为了通过下面的步骤快速理解该业务流程,需要做好以下准备工作:

- 准备一块 NRF52832 开发板及其 SDK。
- 下载最新版本设备端 LLSync SDK。
- 安装 Python、JLink 驱动程序。

控制台操作

新建项目

- 1. 登录物联网开发平台控制台,选择公共实例或者您购买的标准企业实例进入项目管理页面。
- 2. 单击新建项目,在新建项目页面,填写项目基本信息,单击保存。

新建项目	:
项目名称 *	智能灯演示
	支持中文、英文、数字、下划线的组合,最多不超过20个字符
项目描述	选填
	最多不超过80个字符
	保存取消

○ 项目名称: 输入"智能灯演示"或其他名称。

○ 项目描述:按照实际需求填写项目描述。

新建产品

1. 进入该项目的产品列表页面,单击**产品开发 > 新建产品**。



2. 在新建产品页面,填写产品基本信息,单击保存。

新建产品							×
产品名称 *	蓝牙智能灯						
	支持中文、英文	文、数字、下划	线的组合,	最多不超过20)个字符		
产品品类	智慧生活	,	电工	照明	▼ 兆J		v
设备类型	设备	网关	子设备				
认证方式	证书认证	密钥认订	E				
通信方式	Wi-Fi	2G/3G/4G	5G	BLE	LoRaWAN	其它	í
数据协议	数据模板	自定义遗	营传				
描述	选填						
	最多不超过80-	个字符					
			保存	取消			

- 产品名称: 输入"蓝牙智能灯"或其他产品名称。
- 产品类型:选择智慧生活 > 电工照明 > 灯。
- 设备类型:选择"设备"。
- 认证方式:选择"密钥认证"。
- 通信方式:选择"BLE"。
- 其他选项保持默认。

创建数据模板

- 1. 打开已创建的**蓝牙智能灯**产品。
- 2. 单击数据模板 > 导入 JSON 进入"导入 JSON"页面。

产品	品开发 / 蓝牙智	能灯									使	用指南 🖸
1	数据模板		2 设备开发		③ 交互开发		(4) 设备调	at >	5	批量投产		
	标准功能						导入JSON 查看J	SON 添加功能		蓝牙智能	灯 开发中	编辑
	标准功能为系统推	荐,您可按	医恶选择							产品ID		
	功能类型	功能名称	t i i i i i i i i i i i i i i i i i i i	标识符	数据类型	读写类型	数据定义			产品品类	智慧生活-电工照 明-灯	2
	同件	க ளா ×			たな刑	法史	0-关			设备类型	设备	
	唐任	电灯开天	化过	power_switch	1 伟水型	医与	1 - 开			认证方式	密钥认证	
										通信方式	BLE	

3. 打开代码中的 qcloud_iot_explorer_ble\samples\nrf52832\date_template\nrf52832.json ,如果出现乱码请将您的编辑器修 改为 UTF8 编码,将文件中的代码粘贴进来,单击导入。



导入JSON	>
() 注意:导入新的JSON后原产品的数据模板将会被覆盖	
<pre>"start": "00", "start": "0", "step": "1", "unit": "" }], "output": [{ "id": "result", "name": "result", "define": { "define": { "define": { "type:".string", "min": "0", "max": "320" }</pre>	

选择设备开发方式

本文采用标准蓝牙协议(LLSync 协议)方式进行开发,在**设备开发**页面,请选择"基于标准蓝牙协议开发(默认)"的开发方式。如您需要选择"基于自定义蓝牙协议开发"的开发方式,请另参见 H5 自定义开发指南。

数据模板	2 设备开发	> (3) 201	/122 /	4 反用调风			
基于标准蓝牙协议升	发 (默认)						
使用官方标准蓝牙协	议(LLSync 协议)接入:	设备,请查看文档了解如何	基于标准蓝牙协议	开发 LLSync SDK 下讀	找 开发指南▼		
基于自定义蓝牙协议	开发						

