

# 合肥师范学院

## 物联网工程综合实训

题目：基于 Arduino 的智能家居的开发

学    院    计算机学院

专业/班级    2019 物联网工程

学    号    1907431008

姓    名    陈一鸣

指导教师    

2022 年    12    月    27    日

---

## 摘 要

智能家居一直是物联网关注的热点之一，是物联网重要落地平台，智能家居的实现将极大地便利人们的日常生活，有着非常重要的现实意义。我们通过将物联网的应用落实到智能小屋架构上，可以直观地看到数据的采集、数据的迁移、云端处理数据以及整个控制逻辑的构架，本项目囊括了设备端数据的采集、数据上云、云端处理、基于物联网平台的 WEB 页面控制和显示以及构建手机 APP 控制的整个完整流程。

本项目搭建了一个物联网智慧小屋的模型，其中包括了一般的家居所需的装置，包含的传感器件有以下部件：室内温湿度传感器、室外温湿度传感器、PM2.5 传感器（使用 CO2 传感器替代）、可燃气体传感器、土壤湿度传感器、光敏传感器，包含的执行模块有：空调、换气扇、水泵、可调光以及调色的 LED 灯。传感信号是输入信号，由 Arduino 模块进行采集，数据采集后，由无线模块进行传输，在小屋实例中，我们采用了 WiFi 通过 AP 进行传输的方式，通过公网传送到云平台端，实现数据的转发、处理等。云端将控制信号发回给设备端执行。通过智慧小屋功能完整的演示，我们将传感器和执行机构之间的逻辑关系关联起来。目前设定的五个场景是：

1、室内的温度与空调关联，温度高了，开启空调制冷。平时我们开空调都是温度高了，自己用遥控器开启空调。在智慧小屋里，只需要给定相应的参数，就可以按照设定的参数，空调就可以自主运行，开启空调降温 and 去湿。

2、当可燃气体传感器超标时，风扇、蜂鸣器动作。

3、光敏电阻感知外界的光线的强弱，用以判定夜幕降临，在光线暗淡时，拉上窗帘、开启室内的灯光。

4、PM2.5 传感器感知数据超标时，将开启风扇（模拟空气净化器）换气。

5、检测植物的土壤的湿度，在偏低时启动水泵，进行浇灌作业。

物联网大致可以分成：感知层、网络层、平台层和应用层。我们可以直观看看到物联网的分层架构，其与智慧小屋里相对应的关系为：

1、感知层主要是负责通过传感器设备来识别和收集信息，我们通过 Arduino 采集不同接口形式的传感数据。

2、网络层负责安全的把这些信息进行传输，我们通过 AT 指令，将数据通过 WiFi 传送到物联网平台端。

3、平台层负责数据的鉴权、接入和转发，我们以阿里云为例，介绍接入云平台需要的鉴权方式、MQTT 协议。

---

4、应用层负责结合具体的应用需求，利用 IoT Studio 这些先进的可视化工具，将建立服务编排，详细介绍如何创立直观的 Web 显示、调度和控制的页面，并可方便地编制出手机控制的 APP，并在云平台上对数据进行计算、处理、挖掘来实现智能化的物联网应用。

**关键词：**智能家居；远程监控；Arduino；阿里云；物联网

---

## Abstract

Smart home has always been one of the hot spots of the Internet of Things, is an important landing platform of the Internet of things, the realization of smart home will greatly facilitate People's Daily life, has a very important practical significance. By leveraging the Internet of Things onto the smart cabin architecture, The framework of data collection, data migration, cloud processing and the whole control logic can be intuitively seen. This project covers the whole process of device data collection, data cloud, cloud processing, WEB page control and display based on the Internet of Things platform and the construction of mobile APP control.

This project builds a model of the smart house of the Internet of Things, which includes the devices required by the general household. The sensors include the following components: Indoor temperature and humidity sensor, outdoor temperature and humidity sensor, PM2.5 sensor (using CO2 sensor instead), combustible gas sensor, soil moisture sensor, and photosensitive sensor, including the implementation module of air conditioning, ventilation fan, water pump, dimmer and color adjustable LED light. Sensing signals are input signals that are collected by Arduino module and transmitted by wireless module after data collection. In the case of the cabin, WiFi is transmitted through AP and then transmitted to the cloud platform through the public network to realize data forwarding and processing. The cloud sends the control signal back to the device for execution. Through a complete demonstration of the function of the Smart House, we associate the logical relationship between the sensor and the actuator. The five scenarios set so far are:

- 1、The indoor temperature is associated with the air conditioning, the temperature is high, open the air conditioning refrigeration. Usually we open the air conditioning is the temperature is high, they use the remote control to open the air conditioning. In the smart room, only the corresponding parameters need to be given, and the air conditioner can operate independently according to the set parameters, and the air conditioner can be turned on to cool down and dehumidify.

---

2、When the combustible gas sensor exceeds the standard, the fan and buzzer will operate

3、The photosensitive resistor senses the strength of the light outside, to determine the fall of night, when the light is dim, close the curtain, open the indoor light

4、When the PM2.5 sensor perceives data exceeding the standard, the fan (simulated air purifier) will be opened for air exchange.

5、Detect the moisture of the soil of the plant, start the pump when it is low, and carry out the watering operation.

The Internet of Things can be roughly divided into: perception layer, network layer, platform layer and application layer. We can intuitively see the layered architecture of the Internet of Things, and its corresponding relationship with the smart house is as follows:

1. The sensing layer is mainly responsible for identifying and collecting information through sensor devices. We collect sensor data in different interface forms through Arduino.

2. The network layer is responsible for the safe transmission of these information. We transmit data to the Internet of Things platform through WiFi through AT instruction.

3. The platform layer is responsible for the authentication, access and forwarding of data. Taking Ali Cloud as an example, we introduce the authentication mode and MQTT protocol required for accessing the cloud platform.

4. The application layer is responsible for establishing service arrangement based on specific application requirements, using IoT Studio and other advanced visualization tools to introduce in detail how to create intuitive Web display, scheduling and control pages, and preparing mobile phone controlled apps conveniently. And on the cloud platform for data calculation, processing, mining to achieve intelligent Internet of Things application.

**Key Words:** Smart Home; Remote monitoring; Arduino;Ali Cloud;Internet of Things

---

## 目录

摘 要.....	2
Abstract.....	4
1smrat 小屋的整体框架 .....	8
1.1smart 小屋的传感器件与执行机构.....	8
1.2smart 小屋的硬件组成.....	9
1.3smrat 小屋系统架构.....	11
2 嵌入式篇 .....	13
2.1 嵌入式综述 .....	13
2.2 环境搭建 .....	13
2.3 软件烧写 .....	15
2.4 运行测试 .....	16
3 平台篇 .....	17
3.1 物联网云平台简介 .....	17
3.2 物模型 .....	18
3.3MQTT 协议 .....	19
3.4 设备基于 MQTT 接入云的认证方式.....	20
4smart 小屋的软件实现 .....	21
4.1 代码综述 .....	21
4.2 传感器数据采集与逻辑处理.....	22
4.3 阿里云配置 .....	24
5 总结 .....	35
参考文献.....	36

---

## 1smrat 小屋的整体框架

### 1.1smart 小屋的传感器件与执行机构

本项目将物联网总体抽象转化为一个具体的实现，包括硬件的搭建、数据的传感、数据的发送、云端的处理、控制指令回传等功能。

其中传感器件与执行机构如下图 1 所示。

传感器件		执行机构	
序号	名称	序号	名称
1	室内温湿度传感器	1	空调
2	土壤湿度传感器	2	换气扇
3	可燃气体传感器	3	蜂鸣器
4	PM2.5 传感器	4	水泵
5	光敏传感器	5	步进电机（窗帘）
		6	LED 灯

图 1

温湿度传感器两个，其一监测室外的温度，室外温度的传感器处于显示状态，不采集该数据。其二监控室内的温度。这个数据将被 Arduino 采集，并将在室内温度过高的时候提示空调制冷。

PM2.5 传感器监测空气的质量，能够在 PM2.5 数值超过一定的阈值时，开启换气扇通风换气。

可燃气体传感器监控室内的可燃气体的数量，监控燃气是否泄露，在超过一定阈值时，蜂鸣器报警，并将报警事件上报云端，开启换气扇通风换气。

土壤湿度传感器检测土壤的湿度状态。土壤的湿度数值过小时，水泵启动，对植物进行灌溉作业。

光敏传感器检测外界光的强度，检测结果用于决定窗帘电机开启或者是 LED 灯的开关。

空调采用半导体制冷器实现智慧小屋的空调，运行功率大约 70 瓦。

换气扇和 PM2.5 以及可燃气体报警相关，用于换气。

蜂鸣器除了报警功能外，还另外设定了相关的含义：响一声表示上电，响两声表示联网，响三声表示连接到云端成功，成功后才能正常执行传感器逻辑。

水泵在土壤的湿度数值过小时，用于对植物的灌溉。

窗帘电机采用步进电机来控制窗帘的开合，可以按照需求控制开启的幅度。

LED 灯拥有三路独立的红、绿、蓝 LED 灯，可以通过 PWM 调制的方式对每路 LED 灯进行调光以及合成不同色彩。



## 1.2smart 小屋的硬件组成

### 1.Arduino 介绍

选用 Arduino 作为控制主体，负责传感数据的采集、数据的传输、控制指令的下发等功能。采用 Arduino 作为控制平台的原因有普及度高、成本低，扩展版资源丰富、软件易上手学习等。

