

IOTKIT 用户手册

(STM32 版本)



编制人	Jason	审核人	Eric	批准人	Helen
产品名称	IOTKIT	产品编号		文档编号	
会签日期		版本	V2.0		

修改记录

修改时间	修改记录	修改人	版本	备注
2015/08/15	初稿	Jason	V1.0	
2016/04/13		jason	V1.1	
2016-05-21		jason	V1.2	增加基础篇

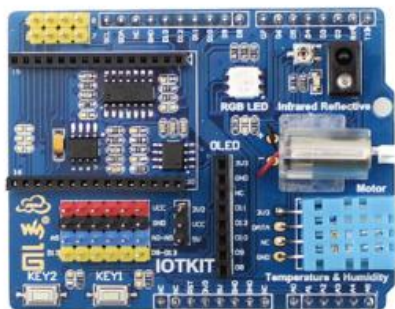
目录

一、	硬件介绍	4
1.1	汉枫 LPB100 模块介绍	5
1.2	乐鑫 ESP8266 模块	7
1.3	IOTKIT 功能板	8
1.4	IOTKIT 底板	9
二、	如何获取软件	11
三、	出厂程序演示	11
3.1	STM32 版本下载出厂程序	11
3.2	如何配置开发板连接到路由器上	13
3.3.1	使用 AriLink 配置	13
3.3.2	使用 SOFT AP 模式	14
3.3	局域网控制和远程控制	14
四、	基础篇	16
4.1	如何安装 KEIL 软件	16
4.2	如何安装 STLINK 驱动	19
4.3	获取出厂程序	21
五、	实例演示	23
5.1	空气质量检测仪	23
5.1.1.	传感器介绍	23
5.1.2.	云服务器后台设置	26
5.1.3.	硬件连接	30
5.1.4.	MCU 代码讲解	30
六、	资料更新	33

一、硬件介绍

这是一款物联网开发套件，提供 WIFI 接口，可兼容 marvell、高通、博通、TI、庆科、汉枫、乐鑫、微雪电子等主流 WIFI 模组，方便用户选择。开发板配套的 WIFI 模组默认烧写的是支持机智云平台的云服务的固件，只需要 MCU 用串口就可以跟 WIFI 模组进行通信，并且实现远程控制。

IOTKIT 采用 Arduino 标准接口设计，兼容 XNUCLEO、Arduino UNO R3、ST NUCLEO 等开发板。板载了常用传感器和一些通用的外设，还预留了 4Pin 传感器接口，方便用户外接更多的传感器模块。



IOTKIT开发版



高通4004



汉枫LPB100



庆科EMW3162



其他WIFI

1.1 汉枫 LPB100 模块介绍

www.yinerda.com

WIFI模块介绍



产品特点

- ☞ 支持802.11b/g/n无线标准
- ☞ 自主开发MCU平台，超高性价比
- ☞ 超低功耗，卓越省电机制，适于电池供电应用
- ☞ 支持UART/SPI/PWM/GPIO数据通讯接口
- ☞ 支持STA/AP/AP+STA工作模式
- ☞ 支持Smart Link智能联网功能（提供APP）
- ☞ 支持无线和远程升级固件
- ☞ 支持WPS功能
- ☞ 支持多TCP链路接入（支持5路）应用
- ☞ 支持外置或内置天线选项
- ☞ 3.3V单电源供电
- ☞ 超小尺寸：23.1mm x 32.8mm x3.45±0.3mm

1.2 乐鑫 ESP8266 模块

硬件

WIFI模块介绍

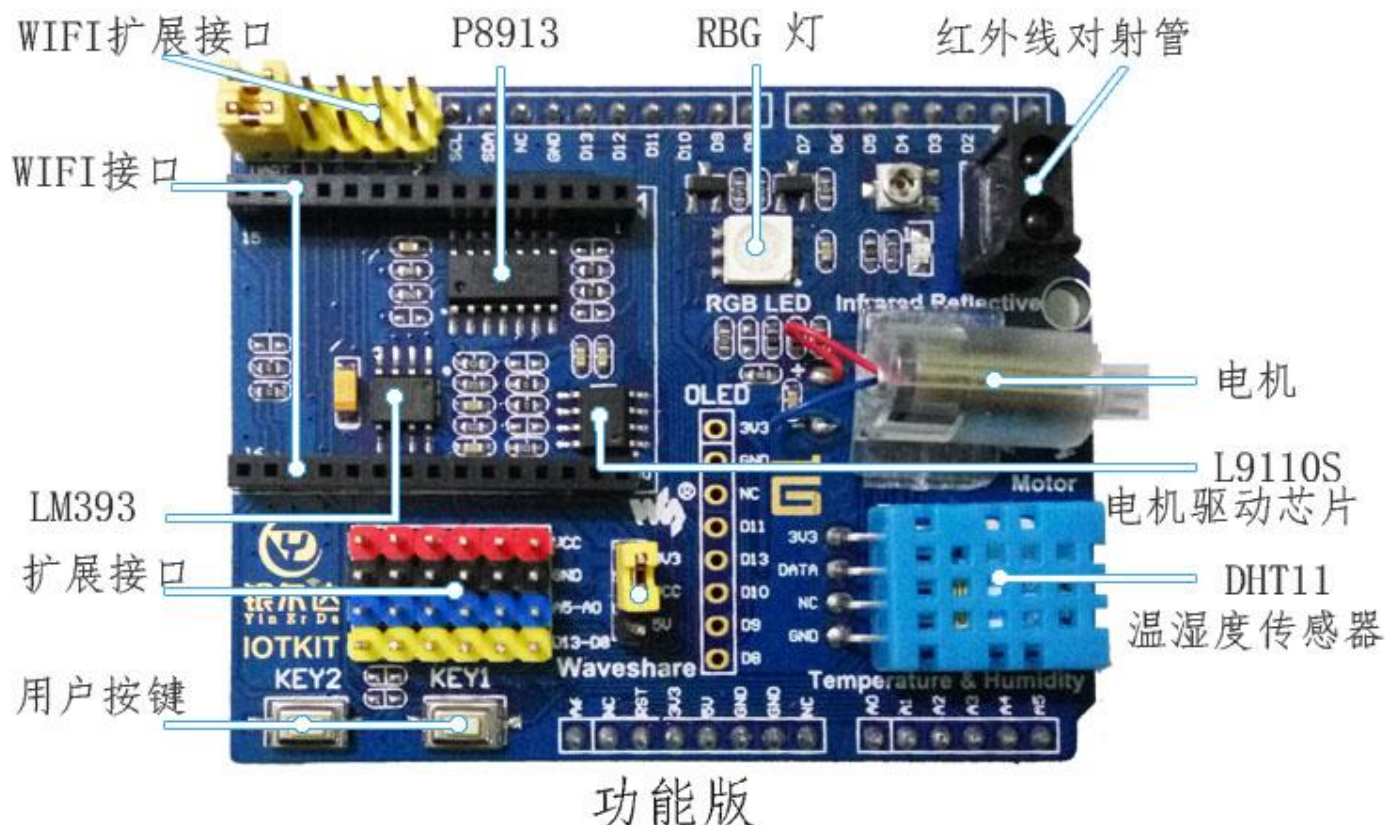


产品特点

- ④ 支持802.11b/g/n无线标准
- ④ ESP8266-12F内置32位微型MCU(Tensilica L106)
- ④ 超低功耗，带16位精简模式，主频支持80/160MHz
- ④ WiFi协议栈只用了20% MIPS，其他可做应用开发
- ④ 支持使用SPI接口的外置Flash，自带4MB Flash
- ④ 支持Smart Link智能联网功能（提供APP）
- ④ 支持无线和远程升级固件
- ④ 支持STA/AP/AP+STA工作模式
- ④ 睡眠电流10uA，2ms之内唤醒连接并传递数据包
- ④ 支持HSPI/PWM/IR/ADC/I2C/UART/I2S通讯接口
- ④ 3.3V单电源供电
- ④ 超小尺寸：16mm x 24mm x3±0.3mm

