

ESP8266 LUA开发基础入门篇  
备份(22)  
ESP8266 SDK开发(33)  
ESP8266 SDK开发基础入门篇  
备份(30)  
GPRS Air202 LUA开发(11)  
HC32F460(华大单片机)物联网  
开发(12)  
HC32F460(华大单片机)学习开  
发(8)  
NB-IOT Air302 AT指令和LUA  
脚本语言开发(27)  
PLC(三菱PLC)基础入门篇(2)  
STM32+Air724UG(4G模组)  
物联网开发(43)  
STM32+BC26/260Y物联网开  
发(37)  
STM32+CH395Q(以太网)物  
联网开发(24)  
STM32+ESP8266(ZLESP826  
6A)物联网开发(1)  
STM32+ESP8266+AIR202/3  
02远程升级方案(16)  
STM32+ESP8266+AIR202/3  
02终端管理方案(6)  
STM32+ESP8266+Air302物  
联网开发(65)  
STM32+W5500+AIR202/30  
2基本控制方案(25)  
STM32+W5500+AIR202/30  
2远程升级方案(6)  
UCOSii操作系统(1)  
W5500 学习开发(8)  
编程语言C#(11)  
编程语言Lua脚本语言基础入  
门篇(6)  
编程语言Python(1)  
单片机(LPC1778)LPC1778(2)  
单片机(MSP430)开发基础入门  
篇(4)  
单片机(STC89C51)单片机开发  
板学习入门篇(3)  
单片机(STM32)基础入门篇(3)  
单片机(STM32)综合应用系列  
(16)  
更多

#### 阅读排行榜

1. ESP8266使用详解(AT,LUA, SDK)(174413)
2. 1-安装MQTT服务器(Windows),并连接测试(105521)
3. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(68041)
4. ESP8266刷AT固件与node mcu固件(67045)
5. 有人WIFI模块使用详解(39503)
6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制----简单的连接通信)(37157)
7. C#中public与private与static(36195)
8. 关于TCP和MQTT之间的转换(35441)
9. android之TCP客户端编程(33109)
10. android服务端+eps8266+单片机+路由器之远程控制系  
统(31724)

#### 推荐排行榜

## 前言

**如果咱自己做产品,APP端肯定是一个程序.**

**如果咱向前面章节似的自己去云端注册设备,然后再把设备的信息(三元组)给用户,**

**然后用户再填写信息接入,这样用户体验不好,过于繁琐,也无疑增加了人工成本.**

**应该是用户打开手机APP,然后APP自己去云平台注册一个设备,然后接入云平台实现通信**

## 动态注册的方式(咱使用的是,一型一密免预注册)

**先解释几个名词:一机一密 ;一型一密(预注册) ;一型一密(免预注册)**

### 一机一密:

**就是咱原先测试的在网页上去手动注册设备,然后把ProductKey、DeviceSecret、 DeviceName**

**给设备,然后设备根据这三个信息组合计算下得到mqtt的连接参数,然后就连接通信了**

[https://help.aliyun.com/document\\_detail/74005.html?spm=a2c4g.11186623.6.585.3bb4111411EN77](https://help.aliyun.com/document_detail/74005.html?spm=a2c4g.11186623.6.585.3bb4111411EN77)

1. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(9)
2. C#委托+回调详解(9)
3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
4. 我的大学四年(6)
5. ESP8266使用详解(AT,LUA, SDK)(6)

## 最新评论

1. Re:2-6-1-视频传输,监控,直播方案-手机连接ESP32的热点,使用微信小程序查看摄像头图像(WiFi视频小车,局域网视频监控)  
赞赞赞,感谢大佬无私奉献  
--SJA2C2A
2. Re:中移动M5311模块使用手册(TCP,MQTT)  
请问你用的usb转ttl是哪一种呢,我用的ch340可是开机串口助手没有SIM识别显示  
--夏洛的网娅



## 一型一密(预注册):

需要咱先在网页上注册上设备,然后把ProductKey、ProductSecret、DeviceName

给设备,然后设备通过https或者mqtt获取DeviceSecret

然后设备根据ProductKey、DeviceSecret、DeviceName这三个信息组合计算下得到mqtt的连接参数,然后就连接通信了

[https://help.aliyun.com/document\\_detail/132111.html?](https://help.aliyun.com/document_detail/132111.html?spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804)

[spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804](https://help.aliyun.com/document_detail/132111.html?spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804)

- **authType**取值为register: 一型一密预注册认证方式, 返回DeviceSecret。物联网平台推送的消息Payload格式如下:

```
{
  "productKey": "xxx",
  "deviceName": "xxx",
  "deviceSecret": "xxx"
}
```

**一型一密(免预注册):** 免预注册就是不用在云端自己注册设备,设备连接上网络以后设备自己去注册.

这样子的话,每个设备烧录同样的程序就可以,不用再配置什么三元组了!

