操作系统实验

**智慧农业**

姓名：xxx

班级：计科23xx

学号：xxxxxxxxxx

目录

[IoT与华为云 3](#_Toc200569897)

[打开浏览器搜索华为云 3](#_Toc200569898)

[创建实例 4](#_Toc200569899)

[配置产品 5](#_Toc200569900)

[配置属性 6](#_Toc200569901)

[配置命令 10](#_Toc200569902)

[进行注册 12](#_Toc200569903)

[硬件同步 15](#_Toc200569904)

[使用脚本生成密钥对 15](#_Toc200569905)

[与此同时，另一边.. 打开小熊派代码 16](#_Toc200569906)

[看看代码 19](#_Toc200569907)

[初始化 19](#_Toc200569908)

[任务线程 20](#_Toc200569909)

[处理函数 22](#_Toc200569910)

[烧录与实操 24](#_Toc200569911)

# IoT与华为云

## 打开浏览器搜索华为云



点击最上方搜索IoTDA



点击控制台



## 创建实例

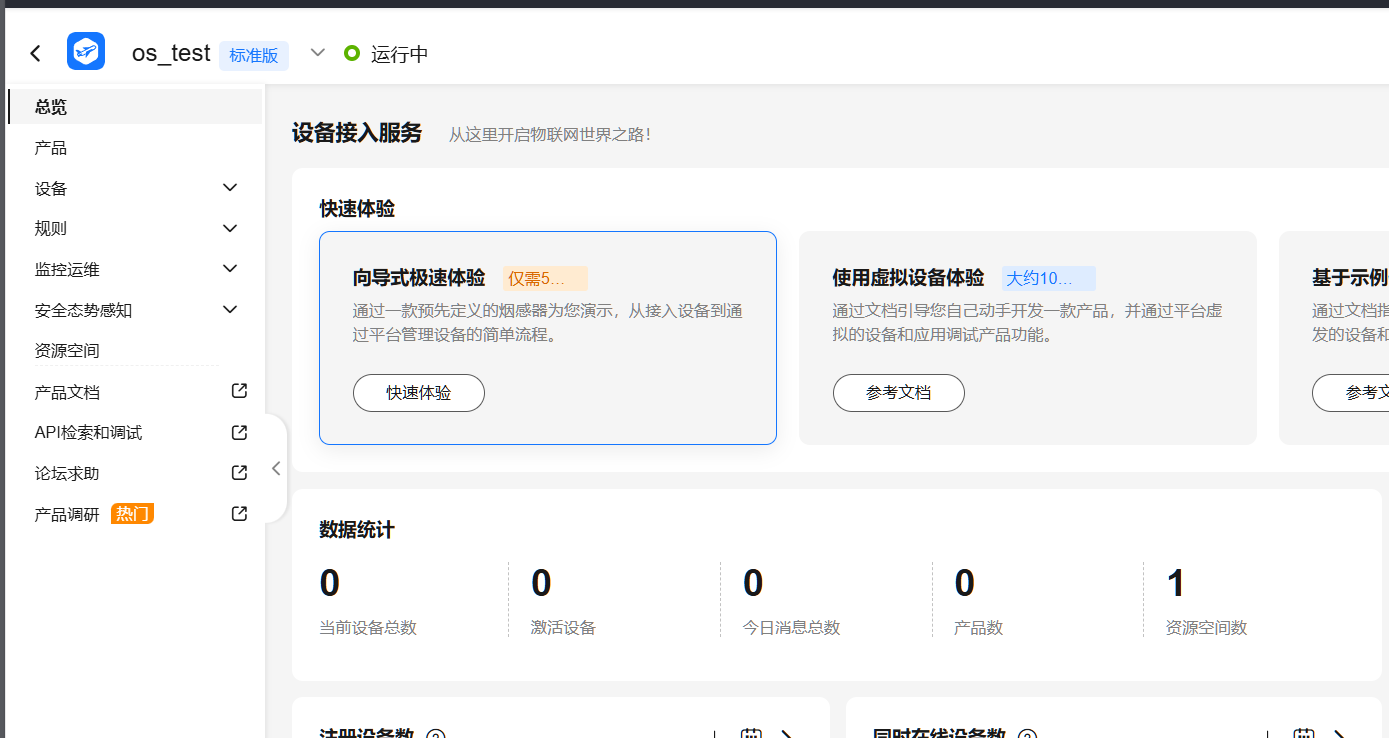
点击开通免费单元



点击右下角立即创建，创建完成后可以回到之前页面

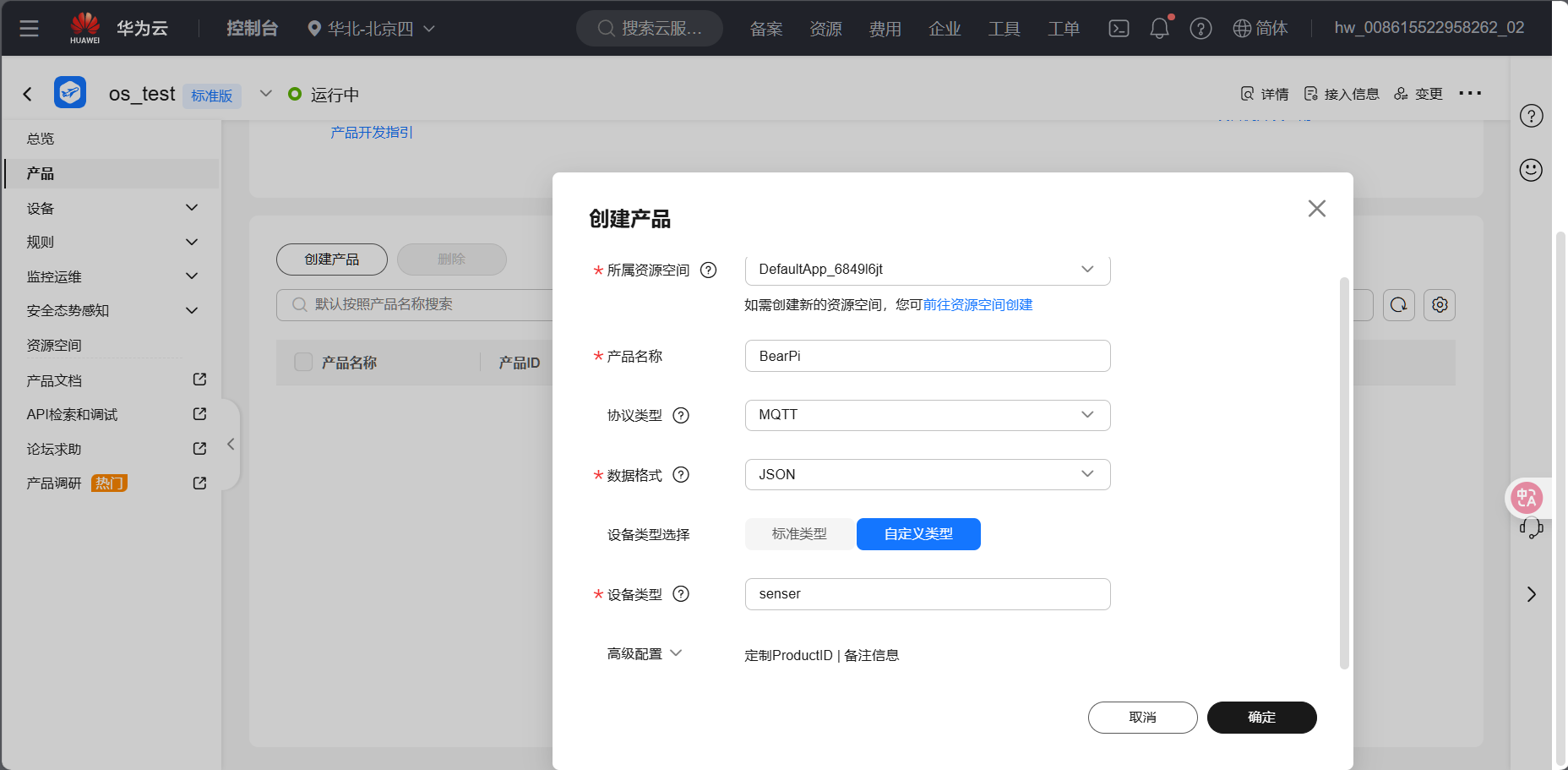


点这个新出现的实例，来进入产品控制台，再点击左侧产品



## 配置产品

进入产品页面之后，点击创建产品



资源空间无需手动调控，后面的MQTT和JSON不要填错就行



成功后下面将会多出来一个产品，点击详情

进入页面后向下翻，点击自定义模型



创建时候，id注意填写 Agriculture 方便和后面的代码配对

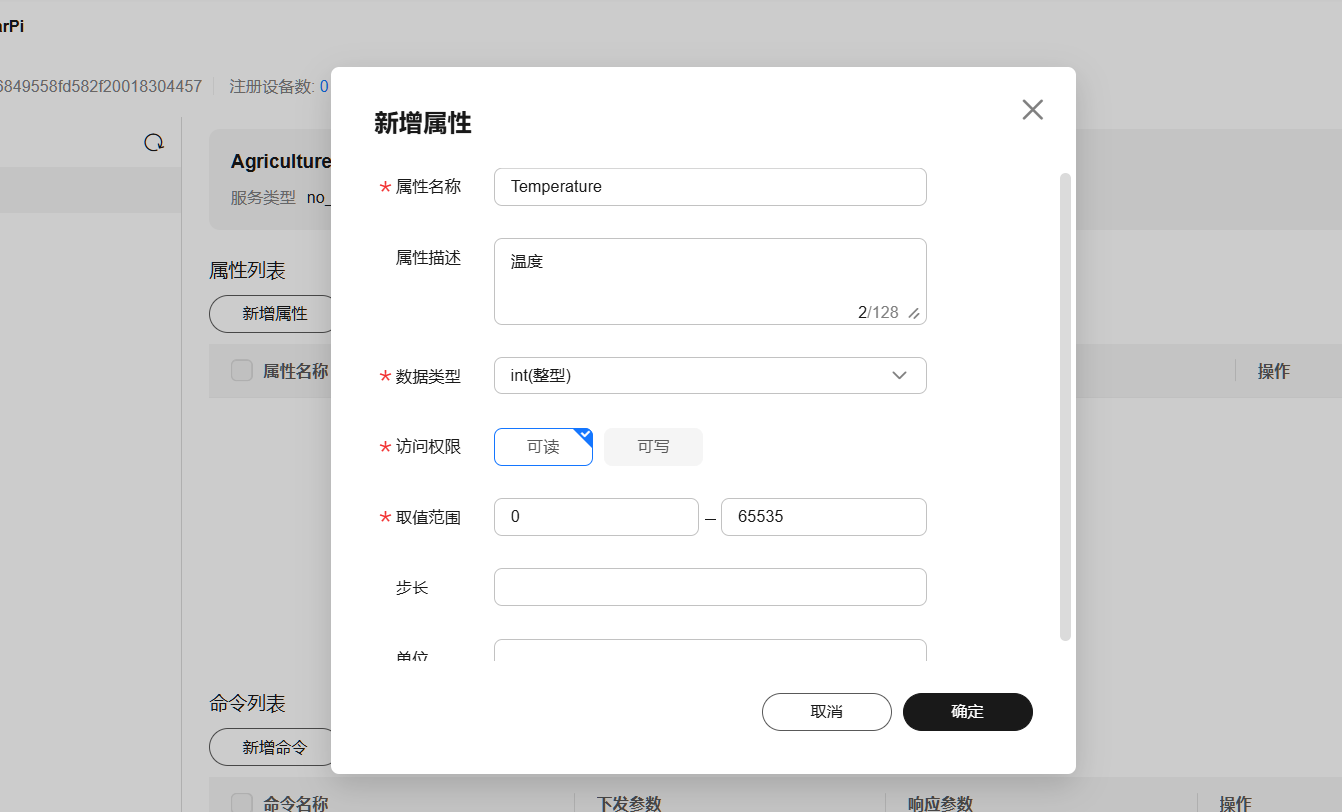


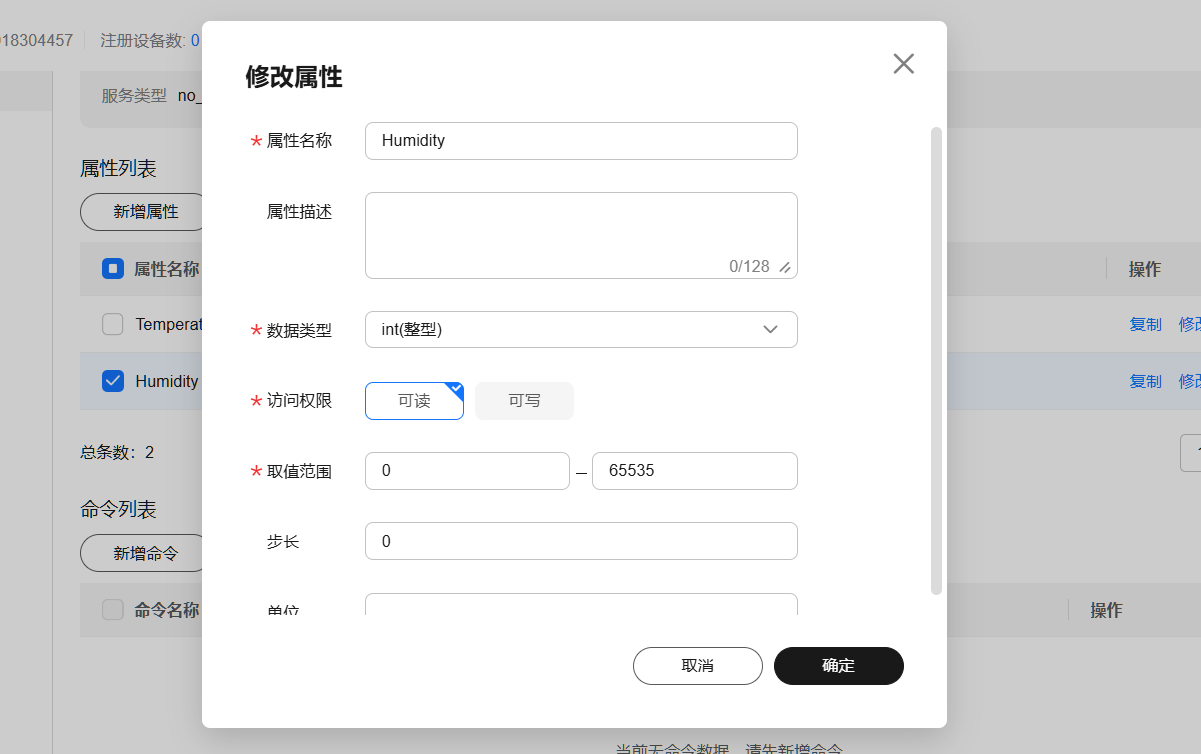
## 配置属性

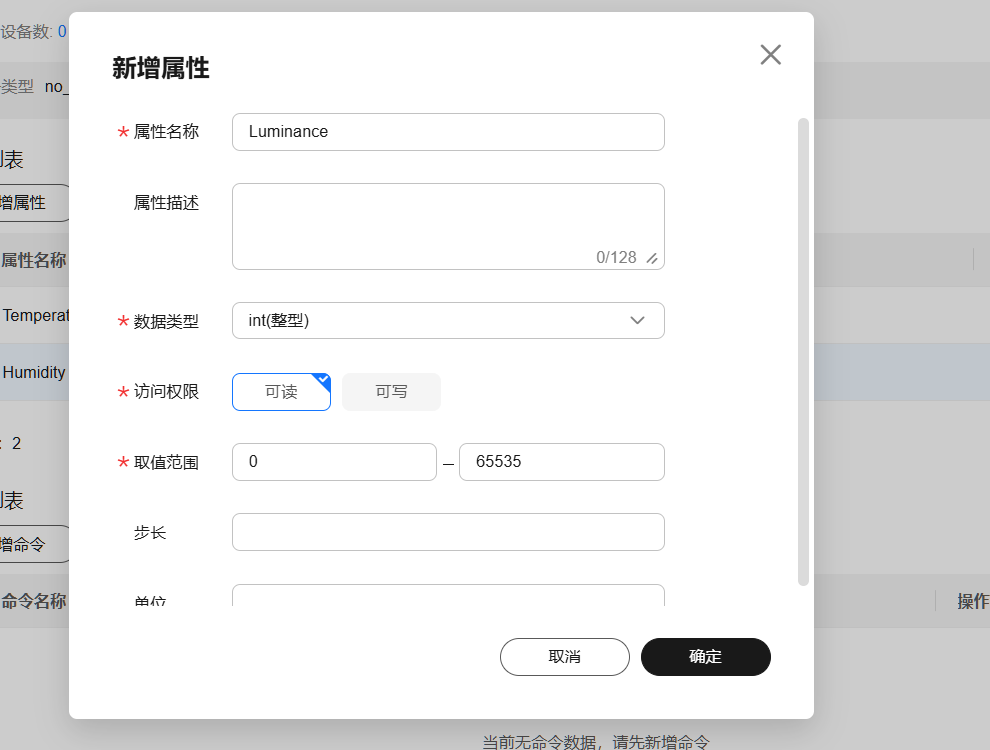
点击新增属性



咱们属性有五个“Temperature”， “Humidity”，“Luminance”，“LightStatus”，“MotorStatus”







