广东青年职业学院

毕业设计

题　　目：物联网智能家居解决方案

所 在 系：计算机工程系

专　　业：物联网应用技术

班　　级：物联网 1 班

学　　号：15050500133

姓　　名：黄大宇

指导教师：孙　波

完成时间：

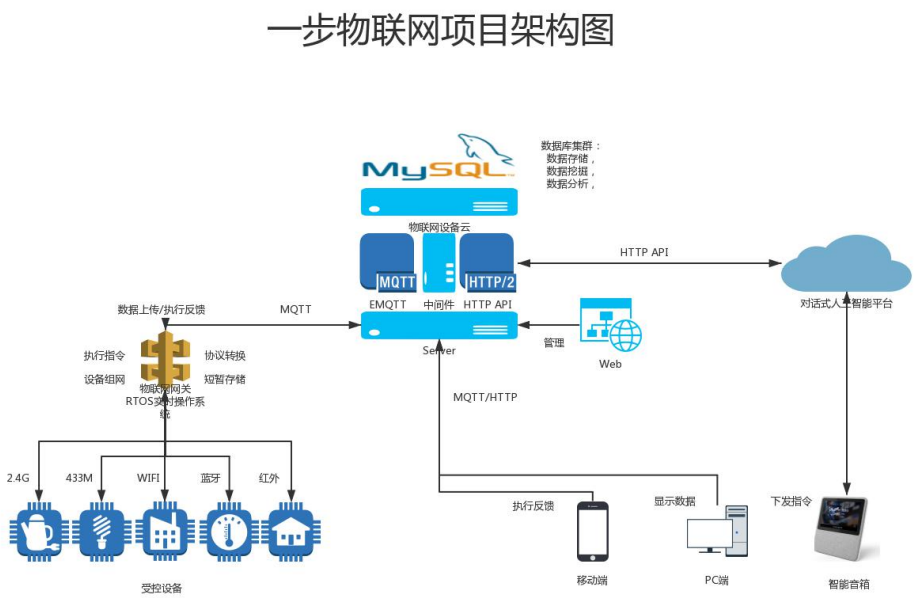
目录

前言

物联网是即互联网和移动互联网之后的第三次科技产业革命。顾名思义，物联网就是物物相连的互联网。物联网的核心和基础仍是互联网，是在互联网技术上的延伸和扩展的网络，其用户端延伸和扩展到了任何物品与物品之间，进行信息交换和通信，达到物物相息。我个人认为，物联网应该细分为 6 层，应用层或服务层，也就是物联网应用于那个行业，如智能农业，智能家居，智能交通；感知层或设备层，也就是具体的传感器，具体的智能设备；网络层或传输层，也就是数据在什么网络中传输，如 HTTP，MQTT，CoAP，WebSocket 等；平台层或者计算层，也就是物联网设备云和云计算层，进行数据计算，分析，挖掘和决策建议。软件成或交互层，也就是 Android 软件，iOS 软件，Web 前端和智能音箱。

1. 项目方案描述

1.1 项目完整架构图



1.2 项目目的

作为一个物联网专业的学生，想要开发一个完整的，全栈的，一站式的物联网项目，从硬件，网络，平台，到软件都是自己搭建和编写，为了增强学习，增加物联网项目经验。

1.3 项目涉及的技术

设备硬件使用 NodeMCU，NodeMCU 基于乐鑫的 ESP8266，使用 Arduino 语言，一个类似 C++的语言；网络使用基于 tcp/ip 的 mqtt 协议和 http1/2 协议；mqtt 代理使用开源的 Emqtt；Server 使用 Spring MVC 架构，用 java 编写，部署到 Tomcat，是一个小的物联网平台；数据库使用 MySql；消息列队中间件使用 kafka；人工对话式人工智能平台接入百度 DuerOS 和阿里巴巴的 AliGenie；人工智能平台与 iot 之间采用 HTTP API 软件使用 Android，java 编写；Web 使用 HTML5；物联网网关使用 RT-Thread RTOS 系统，可连接 WiFi，2.4G 无线网络，433M，蓝牙，蓝牙 Mesh，Zigbee，NFC，RFID，红外等。

1.4 说明

由于工作太忙，时间有限和能力有限，项目并非已经完全实现，但会一步步改进和完善。项目完全开源。

2. 应用层/服务层

2.1 项目应用

物联网的应用领域多达 12 个产业。本项目针对智能家居设计，服务于智能家居。用户可以通过智能音箱，手机软件，web 前端，控制家中的设备，项目中家电设备为台灯和空调，传感器设备为人体红外和温湿度传感器。

