# 一、NB-IoT硬件调试

## （一）烧录软件安装（keil5 MDK）

<https://mp.weixin.qq.com/s/pzQDgNxE5FfYVXx-SZB0Hg>

上述网址有下载地址及教程

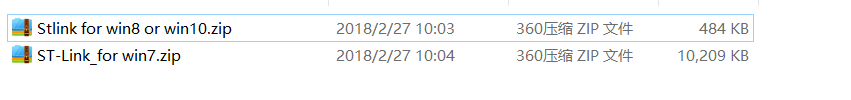
## （二）驱动安装（ST-LINK）

S-Tlink驱动安装

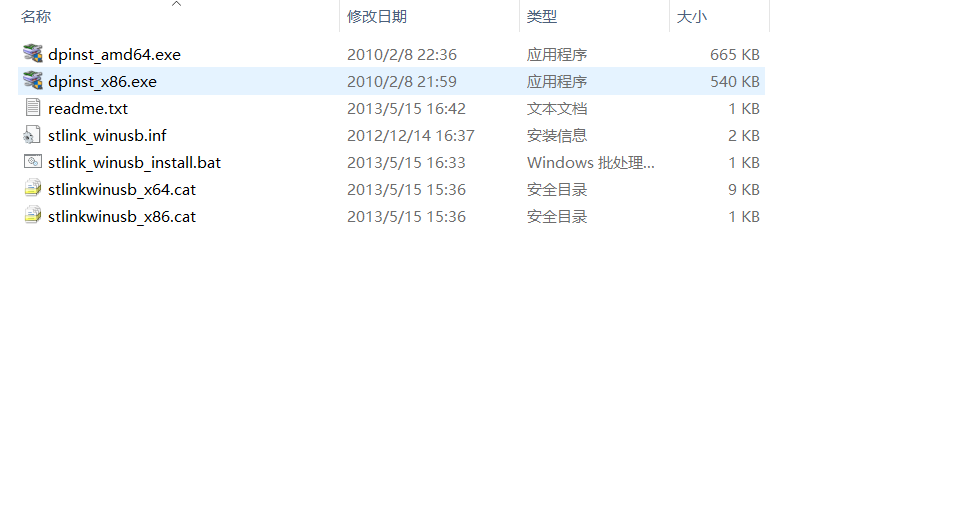
1.下载安装包（百度云），也自行下载

链接：https://pan.baidu.com/s/1506w9fIKxzIDbHvA2kQYaw

提取码：g7wz



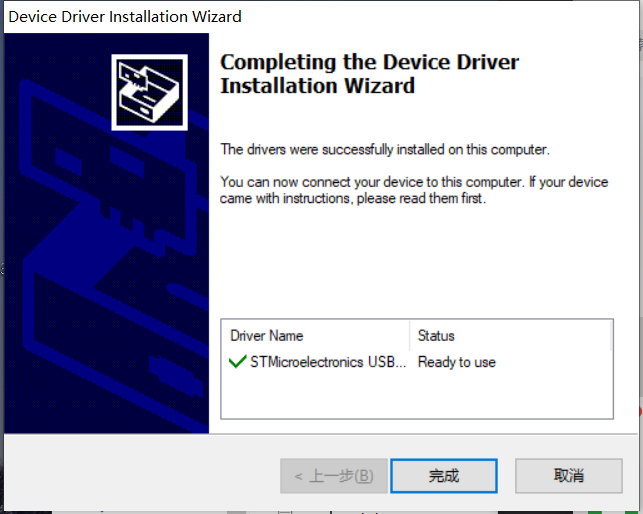
2.安装（管理员打开）



3.



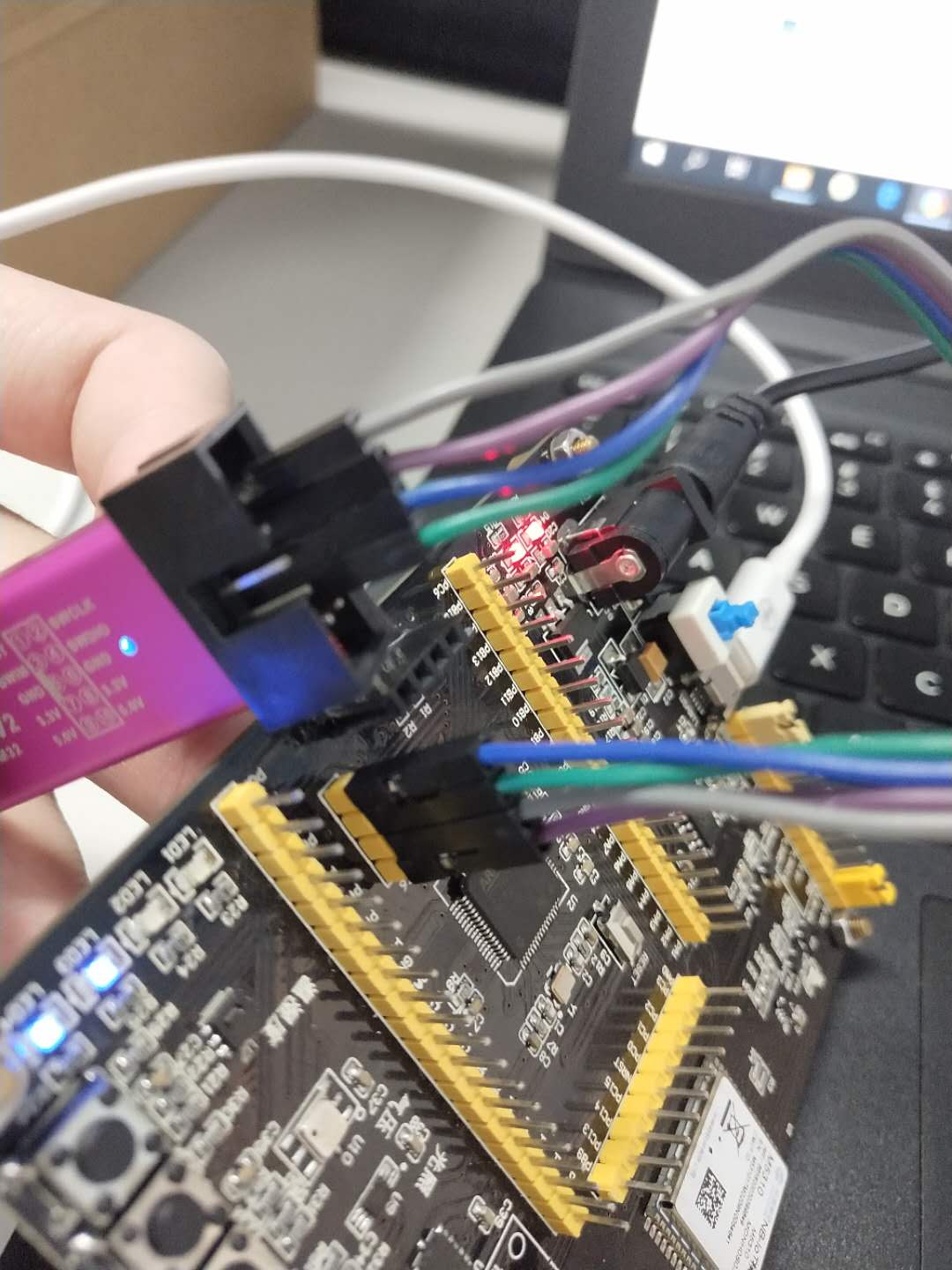
4.



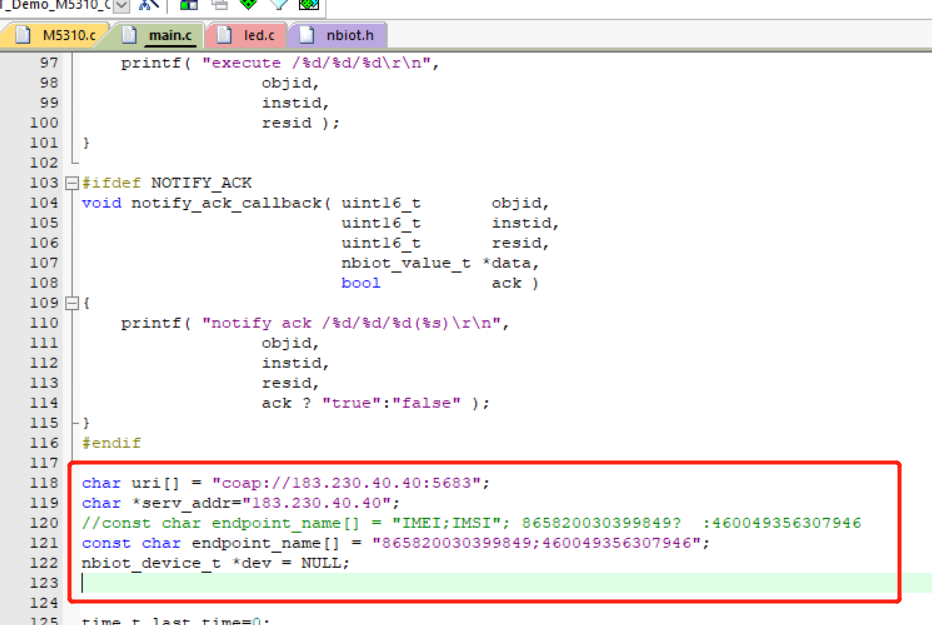
完成。

## （三）程序更改

1.连接



2.代码：**更改为自己开发板上的M5310的IMEI**



char uri[] = "coap://183.230.40.40:5683";

char \*serv\_addr="183.230.40.40";