Arduino 是一款使用简单，集硬件、软件环境于一身的开源开发平台，旨在为智能硬件爱好者，交互艺术设计师以及电子软件工程师，提供简单易用的开发体验。Arduino 包括一个硬件平台——Arduino Board，和一个开发工具——Arduino IDE。两者都是开源的，既可以获得 Arduino 开发板的电路图，也可以获得 Arduino IDE 的源代码。Arduino 的另一个优势是可跨平台，Arduino IDE 能够在主流平台上运行，包括 Microsoft Windows，Linux，Mac OS X。

小屋的硬件分为两部分，亚克力板外壳以及演示用的虚拟家居设备，以及 Arduino 平台为基础的电子器件。系统电路连接图如图 2 所示。

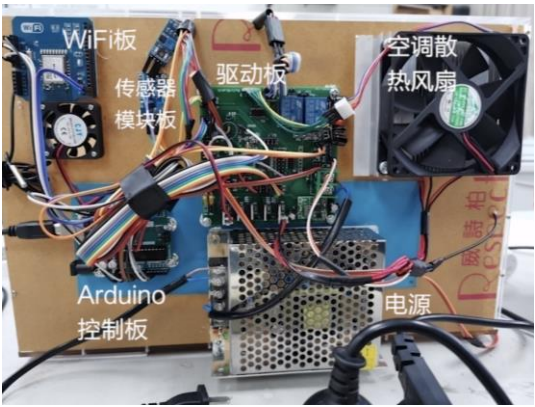


图 2

智慧小屋基于 Arduino 嵌入式开发平台的系统构成如图 3 所示：

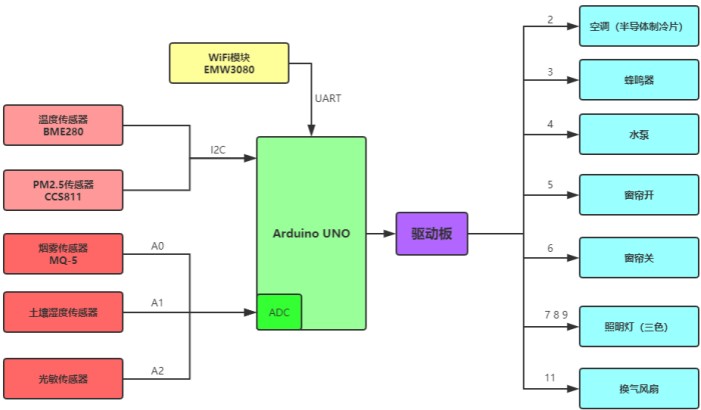


图 3 智慧小屋的硬件系统构成

---

如图所示为智慧小屋的硬件架构图，大致可以分为四个部分，左侧红色的为传感器，右侧蓝色的为执行器，上方黄色的 WiFi 通讯模块和中间绿色的 Arduino UNO 开发板。

右侧执行器中的水泵、空调、照明灯和换气风扇属于功率器件，需要大电流的驱动，Arduino 开发板显然无法提供这么高的功率。驱动板的设计与加入就是为了解决这个问题。

## 2. 传感器介绍

BME280 是一款环境传感器，可感知环境温度、湿度和大气压强，支持 I2C 和 SPI 接口，兼容 3.3V/5V 电平。尺寸小，功耗低、精度高且稳定性好，适用于环境监测、天气预测、海拔高度监测和物联网等应用场景。

CCS811 是一种数字气体传感器，集成了 CCS811 传感器和 8 位 MCU（带模数转换器(ADC)），用来检测室内的空气质量，包括二氧化碳(CO<sub>2</sub>)和广泛的挥发性有机化合物气体(VOCs)。支持 I2C 接口，工作电压为 1.8V~3.6V。

MQ-5 是一款可燃气体传感器，它所使用的气敏材料是在清洁空气中电导率较低的二氧化锡(SnO<sub>2</sub>)。当传感器所处环境中存在可燃气体时，传感器的电导率随空气中可燃气体浓度的增加而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。MQ-5 传感器对液化气、丙烷、氢气的灵敏度高，对天然气和其它可燃蒸汽的检测也很理想。这种传感器可检测多种可燃性气体，是一款适合多种应用的低成本传感器。

土壤湿度传感器是判断土壤中水分含量的多少来判定土壤的湿度大小。如图所示，当土壤湿度传感器探头悬空时，三极管基极处于开路状态，三极管截止输出为 0；当插入土壤中时由于土壤中水分含量不同，土壤的电阻值就不同，三极管的基极就提供了大小变化的导通电流，三极管集电极到发射极的导通电流受到基极控制，经过发射极的下拉电阻后转换成电压。

## 3. WiFi 通信模块

无线通信模块 WiFi 模块包括 WiFi 协议，WiFi 应用和 WiFi 模组硬件电路。

本项目使用 WiFi 通信模块主要有三个原因：其一，无线电波的覆盖范围广。可以实现较大区域的覆盖。其二，传输速度比较快，符合个人和社会信息化的需求。其三，无须布线。

WiFi 模块内置无线网络协议 IEEE802.11b.g.n 协议栈以及 TCP/IP 协议栈。传统的硬件设备嵌入 WiFi 模块可以直接利用 WiFi 联入互联网，是实现无线智能家居、M2M 等物联网应用的重要组成部分。

EMW3080 是上海庆科(MXCHIP)推出的嵌入式 WiFi 模块，高度集成 ARM CM4F，WLAN MAC/Baseband/RF，最高主频 133MHz，内置 256KB SRAM，2M FLASH，3.3V 单电源供电。

#### 4. 硬件连线

本节将阐述硬件接线示意图以及注意事项。

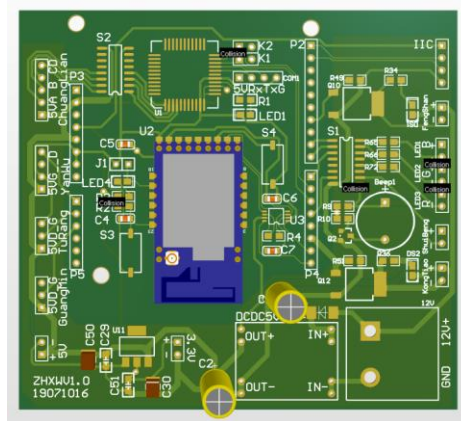


图 4 小屋 PCB 设计三维预览图

图 4 所示为小屋的 PCB 设计三维预览图。中间蓝色的部分为集成了天线的 WiFi 模块，WiFi 模块两侧丝印 P2、P3、P4、P5 为适配了 Arduino UNO 开发板和 Arduino Mega 2560 开发板的排针。驱动板周围的一圈为驱动板对外的各种接口。

驱动板使用 12V 电源提供供电，供电电源最大电流不低于 3A。



图 5 驱动板供电端口

图 5 所示为正确安装驱动板的示例，Arduino UNO 开发板通过排插连接到驱动板后无需额外供电，右下角红框内为驱动板供电端口，12V 与 GND 务必正确连接（在驱动板未与 UNO 板进行连接时单独给开发板供电，驱动板上的 LED 和 LED4 会亮，蜂鸣器会响，当连上 UNO 板并且初始化成功后，蜂鸣器会停止鸣响）。