交互开发配置



配置产品在小程序端的控制面板、快捷入口,若不配置则使用系统默认配置。

 ◇ 数据機板 〉 ◇ 设备开发 〉 ③ 交互开发 	> (4) QBHNDAC > (5) BHBHD/ ²²	
使用官方小程序控制产品 您這座使用千年的官方小爆冲控制产品	使用通用版APP控制产品 意力が現実無法運転二級時 用戶可以使用場所運転通用総APP控制产品。也可以使用开源版APP、自有APP集成APP SOR放制产品	
配置小程序 配置APP		
产品展示图标	0 22	BI
您可以自定义产品在腾讯运连首页-设管列表中展示的产品图标	-	
(快速入口配置) 您可以自定义产品在设备列表-快速功能区域显示的快速操作	0	
回板配置	0 52	B
您可以自定义产品控制面燃的风格、布局、按钮样式等配置		_
设备规定51日 您可以自定义用户在规定BLE设备时的引导图和文素	Ø №	9
扫一扫产品介绍	0 55	81
您可以自定义用户在使用微信扫一扫添加设备时的产品介绍页	_	
智能联动配置 您可以自定义用户在原加智能时,该产品可作为条件或任务的功能属性	0 82	E.

在一些场景中,例如智能手环,在设备绑定期间需要用户操作设备进行确认绑定,此时,可在**设备绑定引导**配置页面,打开需要用户确认选项,并 可配置相关引导文案和引导图。

备绑定引导				
百需要用户确认	是 ▼	預览		
	是否需要用户在智能设备上点击确认绑定	••••• TCloud	22:05 総売设备	100%
导文業	请打开手机篮牙功能,并将得绑定的BLE设备盖于可发现状态。			
空确认引导文案	语能入引导文章。45字符内 请根据BLE设备提示完成规定确认操作。		1	
	谓临入绑压确认引导文案,45字符内	请打开: 备置于:	手机蓝牙功能,并将待绑 可发现状态。	/定的BLE设
定失败提示	 清确认篮牙设备电量是否充足。 清确认手机篮牙功能已开启。 	已发现如下	168 B	
	3. 如仍未成功,请尝试重启手机与设备蓝牙。	MN563	_16BD	连接
		Y0T381	_66UE	連接
	请搁入那定央规定示,100字符内	YOT381	_66UE	连接
定引导图		VOTSSI	1_88UE	谨攒
	建议尺寸: 670px * 670px,支持png、jpg、gi梢戟式,大小不超过500KB			

新建设备



在**设备调试**页面中,单击新建设备,输入设备名"ble_device_01"或其他设备名称,单击保存。

新建设备		×
所属产品	腾讯连连H5小测试	
设备名称 \star		
	支持英文、数字、下划线的组合,最多不超过48个字符	
	保存取消	

设备端操作

硬件简介与连接

开发板实物如下:



指示灯与按键的作用如下:

- 按键1: 广播。
- 按键2: 上报事件。
- 按键3: 擦除 flash 内保存的绑定信息。
- 按键4:上报设备状态或属性,每按一次交替上报。
 - LED1: 开关
 - LED2: action
 - LED3: replay
 - LED4: 未使用

连接实物图操作步骤如下:

- 1.1 使用 USB 线连接开发板左上角串口。
- 1.2 使用 JLink 连接开发板右下角下载线接口。
- 1.3 打开左上角电源开关。



1.4 打开设备管理器,确保串口和 JLink 能够正常识别,若不能识别请重新安装驱动。



准备软件环境

- 1. 前往 NORDIC 官网 下载 nRF5_SDK_15.3.0 及 SoftDevices S132。
- 2. 前往 NORDIC 官网 下载并安装 "nRF Connect for Desktop"。
- 3. 启动后找到"Programmer"并安装。



- 4. 烧录蓝牙协议栈:使用 JLink 连接开发板,打开 Programmer,如下图所示:
 - 在"①"处选择连接到开发板的 JLink。
 - 在"②"处打开需要烧录的蓝牙协议栈文件,路径为