//const char endpoint\_name[] = "IMEI;IMSI"; 865820030399849£ :460049356307946

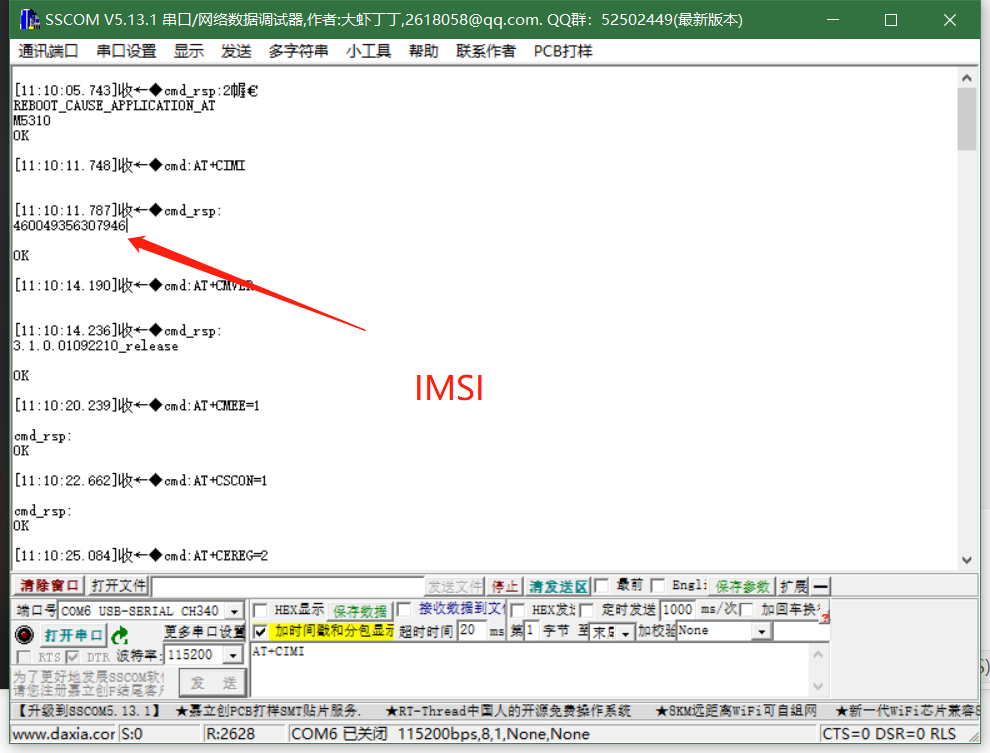
const char endpoint\_name[] = "865820030399849;460049356307946";

nbiot\_device\_t \*dev = NULL;

