

目录

目录	1
一、快速部署系统	2
快速部署，采用docker容器化的方式，拉取镜像，然后运行容器。制作好的docker镜像中包含JDK、Mysql、Redis、EMQX和Nginx服务。	2
1. 安装docker	2
2. 命令窗口运行如下命令(完成后端、前端、数据库、EMQX部署)：	2
3. 扫码下载APP安装,打开APP，跳过登录，配置服务端地址	3
4. 硬件代码烧录到设备	4
二、编译运行和部署系统	5
一、系统要求	5
二、开发工具	5
三、运行	5
四、后端部署	5
五、前端部署	5
六、Android项目打包	5
七、Nginx配置	5
三、APP安装和使用	7
扫码下载APP安装,打开APP，跳过登录，配置服务端地址	7
四、硬件接入系统	8
软硬件交互使用mqtt协议，OTA升级使用http协议。硬件接入系统，需要发布和订阅下面的主题。	8
订阅主题详情	9
发布主题详情	9
字段说明	11
{device_num}是设备编号，唯一标识符。例如：7CDFA1049ADA	11
五、相关开发板	13
如果你有制作好的设备，并且接入系统，可以拿出来分享和销售。作者不参与销售，只是把设备分享出来。需要的朋友自己斟酌购买，高电压设备使用时注意用电安全。	13
开发板	13

一、快速部署系统

快速部署，采用docker容器化的方式，拉取镜像，然后运行容器。制作好的docker镜像中包含JDK、Mysql、Redis、EMQX和Nginx服务。

1. 安装docker

Docker支持windows、linux、macos系统，根据自己电脑或者服务器在官网选择匹配的系统进行安装。安装完成后，命令窗口输入docker --version命令验证是否安装成功。参考文档：<https://docs.docker.com/engine/install/>

```
C:\Users\kerwin>docker --version
Docker version 20.10.7, build f0df350

C:\Users\kerwin>
```

2. 命令窗口运行如下命令(完成后端、前端、数据库、EMQX部署)：

```
docker run \
--name wumei-smart \
--publish 80:80 \
--publish 18083:18083 \
--publish 1883:1883 \
--publish 3306:3306 \
--publish 6379:6379 \
--restart always \
--detach \
registry.cn-chengdu.aliyuncs.com/kerwincui/wumei-smart:1.0
```

```
C:\Users\kerwin>docker run --name wumei-smart --publish 80:80 --publish 18083:18083 --publish 1883:1883 --publish 3306:3306 --publish 6379:6379 --restart always --detach kerwincui/wumei-smart:1.0
```

