

物联网 MQTT 例程使用平台为阿里云物联

注意：本教程仅以目前的界面显示说明，以后界面、部分功能位置可能随阿里云版本变化，使用时请以网站实际界面显示为准

例程基于野火小智 WIFI_ESP8266-12F 模块和 DHT11 温湿度模块，使用时需**先烧录 1MB MQTT 透传 AT 固件**

板载有 ESP8266 的开发板不需要用杜邦线连接单独的 ESP8266 模块，直接使用跳线帽连接好相关引脚就可以使用板载的 ESP8266 了。而对于没有板载 ESP8266 的开发板就需要按照例程的接线方式，来外接一个单独的 ESP8266 模块

对于板载有 ESP8266 的开发板，也可以使用外接 ESP8266 模块，但是要先断开板载 ESP8266 与串口的跳帽，断开之后再使用我们的例程

野火 STM32F103 指南者、F103 霸道和 F407 霸天虎开发板板载了 ESP8266 模块（指南者、霸道 V1、霸天虎 V1 板载的 esp8266 的 Flash 只有 512K，烧录不了 1MB 固件，只能使用基础通信例程，可以通过外接野火小智 WIFI_ESP8266-12F 模块等使用物联网 MQTT 例程）

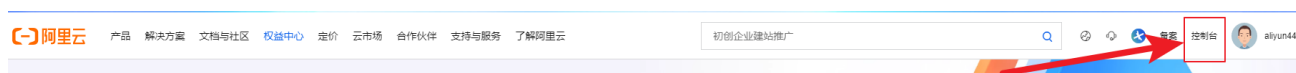
例程的具体引脚分配与接线方式请查看 **ESP8266 模块与 STM32 开发板引脚连接说明.xlsx**，结合实际源码内容查看。