**接下来是两个字符串型**

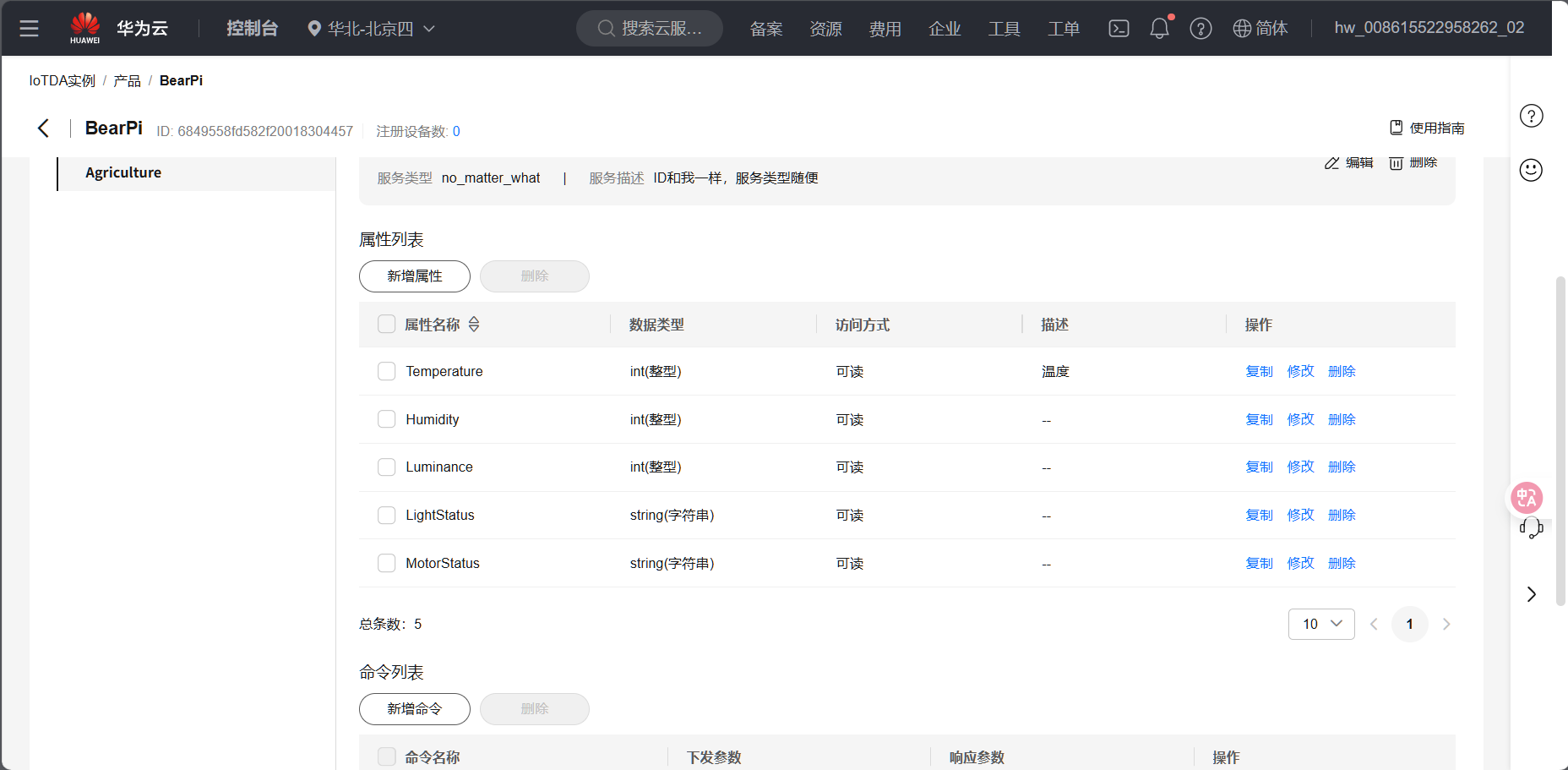




成功后如下图所示



## 配置命令



1. Agriculture\_Control\_light

命令名称 Agriculture\_Control\_light



参数名称：Light

类型： string

长度： 3

枚举值：ON,OFF

//华为云默认枚举值之间选用逗号分隔



1. Agriculture\_Control\_Motor

命令名称： Agriculture\_Control\_Motor

参数名称： Motor

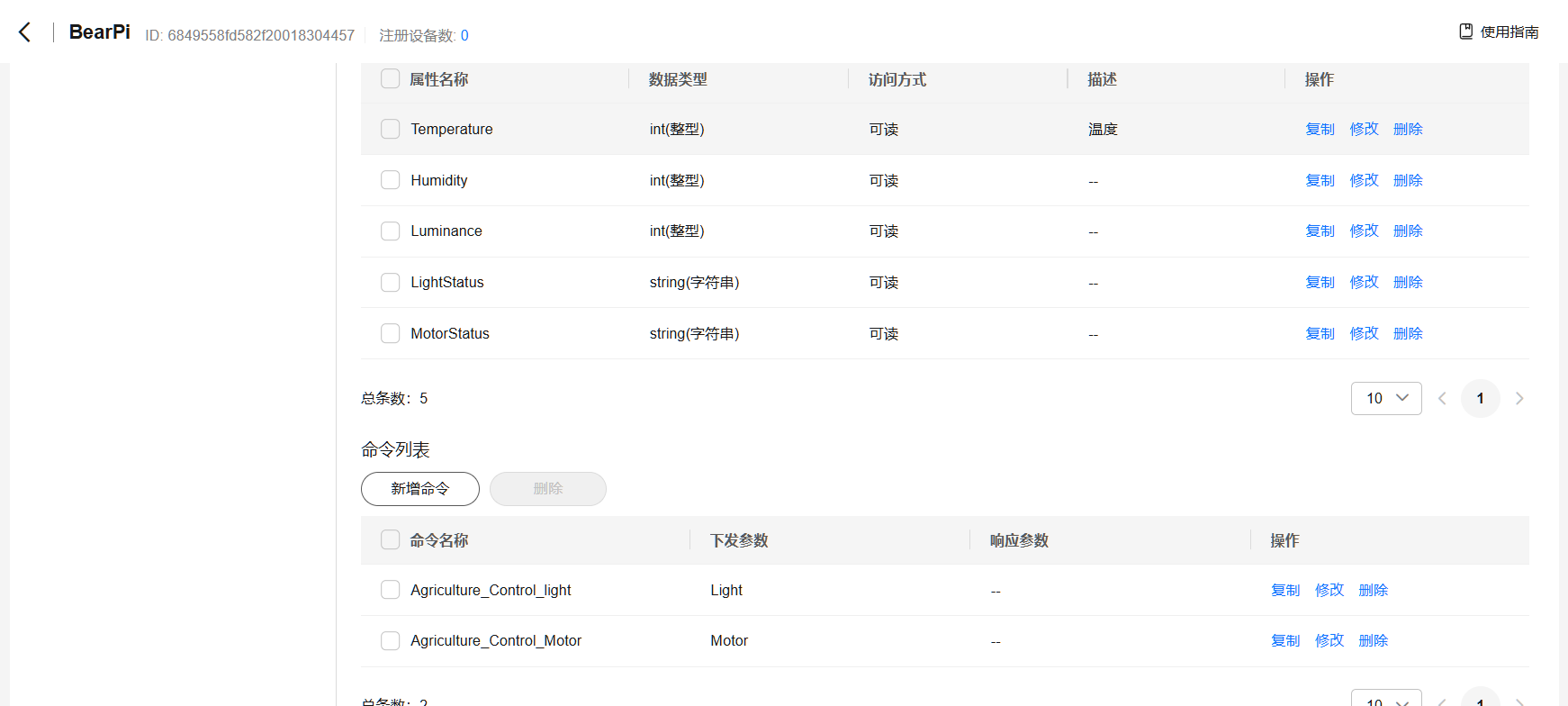
数据名称： string

枚举值： ON,OFF

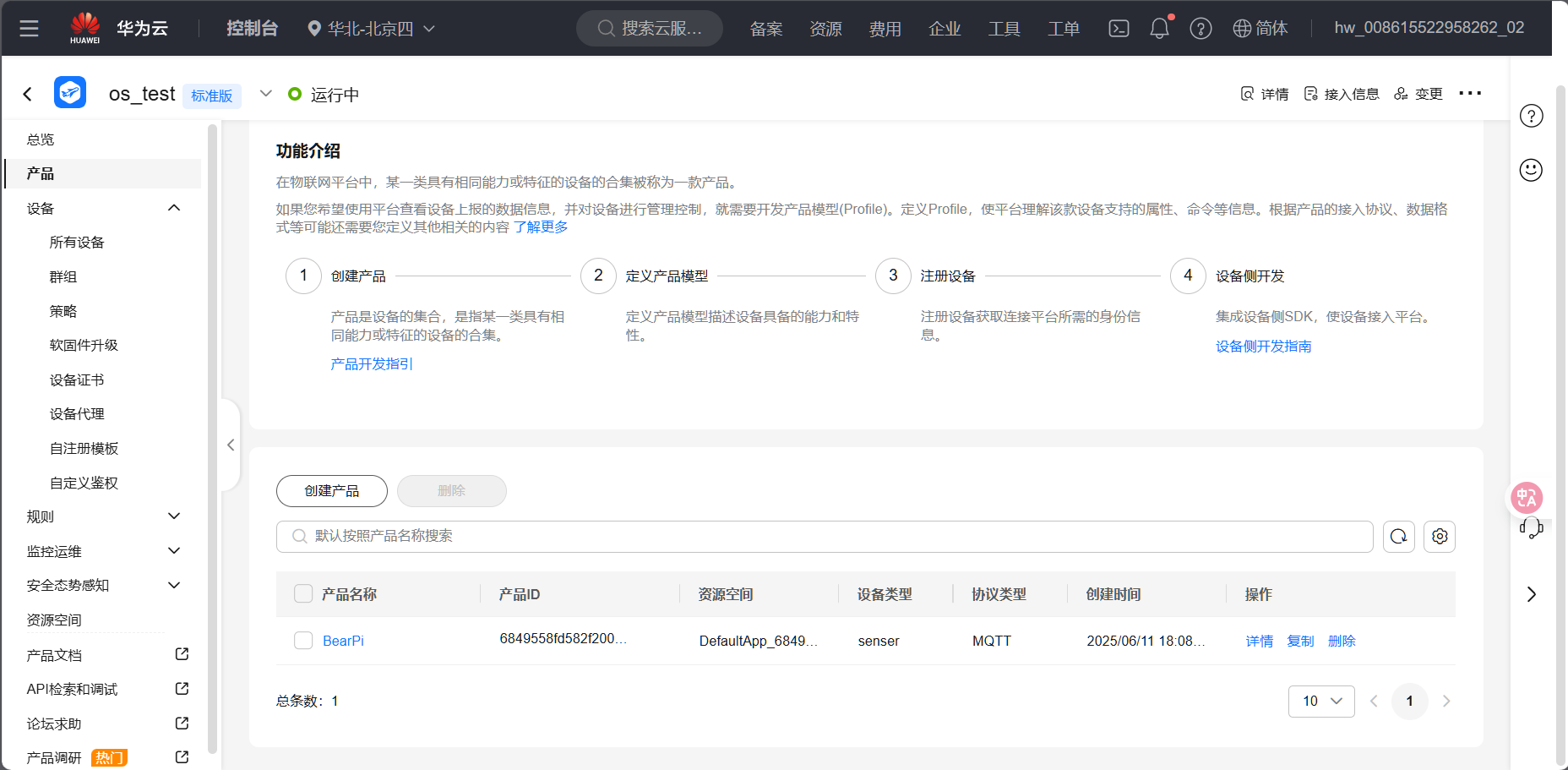


## 进行注册

上述配置全部完成总览：