3. 感知层/设备层

3.1 Raspberry Pi

Raspberry Pi 即是树莓派，俗称迷你卡片电脑。基金会提供了基于 Debian

的 Raspbian 官方系统。是一款基于 ARM 的微型电脑主板，以 SD 卡作为硬盘，具备 PC 的基本功能，树莓派 3 代 B 型采用 Broadcom BCM2837 芯片，ARM Cortex-A53 1.2GHz 64-bit quad-core ARM v8 CPU,GPU 是 Broadcom VideoCrore IV，OpenGL ES 2.0,1080p 30 h.264/MPEG-4AVC 高清解码器，拥有 1GB 运存，4 个扩展 USB，1 个 10/100 以太网接口，802.11n Wireles LAN WiFi,Bluetooth Low Energy 蓝牙，有 40 个可扩展的 GPIO。

树莓派支持众多的物联网操作系统，Windows 10 IOT，Android Thingks 等。在本项目中，用于安装 Raspbian 系统，DuerOS C++/Python SDK 和 HomeAssistant 智能家居管理系统。可以作为智能家居的控制中枢，通过访问 HomeAssistant，可以控制自定义的智能家居设备，也可以控制 HomeAssistant 支持的智能家居厂家的设备，如小米的，博联的，涂鸦的产品等。

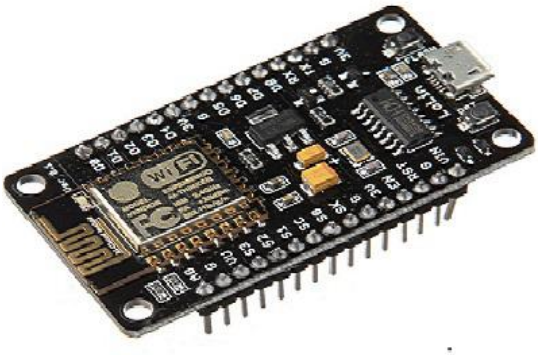
3.1.1 树莓派部分操作指令

a. 更新 Raspbian 系统软件源：sudo apt update
b. 安装运行 DuerOS Python SDK：DuerOS-Modularization
c. 安装运行 HomeAssistant
d. 安装 Python 3.6 环境
e. sudo apt install python3.6
f. sudo apt install python3.6-dev
g. sudo apt install python3-pip
h. 验证安装结果：
i. Python3 -version
j. Pip3 --version
k. 使用 pip3 工具安装 HomeAssistant
l. sudo pip3 install homeassistant
m. 验证安装结果：
n. sudo hass -open-ui
o. 开始使用 HomeAssistant
p. http：//Raspberry pi ip:8123

3.2 Arduino

Arduino 是一款便捷灵活，方便上手的开源电子原形平台，包含硬件各种型号的 Arduino 板和开发软件 Arduino IDE。它构建看开源的 simple I/O 界面版，并且具有使用类似 Java，C 语言的 Processing/Wiring 的开发环境，Ardunio 的编程是通过 Arduino 编程语言（基于 Wiring）和 Ardunio 开发环境（基于 Processing）来实现的。本项目中，主要运用 Arduino IED 作为硬件端的开发环境，也就是搭建 NodeMCU 的开发环境。在 Arduino 中搭建 NodeMCU 的硬件开发环境，需要在配置文件中加入以下 Esp8266 库地址，然后下载安装：
http://arduino.esp8266.com/stable/package_esp8266com_index.json

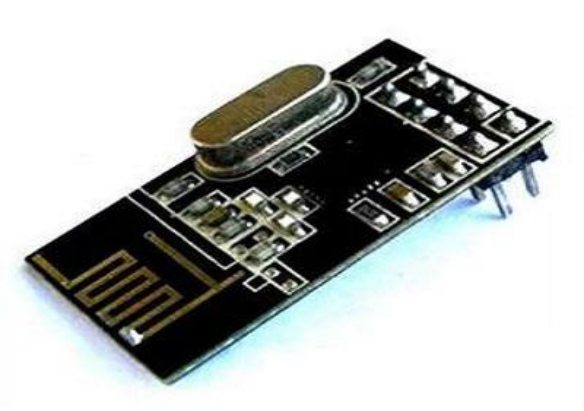
3.3 NodeMCU



NodeMCU 是一个开源的物联网平台，主控芯片是乐鑫的 Esp8266 芯片，可以使用多种语言开发，底层使用乐鑫 Esp8266 sdk 0.9.5 版本，具有 PWM，I2C,1-Wire,ADC 等功能。NodeMCU 是一个快速开发物联网的不二选择。