1. 浏览器搜索阿里云并点击进入登录界面，注册账号并登录

阿里云 登录



2. 登录后点击进入控制台

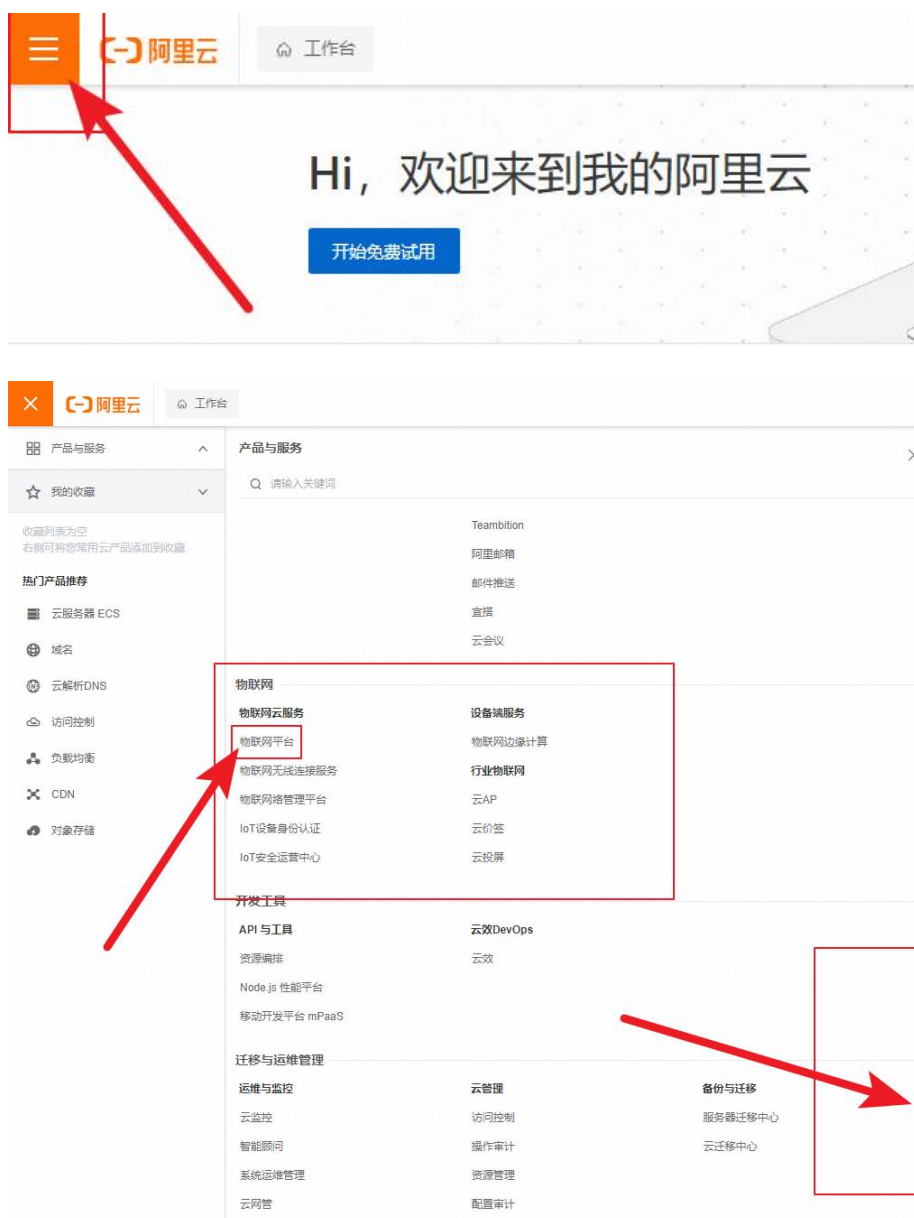


3.使用阿里云物联网需要通过实名认证, 请先完成实名认证和账号安全设置

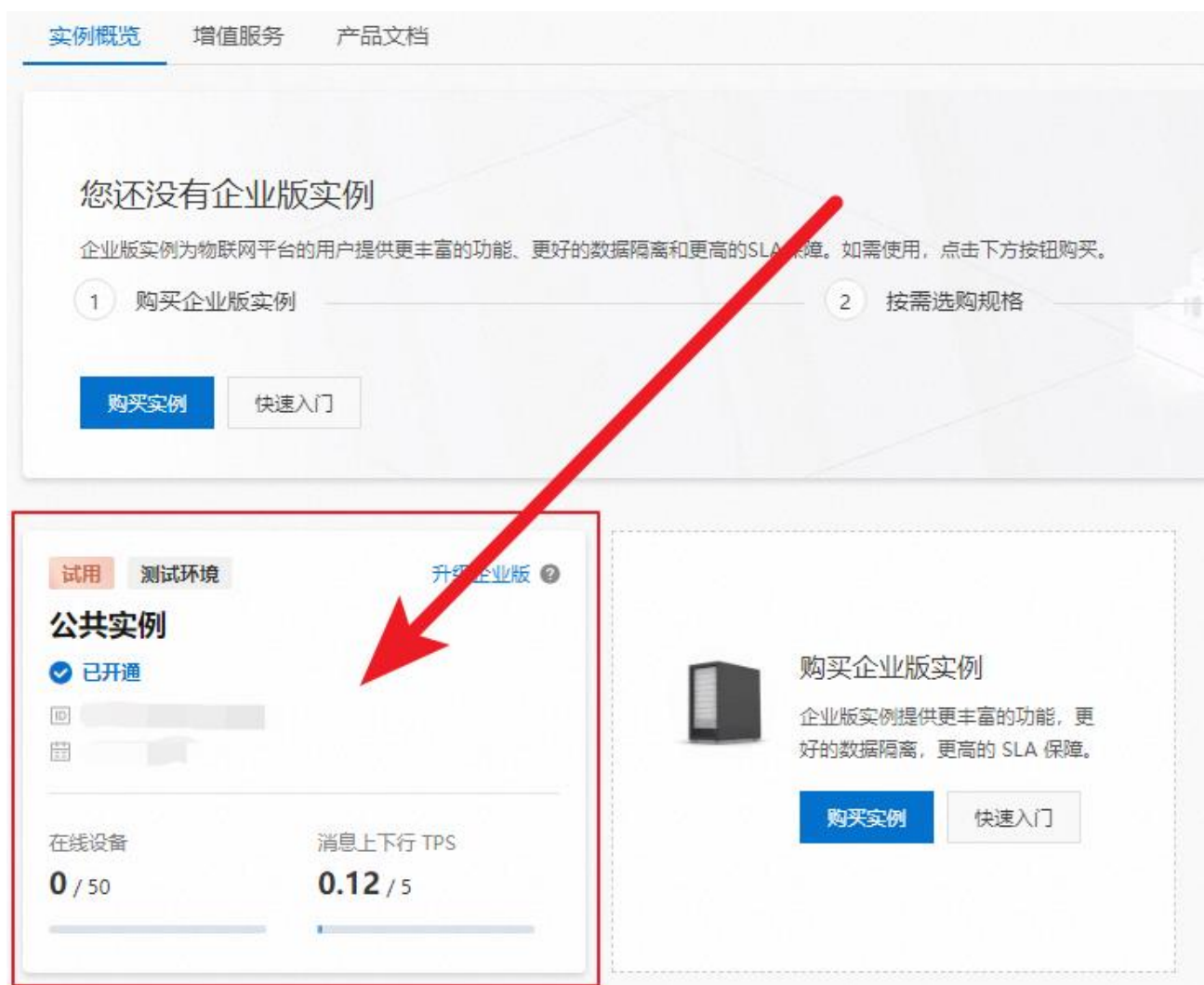
Hi, 欢迎来到我的阿里云



4.点击三横杠里面下拉选物联网平台



5.公共实例是免费，临时试用测试可以开通这个，开通后点击进入实例



6. 点击设备管理里的产品，点击创建产品



7.填入产品名称，选自定义品类，点击确定

← 新建产品 (设备模型)

新建产品 从设备中心新建产品

* 产品名称
温湿度显示设计

* 所属品类
☐ 标准品类 ☒ 自定义品类

* 节点类型
直连设备 网关子设备 网关设备

连网与数据

* 连网方式
Wi-Fi

* 数据格式
ICA 标准数据格式 (Alink JSON)

✓ 校验类型

✓ 认证方式

更多信息

✓ 产品描述

确认 取消

8.产品创建完成后点击前往添加

← 创建产品

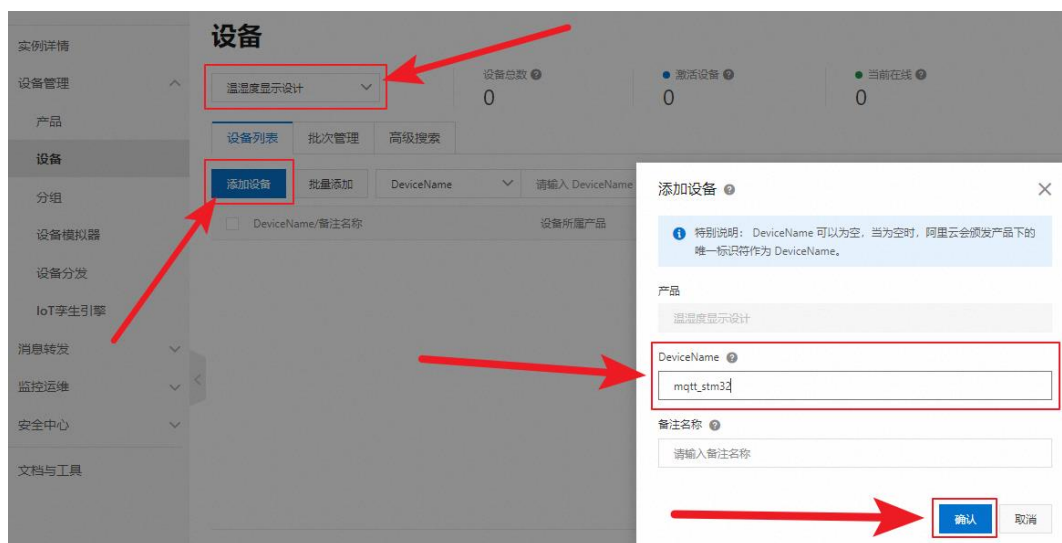
您已成功完成创建产品，接下来您可以：

添加设备
设备归属于某个产品下的具体设备。物联网平台为设备颁发产品内唯一的证书 DeviceName，设备可以直接连接物联网平台，也可以作为子设备通过网关连接物联网平台。
前往添加

功能定义
物联网平台支持为产品定义物模型，将实际产品抽象成由属性、服务、事件所组成的数据模型，便于云端管理和数据交互。产品创建完成后，您可以为它定义物模型，产品下的设备将自动继承物模型内容。
前往功能定义

设备开发
点击下方按钮前往设备开发流程引导，将设备快速接入物联网平台并量产
前往设备开发

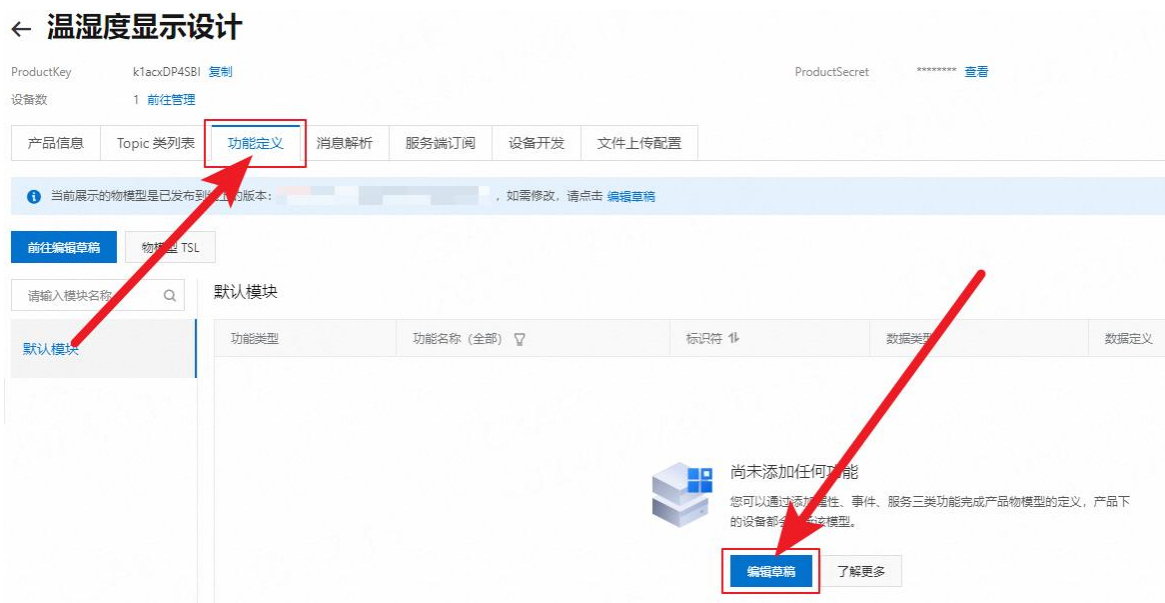
9.选择刚刚创建产品名称，点击创建设备，修改 DeviceName 后点击确认



10.点击回产品，选择刚刚创建产品，点击查看



11.进入功能定义页面，点击编辑草稿



12. 点击添加自定义功能，创建三个物理模型，温度、湿度和 LED 灯，填入相关参数点击确认



添加自定义功能 ×

* 功能类型 ?