返回上一个页面，点击设备下的全部设备，再点击注册设备

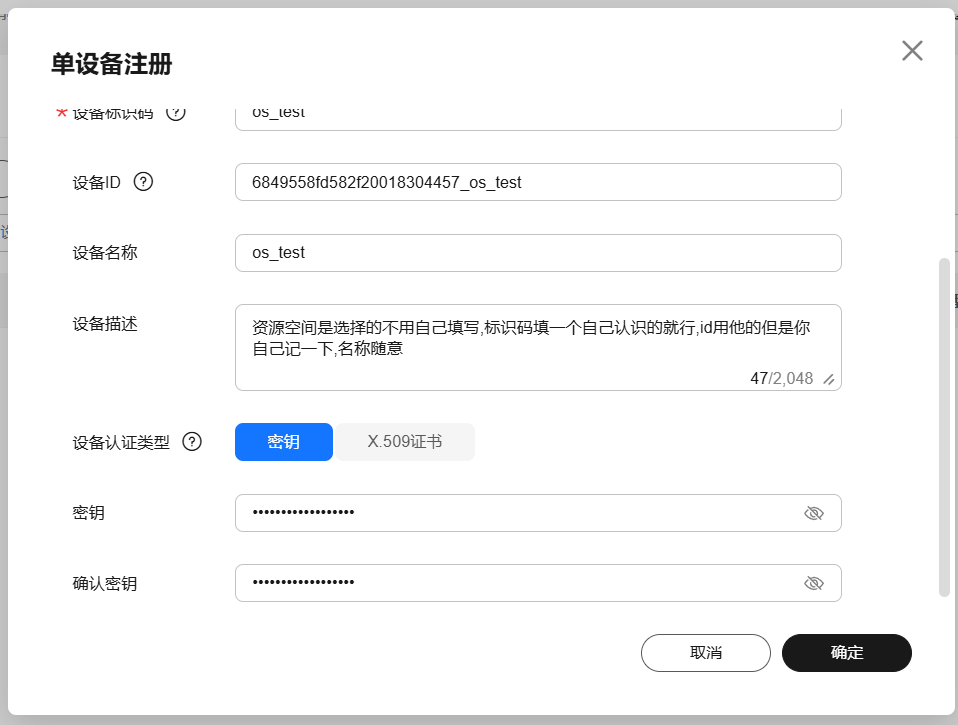




资源空间是选择的不用自己填写,标识码填一个自己认识的就行,**id用他的但是你自己记一下**,名称随意，然后继续向下翻



使用密钥，可以用脸滚出一段大写字母+小写字母+数字的密钥，并且记下来！！！



点击确定，出现成功页面，他还会送你一个文件里面有账户密码

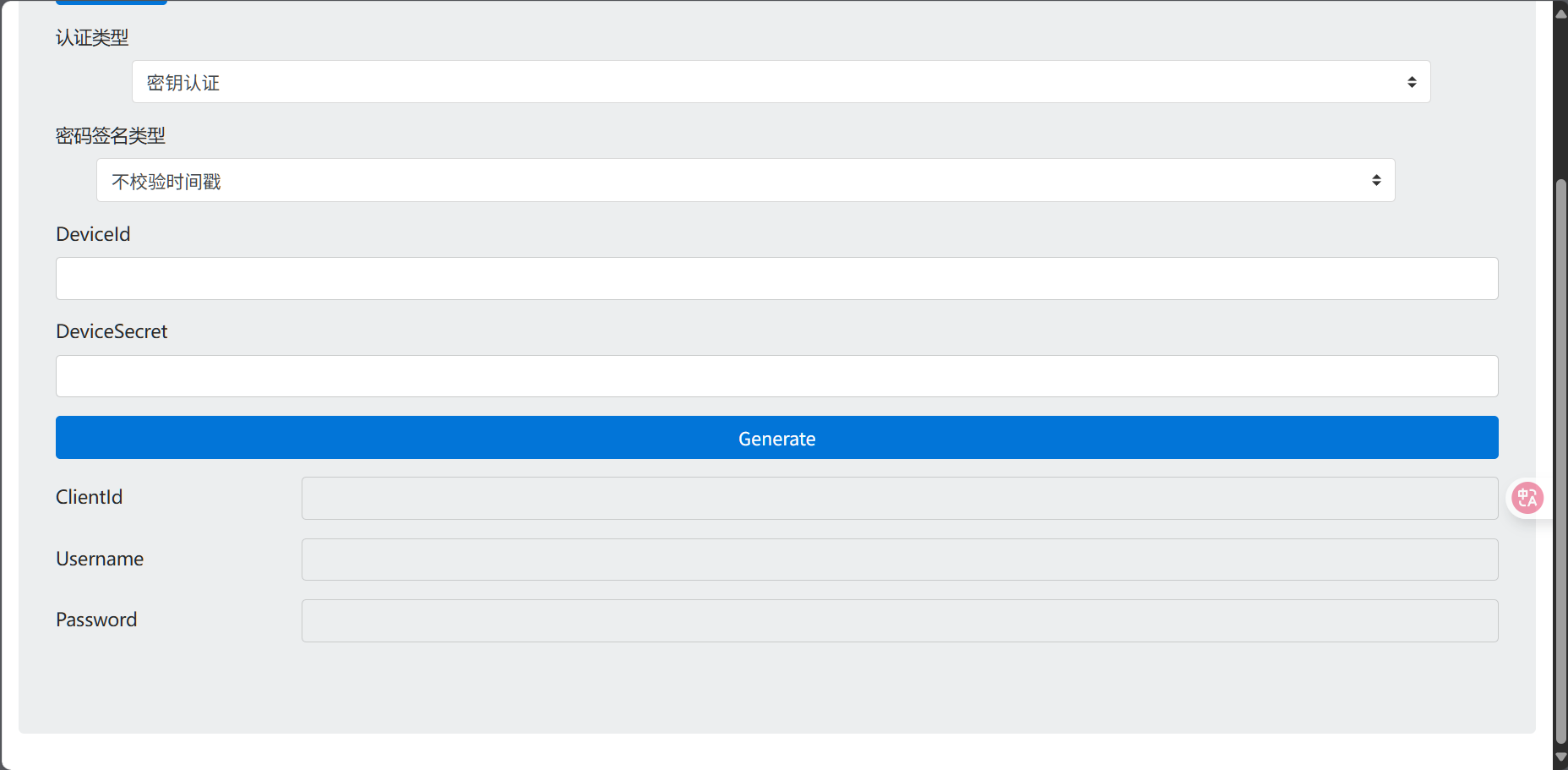


恭喜你，这部分就完成了，接下来讲解如何配置硬件

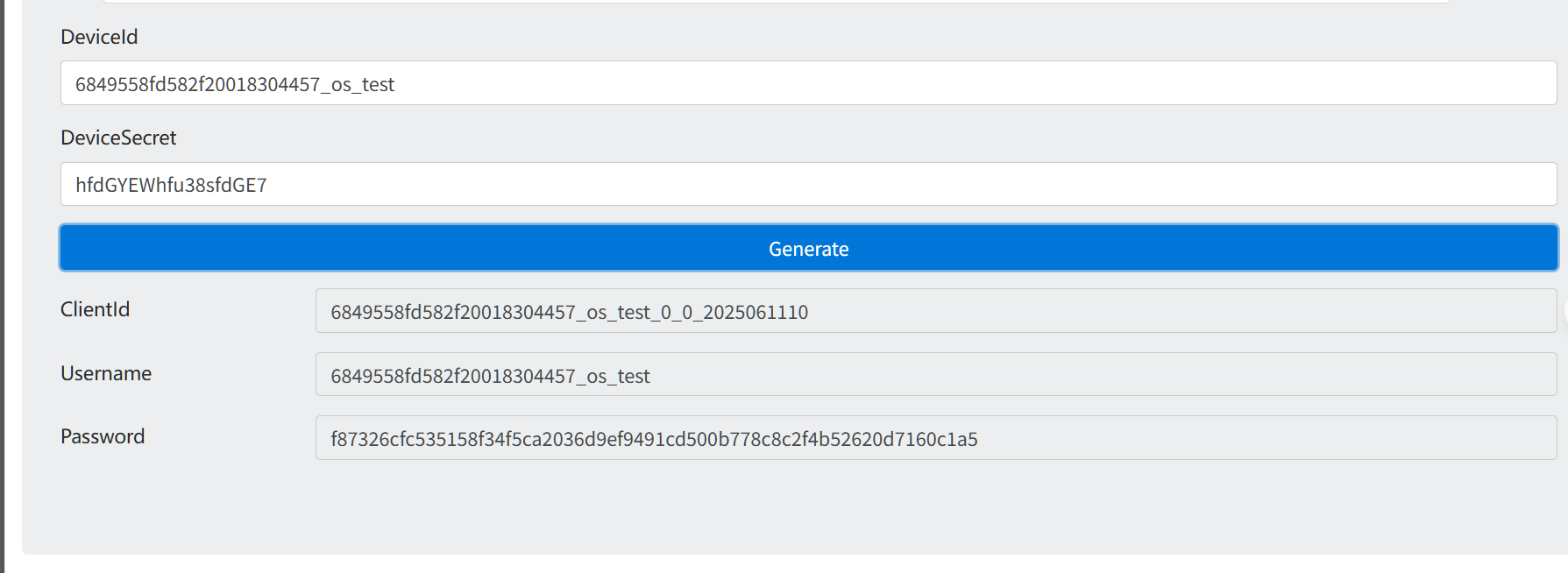
# 硬件同步

## 使用脚本生成密钥对

访问[Huaweicloud IoTDA Mqtt ClientId Generator](https://iot-tool.obs-website.cn-north-4.myhuaweicloud.com/)