1.3 IOTKIT 功能板

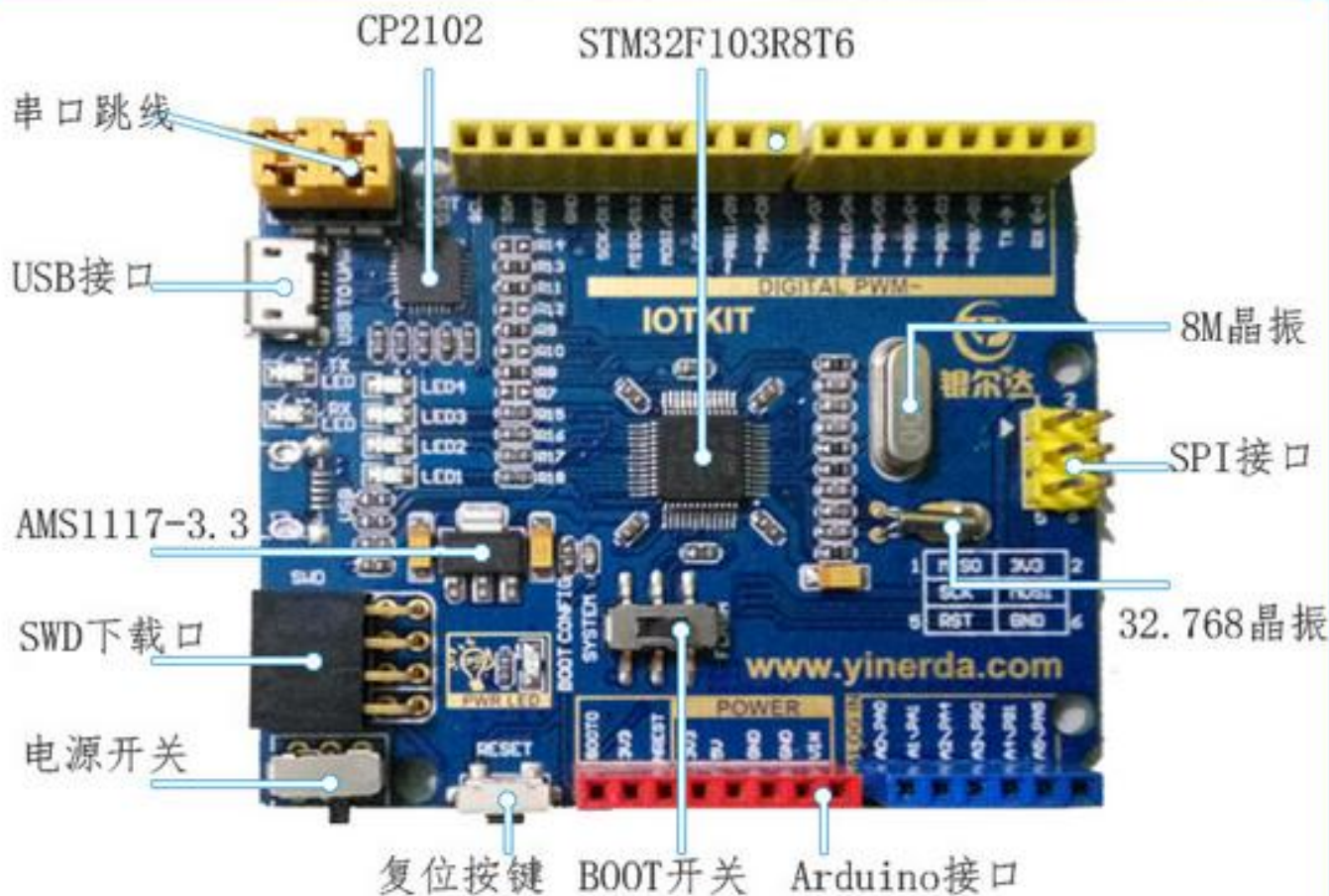
功能板介绍



- ④ WIFI扩展接口: 引出WIFI模块的串口和JTAG下载口。
- ④ WIFI接口: 方便接入多款WIFI模块, 如: LPB100, EMW3162, 乐鑫等
- ④ LM393: 红外对管驱动芯片。
- ④ 扩展接口: 引出Arduino上多余的IO口。
- ④ 用户按键: 方便用户自定义按键。
- ④ DHT11温湿度传感器: 可以实时检测当前环境温湿度。
- ④ L9110S电机驱动芯片: 驱动电机芯片。
- ④ 电机: 可以模拟自动窗帘, 抽水机, 风扇等设备工作。
- ④ 红外线对射管: 红外被遮挡, 就触发报警。
- ④ RGB 灯: 可以变换256*256*256种颜色。
- ④ P8913: 驱动RGB 芯片。

1.4 IOTKIT 底板

STM32底板介绍



STM32底板

STM32F103R8T6:

内核: ARM 32位的Cortex™-M3 CPU, 最高72MHz工作频率。

CP2102: 原装USB转串口芯片。

AMS1117-3.3: 5V转3.3V LDO。

串口跳线: 隔离USB转串口和MCU串口。

USB接口: USB供电接口和USB转串口。

SWD下载口方便STM32下载调试。

电源开关: 方便关闭和打开系统电源。

- ④ 复位按键：系统芯片复位。
- ④ BOOT开关：当使用USB接口下载，拨到SYSTEM, SWD下载拨到FLASH。
- ④ Arduino接口：我们引出标准Arduino接口，方便用户接入其他模块。
- ④ 32.768晶振：STM32实时时钟晶振。
- ④ SPI接口：预留SPI接口,方便二次开发。
- ④ 8M晶振：高速晶振。