属性 服务 事件

* 功能名称 ?

温度

* 标识符 ?

temp

* 数据类型

int32 (整型) ▼

取值范围

-20 ~ 60

步长

请输入步长

单位

摄氏度 / °C ▼

* 读写类型

☒ 读写 ☐ 只读

描述

请输入描述 0/100

确认 取消

添加自定义功能 ×

* 功能类型 ?

属性 服务 事件

* 功能名称 ?

湿度

* 标识符 ?

humi

* 数据类型

int32 (整型) ▼

取值范围

5 ~ 95

步长

请输入步长

单位

相对湿度 / %RH ▼

* 读写类型

☒ 读写 ☐ 只读

描述

请输入描述 0/100

确认 取消

添加自定义功能

×

* 功能类型 ?

属性

服务

事件

* 功能名称 ?

灯状态

* 标识符 ?

led

* 数据类型

bool (布尔型)

▼

* 布尔值

0 - 关

1 - 开

* 读写类型

☒ 读写 ☐ 只读

描述

请输入描述

0/100

确认

取消

13.点击发布上线，勾选确认后，点击确定

默认模块

添加标准功能

添加自定义功能

| 功能类型 | 功能名称 (全部) | 标识符 | 数据类型 | 数据定义 | 操作 |
|------|-----------|-----|------------|------------------------|-------|
| 属性 | 灯状态 | led | bool (布尔型) | 布尔值: 0 - 关 1 - 开 | 编辑 删除 |
| 属性 | 湿度 | | 数值型 | 取值范围: 5 ~ 95 | 编辑 删除 |
| 属性 | 温度 | | 数值型 | 取值范围: -20 ~ 60 | 编辑 删除 |

发布上线

返回

您正在编辑的是草稿，需点击发布后，物模型才会正式生效。

发布物模型上线?

×

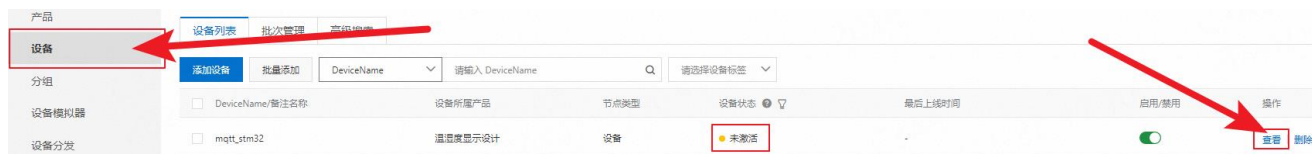
+ 添加发布物模型

☒ 确认已查看当前版本与线上版本的 比对结果

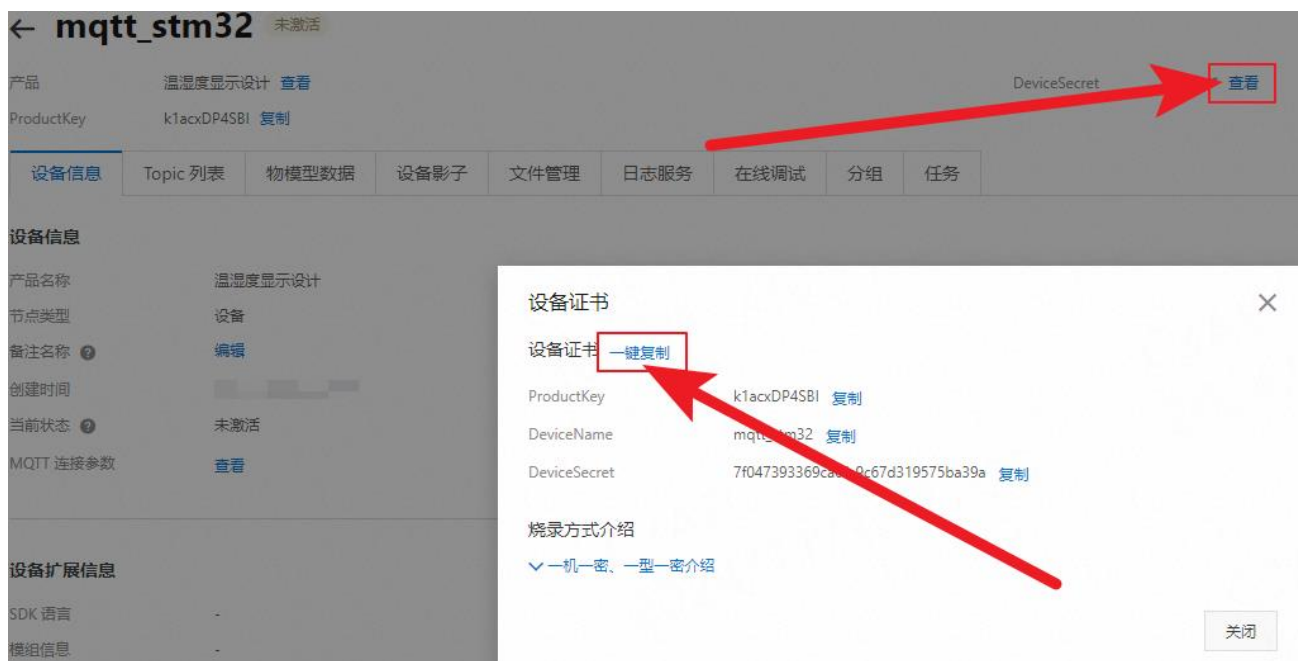
确定

取消

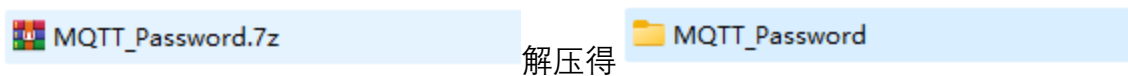
14. 点击设备界面，可以看到设备处于未激活状态，点击查看



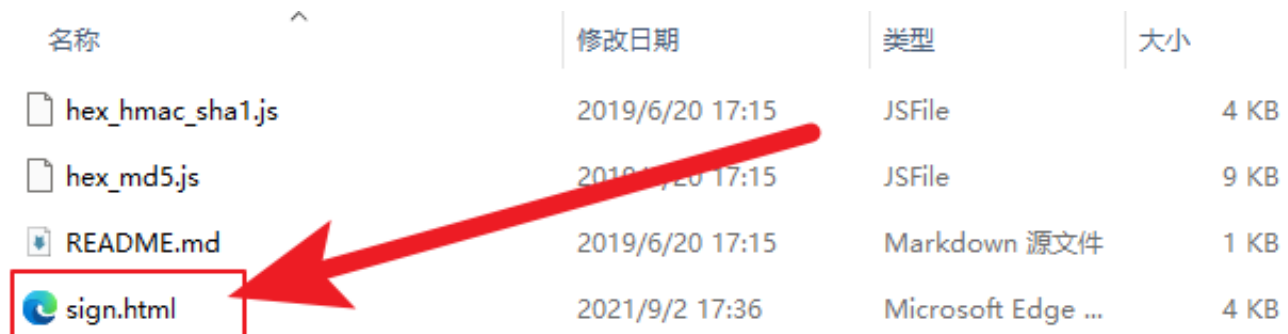
15. 点击查看 DeviceSecret，一键复制阿里云连接三元素，粘贴到记事本中使用