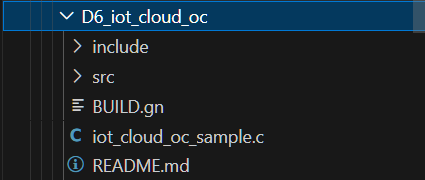
将刚才的账户密码填入两个空格，然后记一下生成的这三个东西





其实我建议就别退出这个页面了，万一记错了或者少个字母可能怪费事的

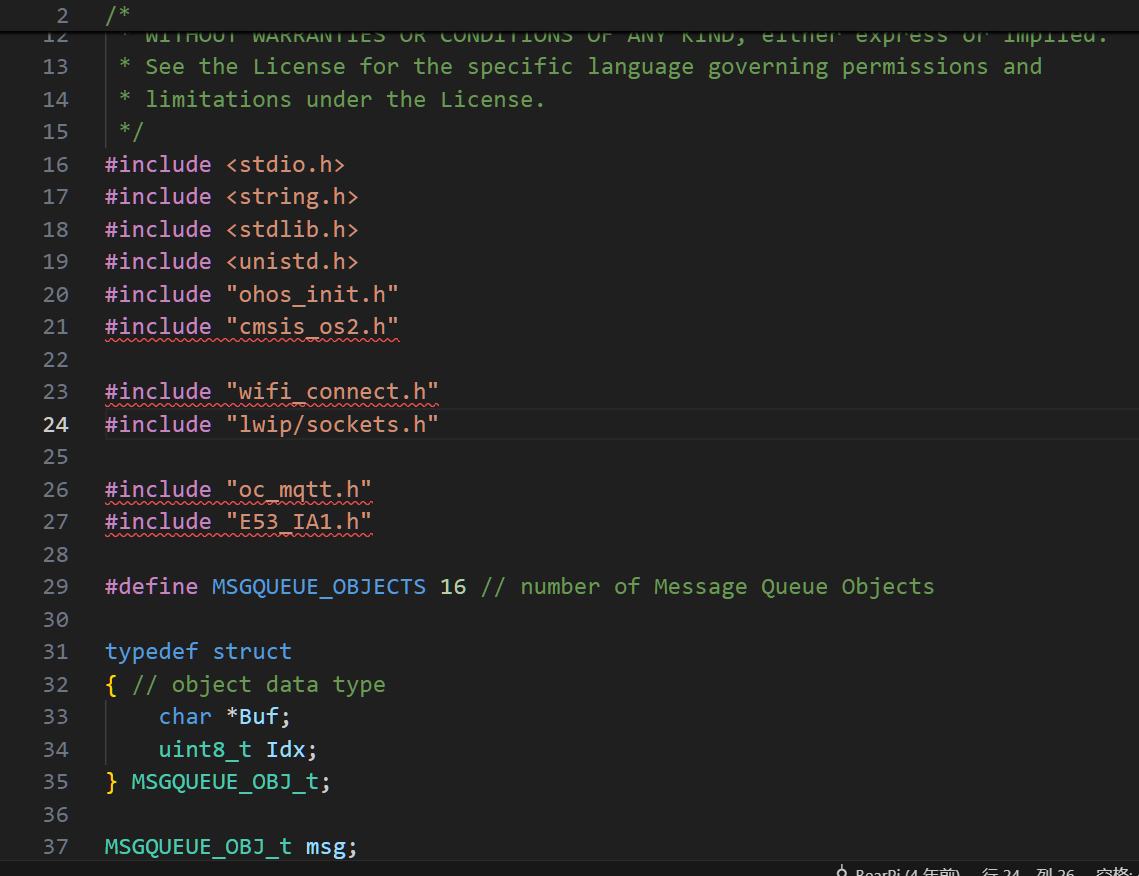
## 与此同时，另一边.. 打开小熊派代码



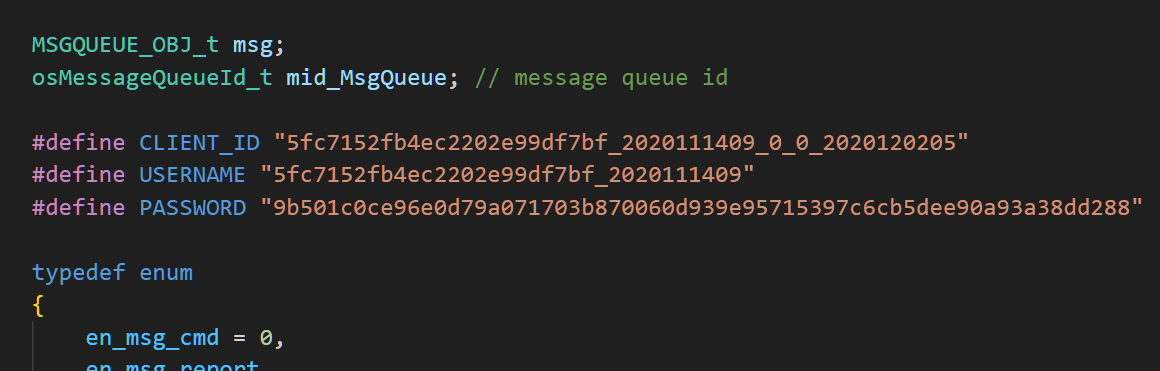
找到这个D6，我把相对路径放下面了

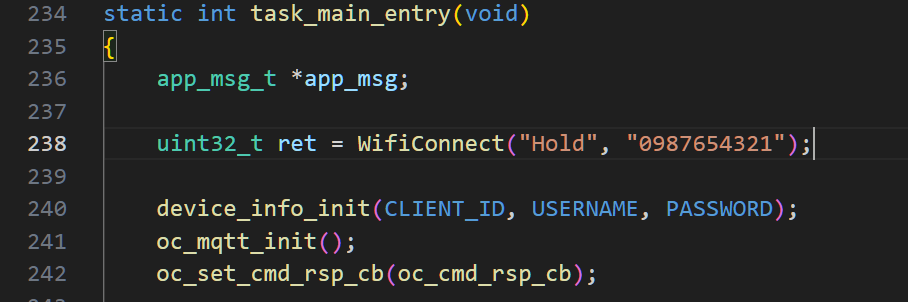
/bearpi-hm\_nano/applications/BearPi/BearPi-HM\_Nano/sample/D6\_iot\_cloud\_oc

长这样，可能会有一堆链接错误，咱们不用管，编译时候会自己处理



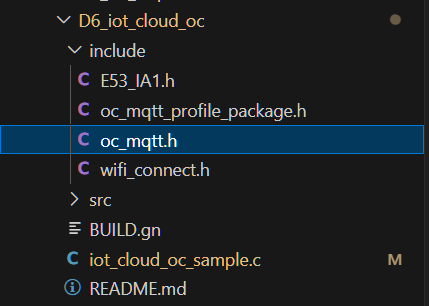
往下翻，找到这里有三个意义不明的宏，简直就像是量身定制的一样，把刚才输出的三个字符串copy过来，copy你自己的，这样才能调试