 $nRF5_SDK_15.3.0_59ac345 \verb| components \verb| softdevice \verb| s132 hex \verb| s132_nrf52_6.1.1_softdevice \verb| hex or s132 \verb| hex \verb| hex or s132 \verb| hex \verb| s132_nrf52_6.1.1_softdevice \verb| hex or s132 \verb| hex \verb| hex or s132 $| hex or s$



○ 单击"③"处Write进行烧写。

nRF Connect v3.6.0 - Progr	ammer		- 🗆 X
■ Select device	•••		8
00002020060 • Serial port: 0	2 com5		File
6&2de583ac8 • Serial port: 0	0&2 ①	File memory layout 🗁	
			🖿 Add HEX file
ſ			C Reload files
1 1			Clear files
			Device
Connect a devic	e to display memory ntents	Drag & drop one or more HEX files here	Erase all
			🖍 Erase & write
			💾 Save as file
Log			• Reset 3
16:14:09.097 Appl	ication data folder: C:\Users\Ad	ministrator\AppData\Roaming\nrfconnect\pc-nrfconnect-pr	🖍 Write
10.14.09.201 0510	קיוויזאראיז איז איז איז איז איז איז איז איז איז		C Read
			Auto read memory
4		• • • • • • • • • • • • • • • • • • •	Enable MCUboot

▲ 注意:

- 蓝牙协议栈只需烧写一次,之后的开发过程中无需重复烧写。
- 新开发板可能没有烧写蓝牙协议栈或烧写的蓝牙协议栈版本不正确,强烈建议按上述步骤烧写一次。

下载例程代码

- 1. 下载 示例程序。
- 2. 拷贝 qcloud-iot-ble-nrf52832 文件夹至 nrf sdk 的示例程序目录下,例如 nRF5_SDK_15.3.0_59ac345\examples\ble_peripheral 。

代码修改与烧录

1. 打开工程

nRF5_SDK_15.3.0_59ac345\examples\ble_peripheral\qcloud-iot-blenrf52832\pca10040\s132\arm5_no_packs\ble_app_blinky_pca10040_s132.uvprojx

2. 打开文件 ble_qiot_ble_device.c ,找到以下三个宏并修改为您自己的设备信息。

nRF_SoftDevice	52 #include "nrf_log_ctrl.h"
qcloud_iot_explorer_ble	53 #include "nrf_log_default_backends.h"
ble giot llsync data c	54 #include "ble_qiot_config.h"
	55
ble_qiot_lisync_device.c	56 #include "flash_storage.h"
ble_qiot_llsync_event.c	57
ble_giot_utils_hmac.c	58 // divece info which defined in explorer platform
hle gigt utils log c	59 #define PRODUCT_ID
bie_qiot_utiis_iog.c	60 #define DEVICE_NAME
ble_qiot_utils_sha1.c	61 #detine SECRET_KEY
ble_qiot_utils_base64.c	62 static wint? t m adv handle -
ble giot utils md5.c	64 BLE GAD ADV SET HANDLE NOT SET: /**/ Advertisi
his gist convice c	65 static wints t m enc advdata[BLE GAP ADV SET DATA (
Die_qiot_service.c	66 static uint8 t m enc scan response data[BLE GAP AD]
ble_qiot_ble_device.c	67



3. 在 控制台 设备调试页面中选择设备信息子页面,将"设备名称"、"设备密钥"、"产品ID"处内容分别填入 DEVICE_NAME ,

SECRET_KEY , PRODUCT_ID 。

品开发 / 蓝牙智能灯							使用指南
数据模板	✔ 设备开发	<u>ج</u> 📎 خ	互开发 🔷	4 设备调试		5 批量投产	
ble_device_01							
设备信息 设备属性	设备日志	设备事件	设备行为	设备上下线日志	在线调试	扩展信息	设备调试日志
设备信息							
设备名称	6	所屋产品	蓝牙智能灯		设备创建时间	2020-11-17 17:07	:45
设备密钥	6	产品ID		Ē	设备状态		
激活时间 -		最后上线时间] -		固件版本	-	