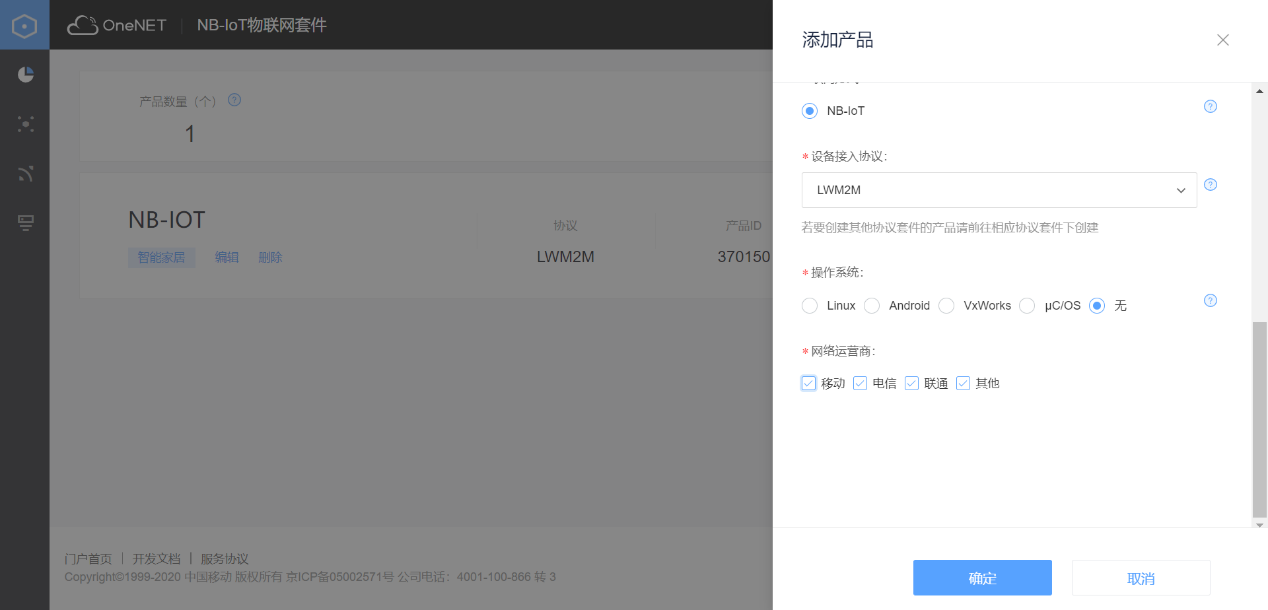
# 二、NB-IoT连接OneNet平台

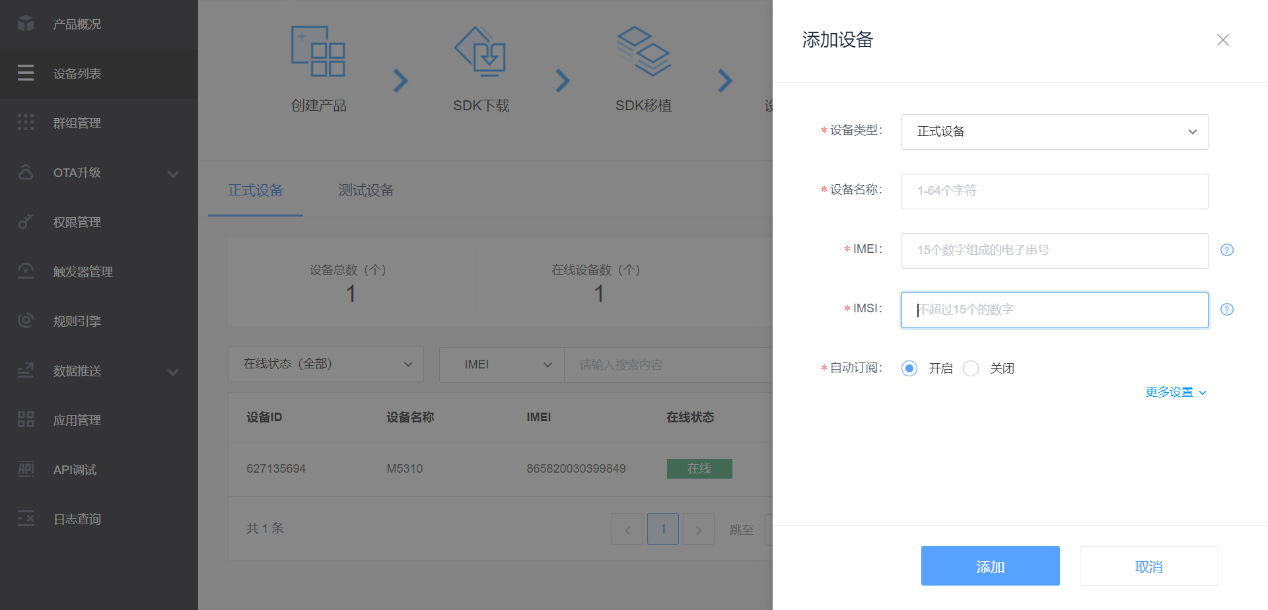
## （一）NB-IoT产品设备的创建

### 1.获取IMIS:数据线连接至电脑后打开串口助手

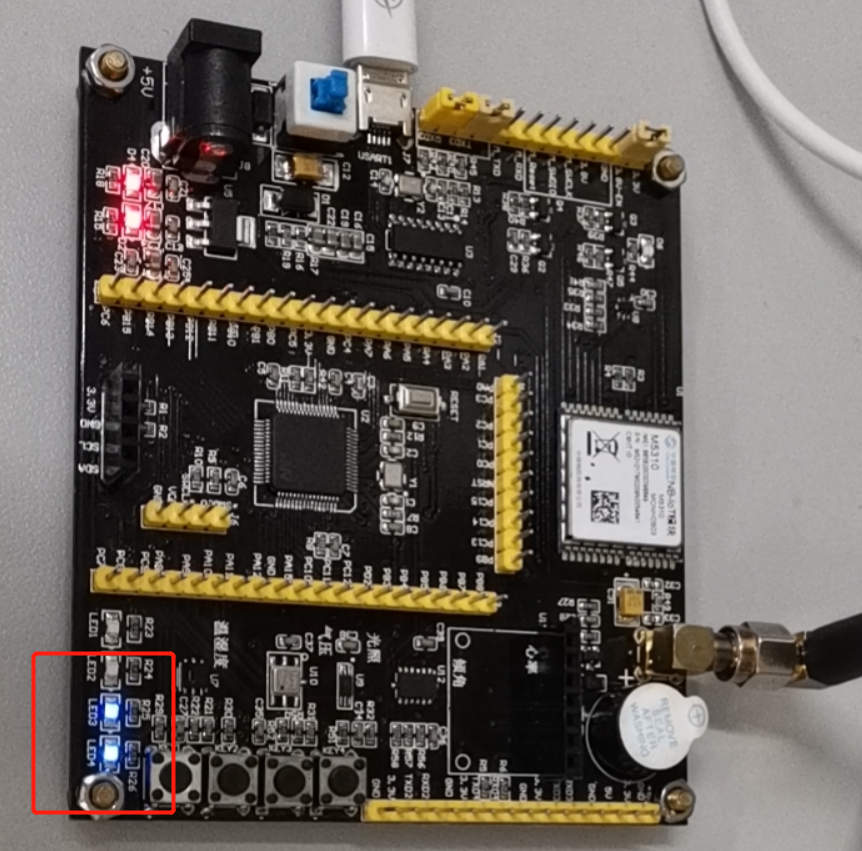


### 2.登录OneNet平台创建NB-IOT产品,添加设备

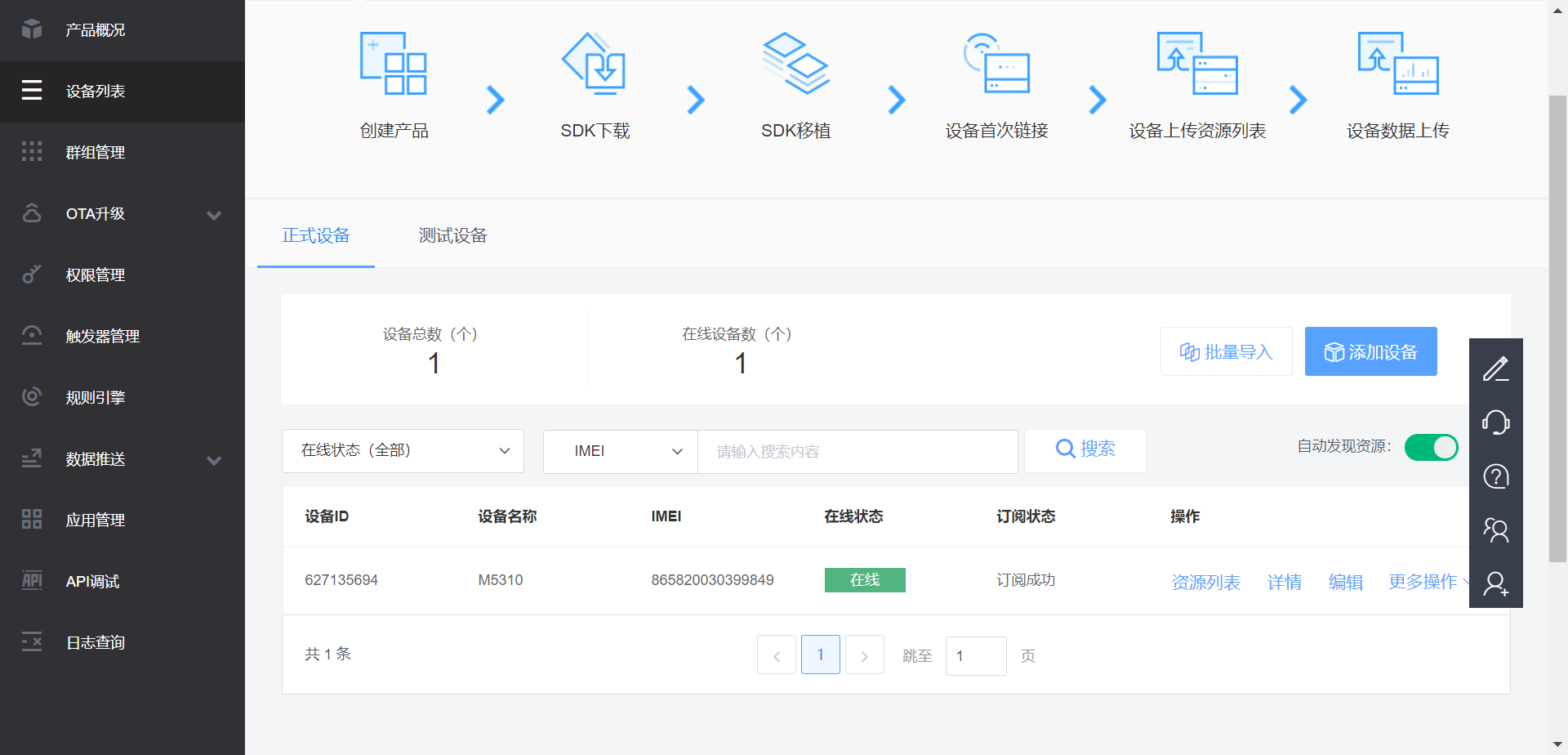




### 3.重新打开开发板



蓝灯出现表示开发板和云平台已经连接好了，并且OneNet平台会显示在线



### 4.测试连接是否成功

调用API （参考文档：<https://open.iot.10086.cn/doc/nb-iot/book/application-develop/list/20batch-query-dev-latest-data.html>）