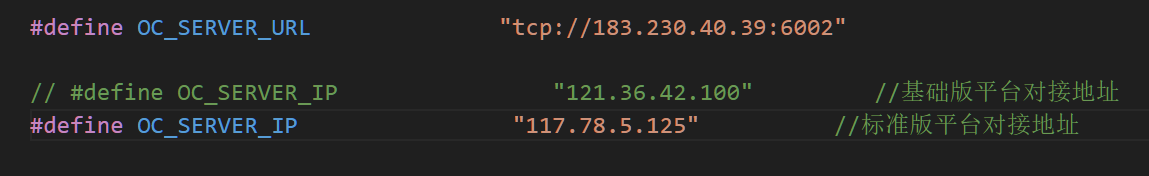


然后别走向下翻，238行你看这里啊

网络连接，改成你们寝室的wifi，或者你手机热点的账号密码。别用校园网，校园网上网要通过web进行上网许可登记，我觉得这个板子好像不太行。

然后我们去找另一个文件，在D6的include文件夹下



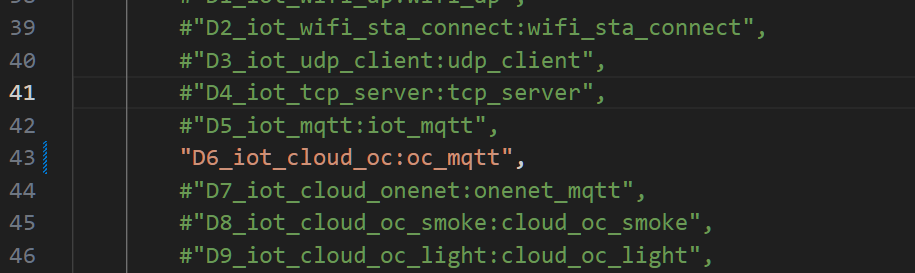
翻到48行

这里有两个选项啊,意思是你是什么版本你就用什么，咱们这边确认一下



可以看到是标准版，那就不改了

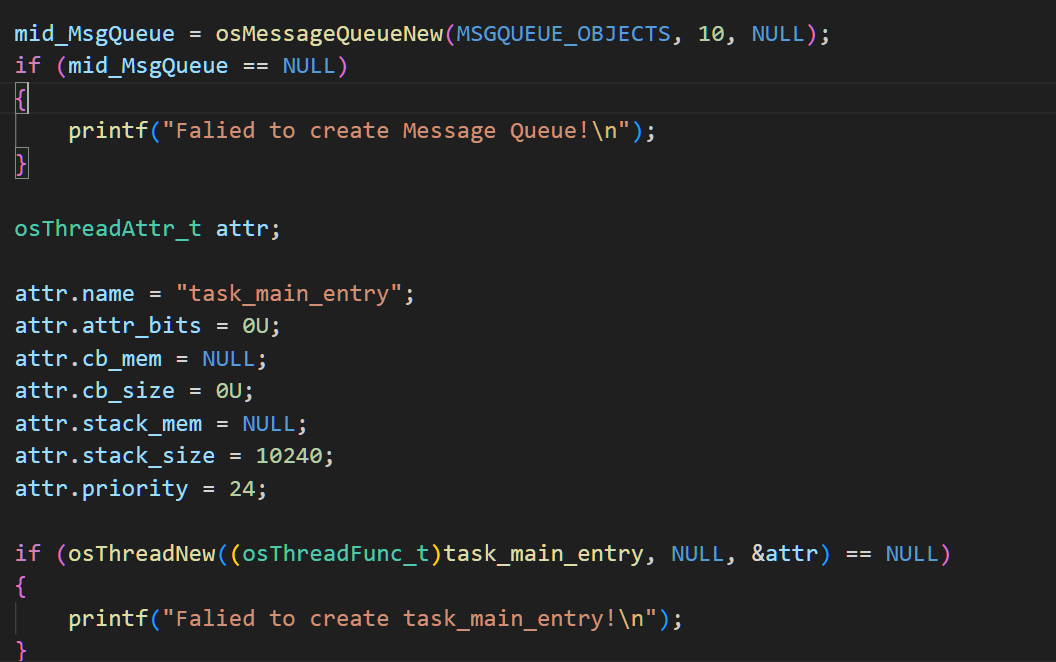
打开总编译文件，将D6添加



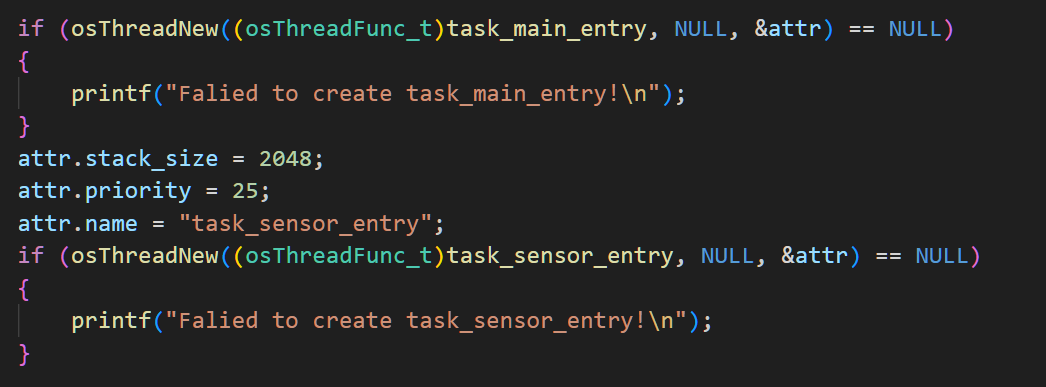
恭喜你到这一步，小熊派同步就结束了，接下来看看代码和现象

# 看看代码

## 初始化



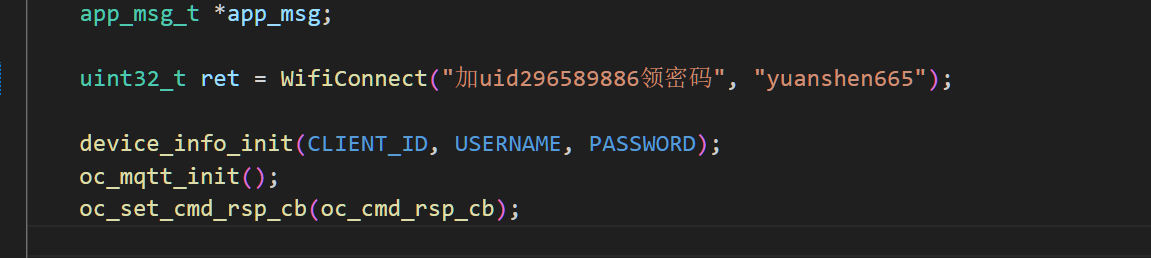
初始化了一个大小为10字节长度为16的消息队列和然后创建了一个任务线程



更改了一下栈内存大小，又创建了另一个任务线程

## 任务线程

task\_main\_entry：*主任务线程*

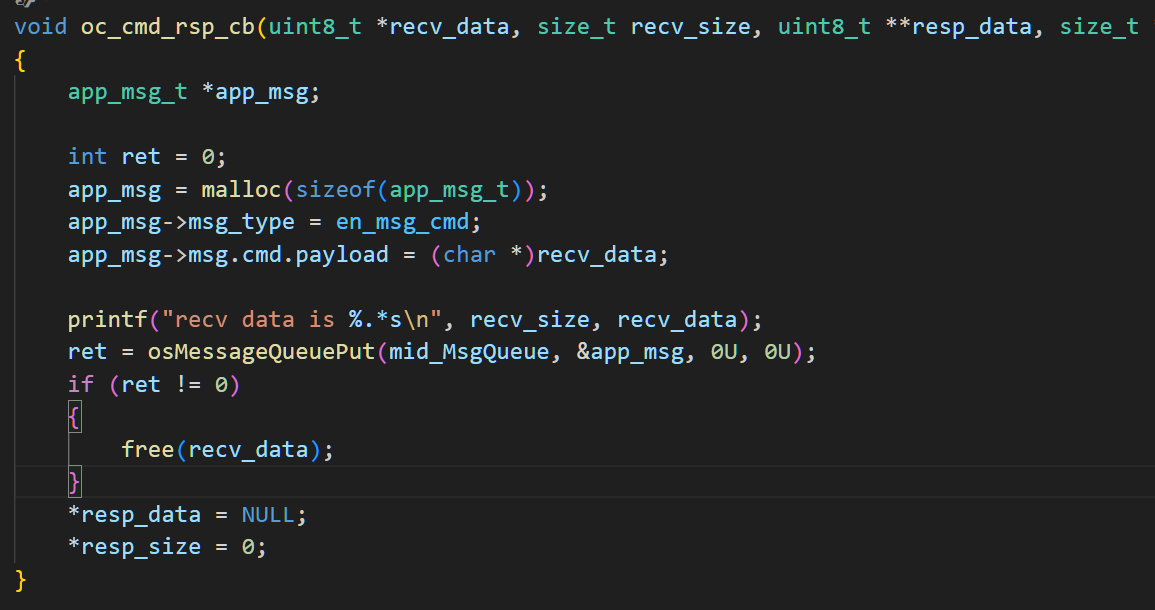




一开始先连接了一个wifi，然后初始化了一下设备号初始化mqtt模块

* 如果之前你有研究过互联网网络相关，比如http那应该对物联网界的mqtt不会太陌生， MQTT是一种轻量级的物联网通信协议，基于发布/订阅模式，支持QoS级别，适用于低带宽、高延迟的网络环境。说白了就是一种基于消息队列的发布订阅协议，一边发送另一边接收。

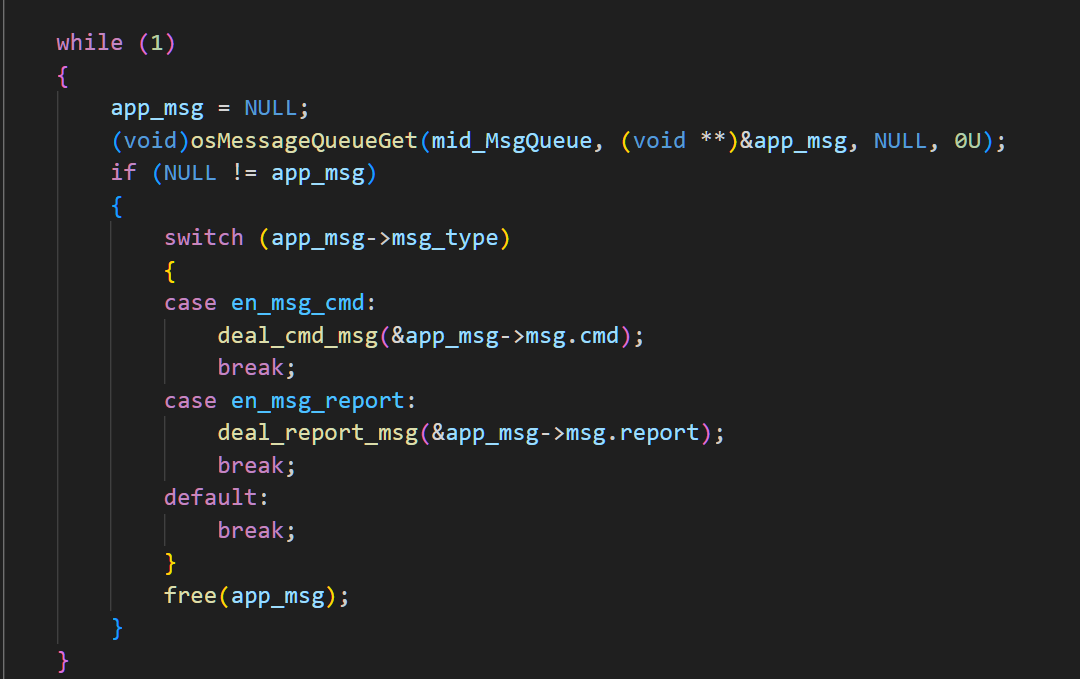
最后一句那个set是一个注册绑定指令，里面传入的方法内容如下



这是一个封装函数，将收到的原始数据（JSON 字符串）封装成 en\_msg\_cmd，投入消息队列，后续在主处理任务里统一解析并执行，也就是说后面，我们通过mqtt接收到的信息都会这么处理。

意思是这样的：我们通过mqtt接收的json内容经过解析之后会通过这个函数被封装成一个message块放进信息队列，然后信息队列去触发/通知，这个task\_main执行任务