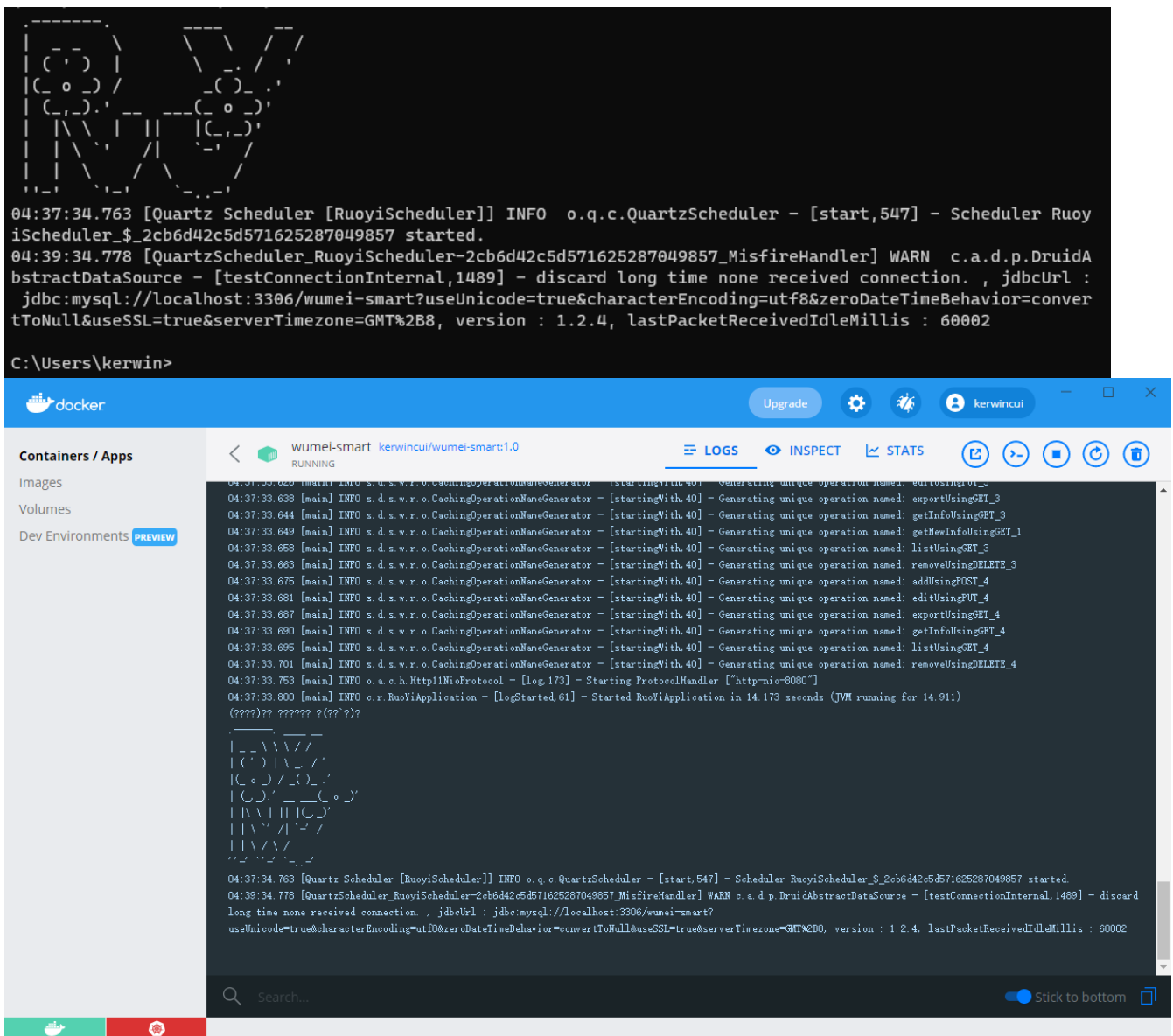
Windows命令窗口运行把 \ 换行符改为 ^。运行命令后，本地没有镜像，就会开始从镜像仓库拉取镜像。制作的docker镜像有500M左右，需要耐心等待下载。

命令中80是Nginx的端口，用于访问前端；18083端口用于访问EMQX控制台；1883端口用于访问EMQX的Borken；3306端口用于访问Mysql；6379端口用于访问Redis。根据需要自行修改容器映射的对应端口号。

命令运行成功后会返回docker容器的ID

```
C:\Users\kerwin>docker run --name wumei-smart --publish 80:80 --publish 18083:18083 --publish 1883:1883 --publish 3306:3306 --publish 6379:6379 --restart always --detach kerwincui/wumei-smart:1.0
2cb6d42c5d5797238195afb2072531ec02272ee4a38b821f416f91ecd3f0e47b
```

docker ps 命令可以查看运行的容器，容器刚启动需要等待20秒左右，里面的服务才能完全启动。 docker logs [containerID] 查看容器内服务是否完全启动（windows也可以通过docker界面的Logs选项查看）。看到RY的图标说明服务全部启动。浏览器输入安装docker的服务器IP就可以访问管理端，本地电脑localhost访问。 注意：容器长时间没有全部启动完成可能是mysql启动失败了，使用命令 **docker restart [containerID]** 重启容器。