二、 如何获取软件

APP

IOS和安卓APP获取



下载SDK

集成文档

DEMO APP下载



扫描二维码安装到手机



下载SDK

集成文档

DEMO APP下载




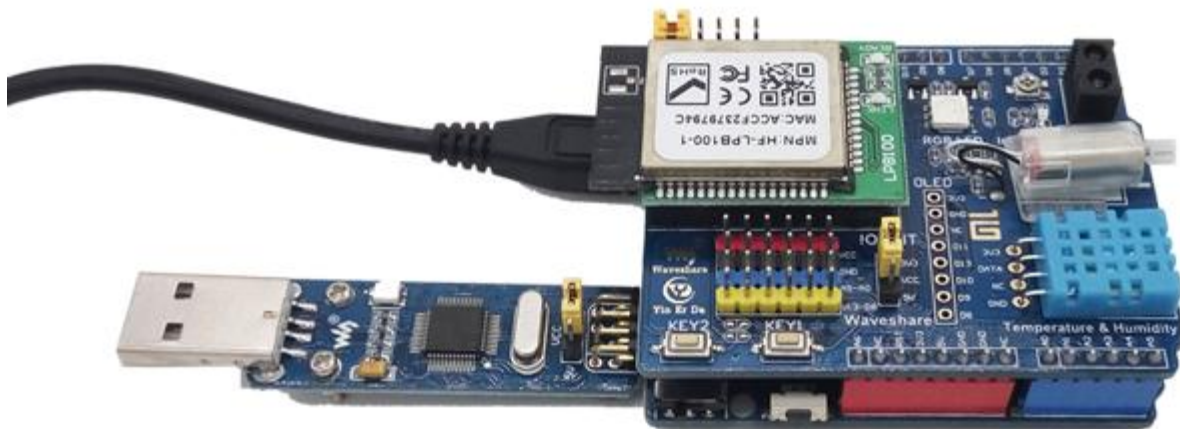
扫描二维码安装到手机

我们提供一套DEMO APP，源码全部开源，供大家免费使用，并且提供集成开发文档，让你快速完成APP的产品开发，

三、 出厂程序演示

3.1 STM32 版本下载出厂程序

 硬件环境：用 USB 给 GOKIT 供电，接上 ST-LINK 仿真器。如下图：



① 软件环境：WIN7 32, 64 位系统，安装 KEIL MDK 软件，需要安装好 STLINK 的驱动。

② 如何使用 KEIL 下载程序：

使用 KEIL MDK 打开我们提供的程序，点击①编译工程，点击②下载程序，点击③仿真。



3.2 如何配置开发板连接到路由器上

3.3.1 使用 AriLink 配置

- 用手机下载 IOE Demo APP，并且将手机连接到板子需要连接的路由器上。
- 打开手机 IOE Demo APP，点击添加设备，同时长按【KEY2】直到【GRB LED】亮绿灯，开启设备的 AirLink 网络配置模式。
- 在 AirLink 网络配置页面，选择相应的 WIFI 模块，在密码输入框中输入路由器的密码。
- 点击【配置】，直到软件弹出设备配置 OK。



第一步：开启设备的AirLink网络配置模式

GoKit设备开启方法：
如上图，开机后长按【KEY2】直到【RGB LED】亮绿灯（Arduino底板亮蓝灯）

其它设备按照自己开发过程设定的方式开启

下一步



3.3.2 使用 SOFT AP 模式

- ④ 短按【KEY2】，让设备进入soft AP模式，此时模块会生产一个名为热点

XPG-GAgent-xxx”密码：123456789

- ④ 打开 IOE Demo app，会进入Soft AP模式配置界面，点击其他，输入可用WiFi的SSID和密码，点击“配置”，当配置成功之后，APP会提示配置成功，模组会自动连接到路由器上。

3.3 局域网控制和远程控制

- ④ 返回设备列表，点击刷新，点击未绑定列表中的设备进行绑定，绑定之后就可以用手机控制板子了。
- ④ 手机关闭WIFI，连接3G或者4G网络，同样也是可以实现板子的远程控制。



四、基础篇

4.1 如何安装 KEIL 软件

➤ 获取软件:

在淘宝详情中找到产品资料下载, 我们提供板子的资料下载连接。

地址 1:

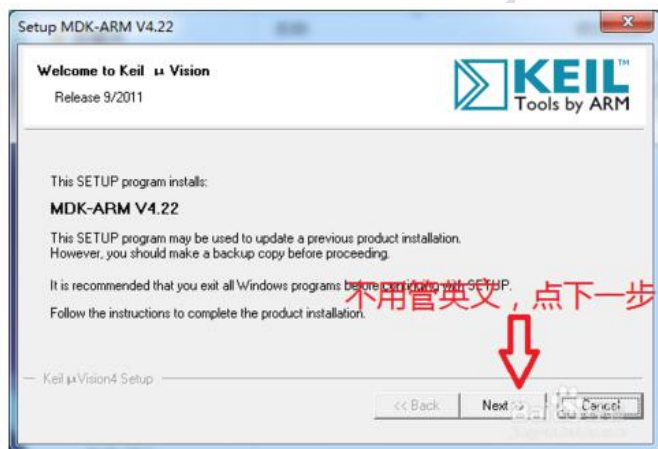
【360 网盘】<https://yunpan.cn/cPGexwQqjsswX> 访问密码 d41e

下载 KEIL 软件, 路径: 物联网 GoKit 开发板资料/软件工具/Keil4.7 版本软件

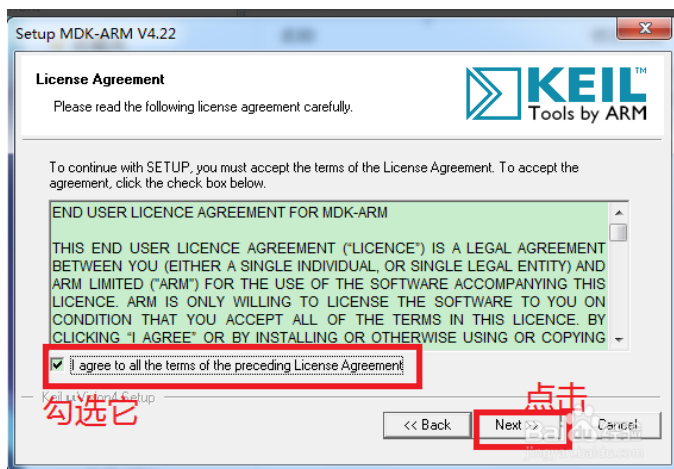


➤ 安装软件

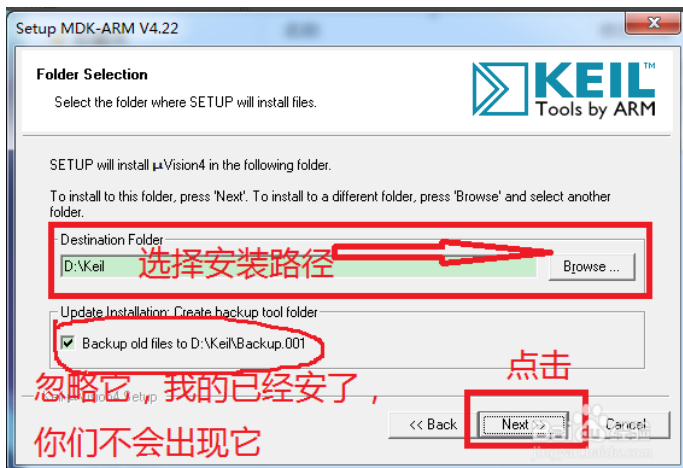
1, 运行 MDKV422.exe 可执行文件, 点击【NEXT】



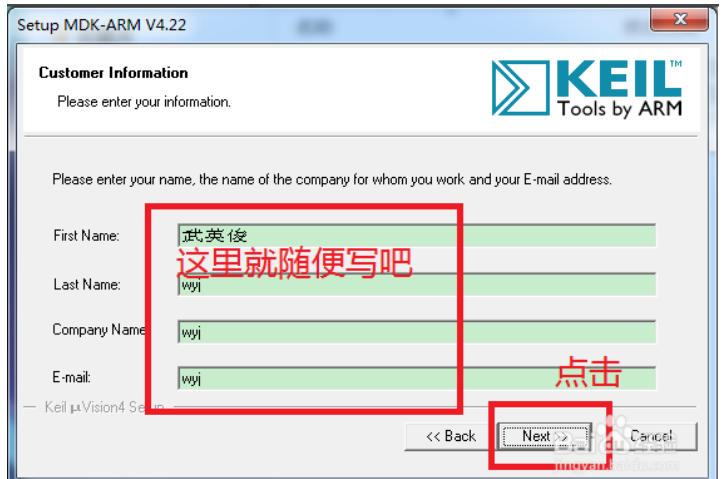
2, 勾选同意协议并点击 next:



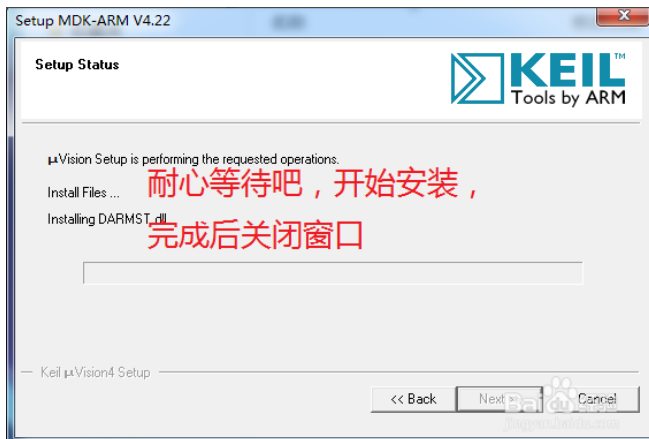
3, 选择自己的安装路径, 点击 next:



4, 自己随便填点字母吧:



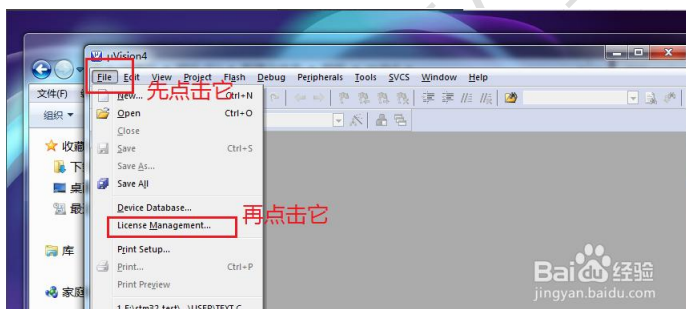
5, 安装中, 完成后关闭窗口



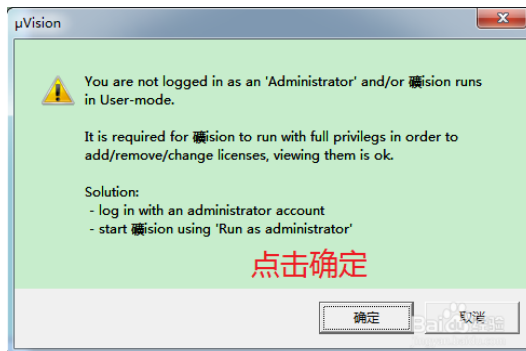
6, 激活软件, 找到 keil4, 右击并以管理员权限运行



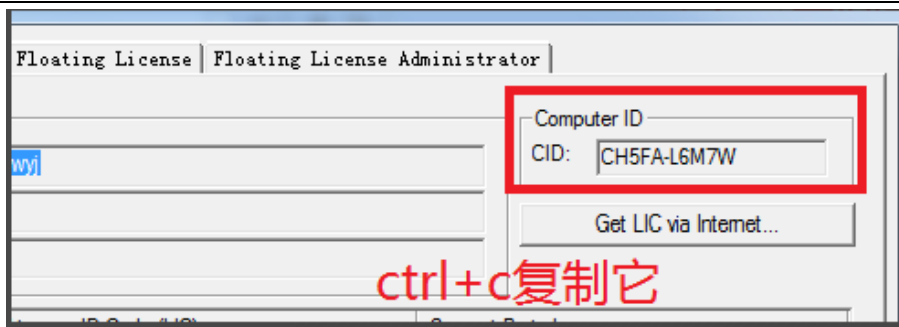
7, 点击 file, 找到 license management 并点击:



8, 点击 "确认"



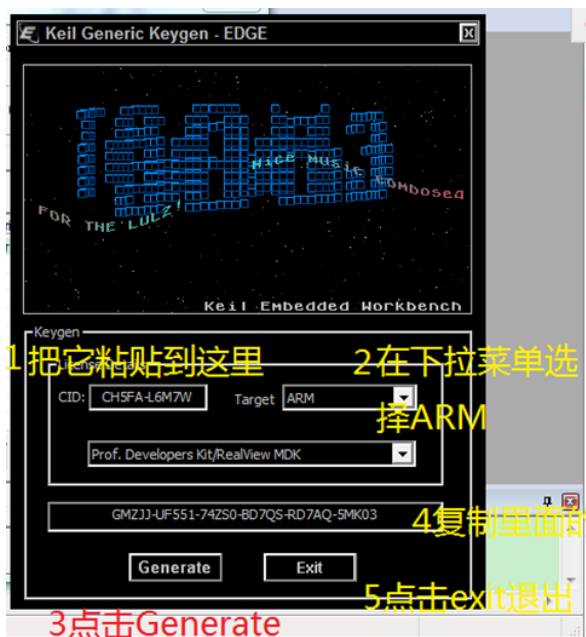
9, 复制自己的 CID:



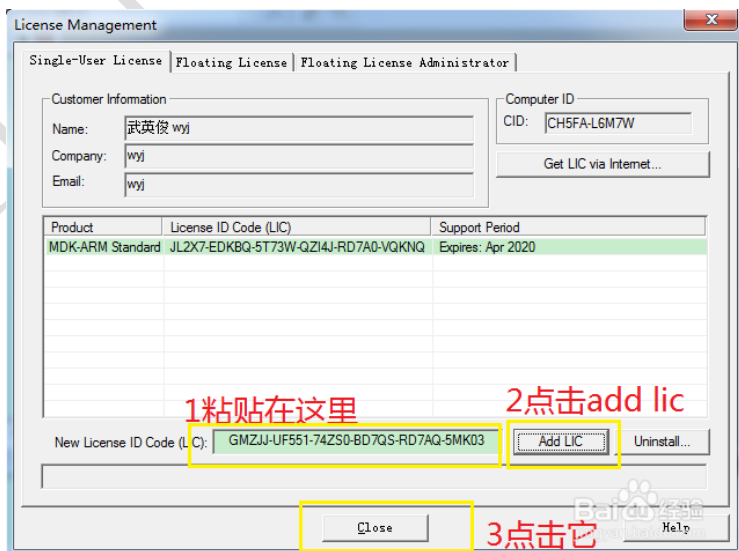
10, 找到注册机存放目录, 运行它:



11, 粘贴到图中位置, 在下拉长的中选择 ARM, 点击 Generate 生成代码, 并复制下来, 关闭注册机:



12, 粘贴在图中位置, 点击 add lic,gx 你已经激活它了:



4.2 如何安装 STLINK 驱动

➤ 演示环境

操作系统: window7(64 位)