项目源码地址

3.4 Esp8266



ESP8266 内置超低功耗 Tensilica L106 32 位 RISC 处理器，CPU 时钟速度最高可达 160 MHz，支持实时操作系统 (RTOS) 和 Wi-Fi 协议栈，可将高达 80% 的处理能力留给应用编程和开发。ESP8266EX 专为移动设备、可穿戴电子产品和物联网应用而设计，通过多项专有技术实现了超低功耗。ESP8266EX 具有的省电模式适用于各种低功耗应用场景。ESP8266EX 集成了 32 位 Tensilica 处理器、标准数字外设接口、天线开关、射频 balun、功率放大器、低噪放大器、过滤器和电源管理模块等，仅需很少的外围电路，可将所占 PCB 空间降到最低。ESP8266EX 的工作温度范围大，且能够保持稳定的性能，能适应各种操作环境。

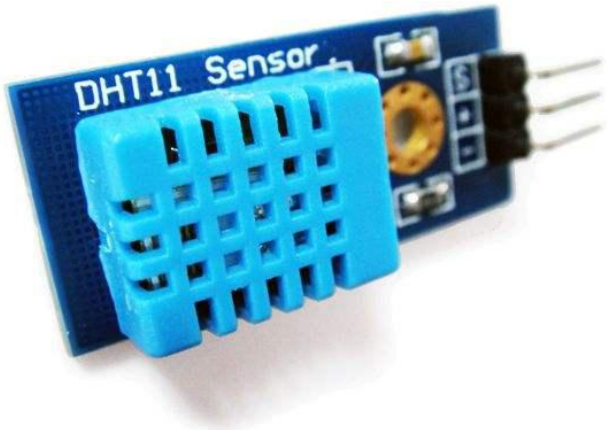
Esp8266 可以使用 AT 指令进行控制芯片，进行开发。也支持 WiFi Mesh 开发，支持 WiFi 自组网。

3.4.1 ESP8266 部分指令

a. 设置 WiFi 应用模式为 AP 兼 Station 模式：
b. AT+CWMODE=<3>
c. 加入 AP 模式（即 WiFi 客户端）：
d. AT+CWJAP=<ssid>,<pwd>
e. 退出与 AT 的连接
f. AT+CWQAP
g. 设置 AP 模式下的参数
h. AT+CWSAP=<ssid>，<pwd>,<chl>,<ecn>
i. 在 Arduino IDE 中可以使用 Esp8266 官方提供的 ESP8266WiFi.h 的库，进行快速开发。

项目源码地址

3.5 DHT 11 温湿度传感器



DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器，它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性和卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接。因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。每个 DHT11 传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式存在 OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，使其成为该类应用中，在苛刻应用场合的最佳选择。产品为 4 针单排引脚封装，连接方便。

Arduino 中可以使用 DHT.h 库开发，readHumidity()函数是读取当前的湿度，readTemperature() 函数是读取当前的温度，单位是 C。

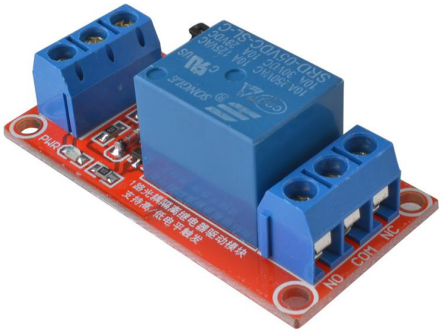
项目源码地址

3.6 红外发射和接收

红外线发射器是一种遥控设备，具有遥控功能。它通过红外线发射管在一定范围内向外发射光线，从而达到控制信号的作用，广泛应用于消费电子、工业和通信等红外线接发器、数据传输技术等领域。本项目使用的红外码库是 Arduino-IRremote ，该库还支持 Esp32 ，Esp8266 和 Arduino 系列板子。

enableIRIn()函数是接收启动器，resume()是接收函数。SendNEC 函数是发送红外信号。

3.7 继电器



继电器（英文名称：relay）是一种电控制器件，是当输入量（激励量）的变化达到规定要求时，在电气输出电路中使被控量发生预定的阶跃变化的一种电器。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系。通常应用于自动化的控制电路中，它实际上是用小电流去控制大电流运作的一种“自动开关”。故在电路中起着自动调节、安全保护、转换电路等作用。

