

阿里 IoT 平台接入

本文档适用于如下系列产品，产品硬件说明请参阅产品手册。

	<p>Elfin-EW 系列 (EW10, EW11) Elfin-EE 系列 (EE10, EE11) Elfin-EG 系列 (EG10, EG11)</p>
	<p>4G 串口系列(HF2411, G43, EG4X)</p>
	<p>Eport 系列 (E10, E20, E20-PIN, E30) Eport Pro 系列 (EP10, EP20, EP20-PIN)</p>
	<p>以太网系列串口服务器 (HF5111A, HF511B)</p>

	<p>Wi-Fi 系列串口服务器 (HF2211, HF2221)</p>
	<p>4G 路由器系列 (HF2421, HF2421G)</p>

目录

1.	概述	4
1.1.	名词解释.....	4
1.2.	设备连接示意图.....	5
2.	阿里 IOT 云端配置.....	6
2.1.	建立设备模型	6
2.2.	透传编写解析脚本	8
2.3.	创建设备.....	11
3.	汉枫 DTU 配置	13
3.1.	G43 设备端配置	13
3.2.	数据通讯测试	17
3.3.	基础版配置.....	19

历史记录

V 1.0 03-28-2019 初版

V 1.1 05-05-2019 更新 Studio

1. 概述

本文将介绍如何通过符合阿里 IoT 接入协议规范的汉枫产品，快速实现将存量串口设备接入阿里 Studio 平台。

在工业、农业、医疗、城市、楼宇、园区等多种不同的场景中，存在着大量的通过串口与外界通讯的存量设备。当在对此类设备进行物联网改造时，往往无法修改设备本身的串口传输协议，而只能将数据解析的工作放到云端完成。为了快速使此类设备接入和使用阿里云 IoT 平台，阿里云 IoT 定义出了可以通过简单配置，即可快速接入阿里云 IoT 平台。

阿里 IoT Studio: <https://iot.aliyun.com/products/iotstudio>

阿里 IoT Studio 文档: <https://studio.iot.aliyun.com/doc>

阿里 IoT Studio 功能演示视频: <http://ftp.hi-flying.com:9000/IOTStudio/>



1.1. 名词解释

物模型: 阿里云 IoT 将任意设备抽象为”属性“，”事件“，”服务“三大要素，设备/事物的三要素抽象描述，我们称为物模型。

属性: 对事物某特征的描述，如电机的转速，灯的亮度，水的温度。

事件: 对事物某状态改变的描述，如开机，重启，报警。

服务: 事物对外提供的能力的描述，如调整电机转速，使灯打开，可以理解为事物对外提供的可被调用的命令。

三元组: 阿里云 IoT 平台为标识设备的唯一性而提供的三个字符串，三个字符串的组合则标识一个阿里云 IoT 平台上的一个唯一设备。三个字符串分别为 ProductKey, DeviceName, DeviceSecret。调试过程请手动配置三元组，批量阶段汉枫可支持提前配置烧录。

ProductKey: 阿里云 IoT 平台为每个种类的产品所分配的一个标识符，如某某型号的冰箱，某某型号的锅炉。可以看作一类设备的标识符。

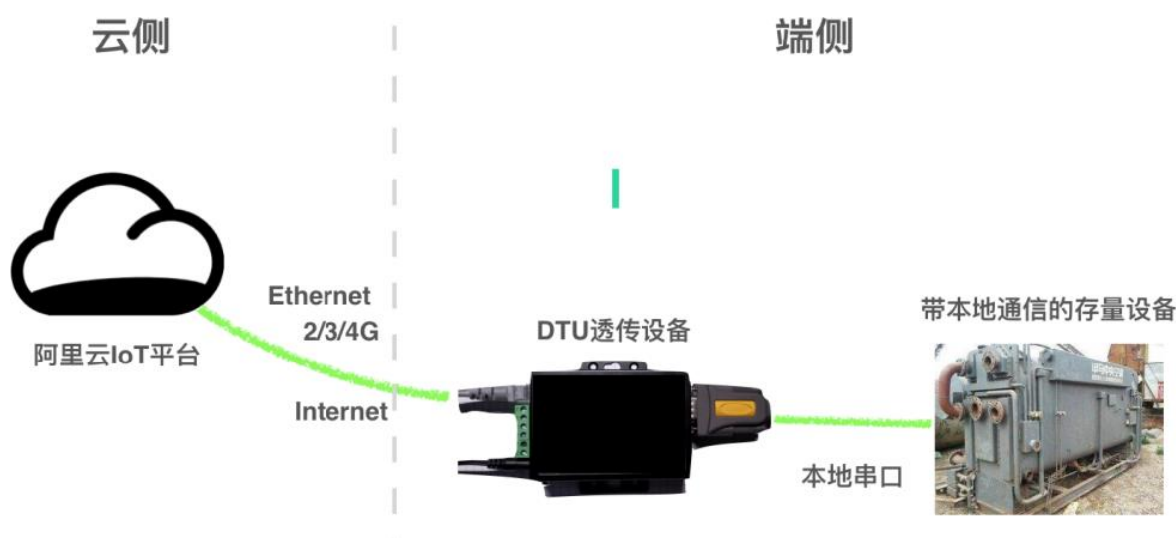
DeviceName: 阿里云 IoT 平台为某类设备中的具体某个设备提供的名称标识符，此标识符可以由用户指定或由平台产生。

DeviceSecret: 阿里云 IoT 平台为某类设备中的具体某个设备提供的密钥，此密钥由平台自动生成，三元组不可泄露。

解析脚本: 阿里云 IoT 平台为透传设备提供了云端数据到物模型数据转换的 js 脚本开发能力，用户可以自己编写符合设备数据传输协议的脚本，用于将设备端的 16 进制数据转换为物模型（平台识别需要使用物模型）。

1.2. 设备连接示意图

本地设备通过串口与 DTU 设备相连，DTU 通过 2/4G，以太网，Wi-Fi 等与平台相连。由 DTU 设备实现阿里云 IoT 的接入协议，设备无需关心改协议，设备三元组将被配置到 DTU 中，由 DTU 代表设备完成数据到 IoT 平台的收发。