### 1.3smrat 小屋系统架构

物联网技术体系架构包括应用层、平台层、网络层、感知层。智慧小屋也采

用了这种架构，应用层与之对应的是各种 WEB、APP，平台层即物联网云平台，网络层采用 WiFi 技术，感知层即用到的各种传感器。可以用一张图形象的加以表示。



图 6

智慧小屋信息流的流动过程如下图所示



图 7 智慧小屋信息流的流动过程

本项目也包含五个场景：

场景一、光敏感电阻和窗帘、灯光的关系

场景二、土壤湿度传感器和水泵的关系

场景三、PM2.5 和风扇的关系

场景四、可燃气体传感器和风扇、蜂鸣器的关系

场景五、温度传感器和空调的关系

## 2 嵌入式篇

### 2.1 嵌入式综述

Arduino Mega 2560 是基于 ATmega2560 的主控开发板。板载资源如图所示：

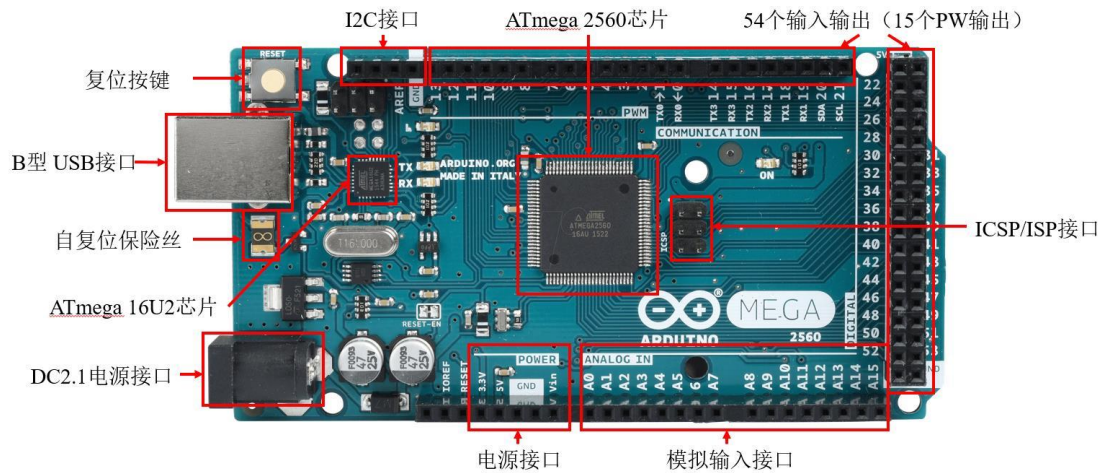


图 8

开发板可以通过 USB-Type-B 进行供电，也可以通过 DC 插头进行供电。若使用 DC 插头进行供电，请务必确认电源接口的正负极为内正外负，电压在 7~12V 之间均可，越接近 7V 越好。本项目通过 USB 接口将开发板与电脑连接，既满足了供电需求也保证了数字通信。

### 2.2 环境搭建

#### 1. 下载 IDE

访问 <https://www.arduino.cc/en/Main/Software>，下载 1.8.16 的免安装包，直接解压文件。

名称	修改日期	类型	大小
drivers	2021/9/3 9:53	文件夹	
examples	2021/9/3 9:53	文件夹	
hardware	2021/9/3 9:53	文件夹	
java	2021/9/3 9:54	文件夹	
lib	2021/9/3 9:53	文件夹	
libraries	2021/9/3 9:54	文件夹	
tools	2021/9/3 9:54	文件夹	
tools-builder	2021/9/3 9:53	文件夹	
arduino.exe	2021/9/3 9:54	应用程序	72 KB
arduino.l4j.ini	2021/9/3 9:54	配置设置	1 KB
arduino_debug.exe	2021/9/3 9:54	应用程序	69 KB
arduino_debug.l4j.ini	2021/9/3 9:54	配置设置	1 KB
arduino-builder.exe	2021/9/3 9:53	应用程序	23,156 KB
libusb0.dll	2021/9/3 9:53	应用程序扩展	43 KB
msvcpr100.dll	2021/9/3 9:53	应用程序扩展	412 KB
msvcpr100.dll	2021/9/3 9:53	应用程序扩展	753 KB
revisions.txt	2021/9/3 9:53	文本文档	96 KB
wrapper-manifest.xml	2021/9/3 9:54	XML 文档	1 KB

图 9

#### 2. 运行 Arduino IDE

运行文件夹中的 arduino.exe 打开编译器如下图所示。

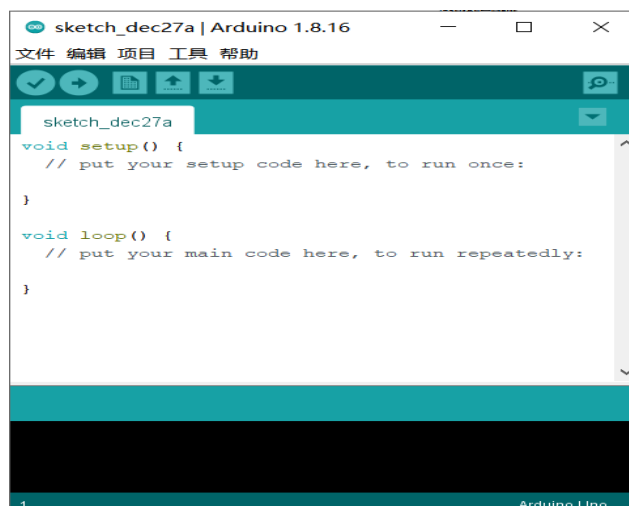


图 10

Arduino IDE 如图 10 所示可分为四个部分：顶部为菜单栏和工具栏，其中工具栏有 6 个快捷按钮，功能如图；中间为编辑区；底部为状态栏，用于输出编译或调试信息。

### 3. 选择开发板型号

在工具菜单下开发板中选择 **Arduino Mega or Mega2560**。

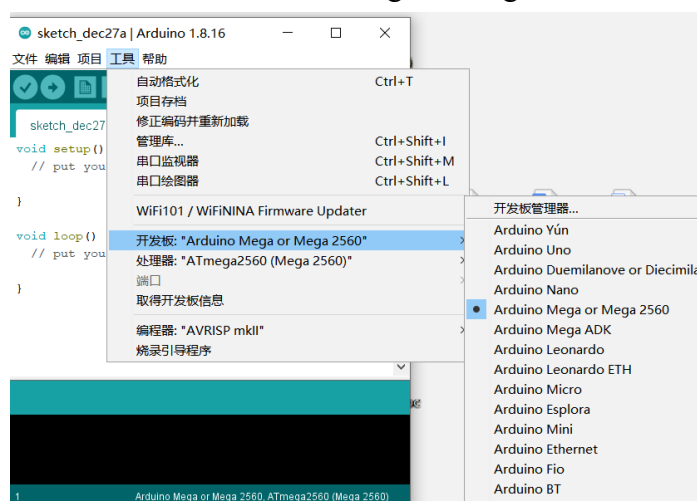


图 11

### 4. 选择端口

接着在端口中选择开发板使用的端口，本实验中连接到了 COM3 端口，根据提示选择对应的端口 COM3（Arduino Mega /Mega 2560）。

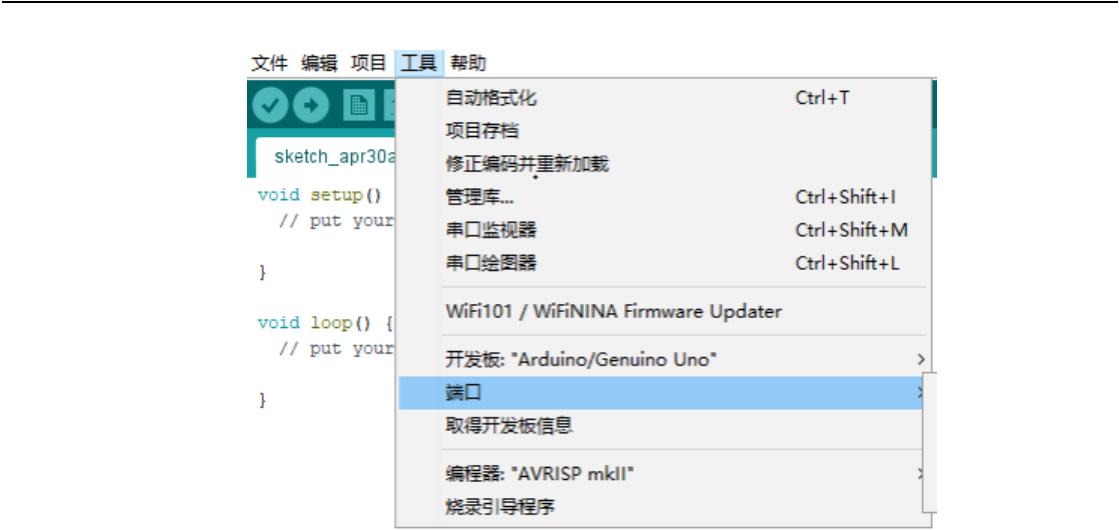


图 12

## 2.3 软件烧写

### 1. 下载资料包

名称	大小	压缩后大小	类型
.			文件夹
Demo			文件夹
Adafruit_BME280_Library-master.zip	455,214	453,905	ZIP 文件
Adafruit_CCS811-master.zip	9,842	8,325	ZIP 文件
Adafruit_Sensor-master.zip	7,830	7,366	ZIP 文件

图 13

如图所示，下载后的压缩包内共有一个文件夹和三个.zip 文件。demo 文件夹内为要烧写到 Arduino UNO 开发板中的代码。其余三个.zip 压缩文件为需要添加到 IDE 中的库文件。