不过每个设备程序里面的需要写上产品的ProductKey 和 ProductSecret 的值,这两个值是固定的.

主要是区分开你注册的时候注册到哪个产品下.

然后设备通过https或者mqtt接口注册设备,

注册的时候服务器会返回ClientID和DeviceToken 等参数

最后按照平台规定的组合MQTT参数

**MQTT连接的方式注册设备:**

[https://help.aliyun.com/document\\_detail/132111.html?](https://help.aliyun.com/document_detail/132111.html?spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804)

[spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804](https://help.aliyun.com/document_detail/132111.html?spm=a2c4g.11186623.2.18.59863f82PBOpQr#task-1545804)

1. 设备发送CONNECT报文，报文中包含动态注册参数，请求建立连接。

说明 目前，动态注册只支持使用TLS建立连接，不支持TCP直连；动态注册时，云端不会校验MQTT连接的Keep Alive（保活时间），因此可以不用设置Keep Alive时间。

MQTT连接域名：

- 对于您购买的实例，接入域名请在**物联网平台控制台**，找到对应的实例，单击实例进入**实例详情**查看。
- 公共实例的连接域名为  `${YourProductKey}.iot-as-mqtt.${YourRegionId}.aliyuncs.com:1883` 。其中：
  - `${YourProductKey}`：请替换为设备所属产品的ProductKey。可登录**物联网平台控制台**，在对应实例的**设备详情页**获取。
  - `${YourRegionId}`：请参见**地域和可用区**替换为您的Region ID。

CONNECT报文的动态注册参数：

- 当设备属于您购买的实例，且使用**一型一密免预注册**认证方式时，动态注册参数如下：

```
mqttClientId: clientId|"securemode=-2,authType=xxxx,random=xxxx,signmethod=xxxx,instanceId=xxxx"  
mqttUserName: deviceName+"&"+"productKey  
mqttPassword: sign_hmac(productSecret,content)
```

- 当设备属于公共实例，或使用**一型一密预注册**认证方式时，动态注册参数如下：

```
mqttClientId: clientId|"securemode=2,authType=xxxx,random=xxxx,signmethod=xxxx"  
mqttUserName: deviceName+"&"+"productKey  
mqttPassword: sign_hmac(productSecret,content)
```

## 拼接注册返回的参数连接MQTT:

[https://help.aliyun.com/document\\_detail/73742.html?](https://help.aliyun.com/document_detail/73742.html?spm=a2c4g.11186623.2.16.1c99426ff6ECtQ#concept-mhv-ghm-b2b)

[spm=a2c4g.11186623.2.16.1c99426ff6ECtQ#concept-mhv-ghm-b2b](https://help.aliyun.com/document_detail/73742.html?spm=a2c4g.11186623.2.16.1c99426ff6ECtQ#concept-mhv-ghm-b2b)

MQTT的CONNECT报文参数

MQTT的参数如下：

```
mqttclientId=12345|securemode=3,signmethod=hmacsha1,timestamp=789|  
mqttusername=device&pk  
mqttpassword=hmacsha1("secret","clientId12345deviceName&deviceProductKey")
```

加密后的Password为二进制转16制字符串，示例结果为：

```
F4FD82A3D602B37FB0FA8B7892F24A477F85 ****
```

- 一型一密免预注册**认证方式：使用ProductKey、DeviceName、ClientId、DeviceToken连接。

```
mqttClientId: clientId|"securemode=-2,authType=connwl"  
mqttUsername: deviceName+"&"+"productKey  
mqttPassword: deviceToken
```