说回刚刚，初始化之后这个任务的具体执行内容如下：



只要消息队列来消息了就只行下面的switch语句，消息队列中来的信息来自另一个传感器任务和我们刚才讲的mqtt绑定。

之后根据消息类型选择执行哪种指令

这两个指令我下面也会讲。



task\_sensor\_entry： *传感器任务线程*



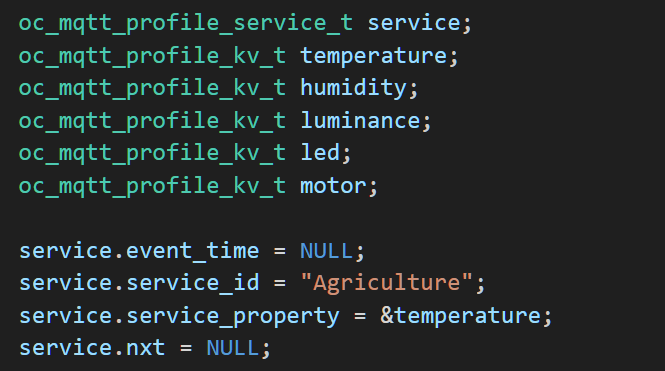
最后有个sleep说明周期三秒

然后从传感器中提取数据指定方法为上传到服务器，然后丢进消息队列，让上面的那个taskmain通过deal\_report方法进行mqtt上传。

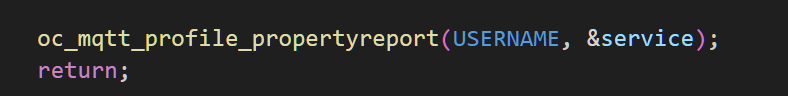
## 处理函数

处理函数有两个

1. deal\_report\_msg *消息上传*



如图声明变量，然后给每个变量赋值，说明自己所属id和变量名称，id就是之前选的那个Agricu那个，所以我说不要改，如果你改了这里也要改



赋值完成后调用api发送即可，发送之后就是服务端响应mqtt的事情了，咱们不用关心

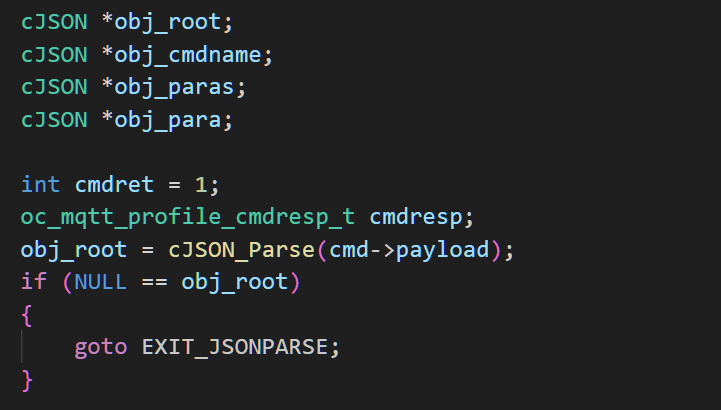
1. deal\_cmd\_msg *硬件控制*

这是一个实现硬件调度的处理函数，其中有两个关键功能，就是响应咱们之前设置的两个方法，一个控制光照一个控制电机

支持两种命令：

* 1. Agriculture\_Control\_light —— 控制板载 LED
  2. Agriculture\_Control\_Motor —— 控制板载电机