16. 从例程文件夹下的 MQTT_Password.7z 解压出 MQTT_Password 文件夹



17. 打开 MQTT_Password 文件夹双击 sign.html 文件



18.软件将阿里云连接的三元素转换为 MQTT 连接的 Client ID、用户名和密码，对应填入相关参数，点击 Generate，复制计算结果供后续使用

无标题 - 记事本

文件 编辑 查看

```
{
  "ProductKey": "k1acxDP4SBI",
  "DeviceName": "mqtt_stm32",
  "DeviceSecret": "7f047393369ca6fa9c67d319575ba39a"
}
```

MQTT协议的password计算器，也就是设备端签名

填入设备信息：

productKey:k1acxDP4SBI

deviceName:mqtt_stm32

deviceSecret:7f047393369ca6fa9c67d319575ba39a

timestamp:1732852282145

clientId:mqtt_stm32

method:hmacsha1

注：此处为设备clientId，并非mqttClientId

点击这里：Generate

计算结果：

mqttClientId: mqtt_stm32|securemode=2,signmethod=hmacsha1,timestamp=1732852282145

username: mqtt_stm32&k1acxDP4SBI

password: 46100E6E696C21DF6336E4D060A9C1B0477FB5FE

复制出来后续用

此处可以随意填，将设备名填入

19.打开物联网 MQTT 例程，填入相关参数

打开 bsp_esp8266_test.h 文件，修改热点名称密钥、服务器域名、填入复制的 Client ID、用户名、密码、主题和发布，框住的内容都需仔细按实际填入

特别注意在 Client ID 内需要加入\\

bsp_esp8266_test.h

```
1 #ifndef __BSP_ESP8266_TEST_H
2 #define __BSP_ESP8266_TEST_H
3
4
5 #include "stm32f10x.h"
6
7
8 /***** 用户需要设置的参数 *****/
9 #define macUser_ESP8266_ApSsid "wifiname" //要连接的热点的名称
10 #define macUser_ESP8266_ApPwd "123456" //要连接的热点的密钥
11
12
13 #define MQTT_BROKERADDRESS "k1acxDP4SBI.iot-as-mqtt.cn-shanghai.aliyuncs.com" //服务器域名
14
15 #define MQTT_CLIENT_ID "mqtt_stm32|securemode=2\\,signmethod=hmacsha1\\,timestamp=1732852282145|" //客户编号
16 #define MQTT_USER_NAME "mqtt_stm32&k1acxDP4SBI" //用户名
17 #define MQTT_PASSWD "46100E6E696C21DF6336E4D060A9C1B0477FB5FE" //用户密码
18 #define MQTT_SUBSCRIBE_TOPIC "/sys/k1acxDP4SBI/mqtt_stm32/thing/service/property/set" //订阅主题
19 #define MQTT_PUBLISH_TOPIC "/sys/k1acxDP4SBI/mqtt_stm32/thing/event/property/post" //消息发布
20
21 /***** 外部全局变量 *****/
22 extern volatile uint8_t ucTcpClosedFlag;
```

其中服务器域名格式需要按物联网平台设备所在地域的修改 Region ID

| 通信类型 | 接入域名格式 | 端口号 |
|------|---|--|
| MQTT | <code>\${YourProductKey}.iot-as-mqtt.\${YourRegionId}.aliyuncs.com</code> | 1883或443。 使用MQTT over WebSocket时，使用443端口。 |

20.按照工程下 [readme.txt](#) 文件内的实验操作进行后，若填入参数都正确，通过串口调试助手查看打印信息，下图为连接成功时的打印

```
正在连接MQTT .....
AT+MQTTCONN=0,"k1acxDP4SBI.iot-as-mqtt.cn-shanghai.aliyuncs.com",1883,0

busy p...
+MQTTCONNECTED:0,1,"k1acxDP4SBI.iot-as-mqtt.cn-shanghai.aliyuncs.com","1883","",0

OK

正在订阅MQTT .....
AT+MQTTSUB=0,"/sys/k1acxDP4SBI/mqtt_stm32/thing/service/property/set",0

OK

配置 ESP8266 完毕
等待接收MQTT LED控制数据
```

同时可以看到设备由未激活状态转为在线状态

| 设备列表 批次管理 高级搜索 | | | | | | | |
|---|-----------------|---------|------|------|--------|-------------------------------------|-------|
| 添加设备 批量添加 DeviceName 请输入 DeviceName 请选择设备标签 | | | | | | | |
| <input type="checkbox"/> | DeviceName/备注名称 | 设备所属产品 | 节点类型 | 设备状态 | 最后上线时间 | 启用/禁用 | 操作 |
| <input type="checkbox"/> | mqtt_stm32 | 温湿度显示设计 | 设备 | ● 在线 | | <input checked="" type="checkbox"/> | 查看 删除 |

21.添加手机设备，实现 esp8266 设备和手机设备之间互相通讯

设备

温湿度显示设计

设备列表

批次管理

高级搜索

添加设备

批量添加

设备总数

1

激活设备

1

当前在线

1

DeviceName

请输入 DeviceName

请选择设备标签

| <input type="checkbox"/> | DeviceName/备注名称 | 设备所属产品 | 节点类型 | 设备状态 | 最后上线时间 | 启用/禁用 | 操作 |
|--------------------------|-----------------|---------|------|------|--------|-------------------------------------|-------|
| <input type="checkbox"/> | mqtt_stm32 | 温湿度显示设计 | 设备 | ● 在线 | | <input checked="" type="checkbox"/> | 查看 删除 |

22.填入手机设备的信息并点击确认

添加设备

特别说明：DeviceName 可以为空，当为空时，阿里云会颁发产品下的唯一标识符作为 DeviceName。

产品

温湿度显示设计

DeviceName

phone

备注名称

手机

确认

取消

23.添加完成后，点击前往查看

添加完成

添加设备成功，接下来您可以：

查看设备信息

设备信息页面包含：设备信息、运行状态、事件管理、服务调用、设备影子、文件管理等设备维度的功能集合。

前往查看

烧入 SDK 至设备

您可以依照您的需求，前往选择不同的语言的 SDK，并复制下方的设备证书，烧入至您的设备。（注：您可以随时在设备信息中，查看此设备的设备证书）

查看 SDK

一键复制设备证书

查看设备信息

完成

24.按之前操作，复制阿里云连接三元素，通过软件获得 MQTT 连接的 Client ID、用户名和密码

```
{
  "ProductKey": "k1acxDP4SBI",
  "DeviceName": "mqtt_stm32",
  "DeviceSecret": "7f047393369ca6fa9c67d319575ba39a"
}
{
  "ProductKey": "k1acxDP4SBI",
  "DeviceName": "phone",
  "DeviceSecret": "e72beb132b0f2881690504065a9568b7"
}
```

[MQTT协议的password计算器，也就是设备端签名](#)