代码：



GET http://api.heclouds.com/devices/datapoints?devIds=12323,12324 HTTP/1.1

#devIds是指设备的id

api-key: WhI\*\*\*\*\*\*\*\*\*\*\*\*\*v1c=#这个是根据Master-APIkey来看的

Content-Type: application/json

测试效果：返回的数据这3个设备，分别是我们的温度，湿度，光照数据。如下图



# 三、电机控制

# STM32F103驱动直流电机

## 实验环境：

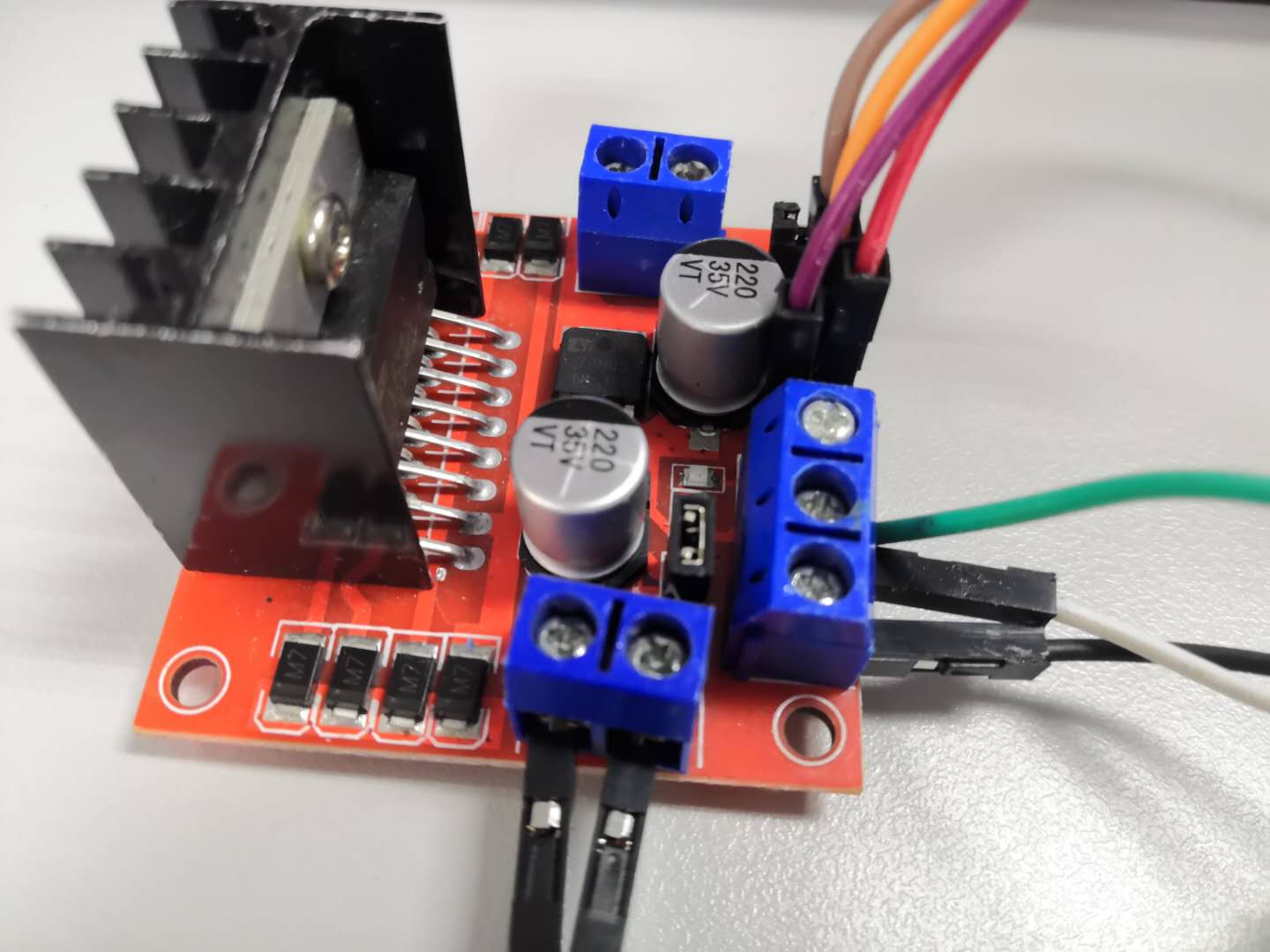
（一）STM32F10X 单片机

（二）L298N驱动模块

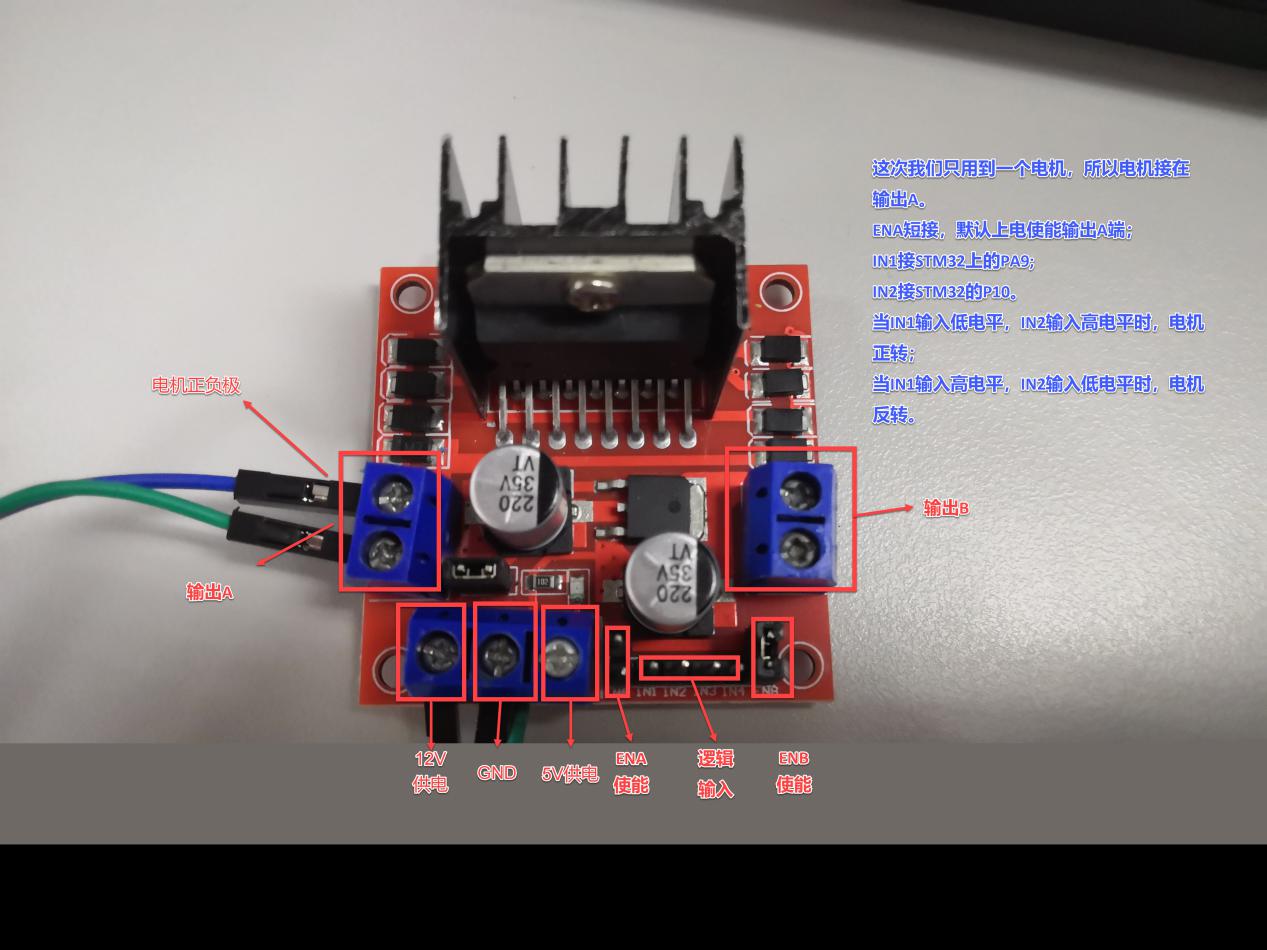
（三）直流电机 1个

## 第一步：L298N正确与STM32接线

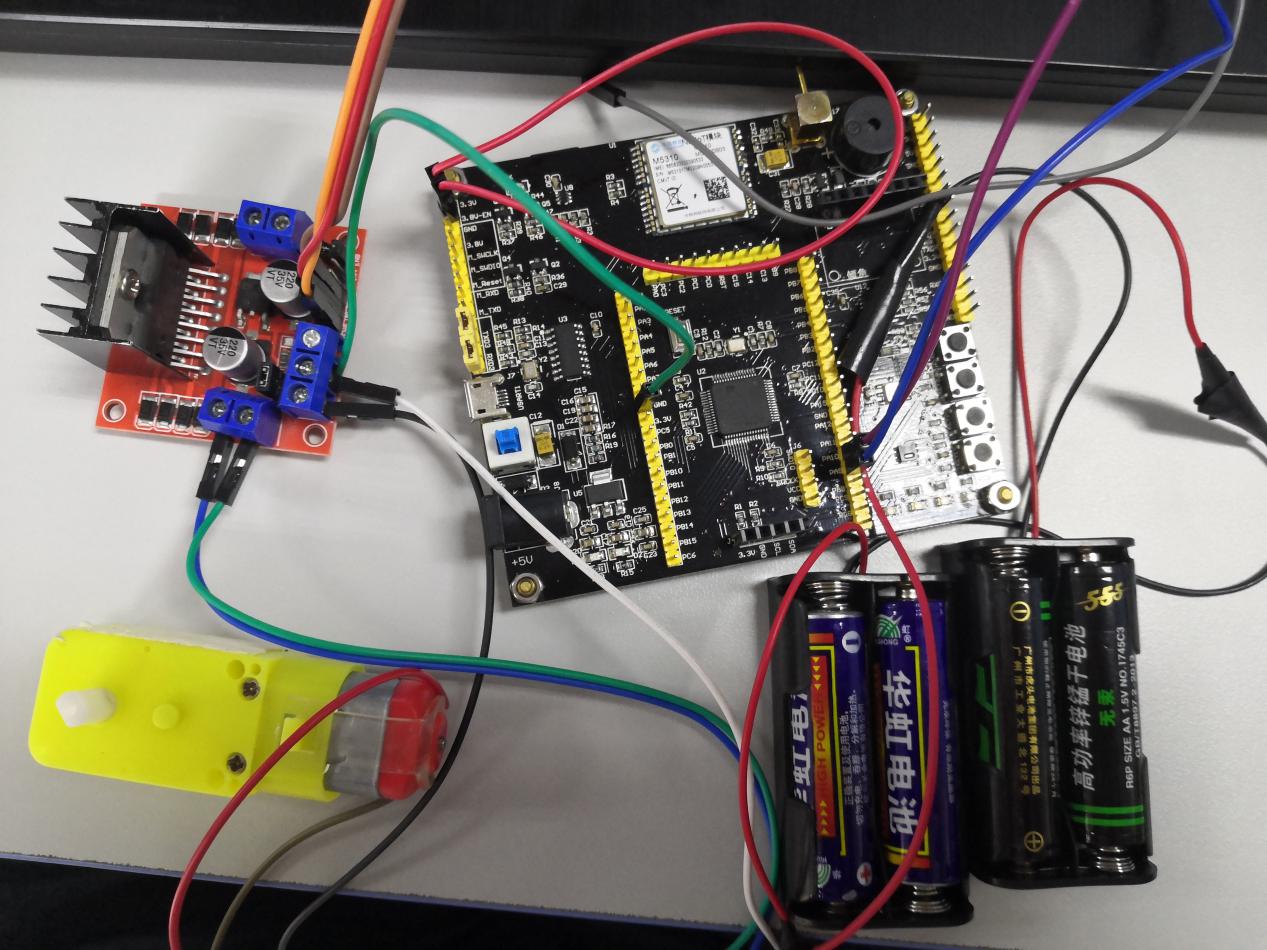
 L298N的模块如下：



可以参考下图进行接线：

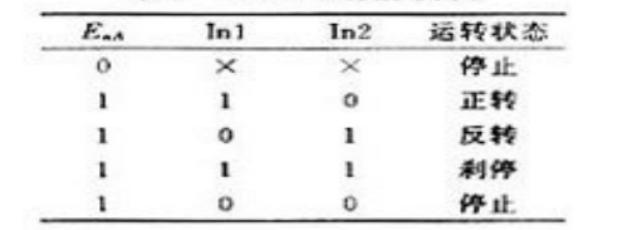


我接的效果图（有点乱）



## 第二步：编写程序

接下来我们可以根据L298N的说明书进行编程



### 1 电机头文件编写

首先编写motor.h头文件：

**#ifndef \_\_MOTOR1\_H**

**#define \_\_MOTOR1\_H**

**#include "stm32f10x.h"**

**#include "stm32f10x\_gpio.h"**

**#define High 1**

**#define Low 0**

**#define IN1(a) if (a) \**

**GPIO\_SetBits(GPIOA,GPIO\_Pin\_9);\**

**else \**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_9)**

**#define IN2(a) if (a) \**

**GPIO\_SetBits(GPIOA,GPIO\_Pin\_10);\**

**else \**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_10)**

**#define IN3(a) if (a) \**

**GPIO\_SetBits(GPIOA,GPIO\_Pin\_11);\**

**else \**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_11)**

**#define IN4(a) if (a) \**

**GPIO\_SetBits(GPIOA,GPIO\_Pin\_12);\**

**else \**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_12)**

**void Motor\_12\_Config(void);**

**void Motor\_1\_STOP(void);**

**void Motor\_1\_PRun(void);**

**void Motor\_1\_NRun(void);**

**void Motor\_2\_STOP(void);**

**void Motor\_2\_PRun(void);**

**void Motor\_2\_NRun(void);**

**#endif**

### 2 电机初始化程序编写

然后编写motor.c

**//单片机头文件**

**#include "motor.h"**

**#include "stm32f10x.h" //**

**#include "stm32f10x\_rcc.h" //**

**#include "delay.h" //**

**void Motor\_12\_Config(void) //定义初始化函数**

**{**

**GPIO\_InitTypeDef GPIO\_InitStructure; //定义GPIO\_InitTypeDef结构体**

**RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOA, ENABLE); //开启引脚时钟**

**GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_9|GPIO\_Pin\_10|GPIO\_Pin\_11|GPIO\_Pin\_12; //定义Pin引脚**

**GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; //通用推挽输出**

**GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; //引脚输出功率**

**GPIO\_Init(GPIOA, &GPIO\_InitStructure); //初始化引脚函数**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_9|GPIO\_Pin\_10|GPIO\_Pin\_11|GPIO\_Pin\_12); //所有引脚拉低即全部引脚设为低电平0：PA2\3\4\5=0**