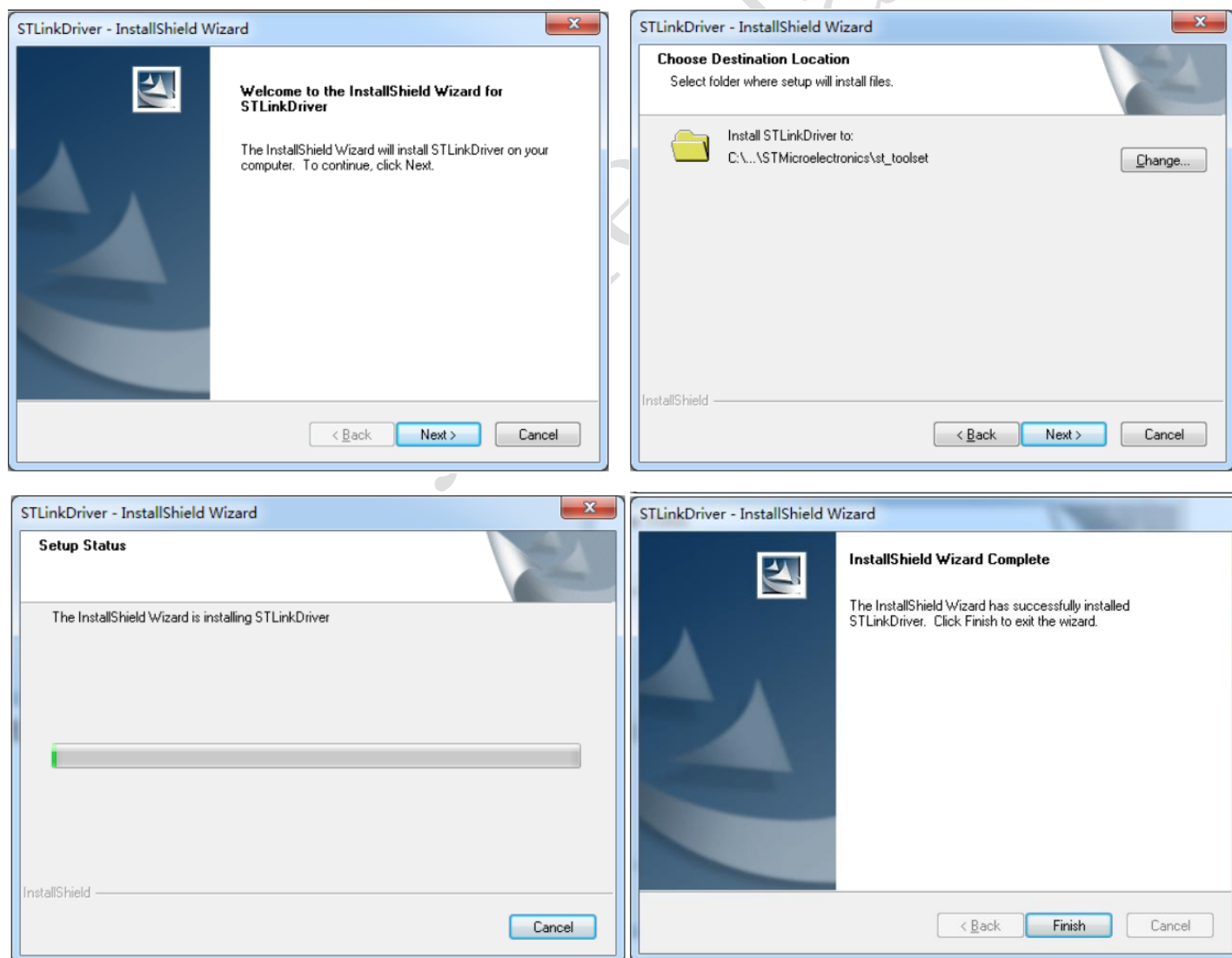
安装软件: st-link_v2_usbdriver.exe

硬件调试下载器: ST-Link v2

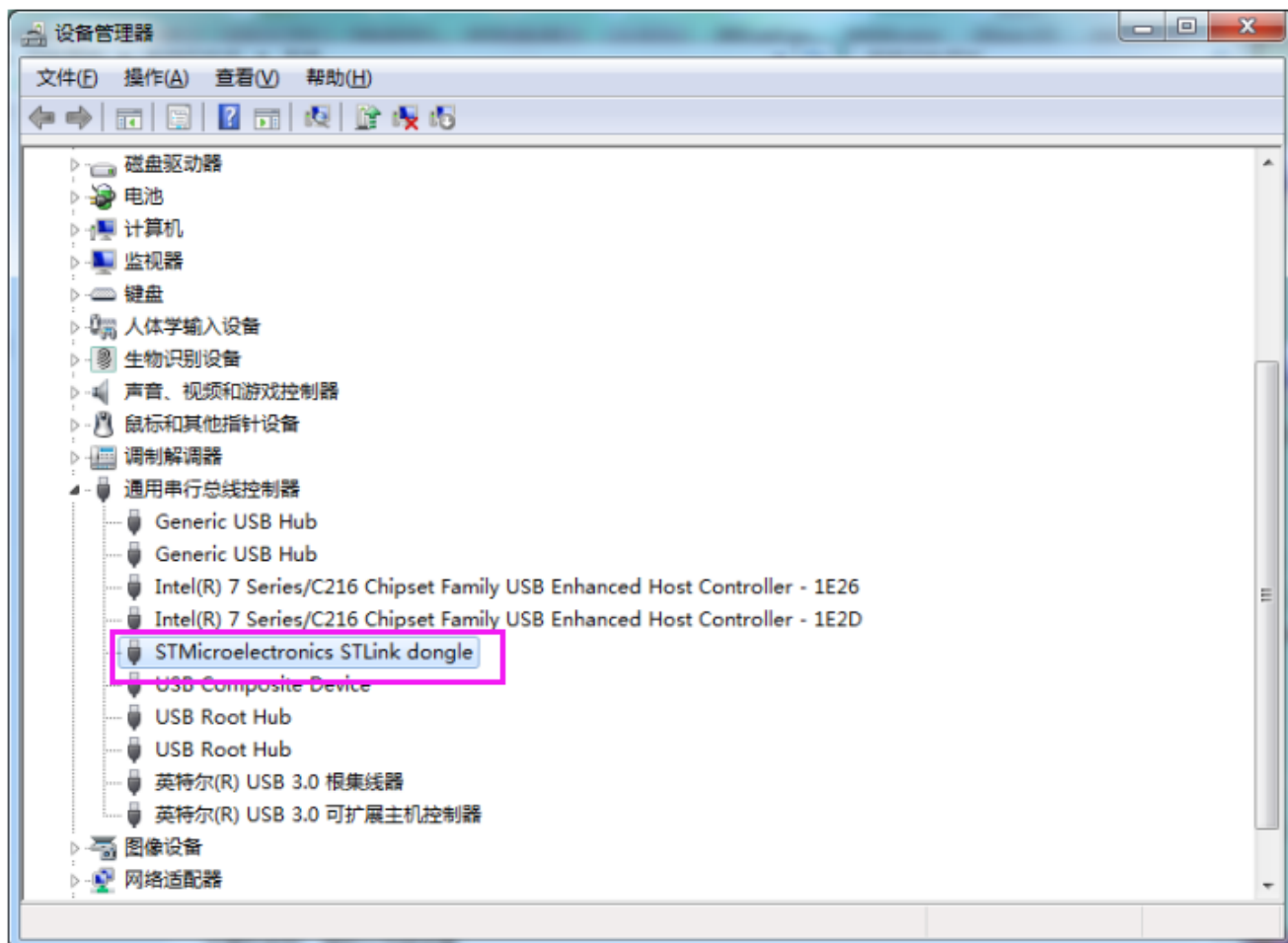
➤ 安装步骤

1, 双击要安装的 st-link_v2_usbdriver.exe 文件

2, 经过一系列安装后



在接入 ST-Link 时，会在设备管理器中的“常用串行总线管理器”中发现 STMicroelectronics STLink dongle。



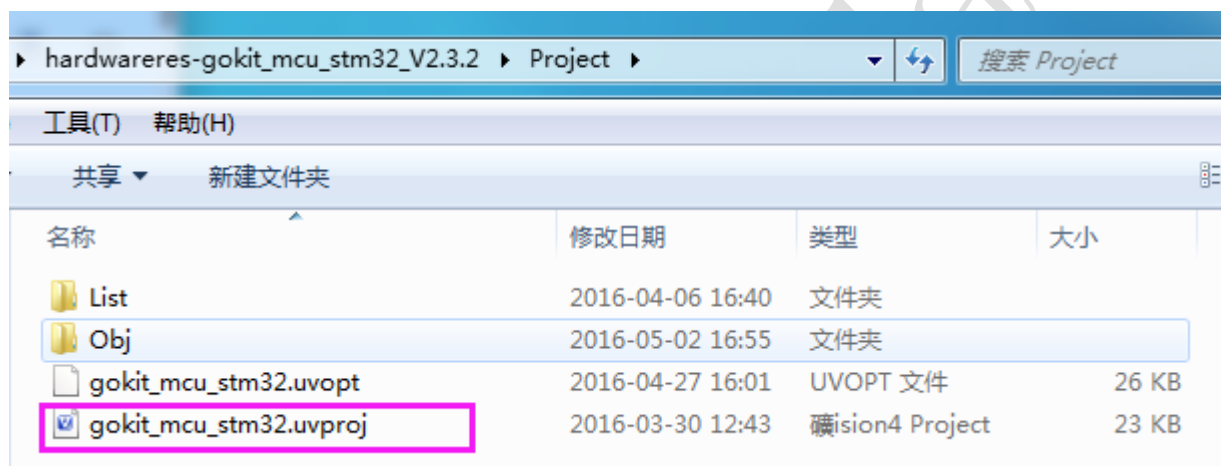
注意：如果正常安装驱动软件后没有在“常用串行总线管理器”中发现 STMicroelectronics STLink dongle，而是在别的项目发现新设备，ST-Link 是不能正常使用的，与你的系统环境有关，可以尝试换一个 USB 口试试，如果所有的 USB 口都不能使用，请重新卸载驱动软件，再次安装，再次尝试，如果这时还没有解决，请更换下电脑尝试，或重装系统。

4.3 获取出厂程序

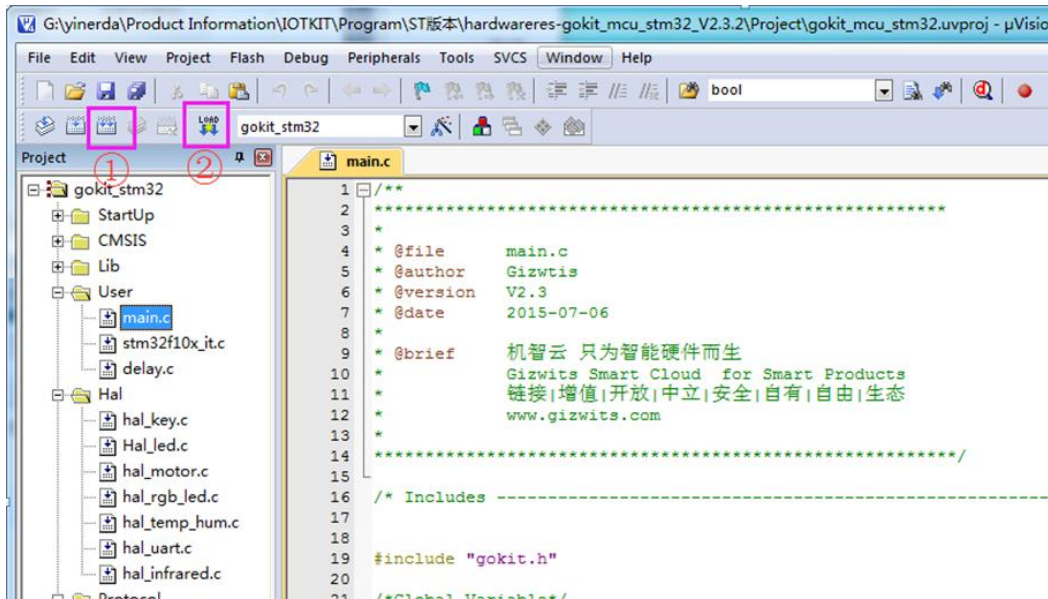