- `mqttClientId` ：格式中  `| |`  内为扩展参数。
- `clientId` 、 `deviceToken` ：设备动态注册时获得的ClientId、DeviceToken，请参见**基于MQTT通道的设备动态注册**。
- `securemode` ：表示目前安全模式，采用**一型一密免预注册**时，固定取值为-2。
- `authType` ：表示认证方式，采用**一型一密免预注册**时，固定取值为regnw1。

## 测试本节一型一密(免预注册)程序

### 1.打开本节的Android程序

- 名称 ^
- Android
  - Android.rar

## 2. 登录自己的云平台打开动态注册

注:选择哪个产品,设备就会注册到哪个产品下

物联网平台 / 设备管理 / 产品

### 产品 (设备模型)

[创建产品](#) [快速入门](#)

产品名称	ProductKey	节点类型	创建时间	操作
TestMqtt	a1m7er1nJbQ	设备	2019/11/10 01:47:20	<a href="#">查看</a> <a href="#">管理设备</a> <a href="#">删除</a>

## 打开动态注册

### ← TestMqtt

ProductKey a1m7er1nJbQ [复制](#) ProductSecret \*\*\*\*\* [查看](#)  
设备数 16 [前往管理](#)

- [产品信息](#)
- [Topic 类列表](#)
- [功能定义](#)
- [数据解析](#)
- [服务端订阅](#)
- [设备开发](#)

当前产品下有设备同时有多个 ClientID, 点击 [查看](#) 可跳转至安全运营中心查看所有风险设备列表。

产品信息 [编辑](#)

产品名称	TestMqtt	节点类型	直连设备	创建
所属品类	路灯照明	数据格式	ICA 标准数据格式 ( Alink JSON )	数据
认证方式	设备密钥	动态注册	已开启 <input checked="" type="checkbox"/>	状态
连网协议	Wi-Fi	产品描述	-	

## 3,查看并替换自己产品的ProductSecret; ProductKey;



提示:有 instanceId , 则填写上实例的 instanceId



iot-060a09wx 基础型

✔ 运行中

ID : iot-060a09wx

到期时间 : 2022/01/21

```
menu_main.xml × MainActivity.java × RegisterDeviceForMqtt.java × MyMqttClient.java × AndroidMa
package com.example.yang.zlappalyreg;

import ...

public class RegisterDeviceForMqtt {
    private static final String TAG = RegisterDeviceForMqtt.class.getSimpleName();

    public static String instanceId = ""; // 替换自己的实例ID(如果没有则不需要填写)
    public static String ProductSecret = ""; // 替换自己的 ProductSecret
    public static String ProductKey = ""; // instanceId 替换自己的 ProductKey
    public static String Region = "cn-shanghai"; // 地域, 根据自己的修改
    public static int MqttPort = 1883; // 端口号
```

#### 4. 下载安装到手机

注:注册的时候设备名字是可以自己规定的



## 5. 输入设备的名称, 然后点击注册设备

我写的是 regapp



## 8. 注册成功以后自动跳转到主页面



## 9.用户可以看下日志,和刷新下云平台设备列表

```

Error  🔍  Regex  Show only selected application
/RegisterDeviceForMqtt: aly_init: MqttIPString:aikalhdH42Z.iot-as-mqtt.cn-shanghai.aliyuncs.com
/RegisterDeviceForMqtt: aly_init: MqttClientId:aikalhdH42Z.regapp|securemode=2,authType=regw1,signmethod-hmacsha1,random=89|
/RegisterDeviceForMqtt: aly_init: MqttUserString:regapp8aikalhdH42Z
/RegisterDeviceForMqtt: aly_init: MqttPwdString:54E630A1F2231B07C74376A86F6332A760A54524
/RegisterDeviceForMqtt: subscribe Err+错误的用户名和密码 (4)
/RegisterDeviceForMqtt: aly_init: MqttIPString:aikalhdH42Z.iot-as-mqtt.cn-shanghai.aliyuncs.com
/RegisterDeviceForMqtt: aly_init: MqttClientId:aikalhdH42Z.regapp|securemode=2,authType=regw1,signmethod-hmacsha1,random=63|
/RegisterDeviceForMqtt: aly_init: MqttUserString:regapp8aikalhdH42Z
/RegisterDeviceForMqtt: aly_init: MqttPwdString:86E52388468E6FDC1B0303194A369F252D218 注册设备
ActivityRegDevice: callback: {"clientId":"3WLjSngFNZE6E81bR5j000100","productKey":"aikalhdH42Z","deviceName":"regapp","deviceToken":"","11604109939253*da3fa823120752c"}
MyMqttClient: sharedCenter: myMqttClient
MyMqttClient: aly_init: MqttIPString:aikalhdH42Z.iot-as-mqtt.cn-shanghai.aliyuncs.com
MyMqttClient: aly_init: MqttClientId:3WLjSngFNZE6E81bR5j000100|securemode--2,authType=connw1|
MyMqttClient: aly_init: MqttUserString:regapp8aikalhdH42Z
MyMqttClient: aly_init: MqttPwdString:"11604109939253*da3fa823120752c 注册之后接入
MyMqttClient: run: Connect Success
/MainActivity: run: 权限被拒绝,下次不会在询问了
```