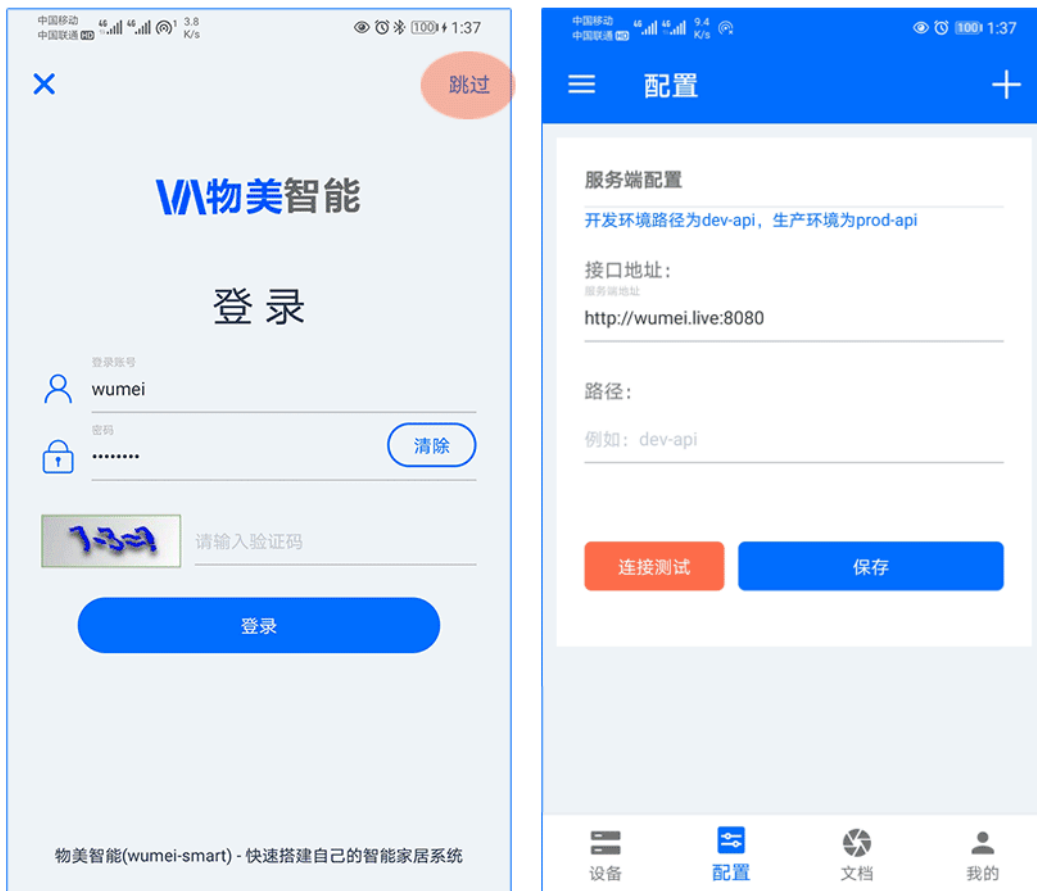


3. 扫码下载APP安装,打开APP,跳过登录,配置服务端地址



• 扫码下载APP

一、快速部署系统



- 配置APP

1. 登录界面可用测试账号登录（wumei admin123），也可以直接跳过登录。打开配置菜单，地址栏填写后端接口地址，后端端口默认为8080，如果前端代理访问，默认端口为80，开发环境路径为dev-api，生产环境为prod-api。确保外网或者局域网内可以访问接口（配置出入站规则，或者直接关闭防火墙）。

2. 点击[连接测试]按钮，测试成功后，单击[保存]按钮。然后关闭APP(不是退出App)，重新打开，这样就连接了服务端。输入默认账号密码登录（admin admin123）。

4. 硬件代码烧录到设备

- [Arduino接入Demo](#)
- [ESP-IDF接入Demo](#)

[物美智能官网](#)