执行完后，把执行结果（ret\_code）通过 oc\_mqtt\_profile\_cmdresp 上报给云端



开局是cJson解析，然后进行指令配对

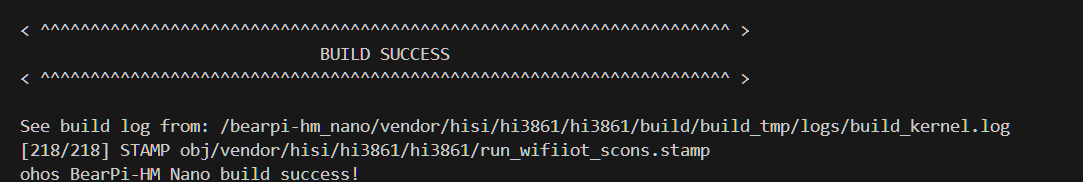


灯光为例，确认指令后会解析这几个指令中的参数，通过cJson提供的方法，然后设置小灯状态，是的没了。另一边也长得一模一样。

恭喜你，现在代码解析也完成了。接下来我们拿板子实操一下。

# 烧录与实操

先编译

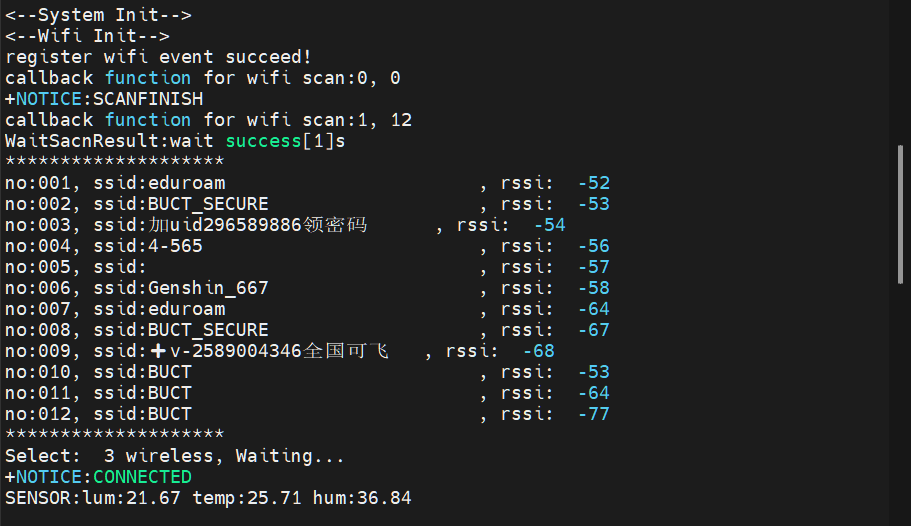


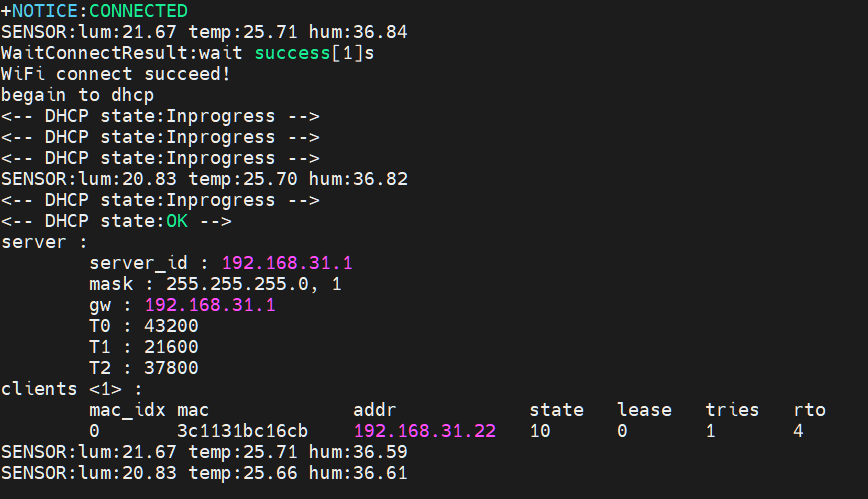
非常成功的编译完成了

准备烧录



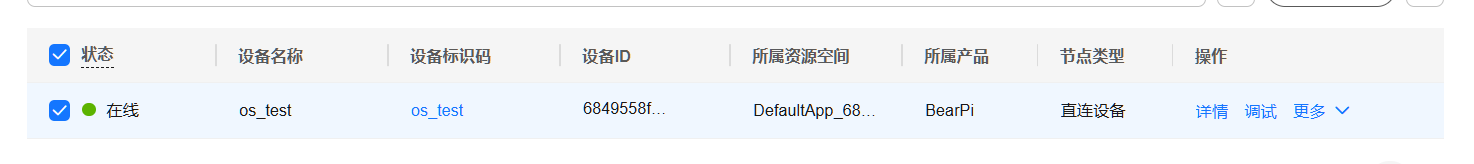
等待烧录完成后，连接终端，可以看到wifi驱动导入



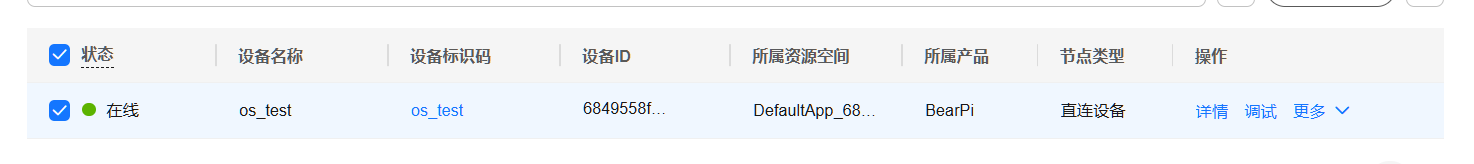


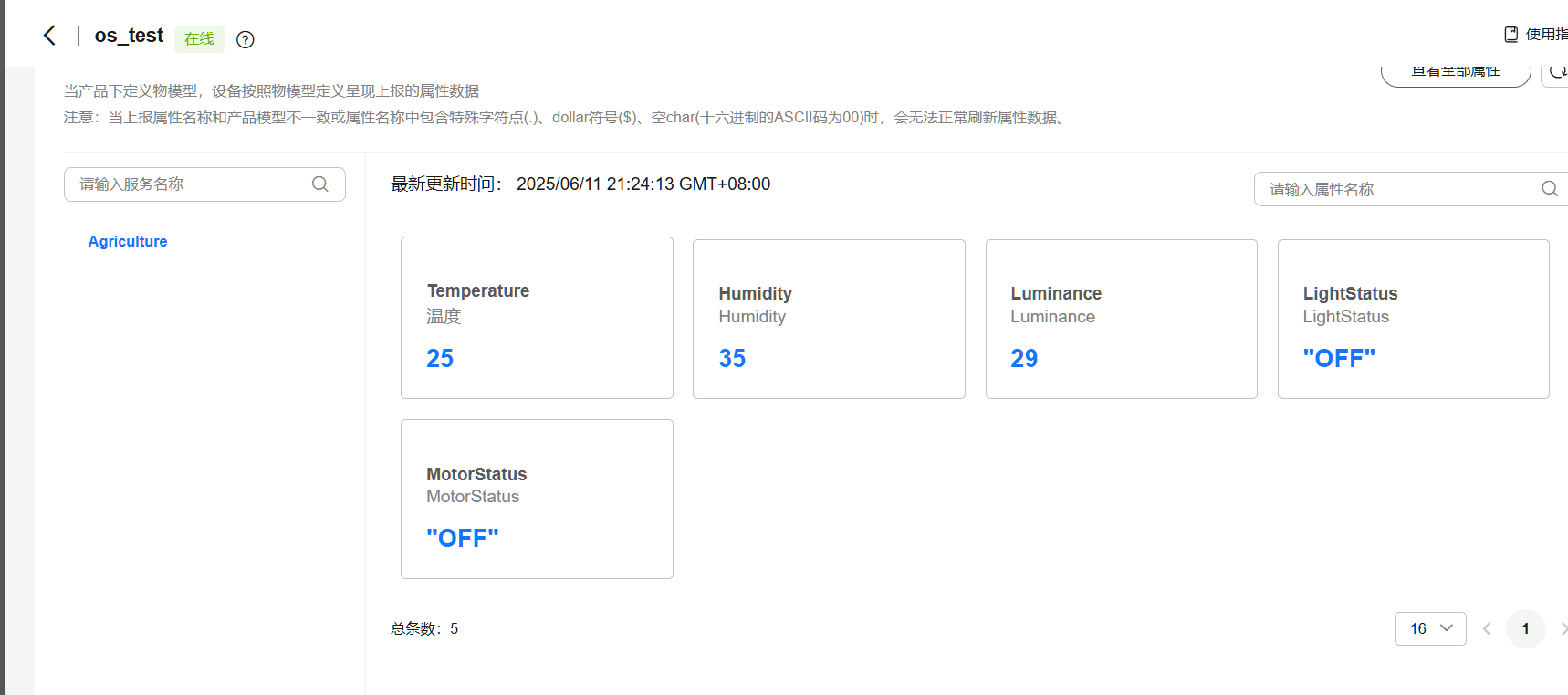