汉枫产品全线支持接入阿里云 IoT 平台（需要更新到最新的固件，部分还未正式发布，需汉枫获取）。

以太网物联	Wi-Fi物联	GPRS 2G物联	4G物联
超级网口 [Eport-E20-PIN] [Eport-E20] [Eport-E30] Linux超级网口 [Eport Pro-EP20-PIN] [Eport Pro-EP20] 以太网串口服务器 [HF5111A] [HF5111B] 以太网多串口服务器 [HF5142A] [HF5142B]	Wi-Fi DTU模块 [Wport-W20] [Wport-W10] Wi-Fi串口服务器 [HF2211] [DTU-H100] Wi-Fi多串口服务器 [HF2221] wifi 路由器 (导轨) [HF8104W]	GPRS串口服务器 [HF2111] [HF2111A]	4G 串口服务器 [HF2411] 4G+WiFi+GPS 串口服务器 [HF2421G] 4G+WiFi串口服务器 [HF2421] 4G路由器(导轨式) [HF8102] [HF8104]
小精灵物联	远程控制		
4G_LTE [Elfin-EG4X] GPRS [Elfin-EG1X] Wi-Fi [Elfin-EW1X] 以太网 [Elfin-EE1X]	Wi-Fi型IO控制 [HF6208] 以太网型IO控制 [HF6508]		

2. 阿里 IOT 云端配置

下面介绍将设备接入阿里 IoT 平台的云端操作步骤。

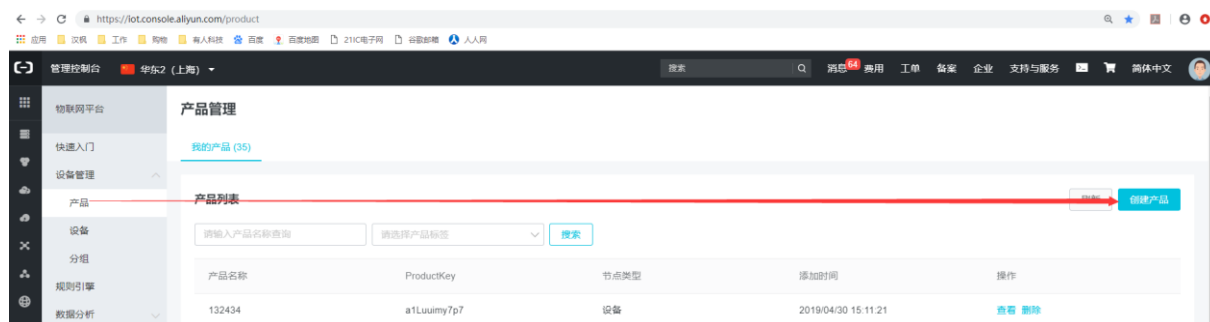
2.1. 建立设备模型

根据对设备的抽象，在云端建立设备模型。在本文中，我们通过一个虚拟的电机变频器设备，来向用户演示数据采集和控制。

设备端物模型抽象如下表：

名称	类型	数据类型	标记符	单位	描述
转速	属性	Int32	speed	rpm	电机当前转速
电流	属性	Int32	current	A	电机当前电流
转速设定值	属性	Int32	setspeed	rpm	电机设定转速值

打开 <https://iot.console.aliyun.com/product>，选择【产品】，点击【创建产品】



配置产品基本信息。

* 产品名称

变频电机

* 所属分类

自定义品类 功能定义

节点类型

* 节点类型

设备 网关

* 是否接入网关

是 否

连网与数据

* 连网方式

蜂窝 (2G / 3G / 4G)

* 数据格式

透传/自定义

* 使用 ID² 认证

是 否

产品名称： 设置产品名称

所属分类： 可选择现有品类或者自定义。选择现有品类的话会创建默认的物模型（属性、服务等）

节点类型： 选设备，不接入网关

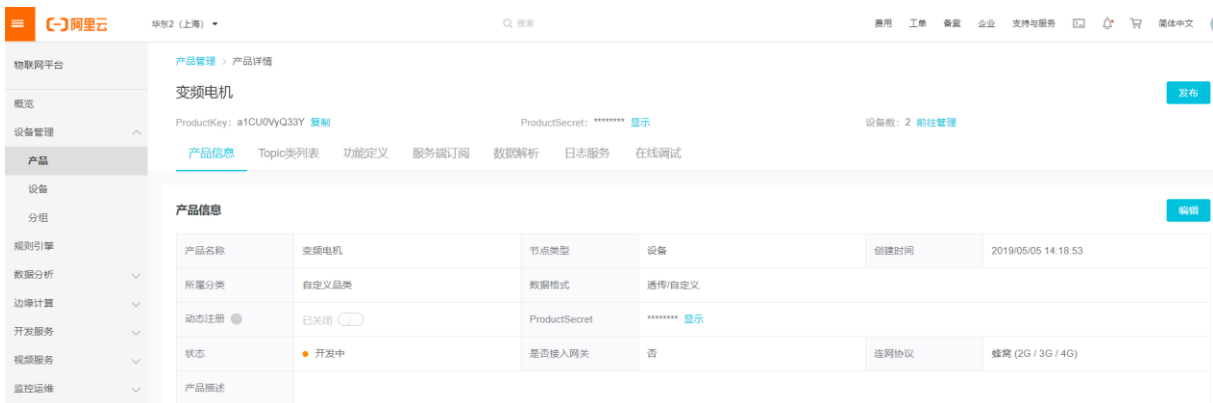
联网方式：选择蜂窝或者以太网、Wi-Fi 等。

数据格式：选透传。

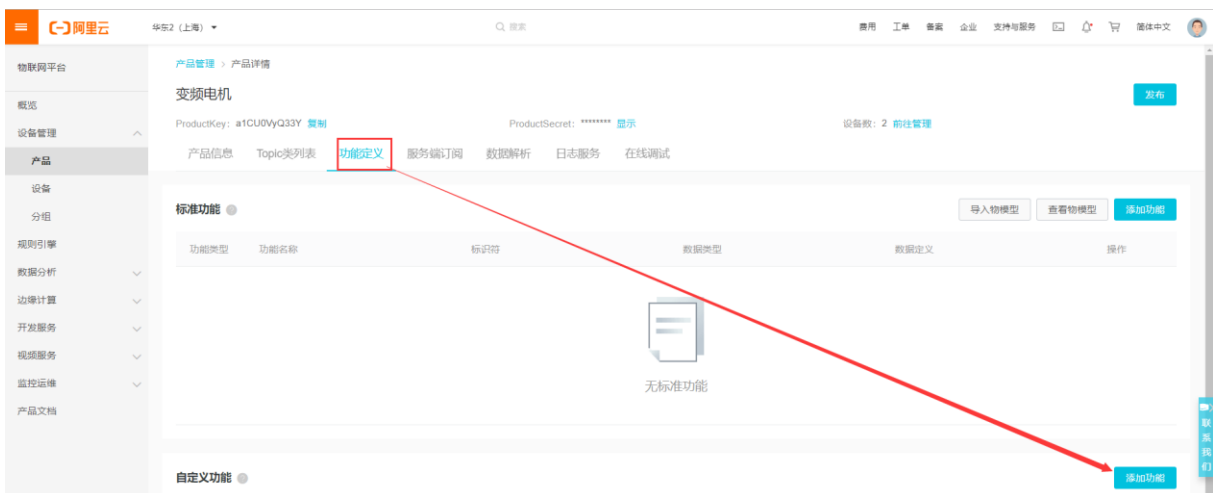
A) 若使用 IoT Studio 功能必须选透传，需要编写脚本，Studio 支持组态可视化，可直接用于数据展现和产品应用)。