二、编译运行和部署系统

一、系统要求

- JDK >= 1.8
- MySQL >= 5.7
- Maven >= 3.0
- Redis >= 3.0
- Node >= 10
- EMQX

二、开发工具

- 后端：IDEA、Eclipse
- 前端：Virtual Studio Code
- 安卓：Android Studio

三、运行

- 后端：1. Mysql新建wumei-smart数据库，导入spring-boot中的sql文件 2. 打开spring-boot项目，自动安装依赖 3. 编辑resources目录下的application-druid.yml，修改数据库配置信息 4. 编辑resources目录下的application.yml，修改redis、mqtt配置，默认账号admin admin123 5. 运行项目
- 前端：1. 安装依赖：执行 `npm install` 命令。强烈建议不要用直接使用 `cnpm` 安装，会有各种诡异的 bug，可以通过重新指定 `registry` 来解决 `npm` 安装速度慢的问题。 `npm install --registry=https://registry.npm.taobao.org`
- 2. 启动项目：执行 `npm run dev` 命令 3. 浏览器打开<http://localhost:80>访问。(默认账户/密码 admin/admin123)

四、后端部署

1. 在spring-boot项目的bin目录下执行package.bat打包Web工程，生成war/jar包文件。spring-boot/ruoyi-admin模块下target文件夹下包含war或jar文件
2. jar部署方式：使用命令执行：`java -jar ruoyi.jar` 或者执行脚本：`spring-boot/bin/run.bat`
3. war部署方式：spring-boot/ruoyi-admin/pom.xml中的packaging修改为war，放入tomcat服务器webapps

五、前端部署

1. 打包正式环境：`npm run build:prod`
2. 打包预发布环境：`npm run build:stage`
3. 构建打包成功之后，会在根目录生成 `dist` 文件夹，里面就是构建打包好的文件，通常是 `***.js`、`***.css`、`index.html` 等静态文件。

六、Android项目打包

1. 修改工程根目录的gradle.properties中的isNeedPackage=true。
2. 添加并配置keystore，在versions.gradle中修改app_release相关参数。
3. 如果考虑使用友盟统计的话，在local.properties中设置应用的友盟ID:APP_ID_UMENG。
4. 使用 `./gradlew clean assembleReleaseChannels`进行多渠道打包。

七、Nginx配置

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
```

二、编译运行和部署系统

```
server {
    listen 80;
    server_name localhost;

    location / {
        root /html/vue;
        try_files $uri $uri/ /index.html;
        index index.html index.htm;
    }