- 在网盘中下载出厂程序



- 解压之后找到找到目录找到 gokit_mcu_stm32.uvproj 文件，安装完 KEIL 之后会自动识别成工程文件：



- 双击打开工程，点击①编译工程，点击②可以直接通过 STLINK 下载程序（硬件连接在前面的出场程序下载中有介绍）

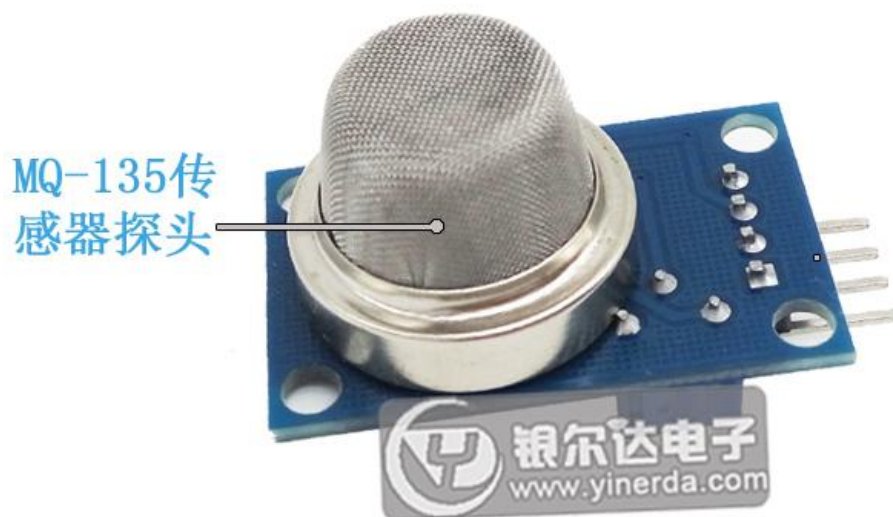


五、 实例演示

5.1 空气质量检测仪

5.1.1. 传感器介绍

产品介绍



特点

广泛的探测范围

高灵敏度/快速响应恢复

优异的稳定性/寿命长

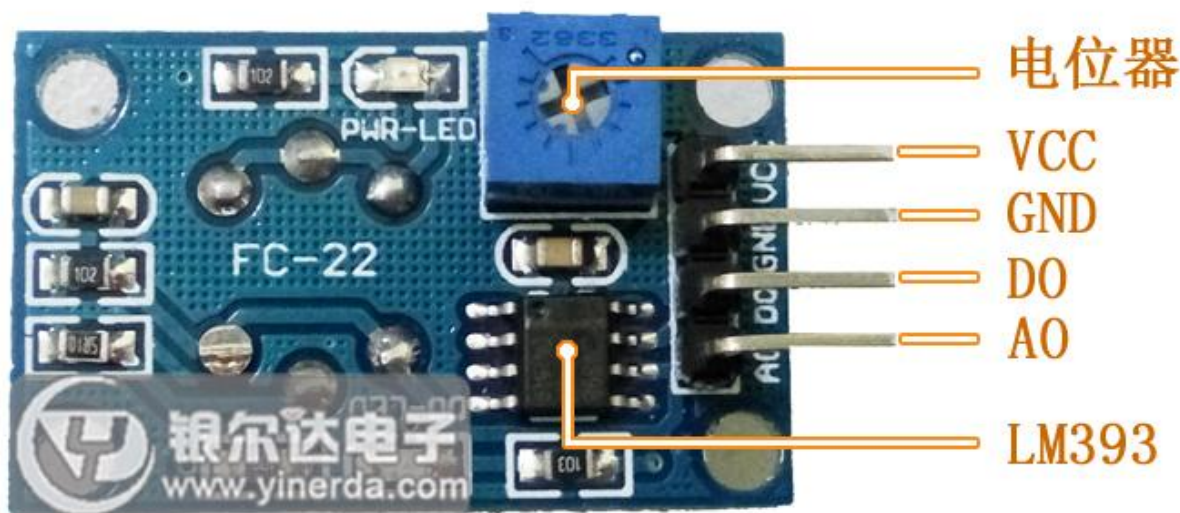
简单的驱动电路

应用

可用于家庭和工厂的气体泄漏检测装置, 适宜于硫化物, 苯系蒸汽, 烟雾等的探测。

www.yinerda.com

硬件解剖



- ④ 电位器: 调节DO输出的灵敏度
- ④ VCC: 5V供电 功耗 (电流150MA)
- ④ DO: TTL数字量0和1输出 (0.1和5V)
- ④ AO: 模拟电压输出 0.1-0.3V相对无污染, 最高浓度电压4V左右
- ④ LM393: 电压比较器, 实现DO输出灵敏度调节