**}**

**void Motor\_1\_STOP(void)//电机1停止函数**

**{**

**IN1(Low);**

**IN2(Low);**

**}**

**void Motor\_1\_PRun(void)//电机1正转函数**

**{**

**IN1(Low);**

**IN2(High);**

**}**

**void Motor\_1\_NRun(void)//电机1反转函数**

**{**

**IN1(High);**

**IN2(Low);**

**}**

**void Motor\_2\_STOP(void)**

**{**

**IN3(High);**

**IN4(High);**

**}**

**void Motor\_2\_PRun(void)**

**{**

**IN3(Low);**

**IN4(High);**

**}**

**void Motor\_2\_NRun(void)**

**{**

**IN3(High);**

**IN4(Low);**

**}**

### 3 主程序编写

最后是控制电机主程序main.c

**/\*\***

**\* Copyright (c) 2017 China Mobile IOT.**

**\* All rights reserved.**

**\*\*/**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**#include <string.h>**

**#include "led.h"**

**#include "key.h"**

**#include "delay.h"**

**#include "motor.h"**

**unsigned int i=0;**

**int main( int argc, char \*argv[] )**

**{**

**//LED灯初始化**

**Led\_Init();**

**//电机驱动初始化**

**Motor\_12\_Config();**

**//按键初始化**

**key\_init();**

**//延时函数初始化**

**delay\_init();**

**//上电流水灯**

**for(i=0;i<=4;i++)**

**{**

**GPIO\_SetBits(GPIOA,GPIO\_Pin\_4|GPIO\_Pin\_5|GPIO\_Pin\_6|GPIO\_Pin\_7);**

**delay\_ms(10000);**

**GPIO\_ResetBits(GPIOA,GPIO\_Pin\_4|GPIO\_Pin\_5|GPIO\_Pin\_6|GPIO\_Pin\_7);**

**delay\_ms(10000);**

**}**

**while(1)**

**{**

**//按键KEY0控制电机正转**

**if(KEY0 == 0)**

**{**

**delay\_ms(20); //按键消抖，时间一般为10-20ms**

**if(KEY0 == 0)**

**{**

**while(KEY0==0){}**

**Motor\_1\_PRun();//电机1正转函数**

**}**

**}**

**//按键KEY1控制电机反转**

**if(KEY1 == 0)**

**{**

**delay\_ms(20); //按键消抖，时间一般为10-20ms**

**if(KEY1 == 0)**

**{**

**while(KEY1==0){}**

**Motor\_1\_NRun();//电机1反转函数**

**}**

**}**

**//按键KEY2控制电机停止**

**if(KEY2 == 0)**

**{**

**delay\_ms(20); //按键消抖，时间一般为10-20ms**

**if(KEY2 == 0)**

**{**

**while(KEY2==0){}**

**Motor\_1\_STOP();//电机1停止**

**}**

**}**

**if(KEY3 == 0)**

**{**

**delay\_ms(20); //按键消抖，时间一般为10-20ms**

**if(KEY3 == 0)**

**{**

**while(KEY3==0){}**

**}**

**}**

**}**

**}**

# 四、读取设备数据和在OneNet平台下发开关命令

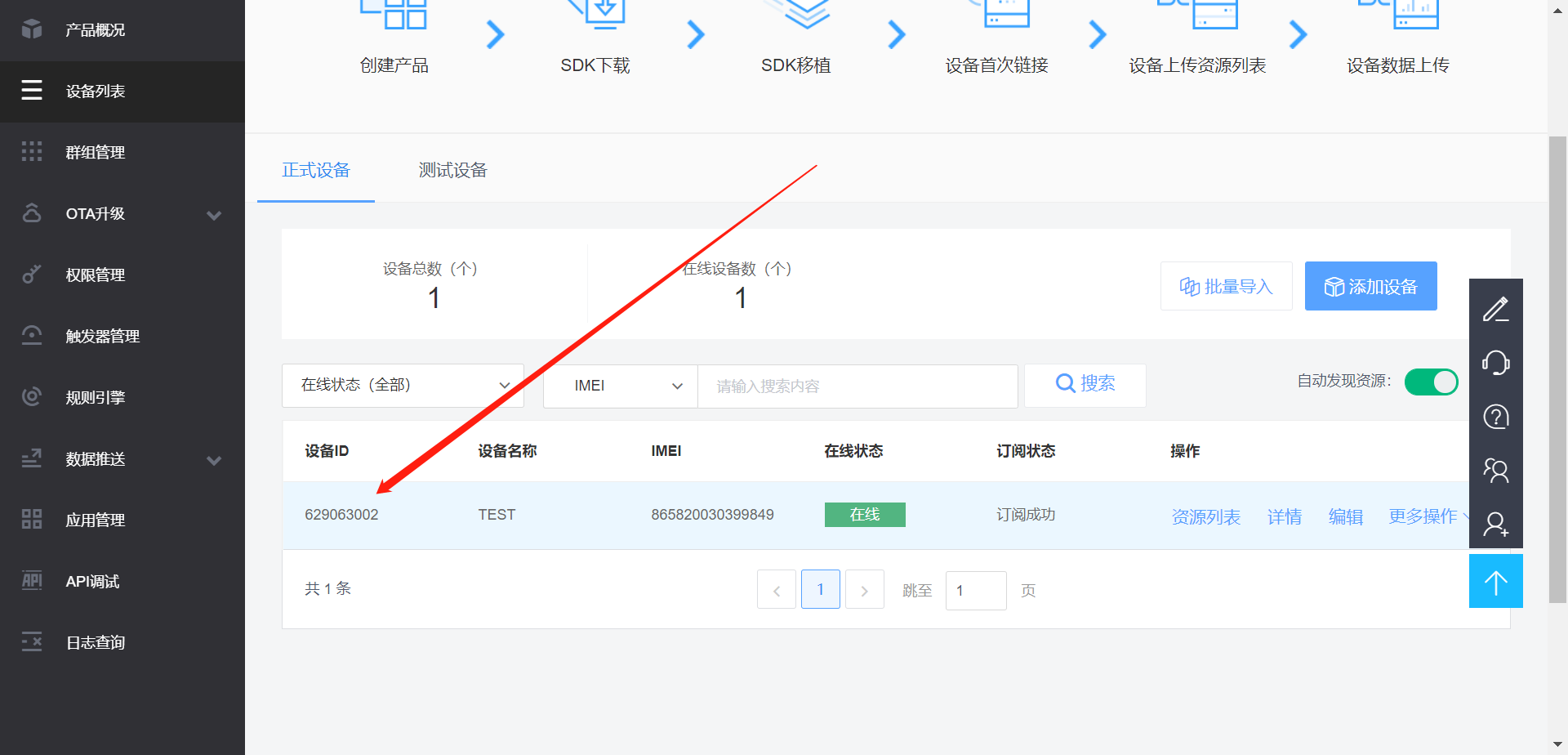
## （一）读取设备数据

1.读取OneNet平台的API

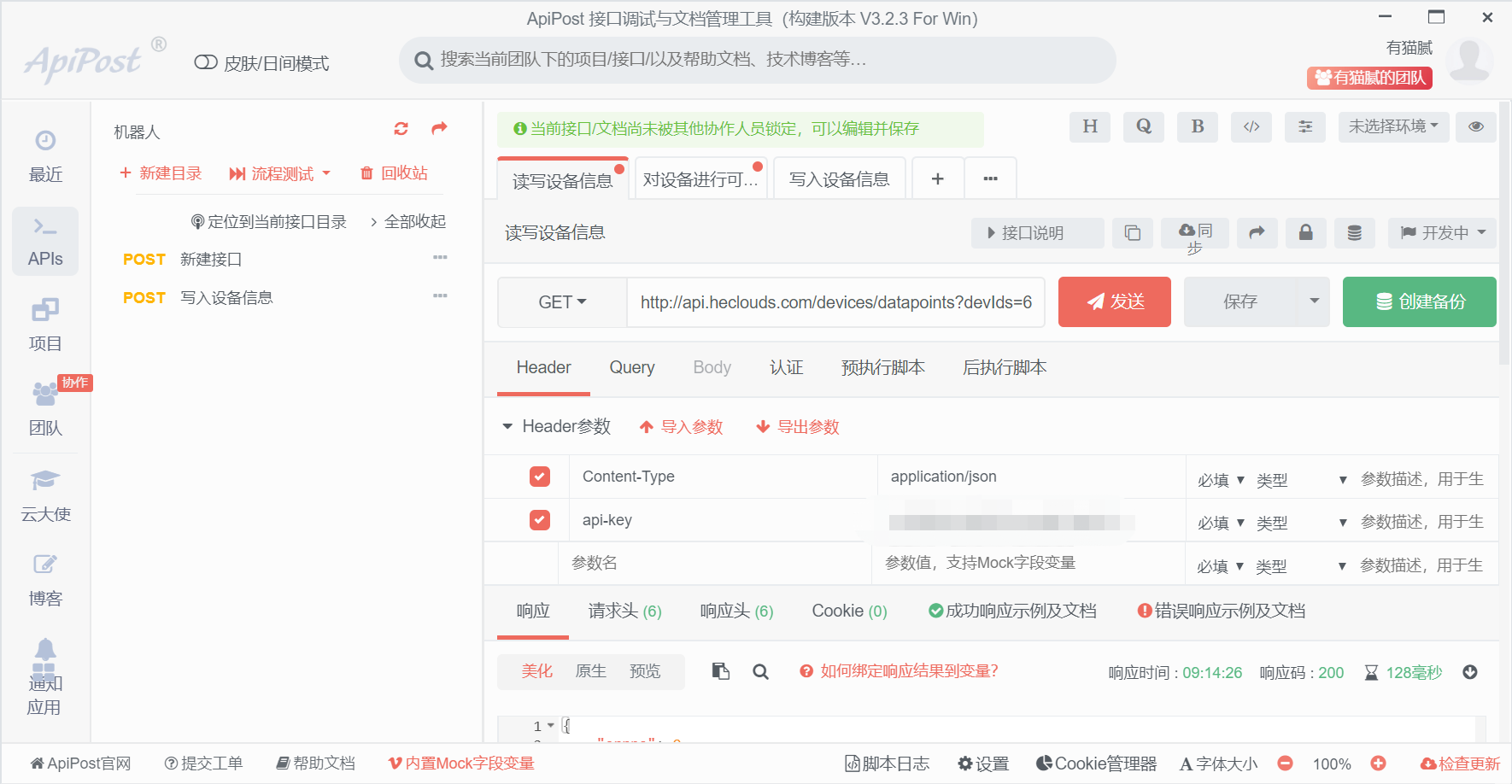


读取的API是:http://api.heclouds.com/devices/datapoints?devIds=629063002

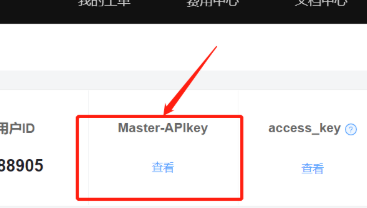
其中devids这个指的是你的设备ID



2. 在headers里面添加一些必要参数，就可以调用了



补充： api-key的是你产品里面的master-apikey



3.写好参数就可以调用了，选择get方式，点击发送

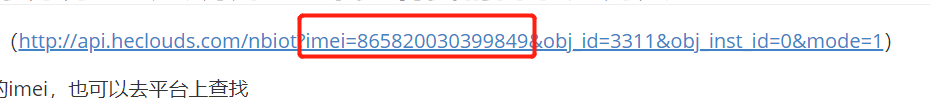


其中value是需要获取到的参数值，例如第一个是我们的光照值，然后第二个是我们灯的状态，false代表就是关灯的意思

## （二）调用API来控制我们的小灯开关

1.API接口是：

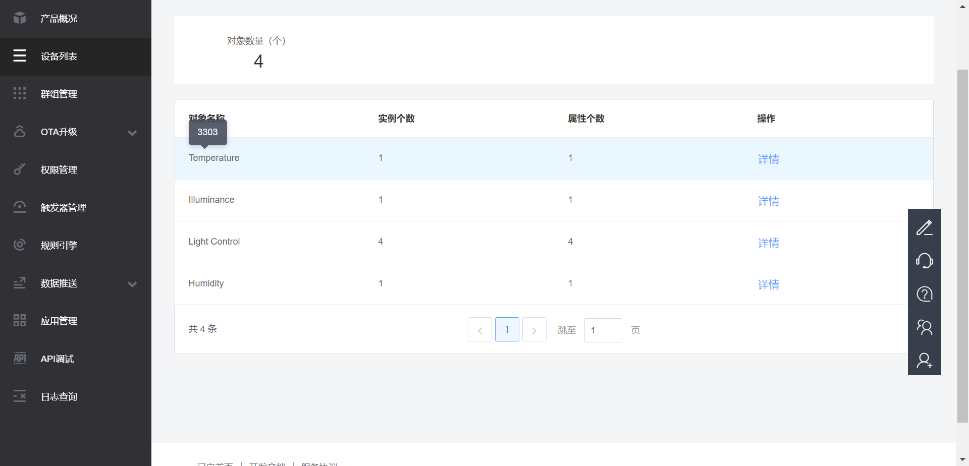
<http://api.heclouds.com/nbiot?imei=865820030399849&obj_id=3311&obj_inst_id=0&mode=1>