本项目中使用的继电器选择低电平触发基，当 NodeMCU GPIO 输出低电平时，接通电源，台灯打开，输出高电平时，台灯关闭。

4. 网络层/传输层

4.1 TCP/IP

TCP/IP 是传输控制协议，也就是网络通信协议，是 Internet 最基本的协议、Internet 国际互联网络的基础，由网络层的 IP 协议和传输层的 TCP 协议组成。TCP/IP 定义了电子设备如何连入因特网，以及数据如何在它们之间传输的标准。协议采用了 4 层的层级结构，每一层都呼叫它的下一层所提供的协议来完成自己的需求。通俗而言：TCP 负责发现传输的问题，一有问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。而 IP 是给因特网的每一台联网设备规定一个地址。

之所以讲 TCP/IP 协议，是因为 MQTT 协议是基于该协议实现的。

4.2 MQTT

MQTT（消息队列遥测传输）是 IBM 开发的一个即时通讯协议，是当今物联网的网络层的重要组成部分，该协议的开发也主要是面向物联网的。MQTT 协议是为大量计算能力有限，且工作在低带宽，不可靠网络的远程传感器设

4.2.1 MQTT 协议特点

a. 使用发布/订阅消息模式，提供一对多的消息发布，解除应用程序耦合；
b. 对负载内容屏蔽的消息传输；
c. 使用 TCP/IP 提供网络连接；
d. 有三种消息发布服务质量：
e. “至多一次”，消息发布完全依赖底层 TCP/IP 网络。会发生消息丢失或重复。这一级别可用于如下情况，环境传感器数据，丢失一次读记录无所谓，因为不久后还会有第二次发送。
f. “至少一次”，确保消息到达，但消息重复可能会发生。
g. “只有一次”，确保消息到达一次。这一级别可用于如下情况，在计费系统中，消息重复或丢失会导致不正确的结果。
h. 小型传输，开销很小（固定长度的头部是 2 字节），协议交换最小化，以降低网络流量；
i. 使用 Last Will 和 Testament 特性通知有关各方客户端异常中断的机制；

4.2.2 MQTT 函数说明

本项目中使用 pubsubclient 库，用于连接订阅 EMQTT 搭建的 Mqtt Broker（mqtt 代理服务器），接收和发送 Mqtt 消息。

a. 创建 mqtt 客户端
b. PubSubClient(ip,port,[callback],client,[stream])
c. 连接打开服务器
d. boolean connect (clientID, username, password)
e. 断开连接
f. void disconnect ()
g. 发送消息
h. int publish (topic, payload)
i. 点阅主题
j. boolean subscribe (topic, [qos])
k. 取消订阅
l. boolean unsubscribe (topic)
m. 判断是否连接
n. int connected ()
o. 获取连接状态
p. int state ()
q. 设置代理服务器 IP 地址和端口
r. PubSubClient setServer (server, port)
s. 设置回调
t. PubSubClient setCallback (callback)
u. 设置客户端信息

v. PubSubClient setClient (client)
w. 设置消息存储流
x. PubSubClient setStream (stream)

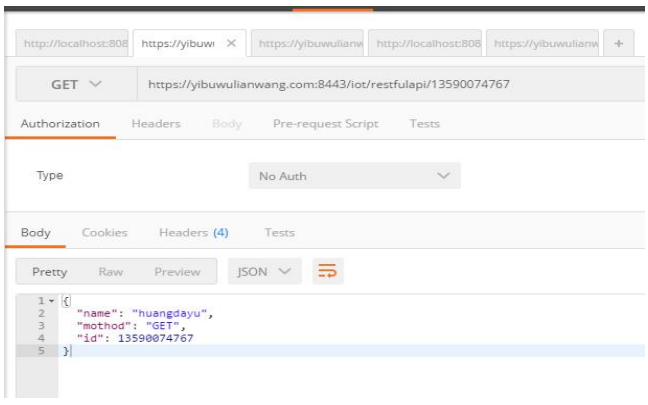
4.3 RESTful API

API 即时应用程序接口，Web API 是网络应用程序接口，网络应用通过 API 接口，可以实现存储服务，消息服务，计算服务等能力，利用这些能力可以进行开发出强大功能的 web 应用。RESTful API 指的是一种接口设计风格，架构风格，不是标准，只是提供了一组设计原则和约束条件，是编程思想。它主要用于客户端和服务端交互类的程序，基于这个设计风格，使交互更加简洁，更有层次，更易于实现缓存等机制。