4. 编译下载并运行。

腾讯连连小程序调试

1. 微信主页面下拉并搜索**腾讯连连**小程序。

17:17 💿 🧟	0.6K/s 🕸 👁 🖄 🗑 🖬 🖬 🖬 C	Ð
小程序		
Q 搜索	小程序	
最近使用		
8		
腾讯连连	A 4 5	
微信(10)	Q (÷
	≡ □ <	



2. 单击下方的"+" > 添加设备。





3. 按下开发板上的按键1开启蓝牙广播,待扫描到蓝牙设备后单击右侧的连接。





4. 在我的设备中打开已添加的蓝牙设备。





5. 单击右侧的**立即连接**连接设备,如连接失败,请再次按下开发板上的**按键1**开启蓝牙广播。





6. 连接成功后单击**电灯开关**,可以看到开发版上**LED1**随之变化。



7. 单击亮度或颜色按钮进行设置。

8. 查看串口打印,可以看到亮度、颜色设置成功。

🔗 腾讯云

▲ SSCOM V5.13.1 申□/网络数据调试器 - □	×
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 PCB打样	
<pre><info> app: evt_id 18 <info> app: evt_id 80 <info> app: dump write data, len 8 00 00 05 22 00 00 00 51 "Q qiot debug: data type 0, effect 0, id 0 qiot debug: type 1, id 2, len 4 qiot debug: set property brightness 81 <info> app:> mtu 1s 20 qiot debug: mtu size 17 <info> app: ble qiot dump: post data, length: 4</info></info></info></info></info></pre>	^
01 00 01 00 <info> app: evt_id 87 <info> app: evt_id 80 <info> app: dump write data, len 6 00 00 03 81 00 02 qiot debug: data type 0, effect 0, id 0 qiot debug: type 4, id 1, len 2 qiot debug: set property color 2 <into> app:> mtu 1s 20 qiot debug: mtu size 17 <info> app: ble qiot dump: post data, length: 4 01 00 01 00 </info></into></info></info></info>	
<info> app: evt_id 87</info>	~
清除窗口 打开文件 发送文件 停止 清发送区 日 最前 English 保存参数	扩展一
端口号 COMLO ▼ IF MEX显示 保存数据 「接收数据到文件 F MEX发送 「定时发送: 1000 ms/次 ▼ 加 ● 医多串口设置 「加时间戳和分包显示, 超时时间: 20 ms 第1 字节 至 末尾 ▼ 加校验 None ▼	回车换行 _?
□ RTS □ DTR 波特车: 115200 U 发送	< ~
S:0 R:848 COM10 已打开 115200bps,8,1,None,None	



设备 OTA 实践

最近更新时间: 2024-11-01 09:43:32

概述

OTA(Over-The-Air)是一种无线下载并安装新软件或更新现有软件的技术。在物联网(IoT)领域,OTA 技术被广泛应用于设备固件的远程 更新,使得设备可以在不需要物理接触的情况下,接收并安装新的固件版本。

<mark>腾讯云物联网开发平台</mark>支持了完善的设备 OTA 能力,本实践以 ESP32S3−BOX3 硬件为载体,详细介绍 OTA 的实现细节。通过本实践可以 快速理解腾讯云物联网开发平台的 OTA 升级步骤和实现流程,使目标设备可以快速集成 OTA 能力。

硬件简介



ESP-BOX 是乐鑫科技推出的面向 AloT、边缘 Al 和 lloT 应用的开发平台。ESP32-S3-BOX、ESP32-S3-BOX-Lite 和 ESP32-S3-BOX-3 是为该平台设计的系列开发板,它们基于乐鑫强大的 ESP32-S3 Wi-Fi + Bluetooth 5(LE) SoC 构建,并拥有紧凑美观的外壳结 构。借助多功能配件和 ESP-BOX 项目内的可靠例程,这些开发板既适用于构建新项目原型,也适用于打造复杂的物联网系统,提供了形式上和 功能上的平衡。