B) 如果不用 IoT Studio 的话可选择 ICA 标准数据格式，此方式无需编写脚本，但平台不做数据解析和展现，只管传输，一般需要客户自行搭建服务器处理数据。

完成后即可看到新定义的产品，点击【查看】即可看到产品详情。



点击【功能定义】，添加自定义功能。



* 功能类型:

属性 服务 事件 ?

* 功能名称:

转速 ?

* 标识符:

speed ?

* 数据类型:

int32 (整数型) v

* 取值范围:

0 ~ 3000

* 步长:

1

单位:

转每分钟 / r/min v

读写类型:

读写 只读

定义完成后的功能列表如下。

自定义功能 ? 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	转速	speed	int32 (整数型)	取值范围: 0 ~ 3000	编辑 删除
属性	电流	current	int32 (整数型)	取值范围: 0 ~ 30	编辑 删除
属性	转速设定值	setspeed	int32 (整数型)	取值范围: 0 ~ 3000	编辑 删除

2.2. 透传编写解析脚本

在产品详情页中点击【数据解析】，进行解析脚本编写。



下面的脚本将设备发送的十六进制数据，转换为阿里 IoT 平台能识别的属性数据格式。

脚本说明参见阿里文档：

https://help.aliyun.com/document_detail/68702.html?spm=5176.11485173.0.0.7bb959afpqogEr

将下列脚本粘贴至【编辑脚本】中，脚本的逻辑为，收到数据的第一个字节为转速值，第二个字节为电流值。

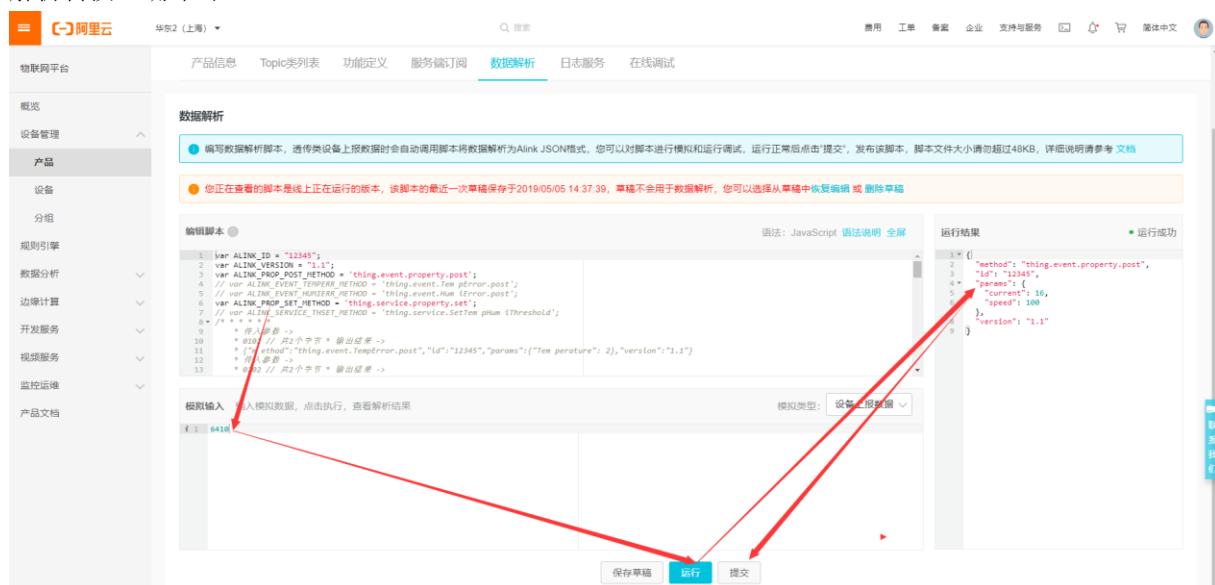
```
var ALINK_ID = "12345";
var ALINK_VERSION = "1.1";
var ALINK_PROP_POST_METHOD = 'thing.event.property.post';
// var ALINK_EVENT_TEMPERR_METHOD = 'thing.event.Tem pError.post';
// var ALINK_EVENT_HUMIERR_METHOD = 'thing.event.Hum iError.post';
var ALINK_PROP_SET_METHOD = 'thing.service.property.set';
// var ALINK_SERVICE_THSET_METHOD = 'thing.service.SetTem pHum iThreshold';
/* *****
 * 传入参数 ->
 * 0102 // 共 2 个字节 * 输出结果 ->
 * {"m ethod":"thing.event.TempError.post","id":"12345","params":{"Tem perature": 2},"version":"1.1"}
 * 传入参数 ->
 * 0202 // 共 2 个字节 * 输出结果 ->
 * {"m ethod":"thing.event.Hum iError.post","id":"12345","params":{"Hum idity":2},"version":"1.1"}
 */
/*此函数用于实现设备上发数据到物模型的转换*/
function rawDataToProtocol(bytes) {
    /*将设备上报的 RAW 数据转换为数组其中 bytes 对象中存储着设备上报 RAW 数据*/
    var uint8Array = new Uint8Array(bytes.length);
    for (var i = 0; i < bytes.length; i++) {
        uint8Array[i] = bytes[i] & 0xff;
    }
    var params = {}; // 定义属性存放对象
    var jsonMap = {}; // 定义模拟 Alink 数据报对象
    /*填写 Alink 数据报协议头部分*/
    jsonMap['version'] = ALINK_VERSION; // Alink 协议版本号
    jsonMap['id'] = ALINK_ID; // 模拟消息 ID
    jsonMap['method'] = ALINK_PROP_POST_METHOD; // 模拟设备上行数据方法->设备属性上报
    /*填写 Alink 数据报属性部分*/
    params['speed'] = uint8Array[0]; // 将收到的第一个字节转换为转速值
    params['current'] = uint8Array[1]; // 将收到的第二个字节转换为电流
    jsonMap['params'] = params; // 将参数打包到数据帧中
    return jsonMap; // 返回时会发送给 IoT 设备管理平台
}
//以下是部分辅助函数
function buffer_uint8(value)
{
    var uint8Array = new Uint8Array(1);
    var dv = new DataView(uint8Array.buffer, 0);
    dv.setUint8(0, value);
    return [].slice.call(uint8Array);
}
function buffer_int16(value)
{