连接过程的报点

然后我们去平台确认连接是否成功



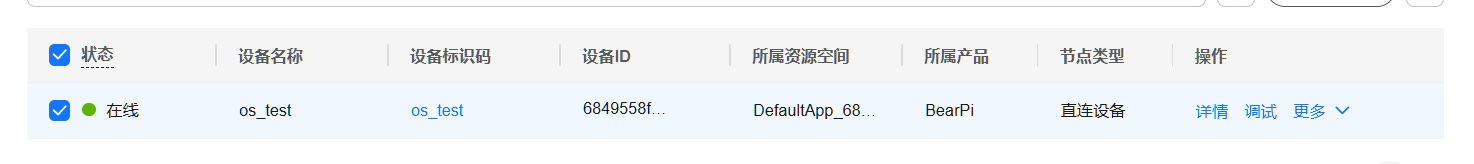
太棒了变成在线了，非常成功。我们再查看一下数据上报，点击详情



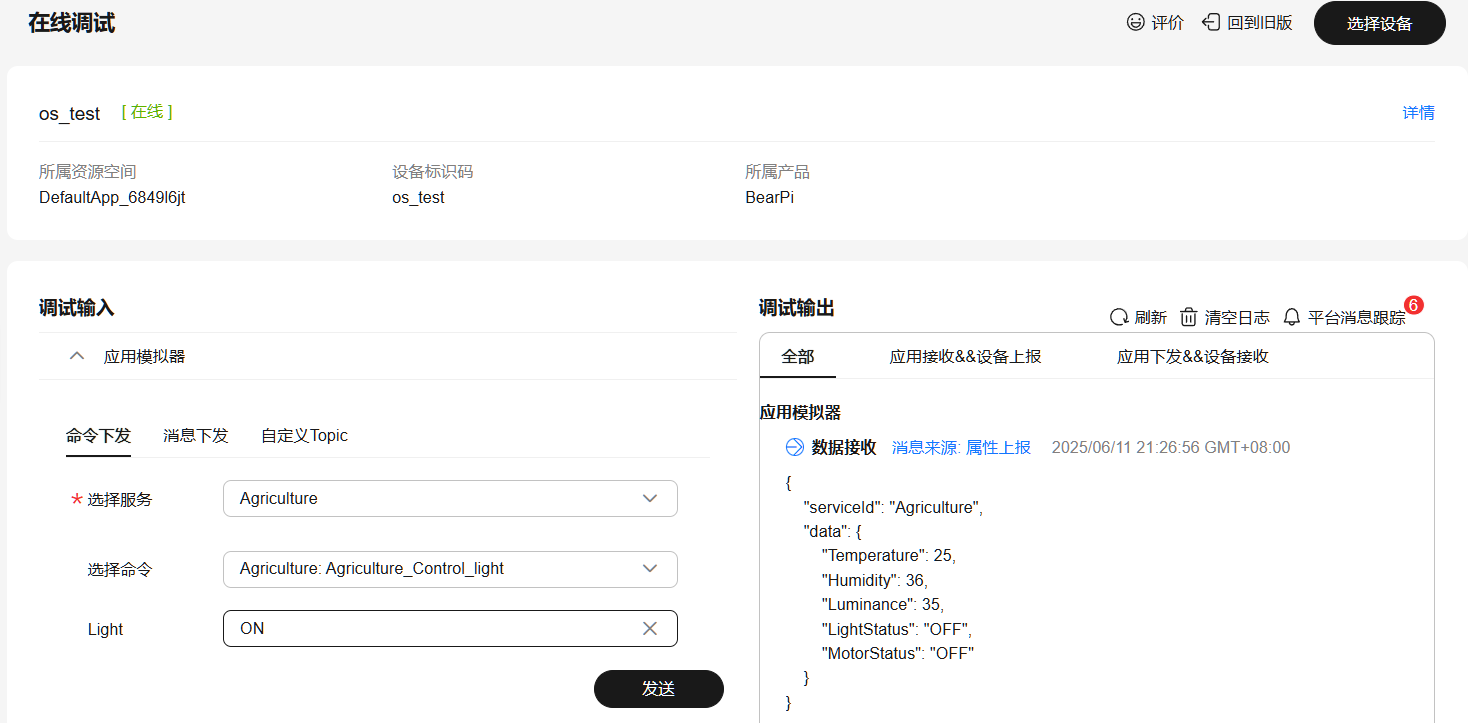


我们能获取到详细的信息

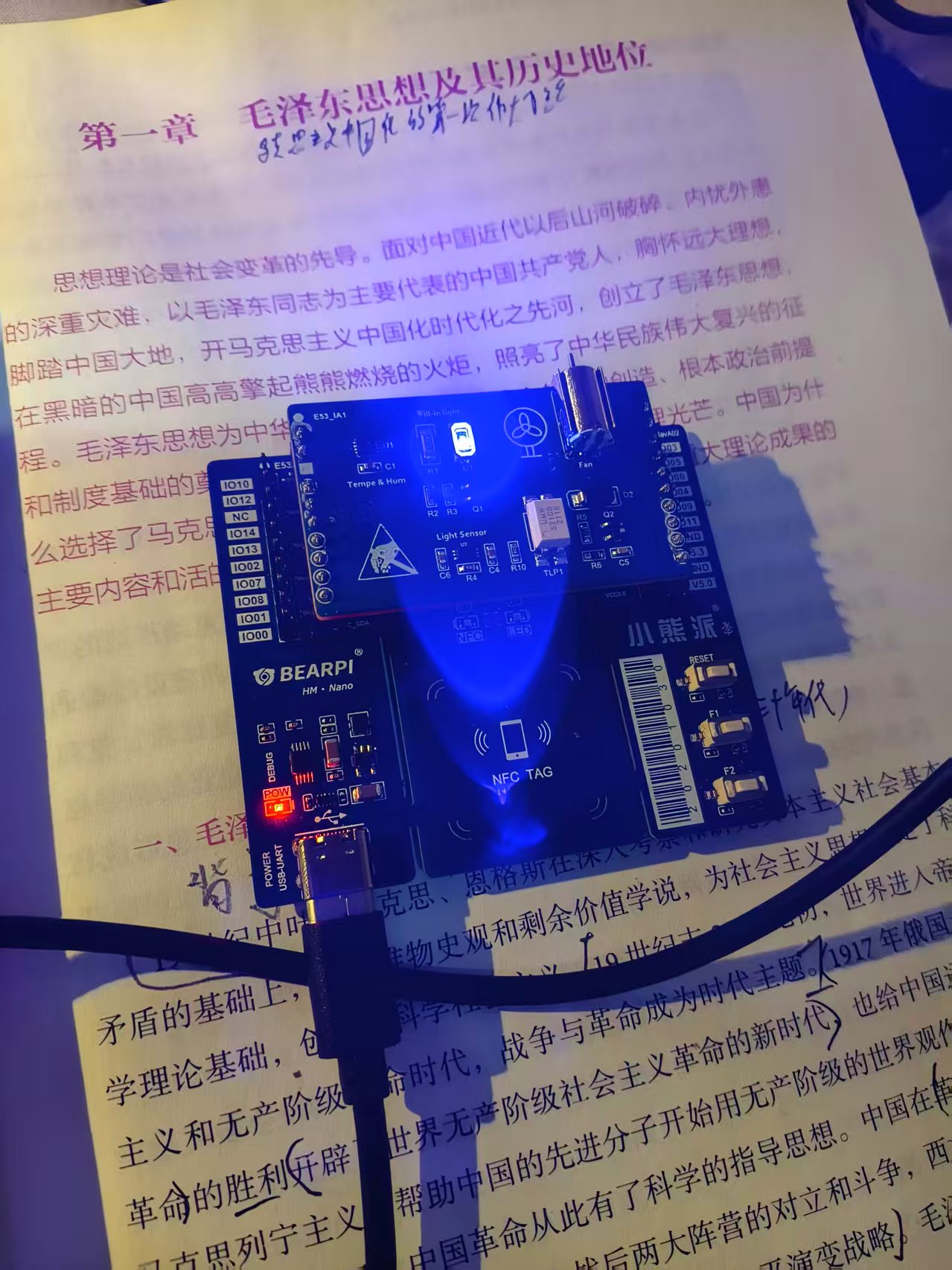
我们再尝试调度功能函数，点击调试



配置命令和值选项



我们选择灯光功能将灯光打开，点击发送，板子亮起奇异的光芒



再点击关闭灯光就会自然关闭

至此所有功能测试完成

恭喜你，你的板子成功实现智慧农业的基本需求，远程监控和远程执行

# 个人感悟

感谢这趟实验课给了我宝贵的os开发板的开发经历和学习机会，在这个过程中我积累了宝贵的实操经验，还提升了自己的开发能力。最后这节课也接近了尾声，我非常不舍老师的悉心教导和组员的真诚帮助，这虽然是我在os走出的第一步，却也是我未来发展至关重要的基石，恭祝操作系统原理课程顺利结束。