关于此硬件的更多资料,请访问 GitHub仓库 获取。

OTA实现原理&升级流程概述

设备在通过MQTT成功连接到云平台后,需要订阅 ota 的 topic 用来接收 ota 升级请求和上报升级进度。





下发的固件信息中包含固件的下载地址、MD5 校验值和固件大小。拿到这些信息后就可以通过 HTTP 来下载固件并校验固件的完整性。一个完整的设备升级流程如下图所示:



() 说明:

这里的升级流程执行的前提是设备成功接入腾讯云物联网开发平台,并订阅了 OTA 的 topic。

实践步骤

1. 静默升级

1. 联系腾讯云物联网开发平台的销售或 提交工单,获取本实践的相关代码(基于乐鑫的 esp-box 开发,客户也可以自行适配)

2. 登录 腾讯云物联网开发平台,新建设备并获取到设备的三元组信息,相关操作可以参考 使用MQTT物模型接入平台 。



← 返回项目列表	产品开发 / LLSync辅助配网测试 腾讯云IoT技术
开发中心	◇ 物模型 > ◇ 迫备开发 > ◇ 交互开发 > ④ 设备调试 > ⑤ 批量投产
🏟 产品开发	← esp32c3_test
(Ô) 应用开发	① 从2023-08-01开始,日志存储时长将由原来的7天改成3天。
1 数据开发	
基础服务	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息
🛞 固件升级	Jud Ho
■ 资源管理	议會信息
🗈 设备管理	设备名称 esp32c3_test L 产品D 3BY0T9692B L 所属产品 LLSync辅助配网测试
铝 场景联动	设备密钥 nMI == ┏ 设备创建时间 2023-05-31 17:28:35 最后上线时间 2024-01-26 15:05:31
数据流转	激活时间 2023-06-01 14:09:17 设备状态 高线 mcu面件版本 mcu_v1.0.0
⊗ 运营分析 👋	模组固件版本 module_v1.0.0

3. 将上一步生成的三元组信息,替换工程中的三元组,并填入当前网络环境的路由器信息,让设备正常接入云平台。

// 腾讯云物联网开发平台三元组信息	
	"产品ID"
	"设备名称"
	"设备密钥"
// 路由器信息	

4. 设备正常上线后,会上报当前的版本号。

← 返回项目列表 terminal_test	产品开发 / LLSync辅助配网测试	腾讯云lo
开发中心	✓ 物模型 〉 ✓ 设备开发 〉 ✓ 交互开发 〉 ④ 设备调试 〉 ⑤ 批量投产	
🗯 产品开发	← esp32c3_test	
(<u>0</u>) 应用开发	 从2023-08-01开始,日志存储时长将由原来的7天改成3天。 	
弐 数据开发		
基础服务	设备信息 在线调试 云端诊断日志 设备云端日志 设备本地日志 扩展信息	
🛞 固件升级		
■ 资源管理	设备信息	
🗈 设备管理	设备名称 esp32c3_test E 产品D 3BY0T9692B E 所属产品 LLSync 辅助配 网	测试
铝 场景联动	设备密钥 nM 2024-01-26 15:** 员备创建时间 2023-05-31 17:28:35 最后上线时间 2024-01-26 15:**	7:30
数据流转	激活时间 2023-06-01 14:09:17 设备状态 在线 mcu固件版本 mcu_v1.0.0	
⊗ 运营分析	模组圆件扳本 module_v1.0.0	

5. 修改工程的系统版本号,重新编译生成一个待升级的固件。

#define FW_RUNNING_MCU_VERSION "mcu_v1.0.1" // 待升级的固件号

重新编译后,生成我们需要的上传云平台的固件:

\$ idf.py build
Executing action: all (aliases: build)
Running ninja in directory E:\GitHub\tc-iot-explorer-for-esp32-box3\examples\tc-iot-explorer-
ota\build
Executing "ninja all"



∧ 注意:

1. 对于 ESP32 系列来说,我们升级只需要工程名.bin即可,对于当前工程来说,我们只需要 build\explorer_ota.bin 。 2. 这里修改版本号只是为了直观的验证升级成功与否,正常的开发流程下需要将待升级的固件上传云平台。

对本次上传的固件进行描述和记录,请输入0-100个字符

提交

重置

6. 将待升级的固件上传到	控制台		
☆ 腾讲云 <u>#发棋式</u>	介 控制台	() 若國	图件大小限制不能满足您的业务需求,可提交工单 22 申请增加文件大小限制
← 返回项目列表 torminal test	固件升级		
terminal_test		固件名称 *	ESP32-BOX3升级测试
开发中心	添加固件		支持中文、英文大小写、数字、部分常用符号(下划线,减号,括弧),必须以中 文、英文或数字开头,长度不超过32个字符
📫 产品开发	固件名称	花园立口 。	
(<u>0</u>) 应用开发	ESP-BOX升级测试	別)掲/ ̄ロロ^	LLSync辅助配网测试 · 上一步创建的产品
式 数据开发	test shuf200	固件类型 *	MCU 🔻
其种服务	test_phy6222	固件版本号*	mcu_v1.0.1 我们固件里填写的版本号
@ 固件升级	EPNode-20230906		仅支持英文字母、数字、点、中划线和下划线,长度限制1~32
日 资源管理	ep-node-test	选择固件*	重新选择固件
设备管理	T-Display-test		文件名称: explorer_ota.bin 文件大小: 1.21MB
旧 场景联动	esp32s3-v1 1 3		MD5: 0ad4ff4eac2c7ff85fcc2f470bc4a255
数据流转			IX 又は、いい、uav .tal .gz .zip .gzip .apk .xz .pack 昭立(昭文)又1+, 又件入小个能通过 2048MB
↔ 运营分析	奉加MESH子设备升级	固件描述	对本次上传的固件进行描述和记录,请输入0-100个字符

7. 确保设备正常上线,然后点击升级固件升级。

博芯模组1.0.9



固件升级				腾讯云loT技术交流群 💬 操作指
添加固件	固件升级	×	全部产品	▼ 请输入固件名称
固件名称 固作	固件名称	ESP-BOX升级测试	添加时间	操 1
ESP-BOX升级测试 MC	所属产品	LLSync辅助配网测试	2024-01-25 15:21:2	29 固件升级 查看详情 删除
test_phy6222 MC	固件类型	MCU	2023-09-27 11:15:5	56 固件升级 查看详情 删除
EPNode-20230906 模组	固件版本号	mcu_v1.0.1	2023-09-06 10:53:4	18 固件升级 查看详情 删除
ep-node-test 模组	批量升级方式 🚯	按固件版本 按设备名称	2023-08-31 15:04:3	32 同件升级 查看详情 删除
	待升级版本号	mcu_v1.0.0 2 -	0000 00 41 47 40 5	
T-Display-test MC	升级范围	全部设备	2023-06-11 17:16:5	74 固件开级 首有详情 關係
esp32s3-v1.1.3 MC	升级确认	静默升级	2023-06-20 10:16:5	51 固件升级 查看详情 删除
奉加MESH子设备升级 MC	超时时长配置 🚯	- 15 + 分钟	2023-06-14 16:48:2	29 固件升级 查看详情 删除
test MC			2022-09-26 11:54:4	16 固件升级 查看详情 删除
博芯模组1.0.9 模组		取消	2022-07-01 16:26:2	24 固件升级 查看详情 删除

() 说明:

- 待升级版本号:等待升级的设备版本号,这里的设备版本号就是设备上电上报的固件版本号,如果这里下拉菜单找不到待升级版本号, 则需要确认设备连接云平台后,是否正确上报了固件版本号。
- 升级范围:选择全部设备,则将当前产品下,上报此版本号的全部设备,都会推送升级请求。选择指定设备,则只给被指定的设备推送 升级请求。
- 升级确认:选择静默升级,则创建完升级任务后,后天会直接推送升级请求给设备,完成设备升级,整个过程用户无感知。选择用户确 认升级,则需要在连连小程序(用户自主小程序)上点击确认升级按钮后,后台才开始下发设备升级请求,完成设备升级。



8. 设备收到固件升级请求后,会从 cos 拉取固件,并上报升级进度。



设备升级界面如下:





同时,在控制台也可以看到当前的升级进度:

设备详情									
全部设备(1)	升级成功(0)	设备离线(0)	待推送(0)	已推送(0)	升级中(1)	升级失败(0)		刷新	请输入设
设备名称		当前版本号		最后更新时间		升级状态	状态详情	操	ſſĘ
esp32c3_test		mcu_v1.0.0		2024-01-26 15:	33:50	下载中	下载中,11%	取	消

9. 设备升级完成后,会校验固件完整性,校验通过后会重启设备,运行新固件。并上报固件版本号:

{"type": "report_version", "report":{"version":"mcu_v1.0.1", "fw_type":"mcu"}}

同时在控制台也可以查询升级任务状态已变为成功:

设备详情						
全部设备(1)	升级成功(1)	设备离线(0)	待推送(0)	已推送(0)	升级中(0)	升级失败(0)
设备名称	当前版本号			最后更新时间		升级状态
esp32c3_test	2c3_test mcu_v1.0.1			2024-01-26 17:30:20 升级成功		
设备详情页也可I	以查看设备当前版	运车号:				



 从2023-08-01开始,日志存储时长将由原来的7天改成3天。 									
设备信息	在线调试 云端	诊断日志	设备云端日志	设备本均	地日志	扩展信息			
设备信息									
设备名称	esp32c3_test 🗖			产品ID	3BY0T989	2B 🖻		所属产品	LLSync辅助配网测试
设备密钥	nMk+F			设备创建时间	2023-05-31	1 17:28:35		最后上线时间	2024-01-26 17:30:30
激活时间	2023-06-01 14:09:17			设备状态	在线			mcu固件版本	mcu_v1.0.1
模组固件版本	module_v1.0.0								

2. 用户确认升级

1. 前面的1~6步骤同上,在创建升级任务时,选择用户确认升级。

固件升级		
固件名称	ESP-BOX升级测试	
所属产品	LLSync辅助配网测	试
固件类型	MCU	
固件版本号	mcu_v1.0.1	
批量升级方式 🛈	按固件版本	按设备名称
待升级版本号	mcu_v1.0.0	▼
升级范围	全部设备	•
升级确认	用户确认升级	
超时时长配置 🛈	- 2	+ 分钟
	保	存取消

2. 以连连小程序为例,找到我们创建的设备,展示设备二维码,使用连连小程序扫一扫添加设备。



← 返回项目列表 terminal_test	产品开发 / LLSync辅助配网测试							
开发中心	✓ 物模型 >	→ 设备开发 >	交互开发	4 设备调试	> (5) 批量技	安产		
🛱 产品开发	产品开发 ① 设备调试提供真实、虚拟设备调试功能,便于测试设备上报、接收数据是否正常,可创建测试设备后进行调试							
(ē) 应用开发								
🕄 数据开发	新建设备虚拟设	後备调试			设备名称 ▼ 输	入设备名称搜索 Q		
基础服务	设备名称	状态	激活时间	最后上线时间	操作	绑定网关		
🛞 固件升级	eagle_long_long_long_lo ng name device	离线	2023-11-16 17:27:18	2023-11-16 19:14:08	调试 二维码 删除	查看		
■ 资源管理								
🗈 设备管理	esp32c3_test	离线	2023-06-01 14:09:17	2024-01-30 14:32:48	调试 二维码 删除	查看		
铝 场景联动								