填入设备信息：

productKey: k1acxDP4SBI

deviceName: phone

deviceSecret: e72beb132b0f2881690504065a9568b7

timestamp: 1732852282145

clientId: phone (注：此处为设备clientId，并非mqttClientId)

method: hmacsha1

点击这里：

计算结果：

mqttClientId: phone|securemode=2,signmethod=hmacsha1,timestamp=1732852282145 (注：可直接用于mqtt.fx的Client ID字段填写)


username: phone&k1acxDP4SBI

password: 198F7200EABA6570C52227D2616E1376BF17E68A

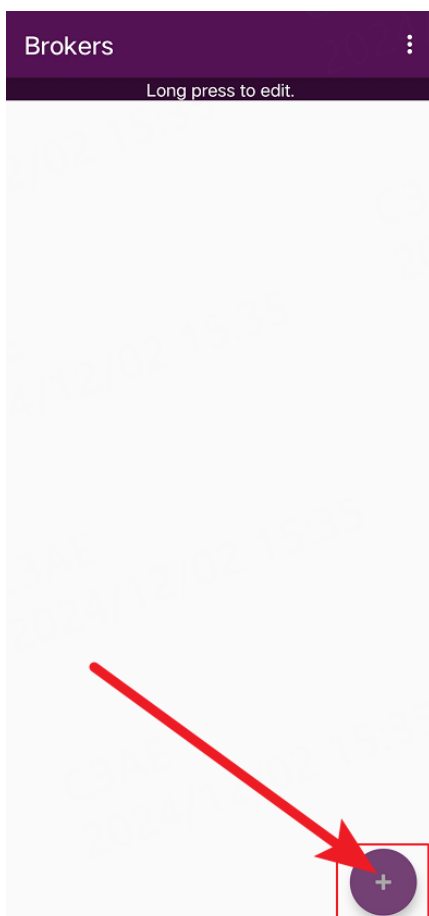
使用设备名来填入

复制出来用

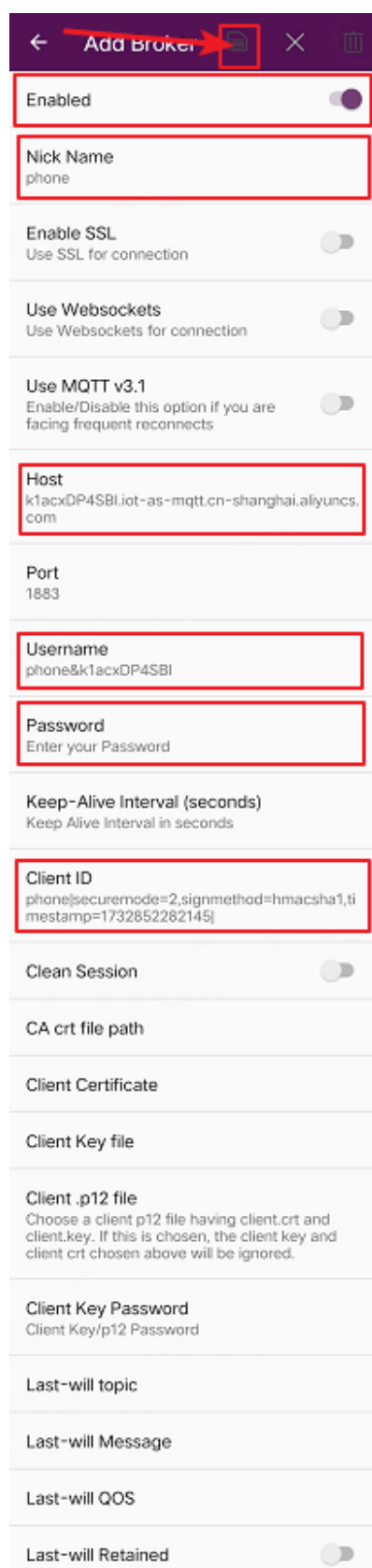
25.使用安卓手机安装例程文件夹下的 MQTT Client APP 软件

 MQTT Client_4.5.1_APKPure.apk

26.打开软件，点击加号图标



27.填入昵称、服务器域名、Client ID、用户名和密码，框住的内容都需仔细按实际填入，最后点击保存



← Add Broker

Enabled

Nick Name
phone

Enable SSL
Use SSL for connection

Use Websockets
Use Websockets for connection

Use MQTT v3.1
Enable/Disable this option if you are facing frequent reconnects

Host
k1acxDP4SBI.iot-as-mqtt.cn-shanghai.aliyuncs.com

Port
1883

Username
phone&k1acxDP4SBI

Password
Enter your Password

Keep-Alive Interval (seconds)
Keep Alive Interval in seconds

Client ID
phone|securemode=2,signmethod=hmacsha1,timestamp=1732852282145|

Clean Session

CA crt file path

Client Certificate

Client Key file

Client .p12 file
Choose a client p12 file having client.crt and client.key. If this is chosen, the client key and client crt chosen above will be ignored.

Client Key Password
Client Key/p12 Password

Last-will topic

Last-will Message

Last-will QOS

Last-will Retained

28.点击刚创建的客户端信息，当界面显示有已连接等字样说明连接成功，否则请检查填入内容是否正确



29.回到设备界面，可以看到手机设备已处于在线状态

设备

温湿度显示设计

设备总数 2 激活设备 2 当前在线 2

设备列表 批次管理 高级搜索

添加设备 批量添加 DeviceName 请输入 DeviceName 请选择设备标签

| <input type="checkbox"/> | DeviceName/备注名称 | 设备所属产品 | 节点类型 | 设备状态 | 最后上线时间 | 启用/禁用 | 操作 |
|--------------------------|-----------------|---------|------|------|--------|-------------------------------------|----------|
| <input type="checkbox"/> | phone 手机 | 温湿度显示设计 | 设备 | ● 在线 | | <input checked="" type="checkbox"/> | 查看 删除 |
| <input type="checkbox"/> | mqtt_stm32 | 温湿度显示设计 | 设备 | ● 在线 | | <input checked="" type="checkbox"/> | 查看 删除 |