```

```

var uint8Array = new Uint8Array(2);
var dv = new DataView(uint8Array.buffer, 0);
dv.setInt16(0, value);
return [].slice.call(uint8Array);
}
function buffer_int32(value)
{
    var uint8Array = new Uint8Array(4);
    var dv = new DataView(uint8Array.buffer, 0);
    dv.setInt32(0, value);
    return [].slice.call(uint8Array);
}
function buffer_float32(value)
{
    var uint8Array = new Uint8Array(4);
    var dv = new DataView(uint8Array.buffer, 0);
    dv.setFloat32(0, value);
    return [].slice.call(uint8Array);
}
    
```

点击【运行】，在运行结果栏中可以看到转换之后的数据，只需要核对 params 中的属性名和值是否解析转换正确即可。



下面的脚本将阿里 IoT 平台发送的属性数据，转换成设备端可识别的十六进制数据。将如下脚本内容增加到上述脚本中。

/*此函数实现由云端下发数据到设备能识别的 16 进制数的转换*/

```

function protocolToRawData(json)
{
    var method = json['method'];
    var id = json['id'];
    var version = json['version'];
    var payloadArray = [];
    if (method == ALINK_PROP_SET_METHOD) // 接收来自 IoT 设备管理平台的“设置设备属性”的命令
    {
        var send_params = json['params'];
        var prop_cur = send_params['setspeed']; // 将设置的具体值抽取出来
        //按照自定义协议格式拼接 rawdata
        payloadArray = payloadArray.concat(buffer_uint8(0x55)); // 第一字节数据头，标识数据功能用户定义
        payloadArray = payloadArray.concat(buffer_uint8(prop_cur)); // 第二字节，具体的设置值
    }
    return payloadArray; // 返回时，将数据发送至设备端。
}
    
```

复制如下测试数据到【模拟输入】，点击【设备接收数据】，点击【运行】，查看执行结果。执行正常之后点击【提交】。

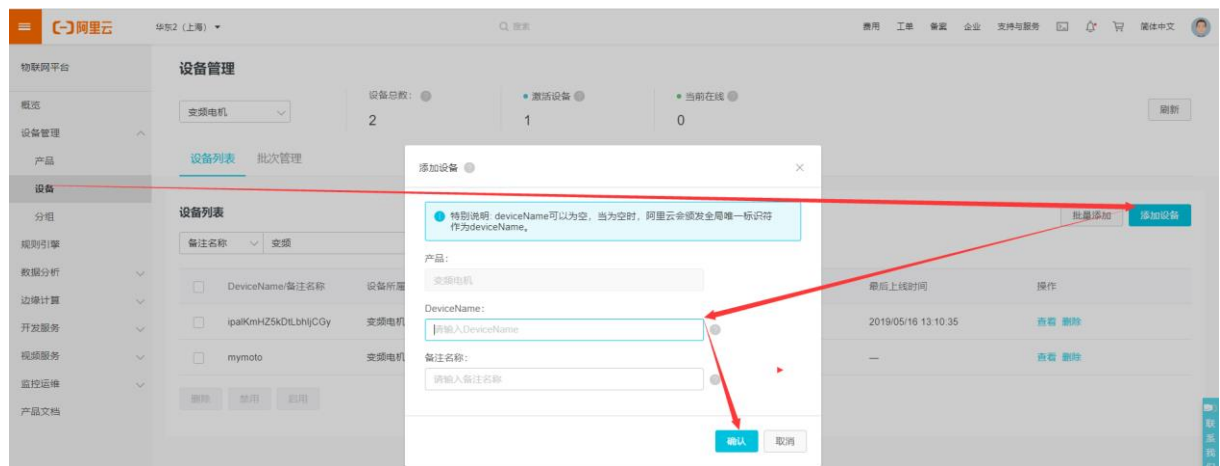
```

{
    "method": "thing.service.property.set",
    "id": "12345",
    "version": "1.1",
    "params": {
        "current": 16,
        "speed": 100
    }
}
    
```



2.3. 创建设备

在完成了产品类型的建立之后，需创建真实的设备，点击【添加设备】。



DeviceName 每个设备都不同，可自动创建随机的或者手动命名，此处按随机填写之后就创建了一个设备，设备的三元组信息如下，此信息需写入到 DTU 中，批量阶段提供三元组给我们，我们出货之前配置好。

A) 手动命名的话建议蜂窝网可使用 IMEI 作为此 DeviceName，以太网/Wi-Fi 可使用 MAC 地址作为 DeviceName。

查看设备证书

① 设备证书用于云端对接入的设备做鉴权认证，请妥善保管！

ProductKey	a1CU0VyQ33Y	复制
DeviceName	ipalKmHZ5kDtLbhjCGy	复制
DeviceSecret	8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I	复制 隐藏

一键复制 关闭

```
{
  "ProductKey": "a1CU0VyQ33Y",
  "DeviceName": "ipalKmHZ5kDtLbhjCGy",
  "DeviceSecret": "8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I"
}
```

设备当前处于未激活状态。

设备列表

DeviceName 请输入DeviceName 请选择设备标签 搜索

DeviceName/备注名称	设备所属产品	节点类型	状态/启用状态	最后上线时间	操作
ipalKmHZ5kDtLbhjCGy	变频器	设备	未激活 <input type="checkbox"/>	—	查看 删除

【设备详情】中可查看设备三元组信息，至此云端配置已经完成，接下来完成 DTU 的配置。

阿里云 华东2 (上海)