() 说明:

如果设备是通过连连小程序配网连接的网络,则可以省略该步骤,因为在配网完成后,会自动将此设备加入连连家庭,可以在连连小程序 上直接找到该设备。

3. 在连连小程序上找到该设备,点击进入设备详情页,会弹出设备升级提示,点击立即升级。

10.02		111 3	
<	智能灯	•••]	$oldsymbol{O}$
			•••
	关		
	发现新固件		
Y.	发现新设备固件,是否升级? (mcu_v1.0.1)		
う。	取消 立即升	·级	
Į O			
Pad			
Red	Green Bille		
	色温		
		-	

🔗 腾讯云

() 说明:

如果这里没有弹出固件升级的提示框,则点击右上角的三个点进入设备详情页,然后点击固件升级来手动检查升级。

4. 连连小程序会显示下载进度,直到下载完成,在设备详情页点击固件升级可以查看到最新的固件版本号。







15:08		al S	86
<	设备详情	•••]	•
设备名称			` >
设备信息			
创建群组			
房间设置			
设备分享			
固件升级	_		>
	已是最新版本		
	当前固件版本为mcu_v1.0.1		
	确定		

断点续传

在弱网环境或者是设备意外断电的情况下,设备下次上电后,上报完当前的固件版本号都会触发设备升级的任务,然后设备就可以收到平台推送的 升级请求,在拉取固件前,设备可以读取当前已下载的固件大小,然后决定是否从当前已下载的固件断点处继续传输。断点续传的流程如下所示:





不论是手动确认升级还是静默升级,设备都是支持断点续传的。在上面的升级过程中,可以认为复位开发板,观察到如下日志,说明设备是从断点 处开始拉取固件:

DBG|1970-01-01 00:00:07|mqtt_client_publish.c|qcloud_iot_mqtt_publish(339): publish topic seq=2289|topicName=\$ota/report/3BY0T9892B/esp32c3_test|payload={"type": "report_version", "report":{"version":"module_v1.0.0", "fw_type":"module"}} INF|1970-01-01 00:00:07|iot_explorer.c|_event_handler(107): publish success, packet-id=2288 DBG|1970-01-01 00:00:07|ota_mqtt.c|_otamqtt_upgrage_cb(111): topic=\$ota/update/3BY0T9892B/esp32c3_test INF|1970-01-01 00:00:07|ota_mqtt.c|_otamqtt_upgrage_cb(112): len=91, topic_msg= {"type":"report_version_rsp","version":"mcu_v1.0.0","result_code":0,"result_msg":"success"} INF|1970-01-01 00:00:07|ota_client.c|_ota_callback(104): Report version success! INF|1970-01-01 00:00:07|iot_explorer.c|process_ota(420): wait for ota upgrade command...



DBG 1970-01-01 00:00:08 ota_mqtt.c _ otamqtt_upgrage_cb(111): topic=\$ota/update/3BY0T9892B/esp32c3_test INF|1970-01-01 00:00:08|ota_mqtt.c|_otamqtt_upgrage_cb(112): len=475, topic_msg= {"version":"mcu_v1.0.1", "downloaded_size":208896} {"version":"mcu_v1.0.1", "downloaded_size":212992} to /data/FW_mcu_v1.0.1.json {"version":"mcu_v1.0.1", "downloaded_size":217088} to /data/FW_mcu_v1.0.1.json DBG 1970-01-01 00:00:11 | mqtt_client_publish.c|qcloud_iot_mqtt_publish(346): publish {"version":"mcu_v1.0.1", "downloaded_size":221184} to /data/FW_mcu_v1.0.1.json {"version":"mcu_v1.0.1", "downloaded_size":225280} to /data/FW_mcu_v1.0.1.json DBG|1970-01-01 00:00:12|mqtt_client_publish.c|qcloud_iot_mqtt_publish(346): publish