## 设备

ESP8266LUASDK

设备总数 5    激活设备 5    当前在线 2

设备列表    批次管理

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	状态/启用状态	最后上线时间
<input type="checkbox"/>	regapp	ESP8266LUASDK	设备	● 在线 <input checked="" type="checkbox"/>	2020/10/31 10
<input type="checkbox"/>	myGSM 作为GPRS或NB_IOT接入使用	ESP8266LUASDK	设备	● 在线 <input checked="" type="checkbox"/>	2020/10/31 10
<input type="checkbox"/>	Mini 作为微信小程序接入使用	ESP8266LUASDK	设备	● 离线 <input checked="" type="checkbox"/>	2020/10/30 20
<input type="checkbox"/>	Android 作为Android接入	ESP8266LUASDK	设备	● 离线 <input checked="" type="checkbox"/>	2020/10/31 09
<input type="checkbox"/>	ESP8266 给8266接入使用	ESP8266LUASDK	设备	● 离线 <input checked="" type="checkbox"/>	2020/10/29 17

删除    禁用    启用

## 源码说明

## 1. 注册设备我封装成了一个java文件

```

package com.example.yang.zlappalyreg;

import ...

public class RegisterDeviceForMqtt {
    private static final String TAG = RegisterDeviceForMqtt.class.getSimpleName();

    public static String ProductSecret = "XXXXXXXXXXXXXX"; // 替换自己的 ProductSecret
    public static String ProductKey = "XXXXXXXXXXXXXX"; // 替换自己的 ProductKey
    public static String Region = "cn-shanghai"; // 地域, 根据自己的修改
    public static int MqttPort = 1883; // 端口号

    public static String MqttIPString = ""; // IP地址
    public static String MqttClientId = ""; // MqttClientId
    public static String MqttUserString = ""; // 用户名
    public static String MqttPwdString = ""; // 密码

    private static RegisterDeviceForMqtt registerDeviceForMqtt;
    private static MqttConnectOptions mqttConnectOptions;
    private static MqttClient mqttClient;
    private OnCallback callback; // 接收信息回调(字符串)

    // 创建一个单任务线程池
    private ExecutorService SingleThreadExecutor = Executors.newSingleThreadExecutor();
    private RegisterDeviceForMqtt() { super(); }
    // 提供一个全局的静态方法
    public static RegisterDeviceForMqtt sharedCenter() {
        if (registerDeviceForMqtt == null) {
            synchronized (MyMqttClient.class) {
                if (registerDeviceForMqtt == null) {
                    registerDeviceForMqtt = new RegisterDeviceForMqtt();
                }
            }
        }
    }
}

```

## 2. 用户调用这个函数注册设备(形参为设备的名字)

```
com / Example / yang / Zlppayreg / RegisterDeviceOnmqtt
AndroidManifest.xml x RegisterDeviceForMqtt.java x ActivityRegDevice.java x Ma
138     });
139     } catch (Exception e) {
140     }
141 }
142
143 public void RegisterDevice(String DeviceName){
144     aly_init(DeviceName);
145     InitMqttOptions();
146     InitMqttConnect();
147
148     SingleThreadExecutor.execute(new Thread(new Runnable() {
149         @Override
150         public void run() {
151             try{
152                 mqttClient.connect(mqttConnectOptions);
153             }
154             catch (MqttException e) {
155                 Log.e(TAG, msg: "subscribe Err"+e);
156             }
157         }
158     }));
159 }
160
```