物联网平台 设备管理 > 设备详情

ipalKmHZ5kDtLbhjCGy 离线

产品: 变频器 [查看](#) ProductKey: a1CU0VyQ33Y [复制](#) DeviceSecret: ***** [显示](#)

[设备信息](#) [Topic列表](#) [运行状态](#) [事件管理](#) [服务调用](#) [设备影子](#) [文件管理](#) [日志服务](#)

设备信息

产品名称	变频器	ProductKey	a1CU0VyQ33Y 复制	区域	华东2 (上海)
节点类型	设备	DeviceName	ipalKmHZ5k... 复制	DeviceSecret	8tKR4i2M12... 复制 隐藏
备注名称	编辑	IP地址	114.87.148.26	固件版本	app-1.0.0-20181203.1837
添加时间	2019/05/05 15:33:22	激活时间	2019/05/05 18:03:13	最后上线时间	2019/05/16 13:10:35
当前状态	离线	实时延迟	测试		

3. 汉枫 DTU 配置

下文分别按蜂窝网和以太网/Wi-Fi 两类产品进行参数配置，进行介绍。

A) 蜂窝网: G43, 此配置方式同样适用于 Elfin-EG10/EG11, Gport-G10/G11/G12, HF2111A, HF2411 等蜂窝网产品) 等

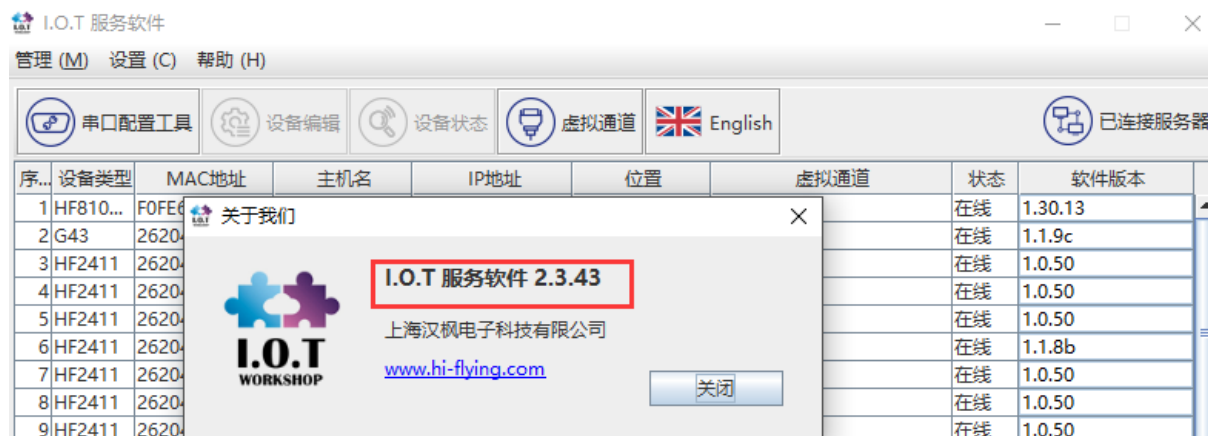
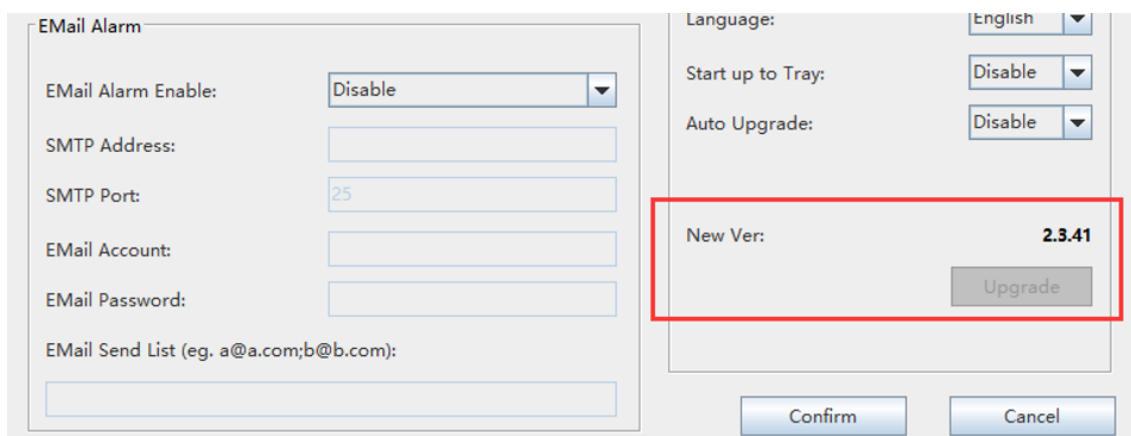
B) 以太网/Wi-Fi: HF5111B, 此配置方式同样适用于 Eport-E10/E20/E30, Eport Pro-EP10/EP20, HF2211, HF2221, HF5111A, HF2421, HF8104 等