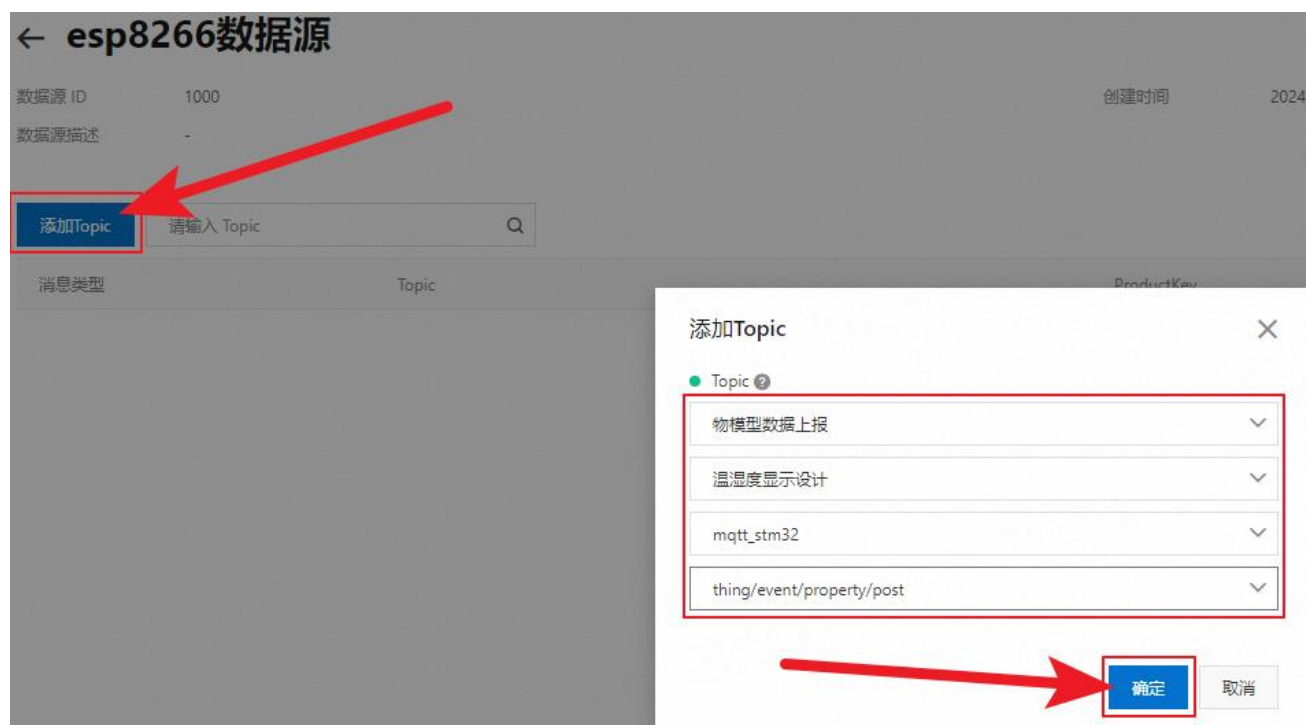
30.点击进入云产品流转界面，创建数据源



31.填入数据源名称，确认后前往编辑



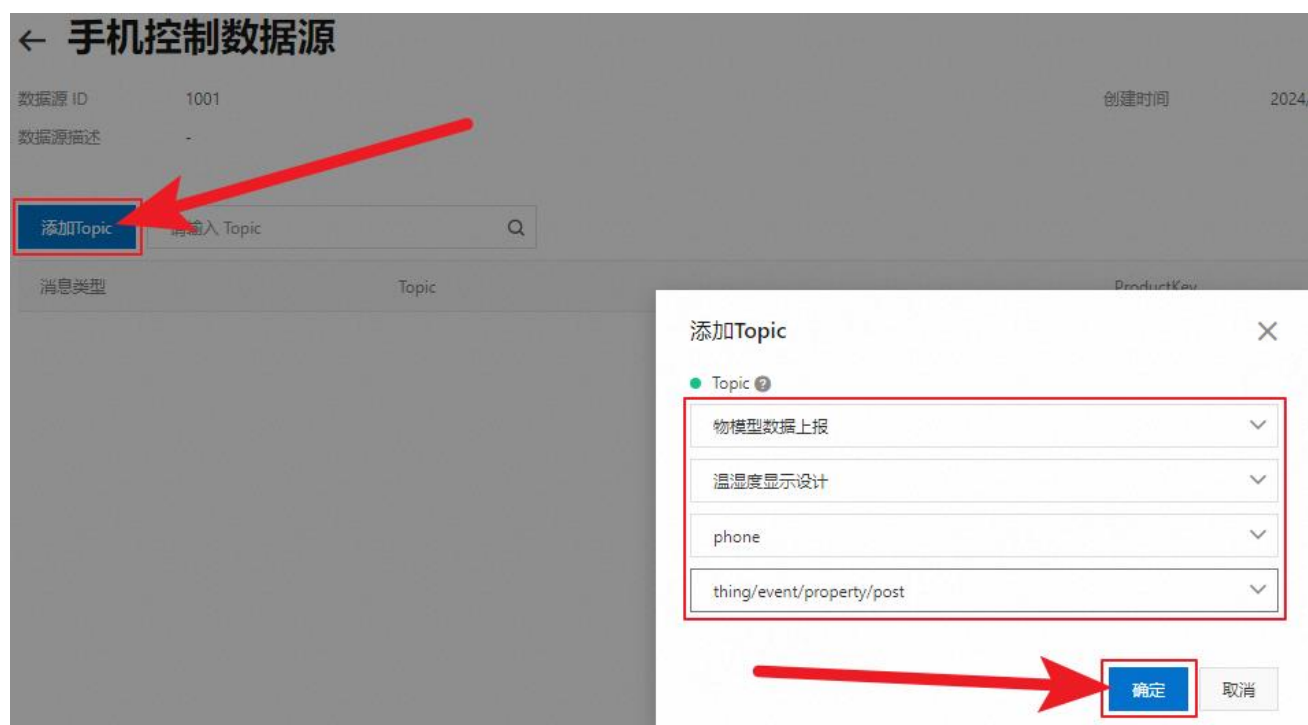
32.添加 Topic，选择物模型数据上报，对应选择 esp8266 设备后点击确定



33.同样步骤创建手机控制数据源



34.添加 Topic，选择物模型数据上报，对应选择手机设备后点击确定



35.返回，进入数据目的界面，点击创建数据目的



36.创建发送到手机和发送到 esp8266 的数据目的

创建数据目的

* 数据目的的名称 ②

发送到手机

数据目的的描述

请输入数据目的描述

0/100

选择操作 ②

发布到另一个 Topic

产品

温湿度显示设计

确定

取消

创建数据目的

* 数据目的的名称 ②

发送到esp8266

数据目的的描述

请输入数据目的描述

0/100

选择操作 ②

发布到另一个 Topic

产品

温湿度显示设计

确定

取消

37.进去解析器界面，点击创建解析器

解析器

数据源

数据目的

创建解析器

导入旧规则

请输入解析器名称

Q

| 解析器名称 | 解析器ID | 解析器描述 | 创建时间 |
|-------|-------|-------|------|
|-------|-------|-------|------|