### 2. 添加库文件

运行 Arduino IDE，在项目菜单下找到“添加.ZIP 库...”如图所示：



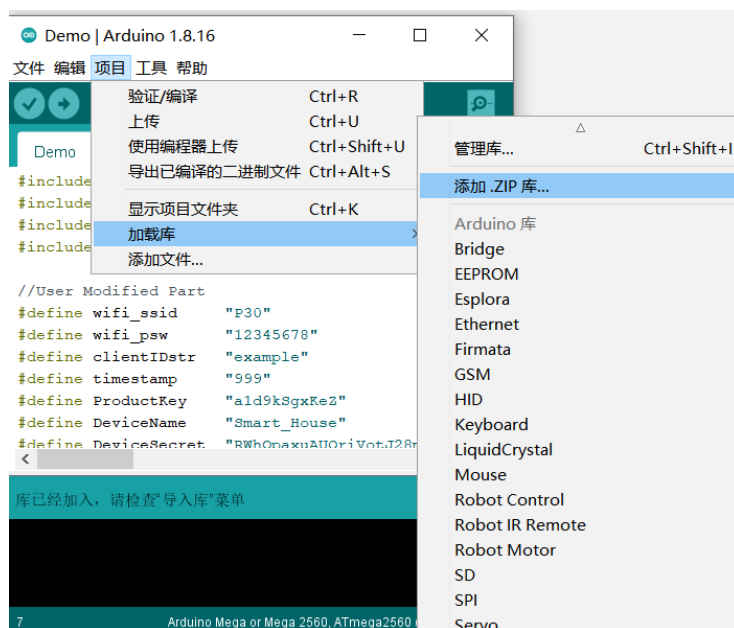


图 14

### 3.修改代码

打开 demo 文件夹中的 Demo.ino，使用三元组等信息替换内容。

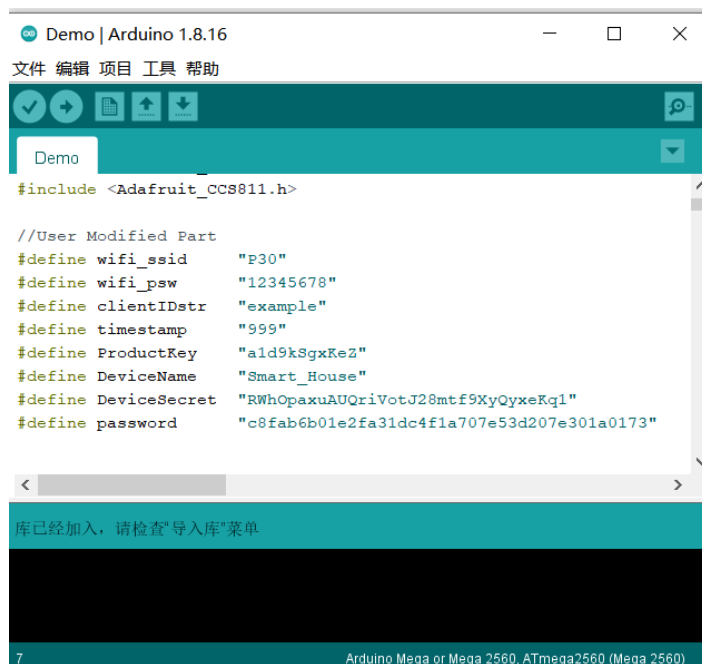


图 1

### 4.上传代码

在给开发板烧写代码时不能连接驱动板，否则程序烧录时会出现报错

## 2.4 运行测试

运行测试分为两部分，嵌入式本地运行测试以及嵌入式和云平台的联调测试。

### □ 嵌入式运行测试：



---

将烧写好的开发板通过排插连接到驱动板（在开发板和驱动板进行联调时，如果驱动板上有跳线帽，应将其拿掉）。开启电源供电。

蜂鸣器会提示当前程序的运行状态：

□响一声代表 IIC 传感器初始化完成，若没有响请检查 IIC 传感器连接。

□响两声代表 wifi 连接完成，若没有响请检查 wifi 配置信息。

□响三声代表与阿里云物联网平台的连接已经建立。若没有响请确认平台 篇部分替换的字段是否正确。

总计六次蜂鸣器的响声过后，小屋开始正常运行各种功能。除可燃气检测联动场景的消除静音需要正确接收下发的消息，其余场景的联动均在本地运行。可以根据场景进行各种测试。检查各类传感器，执行器的状况。

#### □ 平台联调测试

见平台篇

## 3 平台篇

### 3.1 物联网云平台简介

物联网平台是物联网产业链的枢纽，向下接入分散的物联网传感层，汇集传感数据，向上面向应用服务提供商提供应用开发的基础性平台和面向底层网络的统一数据接口，支持具体的基于传感数据的物联网应用。

随着物联网产业的快速发展，产品需求不断变化，对应用程序的快速开发、管理的简化和自动化，以及对应用程序的可靠性和可用性都提出了更高的要求。

PaaS（Platform-as-a-Service）平台即服务特点：

#### 1. 基础平台

PaaS 区别于 IaaS（Infrastructure-as-a-Service，基础设施即服务）和 SaaS（Software-as-a-Service，软件即服务）的最主要特点是，PaaS 提供的是一个基础平台。从传统意义上讲，平台是由应用提供商搭建和维护的，是应用系统部署的基础。而 PaaS 把平台包装成服务，提供给应用提供商，降低了应用提供商的开发成本。

#### 2. 技术支持服务

除基础平台之外，PaaS 服务提供商还提供了该云平台的技术支持，甚至包括对应用系统开发、优化等服务。PaaS 提供的技术支持为之后的应用系统长期、稳定运行提供了技术保障。

#### 3. 平台级服务

PaaS 提供的服务还包括抽象出的元素模型和大量的可编程接口，为提供商的应用开发提供基础。PaaS 为用户提供弹性服务支持，真正实现了资源的动态伸缩、统一运维，提供了更好的平台服务。

云平台端的主要作用是帮助开发者搭建数据通道、方便终端。使用物联网平台的第一步是在云端先创建产品和然后创建对应设备。产品是设备的集合，通常指一组具有相同功能的设备。物联网平台为每个产品颁发全局唯一的 ProductKey。每个产品下可以有成千上万的设备。设备指归属于某个产品下的具体设备。物联网平台为设备颁发产品内唯一的证书 DeviceName。设备可以直接连接物联网平台，也可以作为子设备通过网关连接物联网平台。

网关指能够直接连接物联网平台的设备，且具有子设备管理功能，能够代理子设备连接云端。子设备本质上也是设备。子设备不能直接连接物联网平台，只能通过网关连接。设备能够直接连接物联网平台。

与产品、设备相关的还有一个特别重要的名词是三元组。它是指 ProductKey、DevicesName 和 DeviceSecret。在创建好产品，并且在产品下面添加了一个新设备后，这个设备都会拥有一个在云端唯一的三元组信息，用于鉴权，需要妥善保管好。

### 3.2 物模型

物模型指将物理空间中的实体数字化，并在云端构建该实体的数据模型。物模型本质上是属于功能定义，它是从属性、服务和事件三个维度，分别描述了该实体是什么、能做什么，可以对外提供哪些信息。物模型三个维度的说明：

功能类型	说明
1. 属性（Property）	一般用于描述设备运行时的状态，如环境监测设备所读取的当前环境温度等。属性支持 GET 和 SET 请求方式。应用系统可发起对属性的读取和设置请求。
2. 服务（Service）	设备可被外部调用的能力或方法，可设置输入参数和输出参数。相比于属性，服务可通过一条指令实现更复杂的业务逻辑，如执行某项特定的任务。
3. 事件（Event）	设备运行时的事件。事件一般包含需要被外部感知和处理的通知信息，可包含多个输出参数。如某项任务完成的信息，或者设备发生故障或告警时的温度等，事件可以被订阅和推送。

本项目小屋的功能定义如下：

属性：室内温度、可燃气体传感器读数、光敏传感器读数、土壤湿度、二氧化碳浓度、有机气体浓度，空调制冷开闭，风扇开合，窗帘开合，灯光开关，水

泵开关。

服务：清除报警。当危险状态已经解除，而设备端仍处于可燃气体泄漏报警状态时，用户可以调用该服务清除报警状态。

事件：可燃气体报警。当可燃气体传感器采样到的数值高于设定阈值时，设备向平台上报报警信息。

平台端定义功能如下：

功能类型	功能名称	标识符	数据类型	数据定义
属性	室内温度	RoomTemp	float(浮点型)	取值范围：-50-100
属性	空调制冷	ACTemp	bool(布尔型)	布尔值：关-0；开-1；
属性	可燃气体传感器读数	GasDetector	int32(整数型)	取值范围：0-1024
属性	风扇	Fan	bool(布尔型)	布尔值：关-0；开-1；
属性	蜂鸣器	Buzzer	enum(枚举型)	枚举值：关-0；开-1；静音-2
属性	光敏传感器读数	LightDetector	int32(整数型)	取值范围：0-1024
属性	窗帘	Curtain	bool(布尔型)	布尔值：关-0；开-1；
属性	灯光	Light	bool(布尔型)	布尔值：关-0；开-1；
属性	土壤湿度	SoilHumi	int32(整数型)	取值范围：0-1024
属性	水泵	Pump	bool(布尔型)	布尔值：关-0；开-1；
属性	二氧化碳浓度	eCO2	int32(整数型)	取值范围：400-8192
属性	有机气体浓度	TVOC	int32(整数型)	取值范围：0-1187
事件	可燃气体报警	GasAlarm		事件类型：告警
服务	清除报警	ClearAlarm		调用方式：异步调用

图

“物模型”是用 JSON 格式的数据表述的，称之为 TSL(即 Thing Specification Language)，用于描述产品“是什么、能做什么、可以对外提供哪些服务”。