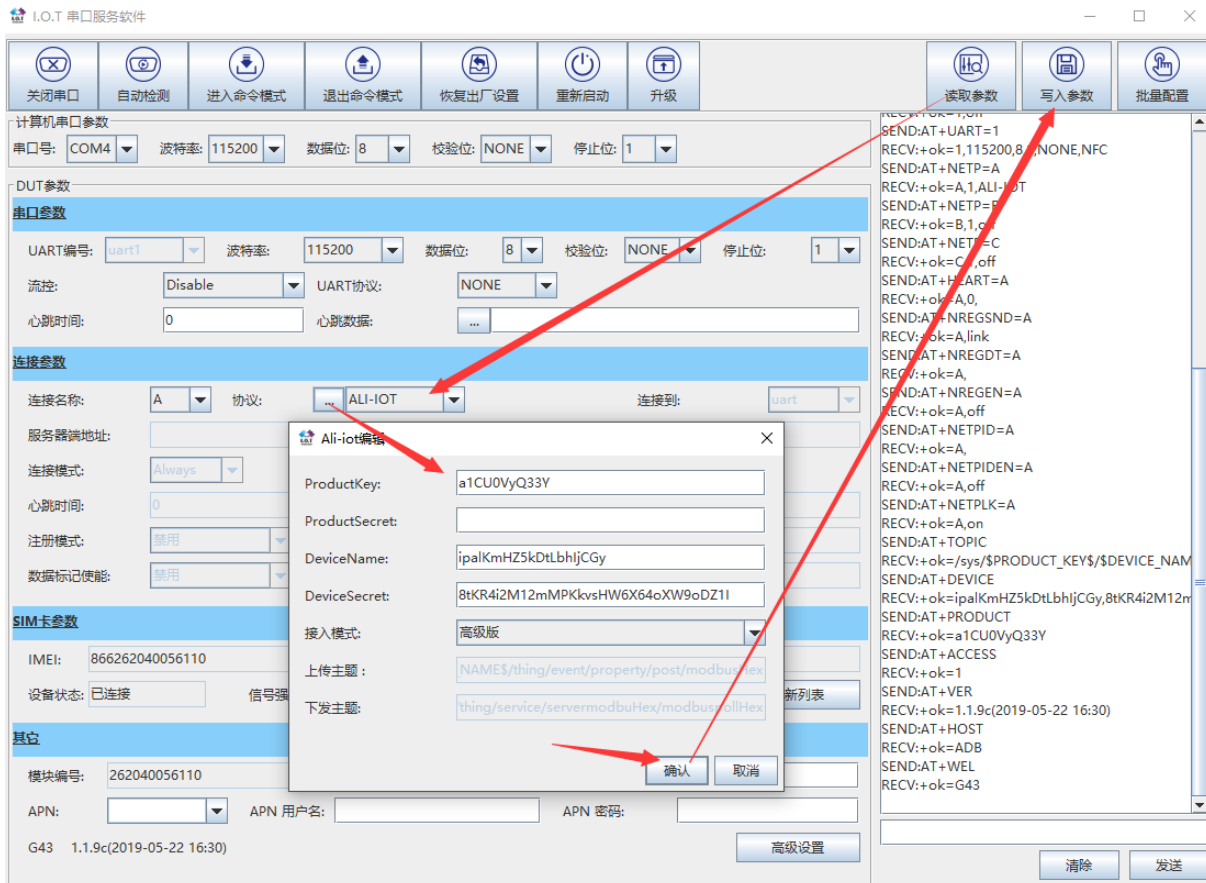
汉枫 DTU 设备的基础使用请参阅相关操作文档进行学习之后再行下文配置。

3.1. G43 设备端配置

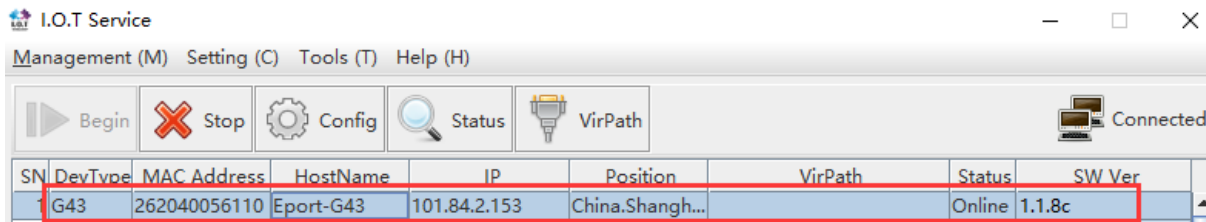
下文选择蜂窝网 Gport-G43 产品进行平台配置。

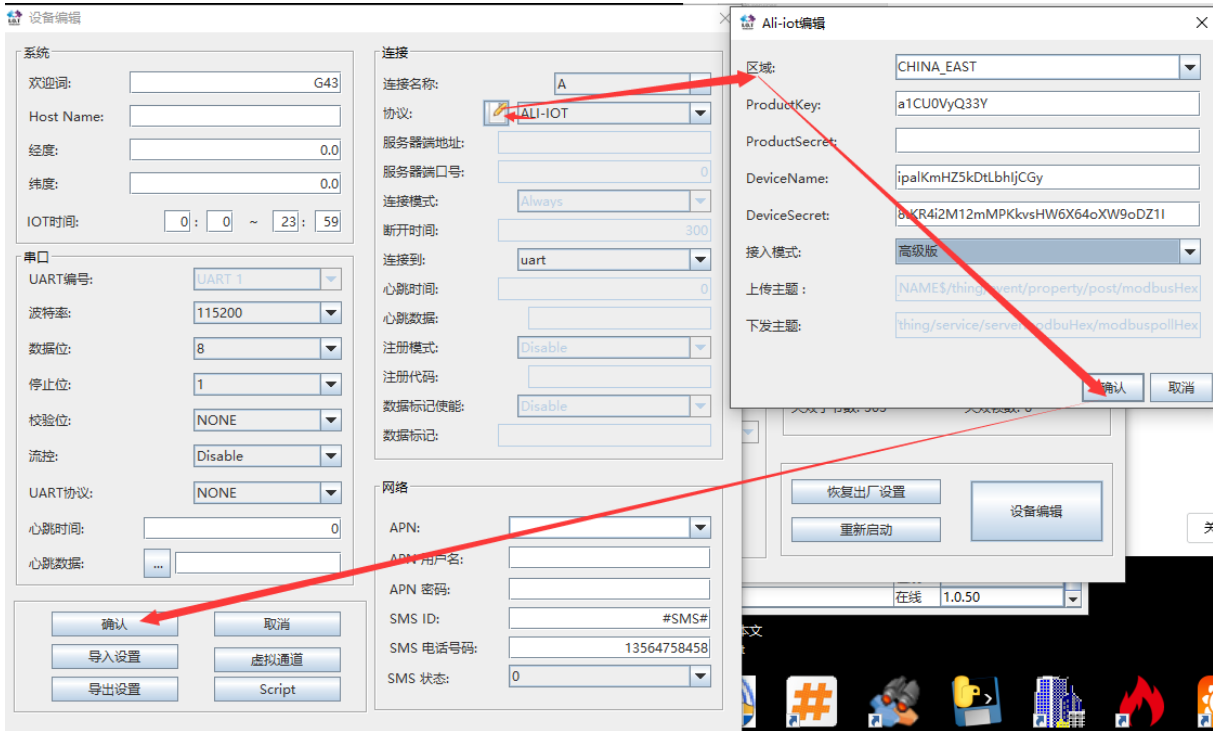
A) 串口方式配置 (工具需更新到最新的版本 2.3.43), 设置完成后重启设备。





B) 网络方式配置，需绑定设备到账户，详见产品基础操作文档。设备绑定到 IOTService 之后，双击此设备，进入产品配置页面，填写 Ali-IOT 相关参数后重启设备。