【特别提醒】

传感器通电后, 需要预热20S左右, 测量的数据才稳定, 传感器发热属于正常现象, 因为内部有电热丝, 如果烫手就不正常了。

5.1.2. 云服务器后台设置

- 进入云服务器后台，创建新产品【空气质量检测仪】，添加如下数据点。

设备数据点

另存为模板

显示名称：污染报警 备注：无	标识名：DO_Sensor	读写类型：只读	数据类型：布尔值	<div>编辑</div> <div>删除</div>
显示名称：污染级别 数据范围：0 - 5 备注：无	标识名：AO_Sensor 分辨率：0.1	读写类型：只读 增量：0	数据类型：数值	<div>编辑</div> <div>删除</div>
显示名称：请求数据上报 备注：无	标识名：Refresh	读写类型：可写	数据类型：布尔值	<div>编辑</div> <div>删除</div>

- 在设备列表中找到当前设置的页面，点击进去可以看到如下页面。
- 1, 产品信息，介绍你产品的详细信息，其中要重点注意的是 **product key**，这一连串的数据在后面写 MCU 代码时需要用到。

空气质量检测仪

申请发布

技术支持: 400-6525-488 800099639 意见反馈

产品管理

产品信息

数据点

虚拟设备

运行状态

产品开发资源

服务

固件升级(OTA)

产测工具

查看统计分析

产品详细信息

产品名称: 空气质量检测仪

产品类型: 家电 (白电)

产品状态: 开发中

通信方式: WiFi

板卡: General Board

通信模块: General Chip

Product Key: ad854438daf24e4599aa8c26b3e48b23

产品密钥: 68d0*****75db 显示完整密钥

创建: 2016-04-18

更新: 2016-04-19

描述: 暂未添加

已绑定的应用 (0)

绑定应用

该设备未绑定任何应用, 请到 应用管理 创建新应用 (创建时系统会自动分配APP ID), 或者点击绑定应用绑定已创建的APP, 授权 APP控制设备。

2, 数据点: 设备数据点, 这个和你需要传输的数据相匹配。

3, 虚拟设备:

- 通过在线设备模拟器, 可以快速验证接口功能的开发。
- 登录状态下启动设备模拟器, 下载并安装 Demo App 或通过扫描模拟器二维码, 即可对模拟设备进行远程控制以及其他数据通信。

产品名称: 空气质量检测仪 did: ZttyRjHhyRLP3PbC27fbcH

DO_Sensor (布)

AO_Sensor (数)

扫码控制设备

通信日志

数据: 000000030600

数据: 000000030600

数据: 000000030600

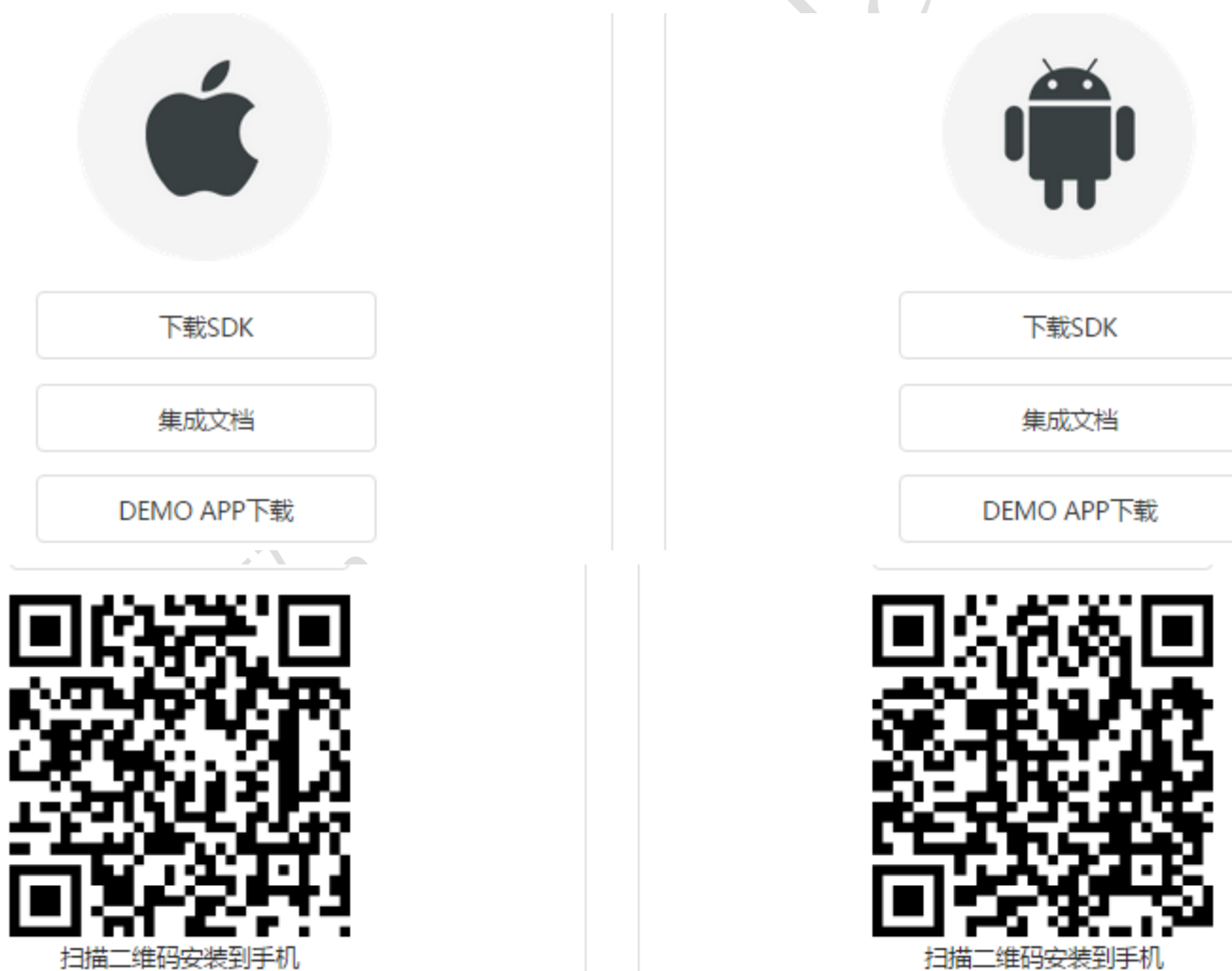
登录成功!