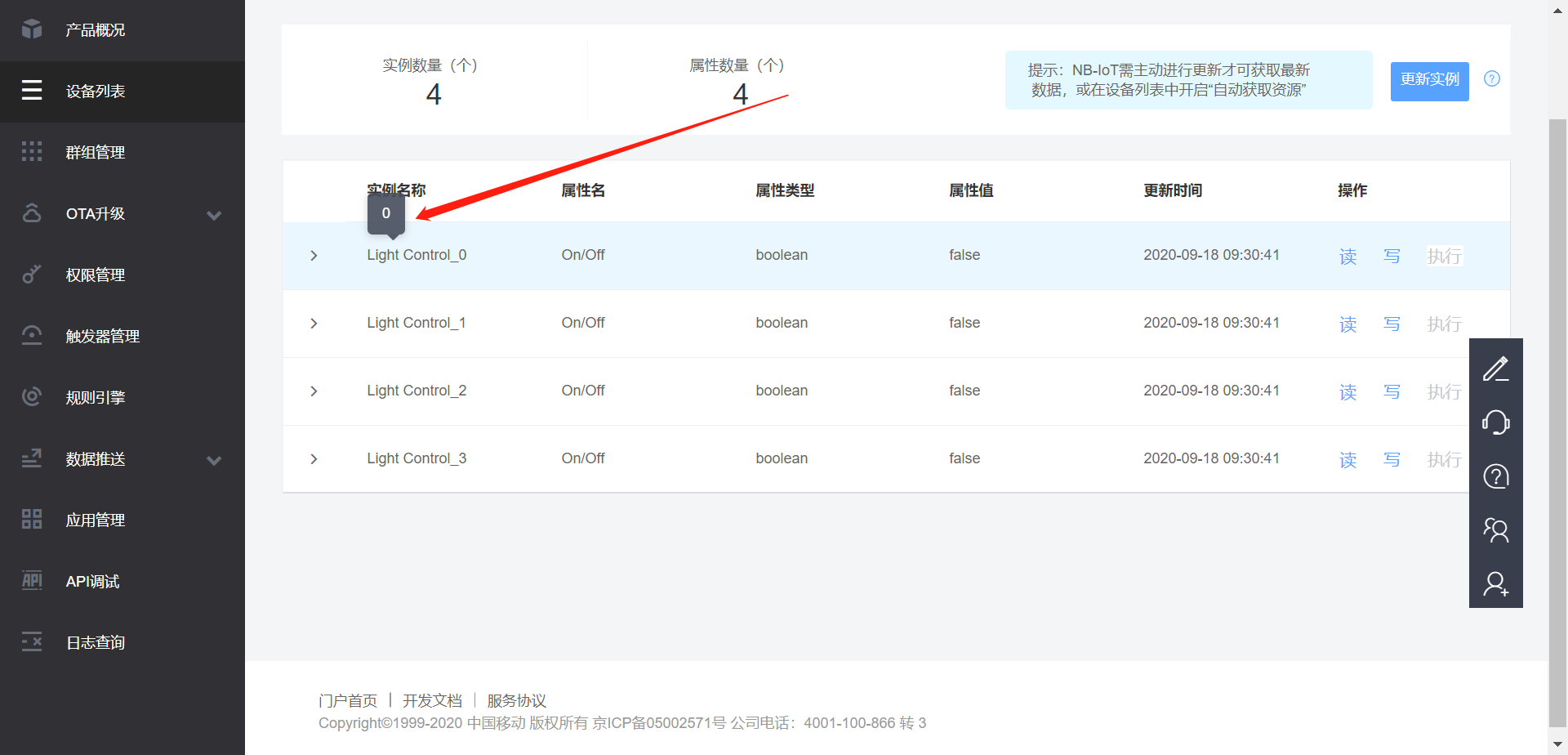
2. imei：红色方框是imei，在平台的设备列表能找到



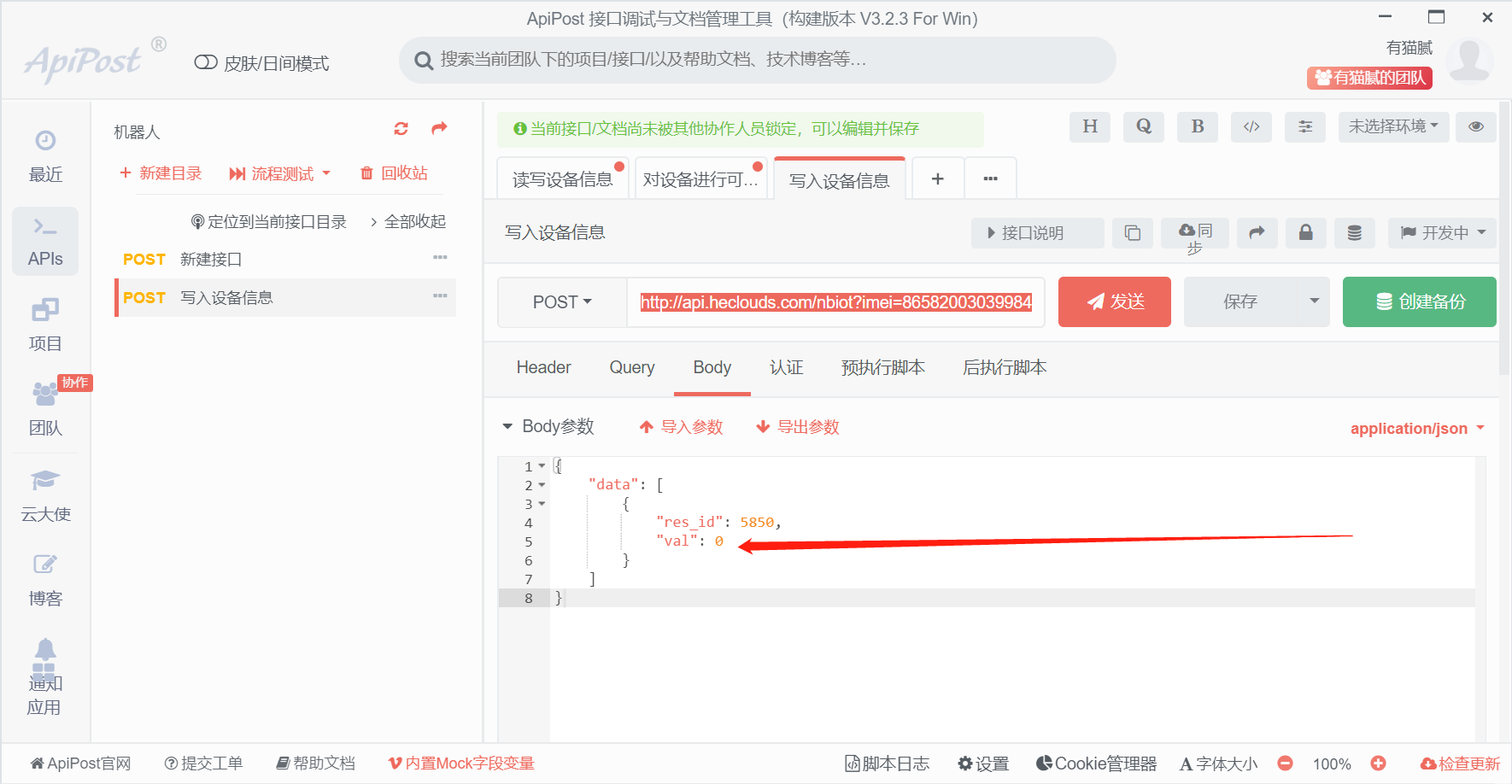
3. Obj\_id是指设备名称的序列，如下图



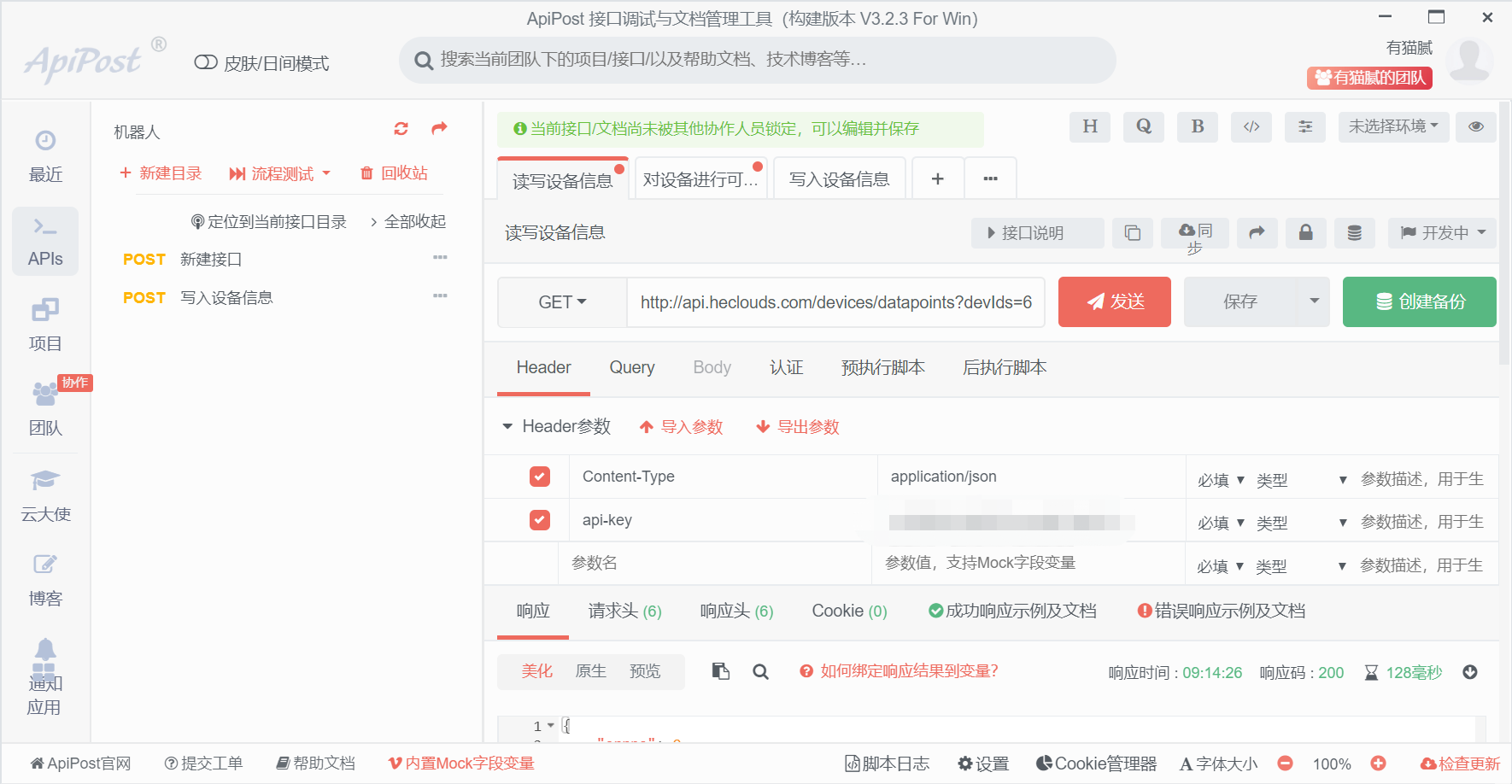
4. obj\_inst\_id指的是我们传感器类型里面的id，例如我的Light-Control里面有4个，然后0就是指的是我的第一个个数，1是第二个个数，以此类推



5. mode是指我发送的数据类型，这里mode只有1或者2,1指的是我发送的是16位进制的内容，2指的是我发送的是时间戳,这个mode与我们最后发送的参数有关，我们要发送控制命令的话，那么Mode选择1即可，然后val：0指的是我们发送低电平过去给开发板，从而做到可以控制开发板的小灯或者延伸的话，进而控制舵机或者电机来实现更多的功能



注意：发送这个post命令，它的请求头和get的请求头是一样的也是这样写



注意二：还有添加query body这个部分，不然是发送失败的，填写对应的参数即可



6. 最后再在我们的body里面写我们要控制的设备，已经发送的命令，例如我现在是控制一个小灯的开

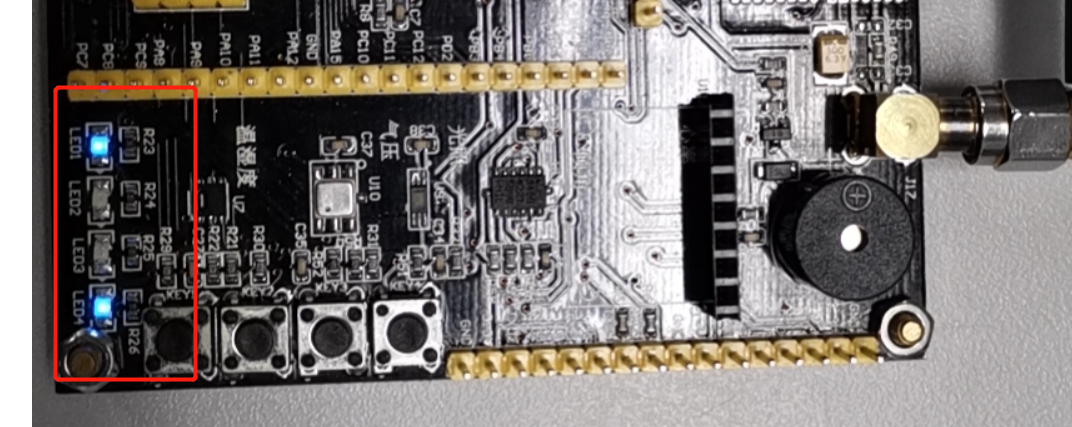




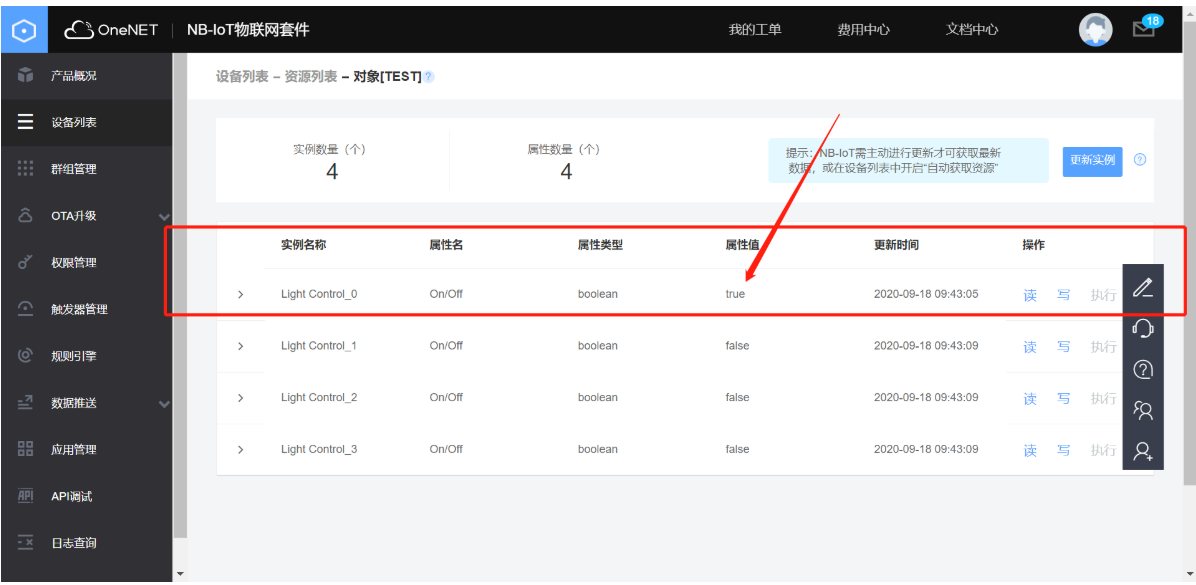
现在蓝灯是一盏灯的状态，其他蓝灯都是关的，我这边点击发送



返回一个成功的参数，这时我们再看我们的开发板



回到云平台，属性值也变化为“true”，说明调用API成功



这时我们再用这个api写代码发送post，get命令都可以，可以做任何我们可以想象到的事情

参考网址一：

<https://open.iot.10086.cn/doc/book/device-develop/multpro/lwm2m/%E5%9F%BA%E4%BA%8EM5310%E6%A8%A1%E7%BB%84%E5%AE%9E%E7%8E%B0NB%E8%AE%BE%E5%A4%87%E6%8E%A5%E5%85%A5%E5%AE%9E%E4%BE%8B%EF%BC%88LWM2M%EF%BC%89.html>

参考网址二：

<https://open.iot.10086.cn/doc/nb-iot/book/application-develop/api-list.html>

# 五、应用端

## （一）代码调试

通过前面的操作，已经成功获取OneNet平台的API了，接下来是调用API实现数据的实时查看和发送命令

1.1

import android.graphics.Color;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.gson.Gson;

import com.shiqi.lianjieonenet.util.Json;

import java.io.IOException;

import java.util.List;

import okhttp3.OkHttpClient;

import okhttp3.Request;

import okhttp3.Response;

public class MainActivity extends AppCompatActivity {

static String time;

static Double wendu;

static Double shidu;

public final String url1 = "http://api.heclouds.com/devices/629063002/datapoints?datastream\_id=3303\_0\_5700&limit=5";