Domain Addr: 服务器地址，根据产品创建的地区，如下华东 2 就选择 CHINA_EAST



ProductKey: 填入之前创建产品的三元组, a1CU0VyQ33Y

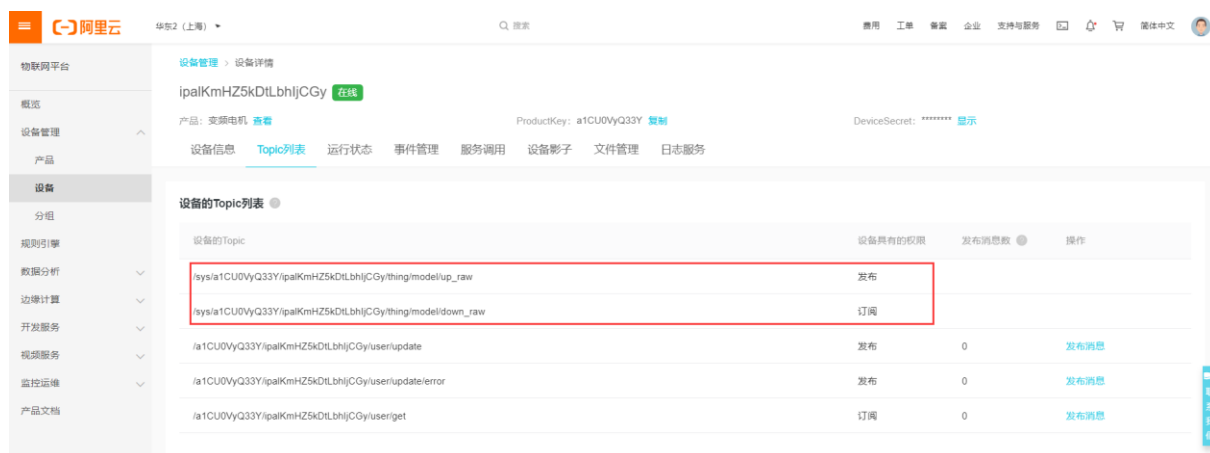
ProductSecret: 悬空不用填。

DeviceName: 填入之前创建产品的三元组, ipalKmHZ5kDtLbhjCGy

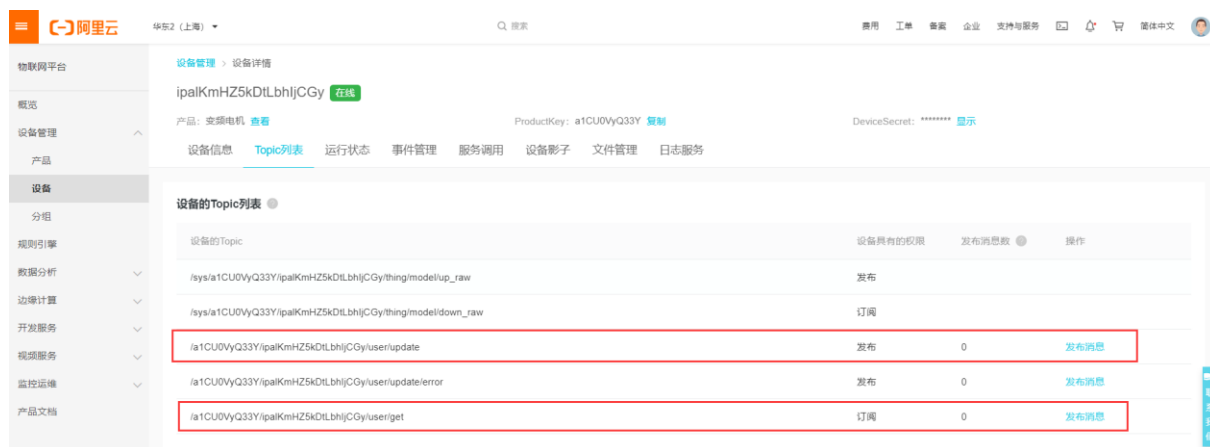
DeviceSecret: 填入之前创建产品的三元组, 8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I

接入模式: 阿里 IoT Studio 使用高级版, 故此处填高级版, 如果自建服务器且只使用阿里 IoT 基础 MQTT 服务, 可选择基础版。

高级版主题固定是用如下阿里服务器定的通道。



基础版主题可修改，默认是如下的。



C) 串口 AT 指令进行配置，配置完成之后 AT+Z 或者断电重启一下。

串口 AT 指令定义如下。

AT+DEVICE=<DeviceName> ,<DeviceSecret> (设置三元组命令)

AT+PRODUCT=<ProductKey>

AT+ACCESS=accessMode

- accessMode=1，高级版，对应阿里 IoT Studio，数据格式必须选透传，由平台做脚本解析。
- accessMode=2，基础版，对应阿里 IoT 基础 MQTT 通讯，如果使用基础版，Topic 可通过 AT+TOPIC 命令进行配置修改 AT+TOPIC=PUBLISH,SUBSCRIBE (设置发布，订阅主题)，数据格式支持 ICA 标准或者透传脚本两种。

高级版设置例子：

AT+NETP=A,1,ALI-IOT

AT+DEVICE=ipalKmHZ5kDtLbhjCGy,8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I

AT+PRODUCT=a1CU0VyQ33Y

AT+ACCESS=1

```
AT+NETP=A,1,ALI-IOT
+ok

AT+DEVICE=ipalKmHZ5kDtLbhjCGy,8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I
+ok

AT+PRODUCT=a1CU0VyQ33Y
+ok

AT+ACCESS=1
+ok
```


基础版设置例子:

AT+NETP=A,1,ALI-IOT

AT+DEVICE=ipalKmHZ5kDtLbhIjCGy,8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I

AT+PRODUCT=a1CU0VyQ33Y

AT+ACCESS=2

AT+TOPIC=/PRODUCT_KEY\$/DEVICE_NAME\$/user/update,/PRODUCT_KEY\$/DEVICE_NAME\$/user/get

```
AT+NETP=A,1,ALI-IOT
+ok

AT+DEVICE=ipalKmHZ5kDtLbhIjCGy,8tKR4i2M12mMPKkvsHW6X64oXW9oDZ1I
+ok

AT+PRODUCT=a1CU0VyQ33Y
+ok

AT+ACCESS=2
+ok

AT+TOPIC=/PRODUCT_KEY$/DEVICE_NAME$/user/update,/PRODUCT_KEY$/DEVICE_NAME$/user/get
+ok
```