例如我在 iot 程序中的基于 Spring 框架实现的 RESTfull 接口：

```java
package com.yibuwulianwang.api;

import org.springframework.web.bind.annotation.PathVariable;

@RestController
@RequestMapping("/restfulapi")
public class RestfulAPIController {
    // 查找
    @RequestMapping(value = "/{id}", method = RequestMethod.GET)
    private Object get(@PathVariable long id,
            @RequestParam(value = "name", defaultValue = "huangdayu") String name) {
        JSONObject json = new JSONObject();
        json.put("mothod","GET");
        json.put("id",id);
        json.put("name", name);
        return json;
    }
    // 增加
    @RequestMapping(method = RequestMethod.POST)
    private Object post() {
        JSONObject json = new JSONObject();
        json.put("mothod","POST");
        return json;
    }
    // 修改
    @RequestMapping(method = RequestMethod.PUT)
    private Object put() {
        return null;
    }

    // 删除
    @RequestMapping(method = RequestMethod.DELETE)
    private Object delete() {
        return null;
    }
}
```

GET 方式请求该接口就会得到以下结果：



本项目获取最新温湿度传感器数据的接口，GET 方式请求：

{"date":"2018-05-02 12:07:27","humidity":"89","id":156,"temperature":"26"}

4.4 HTTPS

HTTPS 是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版。即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL，用于安全的 HTTP 数据传输。SSL 是为网络通信提供安全及数据完整性的一种安全协议，遵循 SSL 协议的 HTTP 就可以对数据进行 SSL 加密和解密。项目中之所以搭建 HTTPS，是因为数据安全和各大人工智能平台之前的通信需要 HTTPS。

Tomcat 中配置 SSL 证书：本人所写详细教程链接

4.5 Json

JSON 是一种轻量级数据交换格式，易于理解，也易于解析和生成。JSON 采用完全独立于语言的文本格式，JSON 建构于两种结构：

a. “名称/值”对的集合（A collection of name/value pairs）。不同的语言中，它被理解为*对象(object)*，纪录(record)，结构(struct)，字典（dictionary），哈希表（hash table），有键列表（keyed list），或者关联数组 （associative array）。

b. 值的有序列表（An ordered list of values）。在大部分语言中，它被理解为数组（array）。

本项目中使用 json 作为通讯数据格式，Mqtt 发送的消息也是 json 格式，RESTful API 通讯也是 json 格式，使用自己定的消息协议，协议如下：

GitHub 链接

```
{
    "clientId": "20369341",
    "deviceId": "Iaideng",
    "deviceType": "light",
    "humidity": "60",
    "messageId": "6d6d6e14-8aee-473e-8c24-0d31ff9c17a2",
    "messageType": "Response",
    "name": "Open",
    "namespace": "Control",
    "payload": "str",
    "publishTime": "2018-04-06 20:01:19",
    "remarks": "RPI-DuerOS",
    "temperature": "20.0",
    "topic": "IOI-MQII"
}
```


messageType:消息类型：请求或者响应

client_id:MQTT客户端ID

topic：MQTT客户端订阅的主题

namespace：指令类别

Discovery 设备发现

Control 设备控制

Query 设备属性查询

name：指令名称

device_id：设备ID

deviceType:设备类型 如：灯，电视，空调

payload:消息

temperature:温度

humidity:湿度

messageId消息的唯一标识符，长度小于128个字符。messageId仅用于标识消息。

publish_time：消息发出时间

remarks：备注信息

5. 平台层/计算层

5.1 Ubuntu

Ubuntu 是一个基于 Linux 内核的操作系统，免费而且开源，是 Debian 操作系统的发行版。Ubuntu 作为本项目的服务器操作系统，稳定，安全且持续更新。其自带强大的安装管理工具 apt，可以安装任何 Linux 软件包，包括协议与更新。

5.1.1 Ubuntu 部分命令

a. 更新系统源：sudo apt update

b. 更新已安装软件包：sudo apt upgrade
c. 升级系统：sudo apt dist-upgrade
d. 我的更多学习 Linux 笔记

5.2 MySQL

MySQL 是一个关系型数据库管理系统，关系数据库将数据保存到不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了灵活性，MySQL 所使用的 SQL 语言是用于访问数据库的最常用的标准化语言，任何一门编程语言都可以使用 SQL 语言对 MySQL 数据库中的数据进行增删改查。

5.2.1 Ubuntu 安装使用 MySQL

a. 安装：sudo apt install mysql-server mysql-common
b. 开启服务：sudo systemctl start mysql
c. 暂停服务：sudo systemctl stop mysql
d. 进入数据库：mysql -uroot -ppassword
e. (root 是账户名，password 是数据库密码)

5.2.2 增删改查 SQL 语句

```
//增
final String sql_insert="insert into iot_temp_hum(date,temperature,humidity) values(?,?,?)";
//删
final String sql_delete= "delete from emp where id < ?";
//改
final String sql_update= "update emp set id = ? where temperature = ?";
//查
final String sql_select= "select * from iot_temp_hum order by id desc limit 1";
```