public final String url2 = "http://api.heclouds.com/devices/629063002/datapoints?datastream\_id=3304\_0\_5700&limit=5";

private TextView txtwendu,txtshidu,txtwendu2,txtshidu2,tishi,shijian;

private EditText etbaojin;

private Button btn,baojin;

private int baojinshu = 35;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

txtwendu = findViewById(R.id.txtwendu);

txtshidu = findViewById(R.id.txtshidu);

txtwendu2 = findViewById(R.id.txtwendu2);

txtshidu2 = findViewById(R.id.txtshidu2);

tishi = findViewById(R.id.tishi);

shijian = findViewById(R.id.txtshijian);

btn = findViewById(R.id.button);

baojin = findViewById(R.id.button2);

etbaojin = findViewById(R.id.etbaojing);

TextView show\_tv = (TextView) tishi.findViewById(R.id.tishi);

show\_tv.setSelected(true);

show\_tv.setText(" 点击“开启应用”即可查看室内温湿度");

show\_tv.setTextColor(Color.WHITE);

//开启应用按钮点击事件

btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

getData();

}

});

//修改报警值

baojin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String bao=etbaojin.getText().toString().trim();

baojinshu = Integer.parseInt(bao);

//其中baojinshu为被转的数字，bao为将被转换的字符串。

getData();

}

});

}

//获取网络端数据，请求方式为Http

public void getData(){

new Thread(new Runnable() {

@Override

public void run() {

try{

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()

.url(url1)

.addHeader("api-key", "LoEEarS4lSeRRLeuwtlK19arUlY=")

.addHeader("Content-Type","application/json")

.build();

Response response = client.newCall(request).execute();

String responseData = response.body().string();

Log.w("test", responseData);

//json提取数据

JsonRootBean app = Json.gson.fromJson(responseData, JsonRootBean.class);

List<Datastreams> streams = app.getData().getDatastreams();