尚未创建任何解析器

解析器需要关联数据源和数据目的，可通过脚本将流转的数据加工处理。

创建解析器

导入旧规则

使用指南

38.esp8266 的数据发送到手机，填入解析器名称，确认后前往编辑

创建解析器

解析器可以对设备上报的数据进行处理，需要关联数据源和数据目的使用，将处理后的数据流转到其他Topic或阿里云产品中。

* 解析器名称

esp8266到手机

解析器描述

请输入解析器描述

0/100

确定

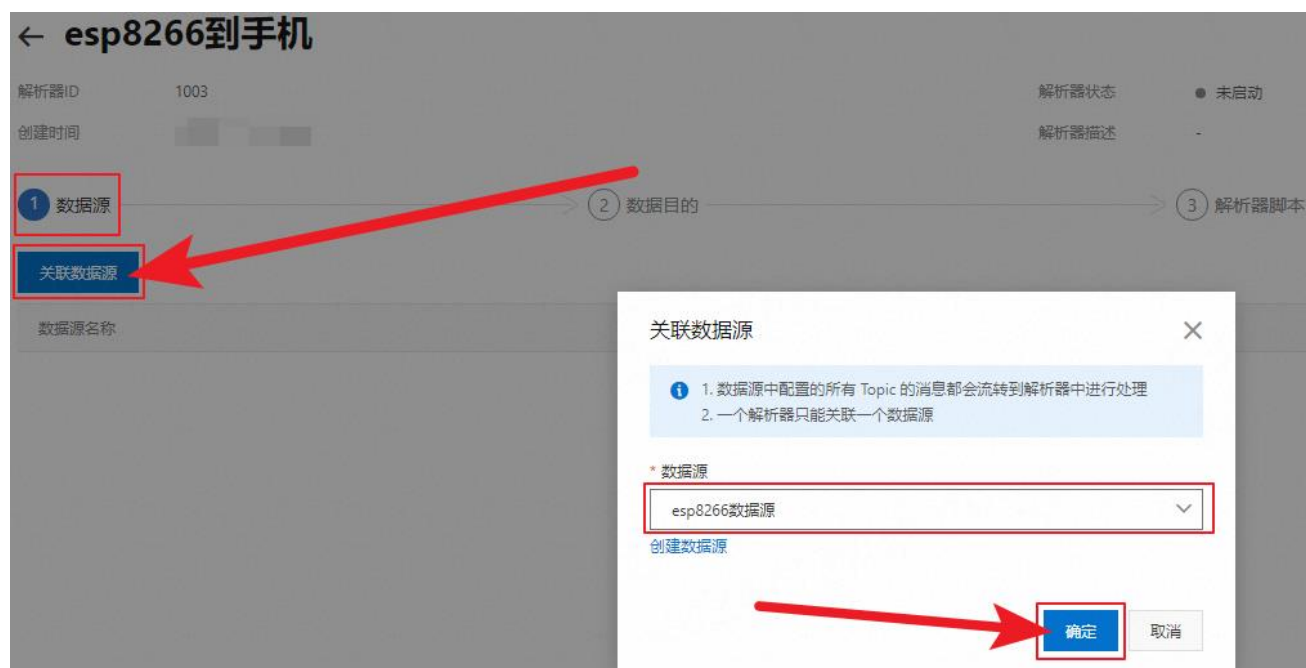
取消

✓

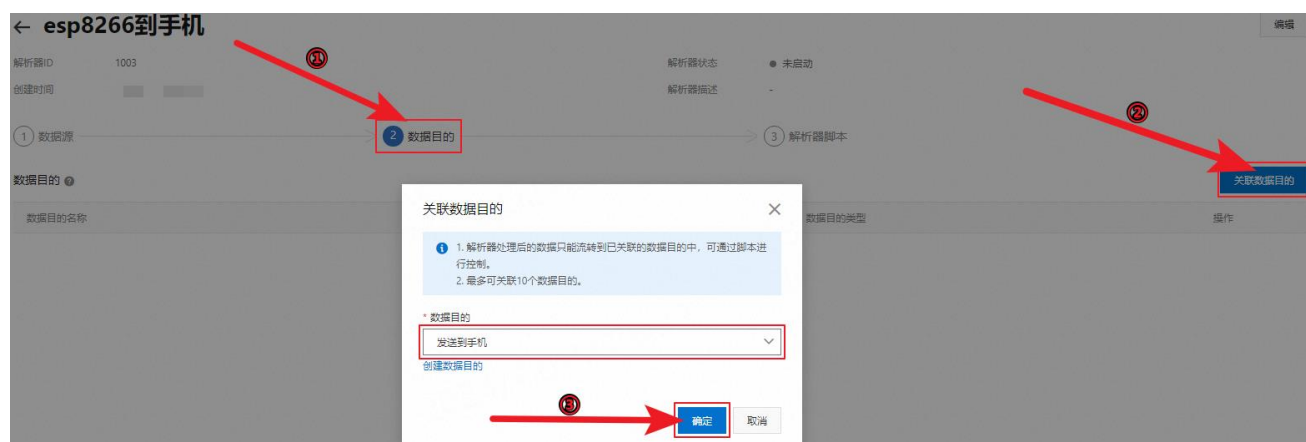
创建解析器成功

前往编辑(2)

39. ①关联数据源，选择 esp8266 设备的数据源



40. ②关联数据目的，选择发送到手机的数据目的



41. ③编辑解析器脚本，调用数据转发到其他 Topic 函数，保存后点击发布

← esp8266到手机

解析器ID1003

创建时间

解析器状态● 未启动

解析器描述-

1 数据源

2 数据目的

3 解析器脚本

编辑脚本 (当前展示为: 草稿)

1 // 草稿页为空时，进入草稿页会生成默认脚本

2 // 如果默认脚本自动保存过，继续绑定数据目的，默认脚本不会自动更新

3 // 此时清空脚本并保存之后，重新进入草稿页即可重新生成包含最新数据目的的默认脚本

4

5 // 设备上报数据内容，json格式

6 var data = payload('json');

7

8 // 流转到另一个Topic

9 writeIotTopic(1004, "/sys/klacxDP4SBI/phone/thing/service/property/set", data);

数据目的ID

对应填你自己设备的信息

发布

调试

保存

42.创建另一个解析器，手机的数据发送到 esp8266，填入解析器名称，确认后前往编辑

创建解析器

1 解析器可以对设备上报的数据进行处理，需要关联数据源和数据目的的使用，将处理后的数据流转到其他Topic或阿里云产品中。

* 解析器名称 ?