5.3 Tomcat

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务程序。Tomcat 为 WebServer 程序提供安全可靠，性能稳定且免费的运行环境。

5.3.1 Ubuntu 中安装 Tomcat

a. 本人笔记详细教程
b. 从清华大学开源镜像站下载软件包
c. `wget https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.8/bin/apache-tomcat-9.0.8.tar.gz`
d. 解压镜像包
e. tar zxvf apache-tomcat-9.0.8.tar.gz
f. 移动到安装目录

g. mv apache-tomcat-9.0.8 /opt/tomcat/
h. 开启服务：
i. /opt/tomcat/apache-tomcat-9.0.8/bin/startup.sh
j. 关闭服务：
k. /opt/tomcat/apache-tomcat-9.0.8/bin/shutdown.sh

5.4 Spring MVC

Spring MVC 是一个 WebServer 应用程序服务框架，提供 Web 应用程序全功能 MVC 模块，使开发者可以快速开发 Web 应用程序。

5.4.1 Spring MVC 有分层思想

a. DAO 层：持久层，主要负责与数据库进行交互。
b. Entity(domain)层：实体层，数据库对应的实体类。
c. Service 层：业务层，负责业务模块的逻辑应用设计。
d. Controller 层：控制层，负责业务模块流程的控制。
e. View 层：交互层，负责前台 jsp，html 页面的表示。

一个 Maven 项目的 SpringMVC 配置文件：web.xml

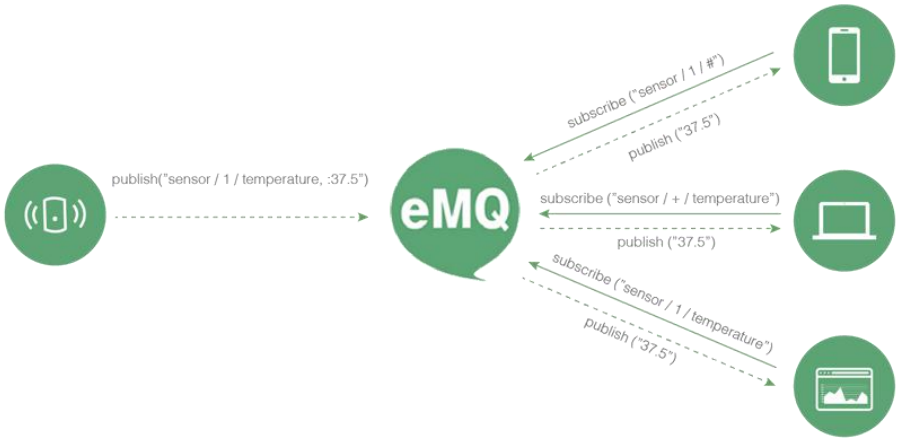
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">

    <servlet>
        <servlet-name>spring</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>spring</servlet-name>
        <url-pattern>/</url-pattern><!-- 拦截器 -->
    </servlet-mapping>

</web-app>
```

spring-servlet.xml

```xml
<!-- 配置扫描的包 -->
<context:component-scan base-package="com.yibuwulianwang.*" />

<!-- 注册HandlerMapper、HandlerAdapter两个映射类 -->
<mvc:annotation-driven />

<!-- 访问静态资源 -->
<mvc:default-servlet-handler />

<!-- 加载了 jdbc.properties 文件 -->
<context:property-placeholder location="classpath:jdbc.properties"/>
<bean id="jdbcConfig" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="location">
        <value>classpath:jdbc.properties</value>
    </property>
</bean>

<!-- 配置了数据库连接的一些参数 -->
<bean id="MyDataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="${jdbc.driverClassName}"/>
    <property name="url" value="${jdbc.url}"/>
    <property name="username" value="${jdbc.username}"/>
    <property name="password" value="${jdbc.password}"/>
</bean>

<!-- jdbcTemplate 一种比原版 jdbc 稍微方便一点的数据库操作类 -->
<bean id="jdbcTemplate" name="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="MyDataSource"/>
</bean>