3.2. 数据通讯测试

经过上述配置之后，在阿里 IoT 的平台可以看到设备已经在线。



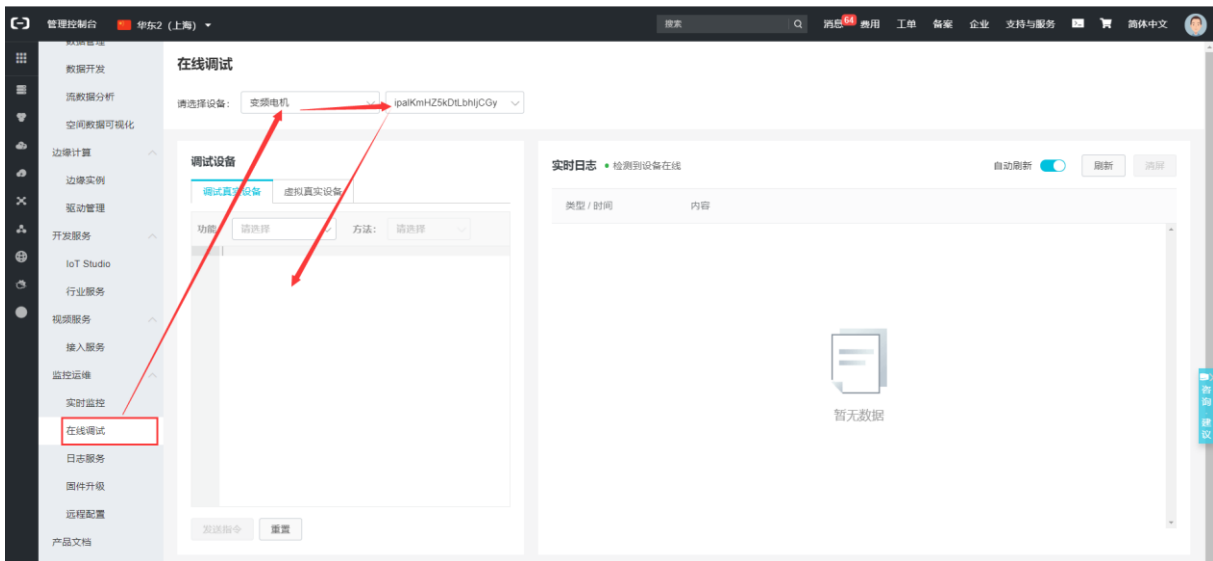
打开串口工具，模拟发送转速和电流两个参数到云端，转速 150，电流 10 安培，则串口工具中下发如下的十六进制数据 96 0A。



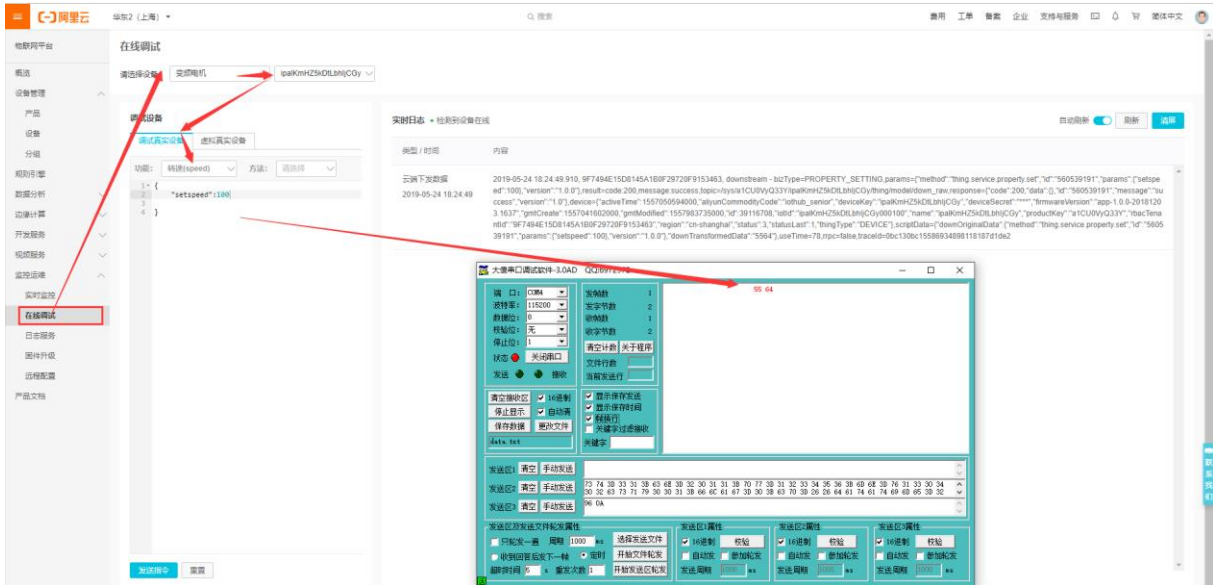
平台能实时查看到设备上报的状态数据。



接下来测试数据下行控制，选择刚才创建好的设备，平台能实时查看到设备上报的状态数据。

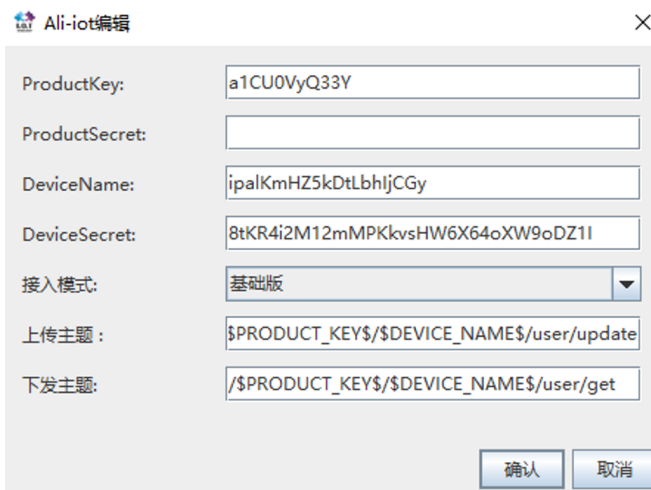


此处下发设置速度为 100，串口就输出 55 64 的数据。至此数据的上下行就都测试跑通。



3.3. 基础版配置

配置设备端使用基础版，如下截图。



点击服务器对应主题，发布消息，设备端可看到串口输出如下对应数据。