    location /prod-api/{
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
```

物美智能官网

三、APP安装和使用

扫码下载APP安装,打开APP,跳过登录,配置服务端地址



- 扫码下载APP



- 配置APP



1.登录界面可用测试账号登录 (wumei admin123) ,也可以直接跳过登录。打开配置菜单,地址栏填写后端接口地址,后端端口默认为8080,如果前端代理访问,默认端口为80,开发环境路径为dev-api,生产环境为prod-api。确保外网或者局域网内可以访问接口(配置出入站规则,或者直接关闭防火墙)。

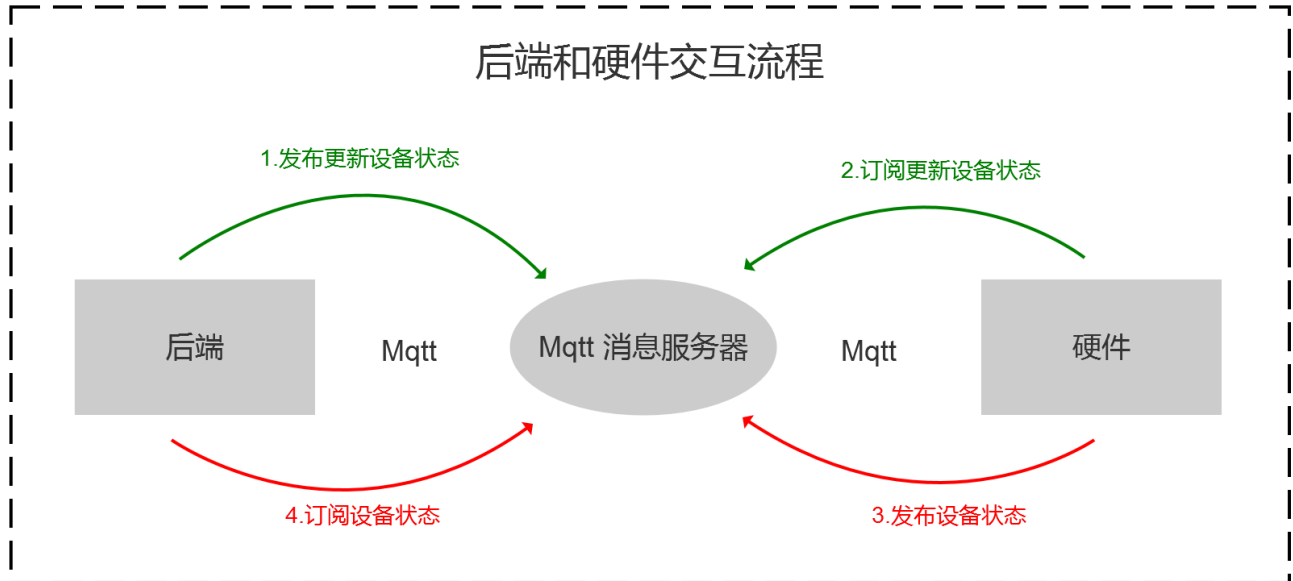
2.点击[连接测试]按钮,测试成功后,单击[保存]按钮。然后关闭APP(不是退出App),重新打开,这样就连接了服务端。输入默认账号密码登录(admin admin123)。

物美智能官网

四、硬件接入系统

软硬件交互使用mqtt协议，OTA升级使用http协议。硬件接入系统，需要发布和订阅下面的主题。

- 硬件订阅主题 1. status/set/{device_num} 更新设备状态 2. status/get/{device_num} 获取设备状态 3. setting/set/{device_num} 更新设备配置 4. setting/get/{device_num} 获取设备配置
- 硬件发布主题 1. device_info 设备信息 2. status 设备状态 3. setting 设备配置 4. offline 设备遗嘱
- 字段说明 1. 设备信息 2. 离线遗嘱 3. 设备状态 4. 设备配置



- [ESP-IDF接入示例 >>](#)
- [Arduino接入示例 >>](#)

订阅主题详情

1. 更新设备状态 订阅主题: status/set/7CDFA1049ADA 返回数据:

```
{
  "deviceNum": "7CDFA1049ADA",
  "relayStatus": 1,
  "lightStatus": 1,
  "isOnline": 1,
  "rssi": -73,
  "deviceTemperature": 0,
  "airTemperature": 0,
  "airHumidity": 0,
  "triggerSource": 0,
  "brightness": 11,
  "lightInterval": 432,
  "lightMode": 0,
  "fadeTime": 259,
  "red": 255,
  "green": 0,
  "blue": 0,
  "remark":""
}
```

2. 获取设备状态 订阅主题: status/get/7CDFA1049ADA 返回数据 : 空

3. 更新设备配置 订阅主题: setting/set/7CDFA1049ADA 返回数据 :

```
{
  "deviceNum": "7CDFA1049ADA",
  "isAlarm": 0,
  "isRadar": 0,
  "isHost": 1,
  "isRfControl": 1,
  "rfOneFunc": 1,
  "rfTwoFunc": 2,
  "rfThreeFunc": 3,
  "rfFourFunc": 4,
  "ownerId": "1",
  "isReset": 0,
  "isAp": 0,
  "isRfLearn": 0,
  "isRfClear": 0,
  "isSmartConfig": 0,
  "radarInterval": 5,
  "isWifiOffline": 0,
  "isOpenCertifi": 0,
  "remark":""
}
```

4. 获取设备配置 订阅主题: setting/get/7CDFA1049ADA 返回数据: 空

发布主题详情

1. 设备信息 发布主题: device_info 发布数据:

```
{
  "deviceNum": "7CDFA1049ADA",
  "categoryId": 2,
  "firmwareVersion": "1.0",
  "ownerId": "1"
}
```

2. 设备状态 发布主题: status 发布数据:

```
{
```

四、硬件接入系统

```
"deviceNum": "7CDFA1049ADA",
"relayStatus": 1,
"lightStatus": 1,
"isOnline": 1,
"rssi": -73,
"deviceTemperature": 0,
"airTemperature": 0,
"airHumidity": 0,
"triggerSource": 0,
"brightness": 11,
"lightInterval": 432,
"lightMode": 0,
"fadeTime": 259,
"red": 255,
"green": 0,
"blue": 0,
  "remark":""
}
```

3. 设备配置 发布主题: setting 发布数据:

```
{
"deviceNum": "7CDFA1049ADA",
"isAlarm": 0,
"isRadar": 0,
"isHost": 1,
"isRfControl": 1,
"rfOneFunc": 1,
"rfTwoFunc": 2,
"rfThreeFunc": 3,
"rfFourFunc": 4,
"ownerId": "1",
"isReset": 0,
"isAp": 0,
"isRfLearn": 0,
"isRfClear": 0,
"isSmartConfig": 0,
"radarInterval": 5,
"isWifiOffline": 0,
"isOpenCertifi": 0,
  "remark":""
}
```

4. 设备遗嘱 发布主题: offline 发布数据:

```
{
"deviceNum": "7CDFA1049ADA",
"isOnline": 0
}
```

字段说明

{device_num}是设备编号，唯一标识符。例如：7CDFA1049ADA

1. 设备信息

```
{
  "deviceNum": "7CDFA1049ADA", 设备编号,唯一标识
  "categoryId": 2,             设备分类, 后端数据 1-wifi通断器,2-智能灯,3-智能门锁,4-智能水阀,5-其他
  "firmwareVersion": "1.0",    固件版本
  "ownerId": "1"              用户ID
}
```

2. 离线遗嘱

```
{
  "deviceNum": "7CDFA1049ADA", 设备编号
  "isOnline": 0                 是否在线
}
```

3. 设备状态

```
{
  "deviceNum": "7CDFA1049ADA", 设备编号
  "relayStatus": 1,            继电器状态, 0-关, 1-开
  "lightStatus": 1,           灯状态, 0-关, 1-开
  "isOnline": 1,              是否在线, 0-关, 1-开
  "rssi": -73,                wifi信号强度 信号极好4格[-55—— 0], 信号好3格[-70—— -55], 信号一般2格[-85—— -70], 信号差1格[-100—— -85]
  "deviceTemperature": 0.0,   设备温度
  "airTemperature": 0.0,     空气温度
  "airHumidity": 0.0,        空气湿度
  "triggerSource": 0,         触发源, 后端字典 0-无、1-按键、2-手机、3-浏览器、4-射频遥控、5-雷达、6-报警、7-定时
  "brightness": 11,          灯亮度
  "lightInterval": 432,      灯渐变间隔
  "lightMode": 0,            灯模式, 后端字典 0-固定颜色 1-七彩渐变 2-七彩动感 3-单色渐变 4-白光 5-暖光
  "fadeTime": 259,          灯渐变时间
  "red": 255,                红灯值0-255, 用于三色混合成其他颜色
  "green": 0,                绿灯值0-255
  "blue": 0,                 蓝灯值0-255
  "remark": ""               备注, 可用于扩展信息
}
```

4. 设备配置