正在登录服务器...

打开手机 IOE DEMO APP, 点击功能列表中的扫一扫, 就可以在 APP 上添加这个虚拟设备。然后测试手机 APP 和云端的通信。



- 4, 运行状态: 显示了当前产品现在设备, 累计上线, 故障设备等一系列产品数据。
- 5, 产品开发资源: 提供 IOS, 安卓的 APP 的 SDK 的下载, 系统根据设置的数据点生成的协议文档, 以及机智云接入的 JSON 文档等资源。



MCU 开发资源

现在，你可以开始开发你的 MCU 程序，机智云根据你在上一步添加的数据点生成了 [《空气质量检测仪-机智云接入串口通信协议文档》](#)，你可以按提示完成你的 MCU 开发。

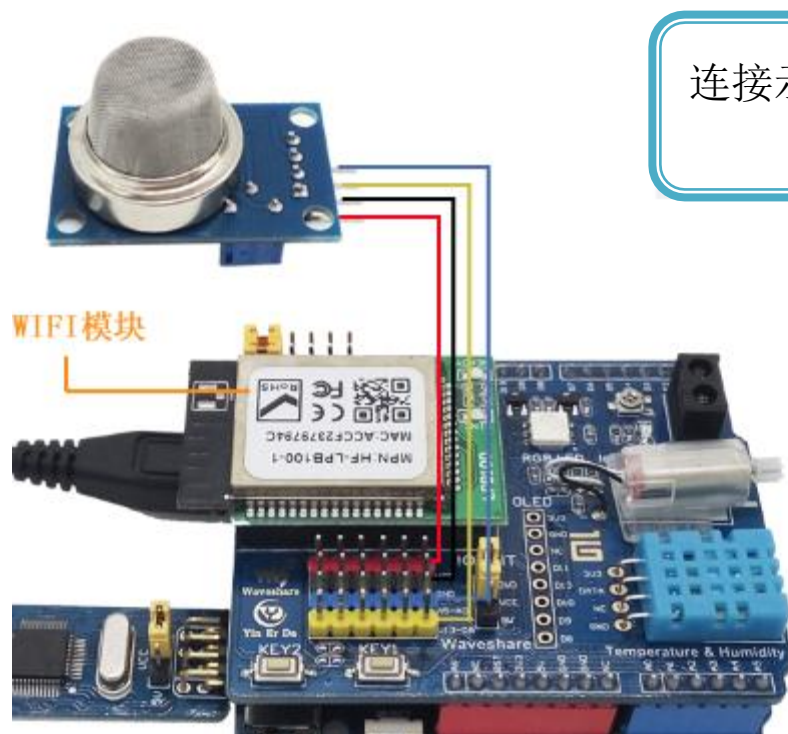
此外，机智云还自动生成了数据点的 [《空气质量检测仪-机智云接入JSON文档》](#)，此文档是对协议的格式化说明，包含每个数据点的ID、描述、数据类型、位置信息等，你可以根据json文档自动化生成MCU头文件，甚至是APP的页面，这将极大的简化你的开发过程。

轻应用开发资源

使用机智云提供的Open APIs（Http / WebSocket），可以快速开发网页或微信应用等基于html的轻应用，用于管理和控制智能设备。

[查看API文档](#)

5.1.3. 硬件连接



连接示意图

- 1, 将传感器用杜邦线连接到 GOKIT 的扩展接口上,

传感器管脚	GOKIT 扩展口
VCC	VCC(5V)
GND	GND
DO	PB6
AO	PA0

- 2, 用 USB 给 GOKIT 供电。
- 3, 使用 STLINK 给 GOKIT 下载程序
- 4, 接上下好机智云固件的 WIFI 模块。

5.1.4. MCU 代码讲解

- 1, 使用 KEIL MDK, 打开我们提供的 DEMO 程序。
- 2, 打开 Protocol.h 文件, 找到 PRODUCT_KEY 的宏定义, 这个就是设备的身份证, 要和云端设置保持一致。

```
#define PRO_VER "00000004"
#define PO_VER "00000004"
#define HARD_VER "02030100"
#define SOFT_VER "02030002"
#define PRODUCT_KEY "ad854438daf24e4599aa8c26b3e48b23"
```

备注:

在机智云后台【产品详细信息】可以找到这个 Product Key。

通信模块: General Chip

Product Key: ad854438daf24e4599aa8c26b3e48b23

产品密钥: 68d0*****75db 显示完整密钥

- 3, 打开 main.c, 找到 int main(void) 函数;

```

40 int main(void)
41 {
42     uint8_t p0_control_buf[MAX_P0_LEN];
43
44     SystemInit();           //stm32系统初始?
45     Delay_Init(72);
46     UARTx_Init();
47
48
49     HW_Init();              //硬件设备的初始化, 这里主要就是按键, 传感器用到的资源初始化。
50     Printf_SystemRccClocks(); //串口打印出当前MCU的时钟信息。
51     SW_Init();              //变量的初始化。
52     while(1)
53     {
54         KEY_Handle();        //按键处理函数
55
56         GizWits_MessageHandle(p0_control_buf, sizeof(WirteTypeDef_t)); //串口介绍到数据之后的解析
57         if(p0Flag == 1)
58         {
59             memcpy((uint8_t *)&WirteTypeDef, p0_control_buf, sizeof(WirteTypeDef_t)); //上报数据到云端
60             GizWits_ControlDeviceHandle();
61             gaterSensorFlag = 1;
62             printf("request DevStatusUpgrade \r\n");
63             Delay_ms(1000);
64             GizWits_DevStatusUpgrade((uint8_t *)&ReadTypeDef, 10*60*1000, 1);
65
66             p0Flag = 0;
67         }
68         if(gaterSensorFlag != 0) //达到定时检测时间, gaterSensorFlag=1
69         {
70             GizWits_GatherSensorData(); //读取传感器的数据
71             gaterSensorFlag = 0;        //清除标志位
72         }
73         GizWits_DevStatusUpgrade((uint8_t *)&ReadTypeDef, 10*60*1000, 0); //定时上报数据给云端。
74     }
75 }
76
    
```

讲解:

- GizWits_MessageHandle(); 手机 APP 推送指令下,WIFI 模块接口到之后, 通过串口转发给 MCU, MCU 接收到一条完整的指令之后都是在这个函数中完成解析的。
- 当 GizWits_MessageHandle()解析之后其中有控制指令的时候, 会把 P0Flag 置 1, 然后 MCU 的主函数检测到 P0Flag 置 1 后, 就通过 GizWits_ControlDeviceHandle 函数去处理控制设备。
- GizWits_DevStatusUpgrade()函数实现的是定时上报, 最后一个参数设置定时是否生效。
- 传感器数据读取函数 Hal_ReadAirMQxxData。

```
98 void Hal_ReadAirMQxxData(void)
99 {
100     float AD_value;
101
102     if(Read_GPIO_DO_Sensor())    //读取DO数据
103     {
104         ReadTypeDef.DO_Sensor = 0;
105     }
106     else
107     {
108         ReadTypeDef.DO_Sensor = 1;
109     }
110
111
112     AD_value = (ADC_ConvertedValue*1000/4096)*0.0033;    //采用DMA快速传输
113     ReadTypeDef.AO_Sensor = AD_value*10;
114
115 }
```

六、 资料更新

更多海量资料陆续在论坛更新中.....

1, 银尔达官方: www.yinerda.com

2, GOKIT 开发者讨论网站:

<http://club.gizwits.com/forum.php?mod=forumdisplay&fid=89>