### 3.程序运行的时,如果没有注册过设备,则跳转到注册页面

```
AndroidManifest.xml x RegisterDeviceForMqtt.java x ActivityRegDevice.java x MainActivity.java x menu_main.xml x MyMqttClient.java
32 @Override
33 protected void onCreate(Bundle savedInstanceState) {
34     super.onCreate(savedInstanceState);
35     setContentView(R.layout.activity_main);
36
37     permissionHelper = new PermissionHelper( activity: this); //权限申请使用
38
39     sharedPreferencesHelper = new SharedPreferencesHelper( context: MainActivity.this, FILE_NAME: "MqttInfo")
40     /* 获取存储的 productKey,deviceName,clientId,deviceToken*/
41     String productKey = sharedPreferencesHelper.getSharedPreferences( key: "productKey", defaultObject: "").toString()
42     String deviceName = sharedPreferencesHelper.getSharedPreferences( key: "deviceName", defaultObject: "").toString()
43     String clientId = sharedPreferencesHelper.getSharedPreferences( key: "clientId", defaultObject: "").toString()
44     String deviceToken = sharedPreferencesHelper.getSharedPreferences( key: "deviceToken", defaultObject: "").toString()
45
46     /* 已经注册了设备*/
47     if (productKey.length()>0 && deviceName.length()>0 && clientId.length()>0 && deviceToken.length() >0){
48         Toast.makeText( context: MainActivity.this, text: "已经注册了设备",Toast.LENGTH_LONG).show();
49         Log.d(TAG, msg: "onCreate: "+productKey+","+deviceName+","+clientId+","+deviceToken);
50         MyMqttClient.sharedCenter().aly_init(clientId,deviceName,deviceToken);
51         MyMqttClient.sharedCenter().setConnect();//连接MQTT
52     }
53     else{
54         Toast.makeText( context: MainActivity.this, text: "请先注册设备,让此APP接入平台",Toast.LENGTH_LONG).show()
55         Intent intent = new Intent( packageContext: MainActivity.this,
56             ActivityRegDevice.class);
57         startActivity(intent);
58         finish();
59     }
60
61     Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar); //右上角菜单
62     setSupportActionBar(toolbar);
63 }
```

### 4.用户点击按钮,调用注册函数注册设备

```
src | main | java | com | example | yang | Learn8266ForLuaALY2 | ActivityRegDevice
MainActivity.java | MyMqttClient.java | strings.xml | RegisterDeviceForMqtt.java | activity_regdevice.xml | ActivityRegDevice.java | Example
39
40 sharedPreferencesHelper = new SharedPreferencesHelper( context: ActivityRegDevice.this, FILE_NAME: "MqttInfo");
41 /* 注册按钮点击 */
42 button1.setOnClickListener(new View.OnClickListener() {
43     @Override
44     public void onClick(View v) {
45         deviceName = editText_deviceName.getText().toString().replace( target: " ", replacement: "" ); // 获取用户输入的信息
46         if (deviceName.length() >= 4) { // 物联网平台规定至少4个字符
47             // 注册回调函数
48             RegisterDeviceForMqtt.sharedCenter().setOnCallback(new RegisterDeviceForMqtt.OnCallback() {
49                 @Override
50                 public void callback(int code, String Msg) {
51                     /* "clientId": "Ntou7r7rQD7fIR4LeZP000100", "productKey": "01m7erInJbQ", "deviceName": "11223344",
52                     if (code == 200) { Log.e(TAG, msg); "callback: "+Msg);
53                     try {
54                         JSONObject jsonObject = new JSONObject(Msg);
55                         String clientId = jsonObject.getString( name: "clientId");
56                         String productKey = jsonObject.getString( name: "productKey");
57                         String deviceName = jsonObject.getString( name: "deviceName");
58                         String deviceToken = jsonObject.getString( name: "deviceToken");
59
60                         sharedPreferencesHelper.put( key: "clientId", clientId);
61                         sharedPreferencesHelper.put( key: "productKey", productKey);
62                         sharedPreferencesHelper.put( key: "deviceName", deviceName);
63                         sharedPreferencesHelper.put( key: "deviceToken", deviceToken);
64                         /* 跳转到主页面 */
65                         Message msg = myHandler.obtainMessage();
66                         msg.what = 1;
67                         myHandler.sendMessage(msg);
68                     } catch (Exception e) {
69
70                     }
71                 }
72             });
73         }
74     });
75     /* 调用动态注册函数 */
76     RegisterDeviceForMqtt.sharedCenter().RegisterDevice(deviceName);
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

## 5.注册完成跳转到主页面