List<Datapoints> points = streams.get(0).getDatapoints();

int count = app.getData().getCount();//获取数据的数量

// wendu = new Integer(1);

wendu = points.get(1).getValue();

Log.w("www","wendu="+wendu);//log输出

final String wendu1 = wendu.toString();//数据类型转换

//textview显示需要回到主线程

txtwendu.post(new Runnable() {

@Override

public void run() {

txtwendu.setText("当前温度：");

}

});

txtwendu2.post(new Runnable() {

@Override

public void run() {

txtwendu2.setText(String.format("%.1f°C",wendu));

}

});

tishi.post(new Runnable() {

@Override

public void run() {

if(wendu>=baojinshu){

tishi.setText("注意！当前室内温度过高！");

tishi.setTextColor(Color.RED);

Toast.makeText(MainActivity.this, "当前温度过高！您可以打开空调降温，防止中暑。", Toast.LENGTH\_SHORT).show();

}

else if(wendu<baojinshu){

TextView show\_tv = (TextView) tishi.findViewById(R.id.tishi);

show\_tv.setSelected(true);

show\_tv.setText("当前室内温度为"+String.format("%.2f°C",wendu));

show\_tv.setTextColor(Color.WHITE);

}

}

});

time = new String();

time = points.get(1).getAt().toString();

// final String time1 =time.substring(0,19);

shijian.post(new Runnable() {

@Override

public void run() {

shijian.setText(" 时间: "+time);

// shijian.setText(" 时间: "+time1);

}

});

}catch (IOException e){

e.printStackTrace();

}

try {

OkHttpClient client1 = new OkHttpClient();

Request request1 = new Request.Builder()

.url(url2)

.addHeader("api-key", "LoEEarS4lSeRRLeuwtlK19arUlY=")

.addHeader("Content-Type","application/json")

.build();

Response response1 = client1.newCall(request1).execute();

String responseData1 = response1.body().string();

JsonRootBean app1 = Json.gson.fromJson(responseData1, JsonRootBean.class);

List<Datastreams> streams1 = app1.getData().getDatastreams();

List<Datapoints> points1 = streams1.get(0).getDatapoints();

int count = app1.getData().getCount();//获取数据的数量

shidu = points1.get(1).getValue();

Log.w("www","shidu="+shidu);

final String shidu1 = shidu.toString();

txtshidu.post(new Runnable() {

@Override

public void run() {

txtshidu.setText("当前湿度：");

}

});

txtshidu2.post(new Runnable() {

@Override

public void run() {

txtshidu2.setText(String.format("%.1f%%",shidu));

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

}).start();

}

}

## （二） build.gradle中增加：

compile 'com.squareup.okhttp3:okhttp:3.4.1'

compile 'com.google.code.gson:gson:2.7'

compile 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0'

compile 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.1'

implementation 'com.google.android.material:material:1.1.0'

## （三）在AndroidManifest.xml中添加：

<uses-permission android:name="android.permission.INTERNET" />

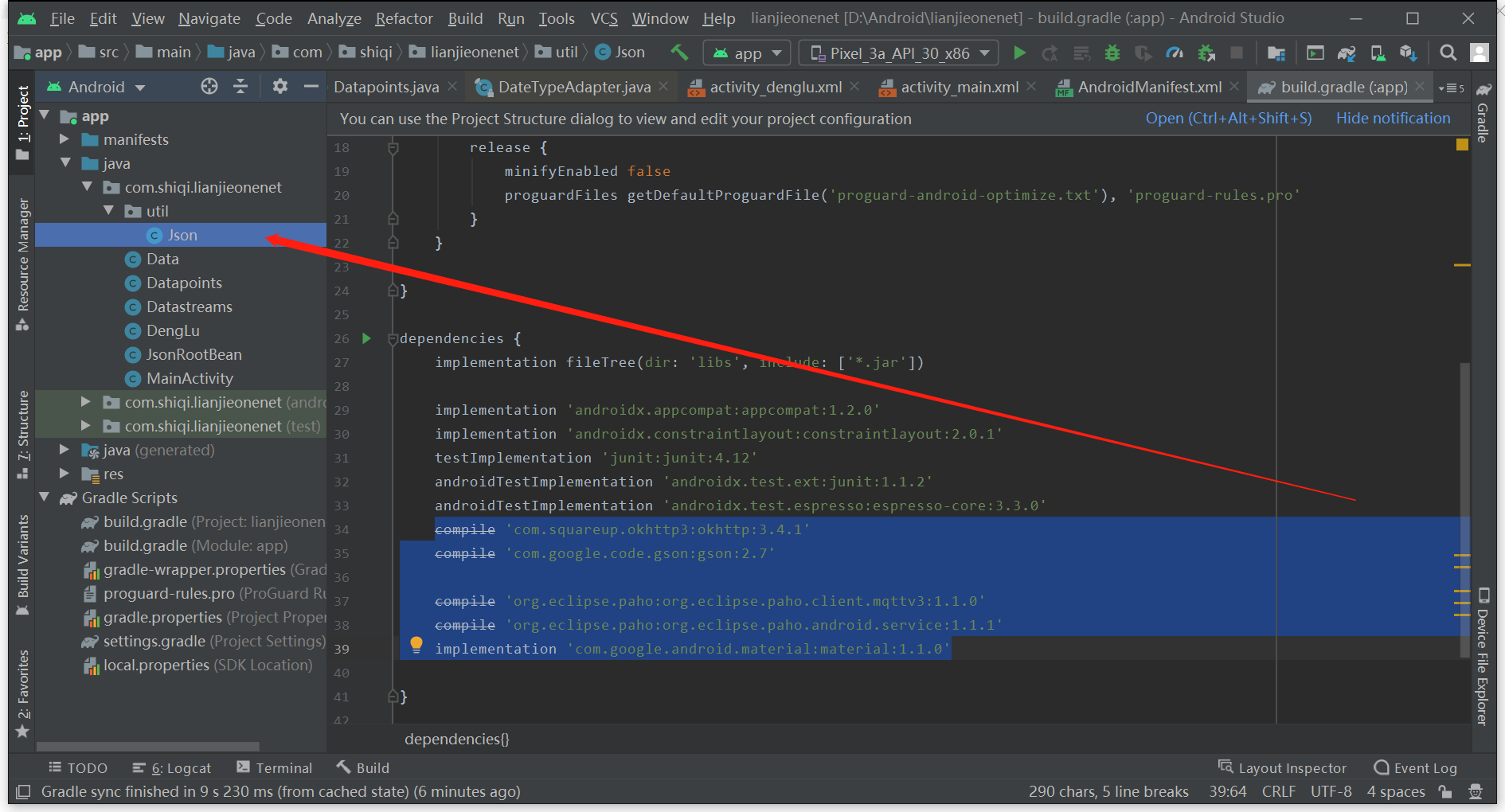
## （四）数据转换

因为我们获取的数据是json格式所以我们还得进行数据转换，这样我们才可以顺利定位到我们需要的文件

需要借用这个网站来进行数据转换[**json 在线转换工具**](http://www.bejson.com/)

用网站自动生成4个类JsonRootBean，Data，Datapoints ，Datastreams

然后因为我们的时间数据类型错误，那么如何才能解决这个错误呢



## （五）创建一个util文件

在这个文件里面添加json类

import com.google.gson.Gson;

import com.google.gson.GsonBuilder;

public class Json {

public static Gson gson;

static {

gson = new GsonBuilder().setDateFormat("yyyy-MM-dd HH:mm:ss").create();

}

}

做到数据类型正确转换，这时再运行我们的Android来看看我们获取到的数据，这里有一点要注意的，我们需要用真机来进行调试，如果用模拟器会出现闪退的现象，这个应该是和网络有关

这时候再打开我们的调试好的APP，点击登录查看







很显然，数据是完全对的上的，那么到这一步就是说明可以正确获取到数据了

## （六）post控件，做到命令的时候发送

//下发命令

public void lightControl(final String imei, final String objId, final int objInstId, final int mode, final int resId, final int val) {

new Thread(new Runnable() {

@Override

public void run() {

String toastStr = null;

try {

String url = String.format("http://api.heclouds.com/nbiot?imei=%s&obj\_id=%s&obj\_inst\_id=%d&mode=%d", imei, objId, objInstId, mode);

String body = String.format("{\"data\":[{\"res\_id\": %d,\"val\": %d}]}", resId, val);

Request request = HttpHandler.INSTANCE.post( url, RequestBody.create(HttpHandler.JSON\_MEDIA\_TYPE, body));

Response response = HttpHandler.INSTANCE.getResponse(request);

String jsonStr = response.body().string();

final JsonRootBean res = Json.gson.fromJson(jsonStr, JsonRootBean.class);

Log.w("test", jsonStr);

switch (res.getErrno()) {

case 0:

// TODO 成功

break;

case 2001:

toastStr = "设备不在线。错误代码：2001";

break;

default:

toastStr = String.format("%s. 错误代码：%d", res.getError(), res.getErrno());

break;

}

} catch (IOException e) {

// TODO 提示连接异常信息

toastStr = "连接失败，请检测手机或设备的网络状态";

e.printStackTrace();

} finally {

if(toastStr != null) {

final String finalToastStr = toastStr;

etbaojin.post(new Runnable() {

@Override

public void run() {

Toast.makeText( MainActivity.this, finalToastStr, Toast.LENGTH\_SHORT).show();

}

});

}

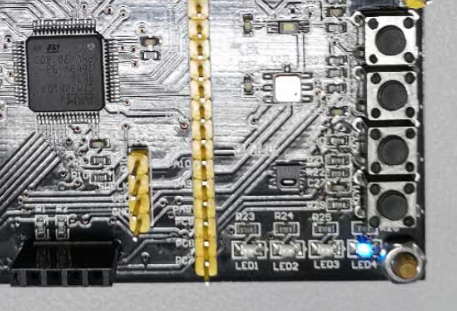
}

}

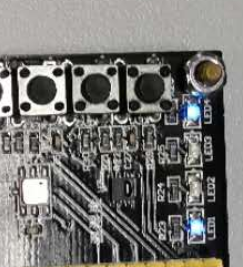
}).start();

}

json来定位，后面再用post来请求



暂时先用LED1来代替伸展功能 ，当你按了伸展按钮的时候



按了收回按钮之后

【