## 3.3MQTT 协议

### 1.协议实现方式

在 MQTT 协议框架中，有三种角色：代理服务器、发布者客户端以及订阅者客户端。发布者客户端也可以是订阅者客户端，甚至可以订阅自己发布的消息。

---

以智慧小屋为例，我们想要在手机端 APP 获取到智慧小屋的温度信息

1.在智慧小屋端通过温度传感器获取到温度信息，通过 WiFi 上传一个智慧小屋主题为“温度”的数据。

2.云端收到这条信息，将其放置到智慧小屋设备的“温度”主题中。

3.由于手机 APP 订阅了智慧小屋“温度”主题，在有智慧小屋新的温度消息传递过来，云端推送了该主题的数据给手机。

MQTT 采用代理的发布/订阅模式实现了发布者和订阅者的解耦(decouple)：空间上解耦：传送消息不需要知道对方的 IP 地址和端口，只需要和服务器通信即可，发布者和订阅者都一样。

时间上的解耦：发布者和订阅者不需要同时在线，由于消息是暂存在服务器的，发布消息的时候，订阅者在不在线是不会影响发布者发布消息的。但是消息的传输必须通过 Topic(主题)来进行，也就是发布者和订阅者都必须是同一个主题。

消息的发布者和订阅者都是客户端，消息代理是服务器，消息发布者可以同时是订阅者。MQTT 中传输的消息分为：主题(Topic)和负载(payload)两部分：

Topic：可以理解为消息的类型，订阅者订阅(Subscribe)后，就会收到该主题的消息内容(payload)

payload：可以理解为消息的内容，是指订阅者具体要使用的内容

## 2. MQTT 协议中的方法

MQTT 协议中定义了一些方法（也被称为动作），来于表示对确定资源所进行操作。这个资源可以代表预先存在的数据或动态生成数据，这取决于服务器的实现。

Connect：等待与服务器建立连接

Disconnect：等待 MQTT 客户端完成所做的工作，并与服务器断开 TCP/IP 会话

Subscribe：等待完成订阅

UnSubscribe 等待服务器取消客户端的一个或多个 topics 订阅

Publish：MQTT 客户端发送消息请求，发送完成后返回应用程序线程

## 3. 网络传输与应用消息

MQTT 会构建底层网络传输：它将建立客户端到服务器的连接，提供两者之间的一个有序的、无损的、基于字节流的双向传输。当应用数据通过 MQTT 网络发送时，MQTT 会把与之相关的服务质量(QoS)和主题名(Topic)相关连。

## 3.4 设备基于 MQTT 接入云的认证方式

MQTT 连接入物联网

阿里云物联网云端提供了两种连接方式：MQTT 客户端直连和使用 HTTPS 认证再连接。

(详情可以参考 [https://help.aliyun.com/document\\_detail/73742.html?spm=a2c4g.11186623.6.657.6e293f86N32RJS](https://help.aliyun.com/document_detail/73742.html?spm=a2c4g.11186623.6.657.6e293f86N32RJS))

### 一、MQTT 客户端域名直连方式

MQTT 客户端域名直连方式更加适合于资源受限的设备使用，需要注意 MQTT Connect 时:Connect 指令中的 KeepAlive 不能小于 30 秒，否则会拒绝连接。建议值为[60 秒, 300 秒]之间。如果同一个设备三元组，同时用于多个连接，可能导致客户端互相上下线。

1、MQTT 采用的是明码传输，安全的级别很低，密码很容易被截取。为了提高安全级别，如果使用下 TLS，需要下载根证书。

2、使用 MQTT 客户端连接服务器，自主接入可以使用开源 MQTT 客户端参考

### 二、HTTPS 认证连接模式

这两种设备认证方式各有其优缺点，MQTT 客户端域名直连方式更加适合于资源受限的设备使用。在本次的项目例程当中，采用的就是 MQTT 客户端域名直连的方式。

## 4smart 小屋的软件实现

### 4.1 代码综述

代码要实现数据采集、逻辑控制、数据上云、指令响应这四方面的功能。本项目中触发阈值的设置尤为重要，本文的设置方法如下图所示。

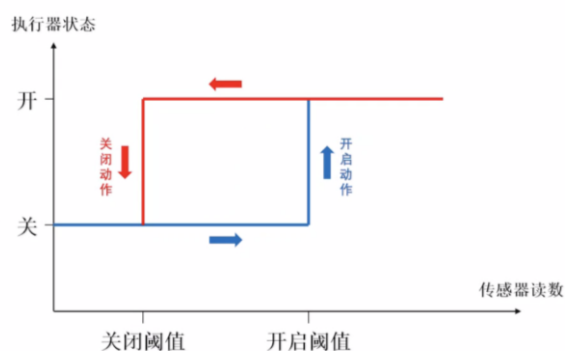


图 2

代码的整体结构如下

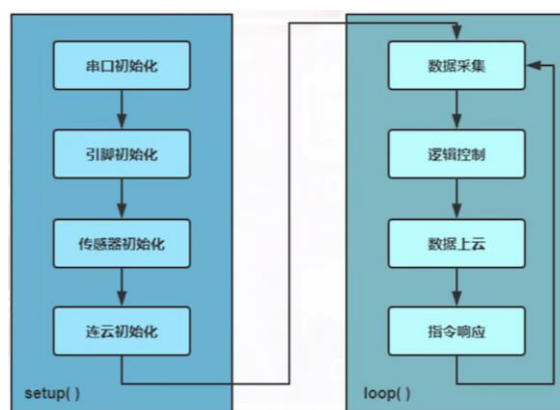


图 3

部分代码示例：

通过蜂鸣器响声获知运行状态

```
void BEEP(int b_time)
{
    for(int i = 1;i <= b_time;i++)
    {
        digitalWrite(BuzzerPin,HIGH);
        delay(100);
        digitalWrite(BuzzerPin,LOW);
        delay(100);
    }
}
```

数据采集

```
//SensorCollect
RoomTemp = bme.readTemperature();
GasDetector = analogRead(GasPin);
LightDetector = analogRead(LightPin);
SoilHumi = analogRead(SoilHumiPin);
if(ccs.available()){
    if(!ccs.readData()){
        eCO2 = ccs.geteCO2();
        TVOC = ccs.getTVOC();
    }
}
```

## 4.2 传感器数据采集与逻辑处理

根据传感器输出值形式的不同，小屋用到的传感器分为模拟传感器和数字传

---

感器。

其中土壤湿度传感器、光敏传感器、可燃气体传感器它们输出电压值为模拟传感器，需经过模拟-数字转换器进行转换。

土壤湿度传感器呈负相关，湿度高，电压低；湿度低，电压高；阈值应设置为读数低，关闭水泵；读数高，开启水泵。

代码示例：

//4.SoilHumi

```
if((SoilHumi > Pump_ON_val)&&(Pump == OFF))
{
    Pump = ON;
    Pump_ON;
}
if((SoilHumi < Pump_OFF_val)&&(Pump == ON))
{
    Pump = OFF;
    Pump_OFF;
}
```

光敏传感器负相关，光照强，电压低；光照弱，电压高；阈值应设置为读数低，白天；读数高，夜晚。

代码示例：

//3.Light

```
if((LightDetector > Light_ON_val)&&(Curtain == ON))
{
    Curtain = OFF;
    Curtain_OFF();
    Light_ON();
    Light = ON;
}

if((LightDetector < Light_OFF_val)&&(Curtain == OFF))
{
    Curtain = ON;
    Curtain_ON();
    Light_OFF();
    Light = OFF;
}
```

}

## 4.3 阿里云配置

### 1.创建产品

<https://iot.console.aliyun.com/lk/summary>



图 4

服务器选择华东 2(上海)，开通物联网平台公共实例(创建需要约 2 分钟)，创建完成后点击进入公共实例管理页面。

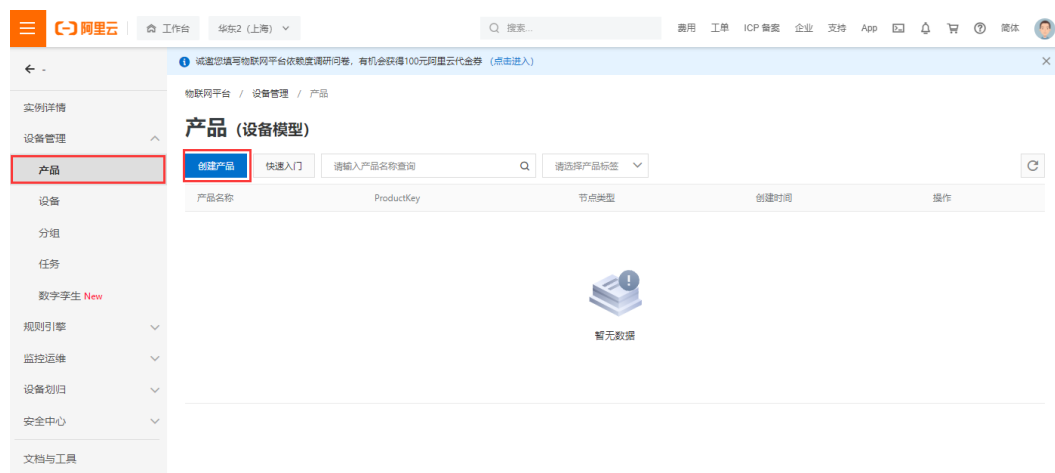


图 5

点击设备管理-产品。在产品界面的左上角找到蓝色的创建产品，新建产品



## ← 新建产品 (设备模型)

新建产品

从设备中心新建产品

\* 产品名称

智慧小屋

\* 所属品类

标准品类

自定义品类

\* 节点类型

直连设备

网关子设备

网关设备

连网与数据

\* 连网方式

Wi-Fi

\* 数据格式

ICA 标准数据格式 (Alink JSON)