```
AndroidManifest.xml | RegisterDeviceForMqtt.java | ActivityRegDevice.java | MainActivity.java | menu_main.xml | MyH
54
55 try {
56     JSONObject jsonObject = new JSONObject(Msg);
57     String clientId = jsonObject.getString( name: "clientId");
58     String productKey = jsonObject.getString( name: "productKey");
59     String deviceName = jsonObject.getString( name: "deviceName");
60     String deviceToken = jsonObject.getString( name: "deviceToken");
61
62     sharedPreferencesHelper.put( key: "clientId", clientId);
63     sharedPreferencesHelper.put( key: "productKey", productKey);
64     sharedPreferencesHelper.put( key: "deviceName", deviceName);
65     sharedPreferencesHelper.put( key: "deviceToken", deviceToken);
66     /* 跳转到主页面 */
67     Message msg = myHandler.obtainMessage();
68     msg.what = 1;
69     myHandler.sendMessage(msg);
70 } catch (Exception e) {
71
72 }
73
74 });
75
76 /* 调用动态注册函数 */
77 RegisterDeviceForMqtt.sharedCenter().RegisterDevice(deviceName);
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

## 6.主页面如果判断有注册的信息执行连接MQTT程序

```
com example yang zlappalyreg MainActivity
AndroidManifest.xml RegisterDeviceForMqtt.java ActivityRegDevice.java MainActivity.java menu_main.xml MyMqttClient.java
40 /*获取存储的 productKey,deviceName,clientId,deviceToken*/
41 String productKey = sharedPreferencesHelper.getSharedPreferences( key: "productKey", defaultObject: "").toString()
42 String deviceName = sharedPreferencesHelper.getSharedPreferences( key: "deviceName", defaultObject: "").toString()
43 String clientId = sharedPreferencesHelper.getSharedPreferences( key: "clientId", defaultObject: "").toString()
44 String deviceToken = sharedPreferencesHelper.getSharedPreferences( key: "deviceToken", defaultObject: "").toString()
45
46 /*已经注册了设备*/
47 if (productKey.length()>0 && deviceName.length()>0 && clientId.length()>0 && deviceToken.length() >0){
48     Toast.makeText( context: MainActivity.this, text: "已经注册了设备",Toast.LENGTH_LONG).show();
49     Log.d(TAG, msg: "onCreate: "+productKey+","+deviceName+","+clientId+","+deviceToken);
50     //初始化 MQTT 参数
51     MyMqttClient.sharedCenter().aly_init(clientId,deviceName,deviceToken);
52     //连接MQTT
53     MyMqttClient.sharedCenter().setConnect();
54 }
55
56 else{
57     Toast.makeText( context: MainActivity.this, text: "请先注册设备,让此APP接入平台",Toast.LENGTH_LONG).show();
58     Intent intent = new Intent( packageContext: MainActivity.this,
59         ActivityRegDevice.class);
60     startActivity(intent);
61     finish();
62 }
63
64 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);//右上角菜单
65 setSupportActionBar(toolbar);
66 }
```

## 结语

之所以我把注册时的设备名称拿出来,是因为咱后面可以和手机验证码结合.

用户拿到手机APP之后用户需要用手机号登录注册,然后呢设备的名字就用手机号代替.

注意: 产品的 ProductSecret 和 ProductKey 别泄露了哈!!!!

如果让别人知道了之后,别人也可以在你的平台上动态注册设备并接入了!!!!

分类: [HC32F460\(华大单片机\)物联网开发](#)

好文要顶 关注我 收藏该文  

 杨奉武  
关注 - 1  
粉丝 - 693

0 0

« 上一篇: [002-HC32F460\(华大\)+Air724UG\(4G GPRS\)基本控制篇\(阿里云物联网平台\)-在阿里云物联网平台上一型一密动态注册设备\(HC32F460+ Air724UG\)](#)

posted on 2021-12-20 14:10 杨奉武 阅读(0) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑 预览

B    