```
{
  "deviceNum": "7CDFA1049ADA", 设备编号
  "isAlarm": 0,              是否打开报警, 0-否, 1-是
  "isRadar": 0,              是否打开雷达, 0-否, 1-是
  "isHost": 1,               是否托管模式, 暂时用不到
  "isRfControl": 1,          是否启用射频遥控, 0-否, 1-是
  "rfOneFunc": 1,            射频遥控第一个键功能, 后端字典 0-无, 1-继电器通断, 2-开关灯, 开关雷达, 4-报警开关, 5-智能配网
  "rfTwoFunc": 2,            射频遥控第二个键功能, 后端字典 0-无, 1-继电器通断, 2-开关灯, 开关雷达, 4-报警开关, 5-智能配网
  "rfThreeFunc": 3,          射频遥控第三个键功能, 后端字典 0-无, 1-继电器通断, 2-开关灯, 开关雷达, 4-报警开关, 5-智能配网
  "rfFourFunc": 4,          射频遥控第四个键功能, 后端字典 0-无, 1-继电器通断, 2-开关灯, 开关雷达, 4-报警开关, 5-智能配网
  "ownerId": "1",           用户ID
  "isReset": 0,              是否重启, 0-否, 1-是
  "isAp": 0,                 是否打开AP配网, 0-否, 1-是
  "isRfLearn": 0,            是否打开射频遥控配对, 0-否, 1-是
  "isRfClear": 0,           是否清楚已匹配的遥控, 0-否, 1-是
  "isSmartConfig": 0,        是否打开智能配网, 0-否, 1-是
  "radarInterval": 5,        雷达感应间隔
  "isWifiOffline": 0,        是否启用离线模式, 0-否, 1-是
  "isOpenCertifi": 0,        是否使用证书加密, 0-否, 1-是
  "remark": ""               备注, 可用于扩展信息
}
```

四、硬件接入系统

物美智能官网

五、相关开发板

如果你有制作好的设备，并且接入系统，可以拿出来分享和销售。作者不参与销售，只是把设备分享出来。需要的朋友自己斟酌购买，高电压设备使用时注意用电安全。

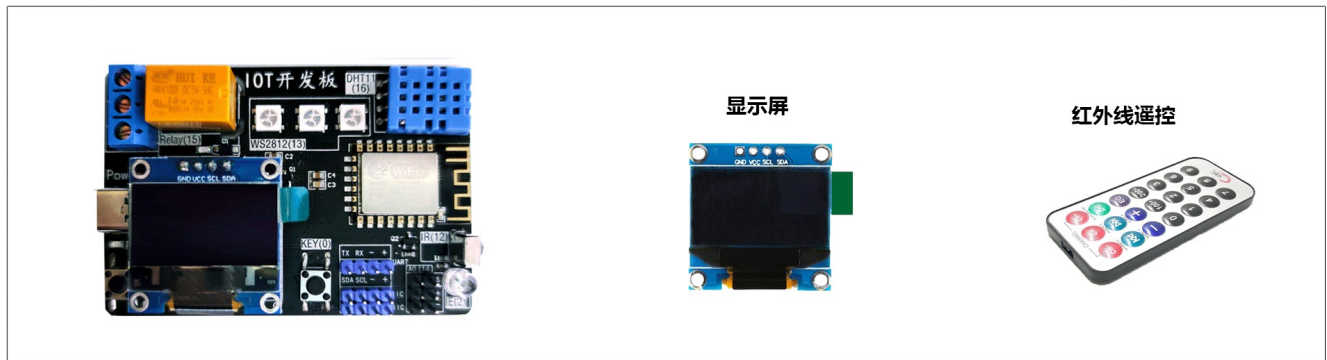
开发板

1. 智能开关 (wifi断路器)



- 介绍：手机、电脑远程控制、遥控配对、清码和控制、空气温湿度监控、雷达感应和报警、220V和5V电压供电，阻性负载2500W，感性负载250W
- 购买方式：小驿物联 QQ:2985672336 淘宝购买 >>

2. 物联网开发板



- 介绍：板载继电器，温湿度(DHT11),RGB(ws2812)1600万色，红外接收，红外发送，12864OLED显示，用户按键，引出两路IIC，一路串口，两个io口
- 购买方式：淘宝购买 >>

3. 物联网入门学习套件



- 介绍：从零入门，学习物联网开发。搭配多种传感器，教程和资料丰富。
- 购买方式：小驿物联 QQ：2985672336 淘宝购买 >>