✓ 校验类型

✓ 认证方式

更多信息

✓ 产品描述

图 6

单击确认后等待几秒，在产品列表中将看到新创建的智慧小屋，单击查看进入产品详情界面

## 产品 (设备模型)

创建产品	快速入门	请输入产品名称查询	Q	请选择产品标签	↕	↻
产品名称	ProductKey	节点类型	创建时间	操作		
智慧小屋	gg36CKc76UJ	设备	2021/09/30 19:07:23	查看	管理设备	删除

## ← 智慧小屋

发布

ProductKey gg36CKc76UJ 复制

ProductSecret \*\*\*\*\* 查看

设备数 0 前往管理

产品信息	Topic 类列表	功能定义	数据解析	服务订阅	设备开发
------	-----------	------	------	------	------

产品信息 编辑

产品名称	智慧小屋	节点类型	直连设备	创建时间	2021/09/30 19:07:23
所属品类	自定义品类	数据格式	ICA 标准数据格式 (Alink JSON)	数据校验级别	弱校验
认证方式	设备密钥	动态注册	已关闭	状态	开发中
连网协议	Wi-Fi	产品描述	-		

标签信息 编辑

产品标签: 无标签信息

图 7

产品详情页面上方找到功能定义，单击进入功能定义选项卡。

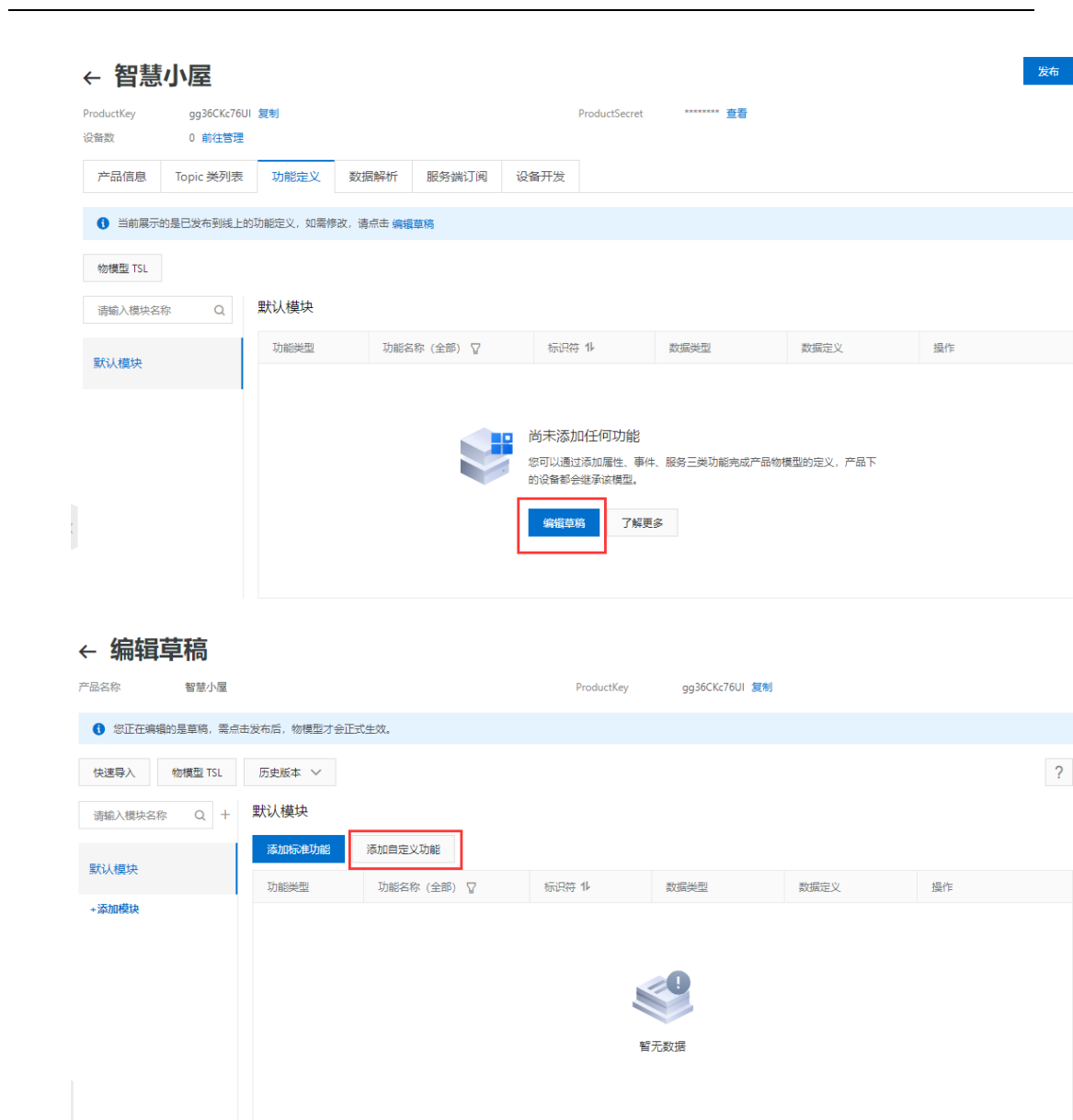


图 8

点击蓝色的“编辑草稿”。共有两种添加功能的方式，第一种方式为手动添加各个功能，该方式相对复杂，但操作后可增加对各项功能的了解；第二种方式为通过导入物模型文件批量添加功能。

默认模块

添加标准功能

添加自定义功能

功能类型	功能名称 (全部)	标识符	数据类型	数据定义	操作
属性	有机气体浓度 自定义	TVOC	int32 (整数型)	取值范围: 0 ~ 1187	编辑   删除
属性	类二氧化碳浓度 自定义	eCO2	int32 (整数型)	取值范围: 400 ~ 8192	编辑   删除
属性	窗帘状态 自定义	Curtain	bool (布尔型)	布尔值: 0 - 关 1 - 开	编辑   删除
属性	灯光状态 自定义	Light	bool (布尔型)	布尔值: 0 - 关 1 - 开	编辑   删除
属性	光敏传感器读数 自定义	LightDetector	int32 (整数型)	取值范围: 0 ~ 1023	编辑   删除
属性	蜂鸣器状态 自定义	Buzzer	enum (枚举型)	枚举值: 0 - 关闭 1 - 开启 2 - 静音	编辑   删除
属性	风扇状态 自定义	Fan	bool (布尔型)	布尔值: 0 - 关 1 - 开	编辑   删除
属性	可燃气体传感器读数 自定义	GasDetector	int32 (整数型)	取值范围: 0 ~ 1023	编辑   删除
属性	水泵状态 自定义	Pump	bool (布尔型)	布尔值: 0 - 关 1 - 开	编辑   删除
属性	土壤湿度传感器读数 自定义	SoilHumi	int32 (整数型)	取值范围: 0 ~ 1023	编辑   删除

< 1 2 >

共有 12 条, 每页显示: 10

默认模块

添加标准功能

添加自定义功能

功能类型	功能名称 (全部)	标识符	数据类型	数据定义	操作
属性	空调状态 自定义	AC	bool (布尔型)	布尔值: 0 - 关 1 - 开	编辑   删除
属性	室内温度 自定义	RoomTemp	float (单精度浮点型)	取值范围: -100 ~ 100	编辑   删除