<!-- 视图解析器 -->
<bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/view/"></property>
    <property name="suffix" value=".html"></property>
</bean>
```

5.5 EMQTT

EMQTT 是中国开发者使用 Erlang/OTP 语言平台设计，支持百万级连接的分布式集群，发布订阅模式的开源物联网 MQTT 消息服务器，完整支持 MQTTV3.1/V3.1.1 协议规范，扩展支持 WebSocket，Stomp，CoAP，MQTT-SN 或私有 TCP 协议。支持双节点负载均衡或多节点分布式集群，支持扩展模块与插件，支持跨平台部署。



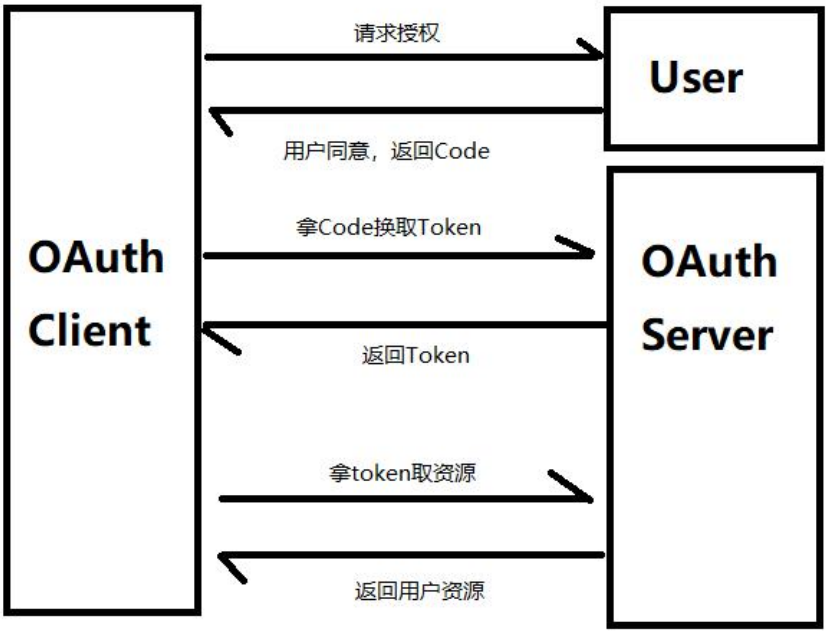
a. 下载 linux 通用安装包
b. wget http://emqtt.com/static/brokers/emqttd-ubuntu16.04-v2.3.6.zip
c. 解压
d. sudo unzip emqttd-ubuntu16.04-v2.3.6.zip
e. 运行服务
f. ./emqttd/bin/emqttd start
g. 查看状态
h. ./emqttd/bin/emqttd_ctl status
i. 停止服务
j. ./emqttd/bin/emqttd stop

官方文档

5.6 OAuth

OAUTH 协议为用户资源的授权提供了一个安全的、开放而又简易的标准。同时，任何第三方都可以使用 OAUTH 认证服务，任何服务提供商都可以实现自身的 OAUTH 认证服务，因而 OAUTH 是开放的。OAUTH 是一种开放的协议，为桌面、手机或 web 应用提供了一种简单的，标准的方式去访问需要用户授权的 API 服务。

授权码模式的 OAuth2.0 的流程图：



Java 实现 OAuth Server 的代码在 com.yibuwulianwang.oauth.serve 中
Java 实现 OAuth Client 的代码在 com.yibuwulianwang.oauth.client 中

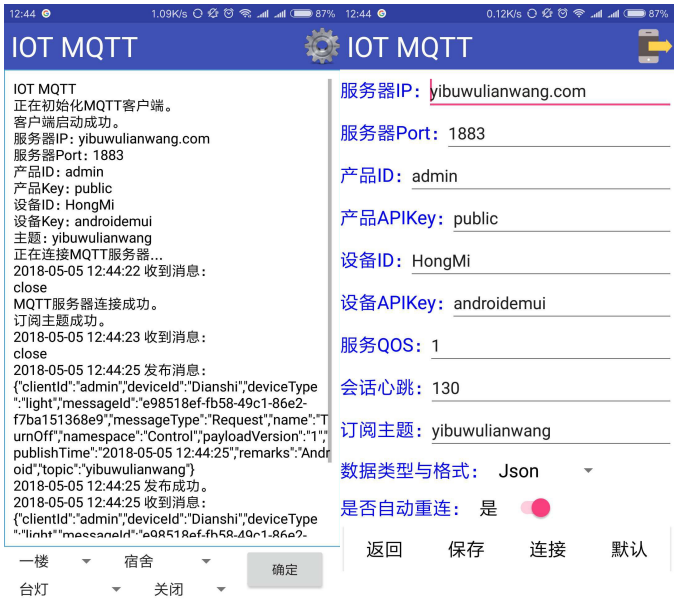
6. 软件层/交互层

6.1 Android

Android 是一个基于 Linux 内核的自由及开放源代码的操作系统，主要使用于移动设备。Android 的底层是 C/C++实现，应用层可以是 Java，Kotlin 等语言实现的 APP。

本软件中使用 Android APP 进行指令下发，消息接收和数据展示。

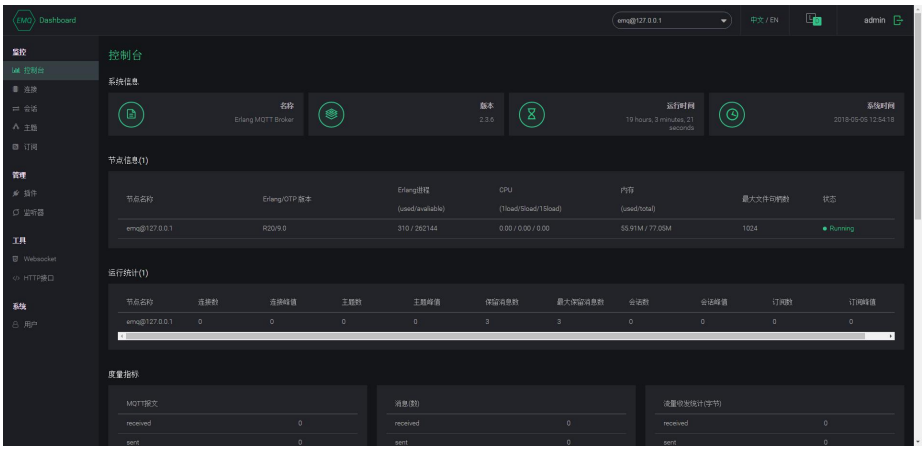
软件源码



6.2 Web

Web 是一种基于超文本和 HTTP 的，全球性的，动态交互的，跨平台访问的分布式图形信息系统，为浏览者在 Internet 上查找和浏览信息提供图形化，易于访问的直观界面。

本项目 Emqtt 管理界面



本项目温湿度传感器数据展示界面：

温湿度

时间	温度	湿度	操作
2018-05-02 12:07:27	26	89	刷新

6.3 Ajax

Ajax 是一种创建交互式网页应用的网页开发技术，是异步的 JavaScript 和 XML，是一种用于创建快速动态网页的技术，是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。通过在后台与服务器进行少量数据交换，Ajax 可以使网页实现异步更新，这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新，这比传统的网页更加优化，更加快速。

本项目中使用 Ajax 访问 RESTful API 进行数据资源获取，并展示到 Web 中。

全部代码：