手机到esp8266

解析器描述

请输入解析器描述

0/100

确定 取消

43. ①关联数据源，选择手机设备的数据源

手机到esp8266

解析器ID 1004

创建时间

解析器状态 未

解析器描述 -

1 数据源 2 数据目的 3 解析器

关联数据源

数据源名称

关联数据源

1. 数据源中配置的所有 Topic 的消息都会流转到解析器中进行处理
2. 一个解析器只能关联一个数据源

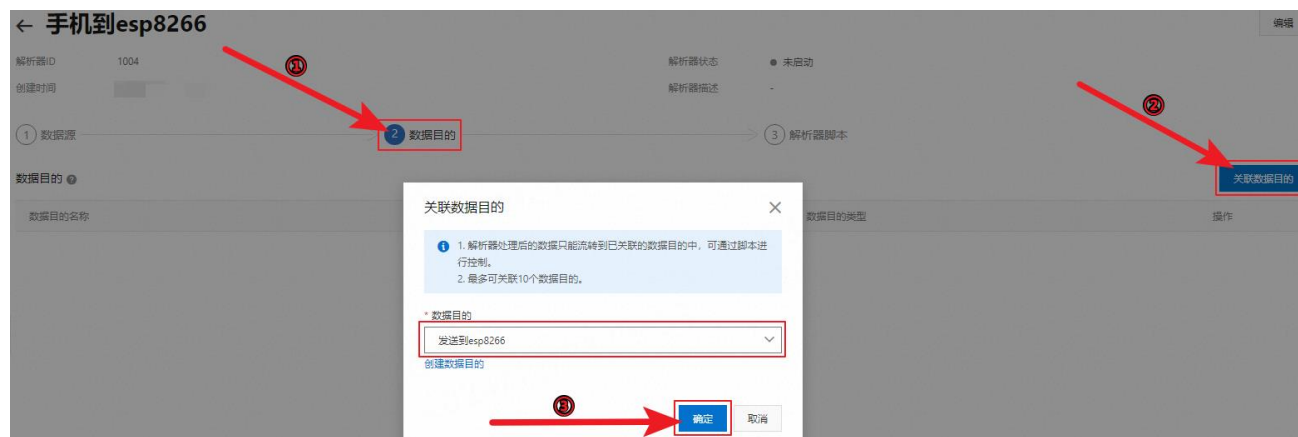
* 数据源

手机控制数据源

创建数据源

确定 取消

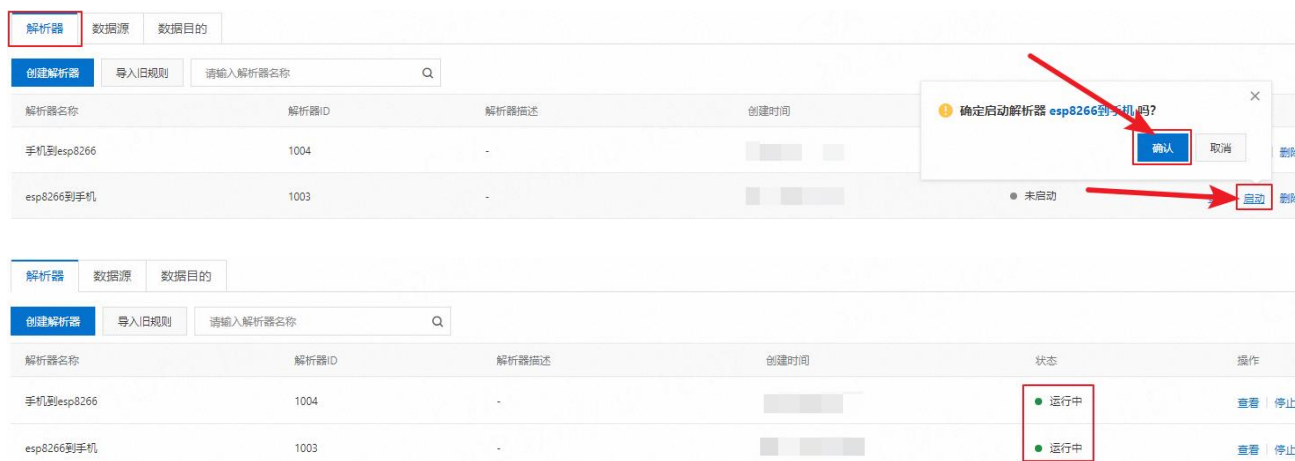
44. ②关联数据目的，选择发送到 esp8266 的数据目的



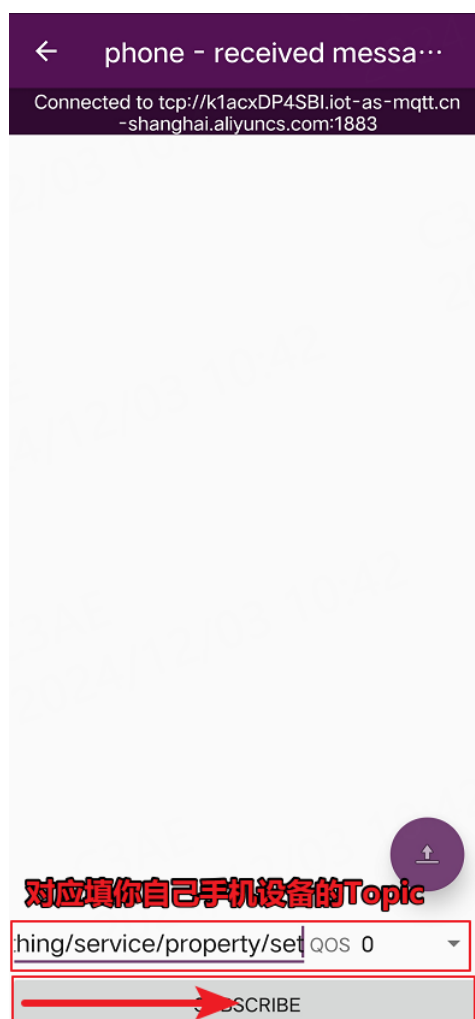
45. ③编辑解析器脚本，调用数据转发到其他 Topic 函数，保存后点击发布



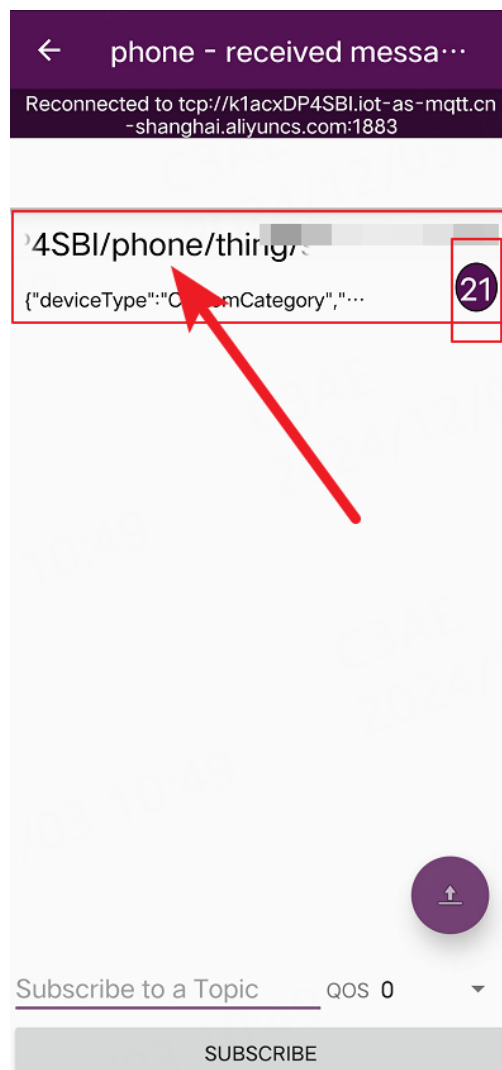
46.返回到解析器界面，将刚创建的两条解析器启动



47.手机上订阅设备属性设置 Topic， /sys/k1acxDP4SBI/phone/thing/service/property/set， 对应填你自己手机设备的 Topic， 然后点击订阅



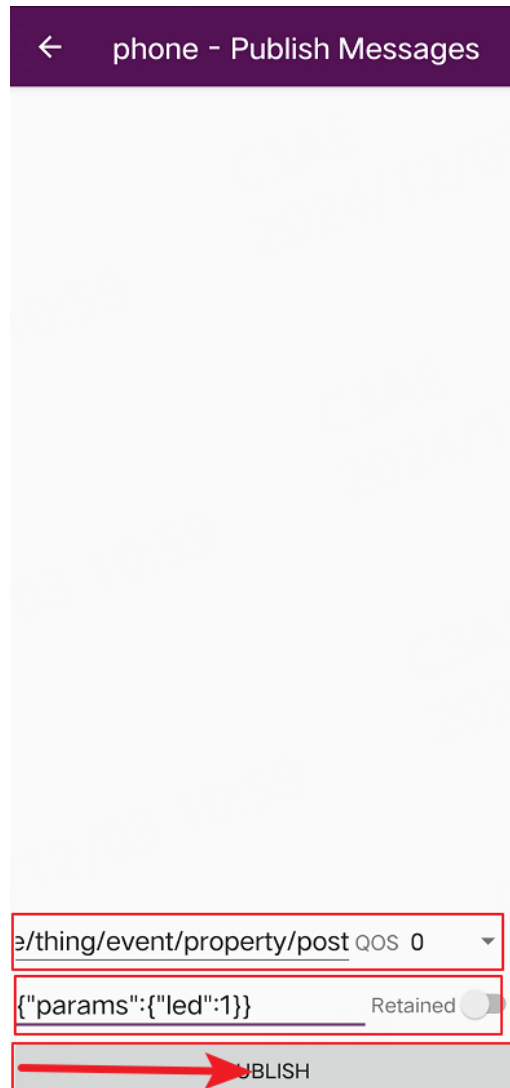
48.订阅成功后可以看到 esp8266 发送到手机的数据，可以点击进去查看数据



49. 点击右下角的向上箭头图标，进入消息发布界面



50.填入设备属性上报 Topic，填入要发送的 LED 控制数据，{"params":{"led":1}}控制 LED 打开， {"params":{"led":0}}控制 LED 关闭， /sys/k1acxDP4SBI/phone/thing/event/property/post 对应填你自己手机设备的 Topic，然后点击发布



The screenshot shows a mobile application interface titled "phone - Publish Messages". At the bottom, there are two input fields and a "PUBLISH" button. The first input field contains the topic "/sys/k1acxDP4SBI/phone/thing/event/property/post" and has "QOS 0" and a dropdown arrow to its right. The second input field contains the message '{"params":{"led":1}}' and has a "Retained" toggle switch to its right. A red arrow points to the "PUBLISH" button.

← phone - Publish Messages

/sys/k1acxDP4SBI/phone/thing/event/property/post QOS 0

{"params":{"led":1}} Retained

PUBLISH

51.查看串口打印，可以看到 esp8266 设备接收到手机设备发送的 LED 控制数据并将 LED 点亮

```
读取DHT11成功!  
湿度为45.0 %RH ， 温度为 27.1℃  
LED灯状态为 1  
  
正在发布消息 .....  
AT+MQTTPUB=0,"/sys/klacxDP4SBI/mqtt_stm32/thing/event/property/  
post",{"params":{"temp":27,"humi":45,"led":1},"version":  
"1.0.0"},0,0  
  
OK  
  
接收到LED控制数据，等待消息发布  
LED：1  
  
读取DHT11成功!  
湿度为45.0 %RH ， 温度为 27.1℃  
LED灯状态为 1
```