< 1 2 >

共有 12 条, 每页显示: 10

图 9

全部添加完成后的属性如图所示，共 12 条属性。

2.添加设备

在左侧导航栏中找到设备，单击进入设备管理页面，单击添加设备。添加设备如图。

添加设备

特别说明：DeviceName 可以为空，当为空时，阿里云会颁发产品下的唯一标识符作为 DeviceName。

产品

智慧小屋

DeviceName

Smart\_House

备注名称

请输入备注名称

确认

取消

图 10

单击确认后，弹出添加完成窗口。单击一键复制设备证书。新建记事本，将复制的设备证书三元组粘贴到记事本中。

添加完成

添加设备成功，接下来您可以：

查看设备信息

设备信息页面包含：设备信息、运行状态、事件管理、服务调用、设备影子、文件管理等设备维度的功能集合。

前往查看

烧入 SDK 至设备

您可以依照您的需求，前往选择不同的语言的 SDK，并复制下方的设备证书，烧入至您的设备。（注：您可以随时在设备信息中，查看此设备的设备证书）

查看 SDK

一键复制设备证书

完成

设备证书.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
{
  "ProductKey": "gg36CKc76UI",
  "DeviceName": "Smart_House",
  "DeviceSecret": "ccced824b852c351bfba4534831c98bd"
}
```

第 1 行, 第 1 列 100% Windows (CRLF) UTF-8

图 11

### 3. 计算哈希值

复制以下文字，粘贴到记事本中，如图。

clientIDstr example

timestamp 999

clientIdAAAdeviceNameBBBproductKeyCCCtimestampDDD

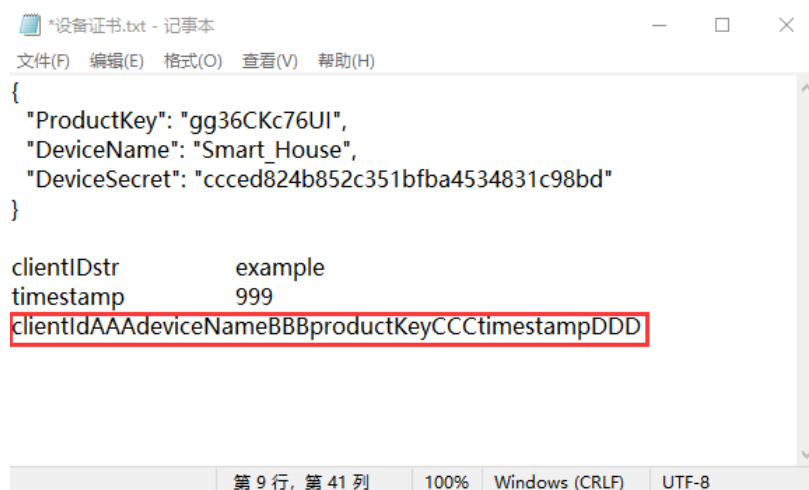


图 12

利用已有字段对红框内最后一行的信息进行替换：

□AAA 替换为 clientIDstr 后面的内容

□BBB 替换为三元组的 DeviceName

□CCC 替换为三元组的 ProductKey

□DDD 替换为 timestamp 后面的内容

替换后的效果如图所示。

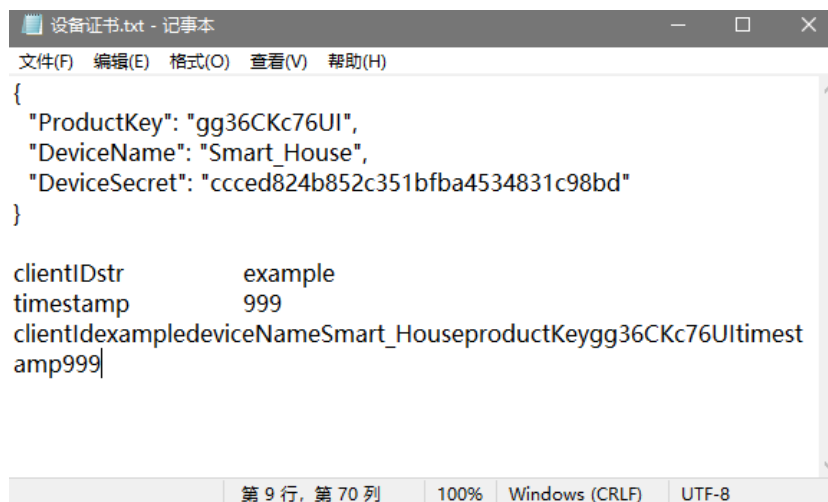


图 13

访问链接打开签名算法计算工具：

<http://iot-face.oss-cn-shanghai.aliyuncs.com/tools.htm>

将最后一行的内容填写到明文中（注意明文的末尾不能包含空格，否则会得

到完全不同的哈希值），并将 DeviceSecret 填写到秘钥中，单击 hmac-sha1 按钮生成哈希值如图。

HmacMD5,HmacSHA1,HmacSHA256 签名算法验证

明文:

秘钥:

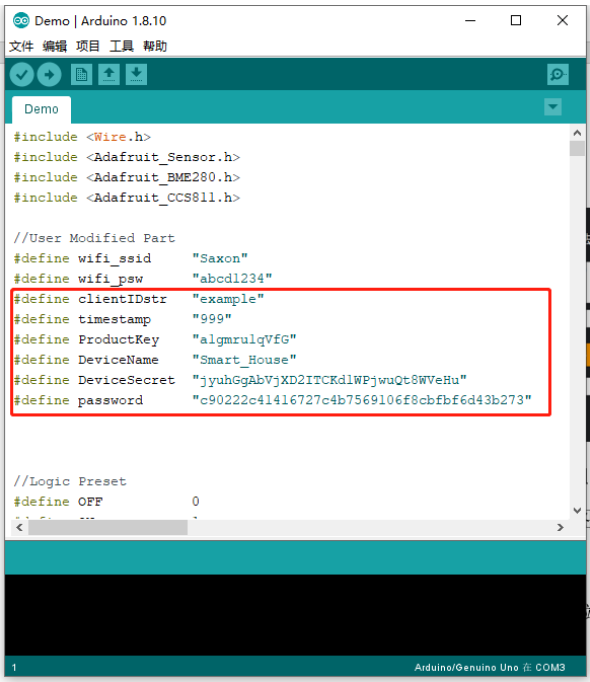
哈希值:

图 14

复制生成的哈希值到记事本中，作为 password 字段替换代码内容。

#### 4. 修改代码

打开代码，找到用户自定义部分如图。



```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <Adafruit_CCS811.h>

//User Modified Part
#define wifi_ssid "Saxon"
#define wifi_psw "abcd1234"
#define clientIDstr "example"
#define timestamp "999"
#define ProductKey "algrulqVFG"
#define DeviceName "Smart_House"
#define DeviceSecret "jyuhGgAbVjXD2ITCKd1WPjwuQt8WVeHu"
#define password "c90222c41416727c4b7569106f8cbfbf6d43b273"

//Logic Preset
#define OFF 0
```

前文中，我们已经对 wifi 的配置信息进行了替换，此处我们需要替换剩下的内容：

- clientIDstr 和 timestamp 若没有改变，不需要修改。
- 三元组内容按照记事本中的内容替换。
- password 使用刚刚生成的哈希值进行替换。

单击保存，将修改好的代码烧入开发板。平台的相关配置工作就完成了。

#### 5. 查看运行状态

小屋正常运行后，我们可以在物联网平台查看所有的状态信息。

打开物联网平台，在设备管理页面找到新添加的 Smart\_House，此时的设备属性应显示为在线。若不显示在线，且小屋上电后发生了响三声的动作，说明烧入了错误的三元组。

## 6. 下发静音指令

小屋的蜂鸣器被触发的情况下，可以通过平台下发的静音指令消除报警声音。

在物联网平台左侧的导航栏，监控运维中找到在线调试，进入在线调试页面如图所示。

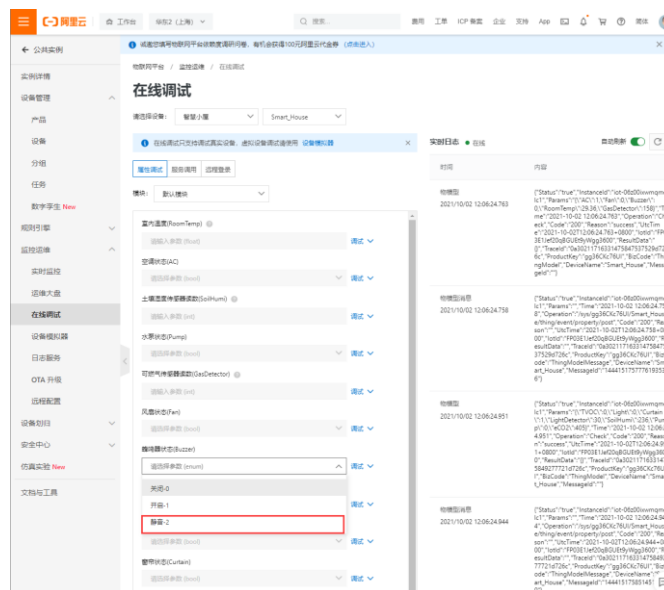


图 15

选择设备为“智能小屋” - “Smart\_House”，点击属性调试，在蜂鸣器状态中选择“静音-2”单击“设置”指令，等待几秒钟即可。

## 4.4 数据上云与指令响应

AT 指令

3GPP

Wifi 配置指令：AT+WJAP=Saxon,abcd1234

指令回写：AT+UARTE=ON

通过 AT 指令，我们实现与 WiFi 模组的交互过程，下达 AT 指令通过模组连接到手机的热点，通过回复的信息确认连接成功。



代码实现:

```
#define wifi_ssid      "P30"
#define wifi_psw       "12345678"
```

/ATcmd Format

```
#define AT              "AT\r"
#define AT_OK           "OK"
#define AT_REBOOT       "AT+REBOOT\r"
#define AT_ECHO_OFF     "AT+UARTE=OFF\r"
#define AT_MSG_ON       "AT+WEVENT=ON\r"
```

```
#define AT_WIFI_START   "AT+WJAP=%s,%s\r"
#define AT_WIFI_START_SUCC "+WEVENT:STATION_UP"
```

```
bool check_send_cmd(const char* cmd,const char* resp,unsigned int timeout)
{
    int i = 0;
    unsigned long timeStart;
    timeStart = millis();
    cleanBuffer(ATbuffer,BUF_LEN);
    Serial.print(cmd);
    Serial.flush();
    while(1)
    {
        while(Serial.available())
        {
            ATbuffer[i++] = Serial.read();
```



---

```
        if(i >= BUF_LEN)i = 0;
    }
    if(NULL != strstr(ATbuffer,resp))break;
    if(((unsigned long)(millis() - timeStart > timeout * 1000)) break;
}

if(NULL != strstr(ATbuffer,resp))return true;
return false;
}

bool WiFi_init()
{
    bool flag;

    flag = check_send_cmd(AT,AT_OK,DEFAULT_TIMEOUT);
    if(!flag)return false;

    flag = check_send_cmd(AT_REBOOT,AT_OK,20);
    if(!flag)return false;
    delay(5000);

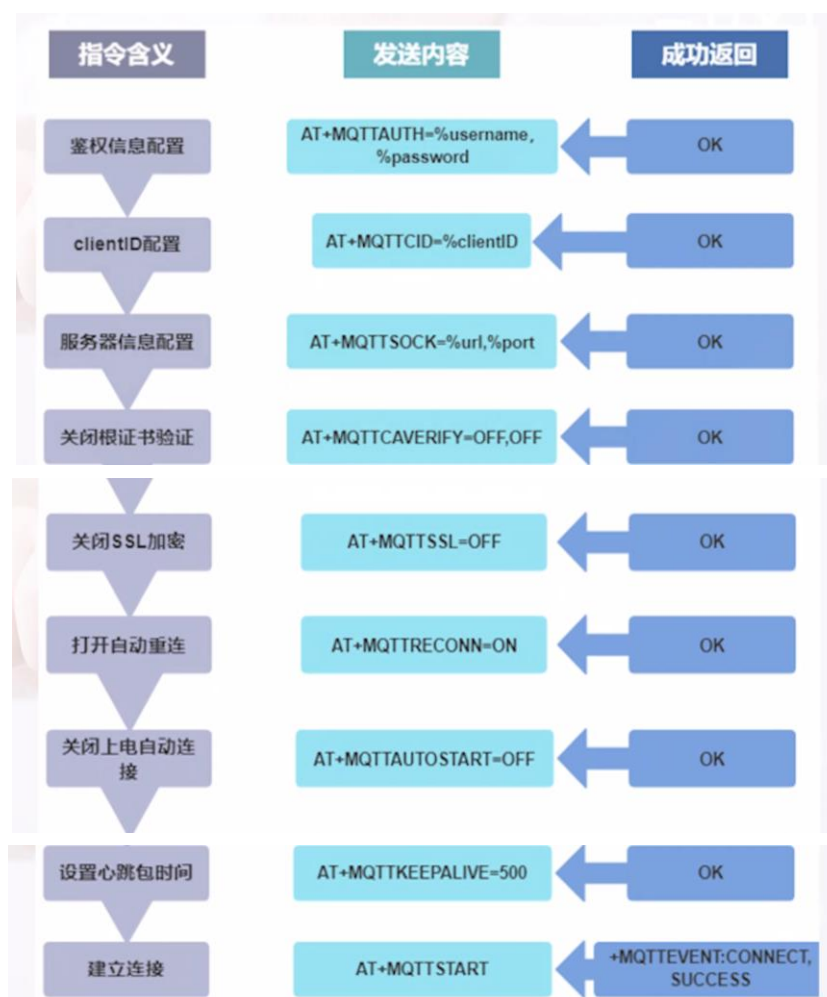
    flag = check_send_cmd(AT_ECHO_OFF,AT_OK,DEFAULT_TIMEOUT);
    if(!flag)return false;

    flag = check_send_cmd(AT_MSG_ON,AT_OK,DEFAULT_TIMEOUT);
    if(!flag)return false;

    cleanBuffer(ATcmd,BUF_LEN);
    snprintf(ATcmd,BUF_LEN,AT_WIFI_START,wifi_ssid,wifi_psw);
    flag = check_send_cmd(ATcmd,AT_WIFI_START_SUCC,20);
    return flag;
}
```

WiFi 连接成功，下一步要建立与物联网云平台的 MQTT 连接。

MQTT 通信指令



## 5 服务规则编排

### 5.1IoT Studio 服务编排

#### 1.什么是 IoT Studio

IoT Studio 是阿里云 IoT 提供的一站式物联网应用开发平台，提供丰富的设备端、服务端、应用端开发能力。

IoT Studio 提供流式的服务编排能力，以可视化拖拽的方式编排开发服务端应用。

IoT Studio 提供可视化的 Web App 搭建能力，可以通过拖拽，配置的方式快速开发前端应用。



## 2.什么是服务开发

服务开发承载着应用的业务逻辑

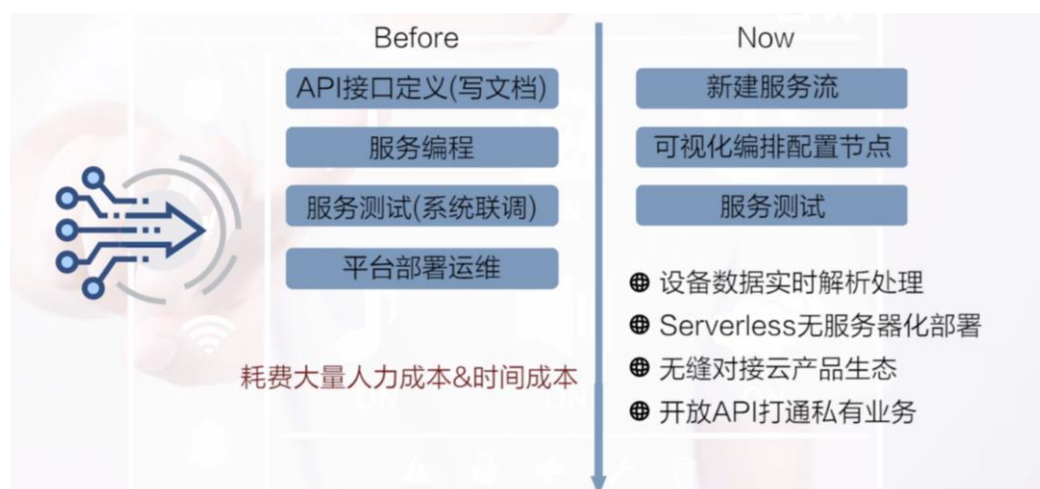


海量的数据在服务端得到分析和处理

2018年双11，背后靠强大的服务应用和计算资源支撑，

杭州城市大脑，海量交通探头、信号灯接受AI和服务应用的实时调度

## 3.IoT Studio 服务开发特点



流式服务可视化编排

---

可视化拖拽配置即可快速完成服务 API 的创建，无码化开发体验。

提供服务流全流程调试工具。

服务流自动部署托管，免运维压力烦恼。

#### 4.IoT Studio 服务开发案例

智能家居开关门控制

什么时候需要开走廊灯？

想控制加湿器根据天气情况自动执行开启或关闭，怎么办？

可获取网络天气查询服务，集成 AI 图像识别等丰富的开放能力

## 5.2 服务端搭建工作台一览

## 6 总结

回顾本次项目的开发过程，其中遇到许多挫折困难，作为一名物联网专业的学生，在理论知识的储备上有很大的进步空间，对于新事物的掌握远远不够，阿里云的许多帮助文档都得细心琢磨，不断摸索，更让我明白了活到老学到老的重要性。伴随着物联网时代到来的，不止是百亿级的终端设备，更是数据洪流。因此，物联网若想继续发展，从互联阶段迈入智能阶段，需要做的便是将这些数据更进一步的收集、分类、处理。数据的管理与计算分析能力无疑将会是物联网平台竞争的差异化所在，例如数据清洗服务、深度学习、人工智能等。因此需要我们青年的不断努力，努力前行，不忘初心，牢记使命。

## 参考文献

- [1] 崔阳，张维华，白云峰．一种基于 Arduino 的智能家居控制系统 [J]. 电子技术应用，2014, 40(4): 123-125.
- [2] 朱敏玲，徐雅斌．基于云平台的智能家居气象站的研究与设计 [J]. 电视技术，2015, 39(12): 23-25.
- [3] 李宇，王卫星，陈润泽．基于 ZigBee 的物联网智能家居系统 [J]. 电子测试，2016, 1(5): 71-75.
- [4] 陈凯旋，周世恒，陈涛，等．基于 Arduino 与 One NET 云平台的简易智能家居系统设计 [J]. 物联网技术，2019, 0(12): 88-90+93.
- [5] 张加书．基于 ARM\_ONENET 云平台的智能家居产品设计 [D]. 太原市：中北大学，2019.
- [6] 王耀楠. 基于云服务平台的智能家居管理系统[D]. 成都市：电子科技大学，2019.

- 
- [7] 聂琿 , 陈海峰 , 周豪 .基于 NB-IoT 的环境监测系统 [J]. 实验技术与管理 , 2020, 37(5) : 89-93+110.
- [8] 徐仲 , 孙先松 . 基于 NB-IoT 的环境温湿度监测系统设计[J]. 物联网技术 , 2020, 1(2) : 20-22.