```
<!DOCTYPE html>
<head>
<meta charset="utf-8" />
<title>TempHum</title>
<base target="_blank" />
<link rel="stylesheet"
    href="https://cdn.bootcss.com/bootstrap/3.3.4/css/bootstrap.min.css">
<script src="https://cdn.bootcss.com/jquery/1.11.2/jquery.min.js"></script>
<script
    src="https://cdn.bootcss.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
<script type="text/javascript">
    var url = 'temphum/get/findLast';
    $.getJSON(url, function(data) {
        $('#jsonDataList').append('<h3>时间：'+data.date+'</h3>');
        $('#jsonDataList').append('<h3>温度：'+data.temperature+'</h3>');
        $('#jsonDataList').append('<h3>湿度：'+data.humidity+'</h3>');
    });
</script>
<style type="text/css">
#tetle {
    color: #0000FF;
    padding-left: 10px;
    font-size: 30pt;
    font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>
    <h1 id="tetle" class="text-muted">温湿度</h1>
    <div id="jsonDataList"></div>
</body>
</html>
```

6.4 DuerOS

DuerOS 是百度公司度秘事业部研发的对话式人工智能平台（系统），搭载 DuerOS 的设备可以与用户进行自然语言的对话式交互，实现影音娱乐，信息查询，生活服务，出行路况等 10 大类目 200 多项功能，DuerOS 广泛适用于智能音箱，手机等小家电和各种场景及设备。DuerOS 还推出了小度智能设备平台和小度技能开放平台，在 DuerOS 技能平台中，个人开发者可以开发技能，分内容播报，自定义和智能家居技能。我个人已经在平台上线了两个自定义技能，分别是“生肖查询”和“猜字谜小游戏”。

本项目中开发的是智能家居技能，通过该技能，可以控制项目规划中的台灯，空调，也可以查询温湿度传感器的数据并通过搭载 DuerOS 的智能音箱播放出来。平台服务配置如下：

6.5 AliGenie

AliGenie 是阿里巴巴第一代中文人机交流系统，它具有自然语言处理、开放平台、深度学习、知识图谱等等人工智能能力，让 X1 具有理解力、执行力甚至是进化力。开发者平台是阿里巴巴人工智能实验室推出的一款面向国内外企业、中小创业团队和个人开发者的语音平台。AliGenie 开发者平台主要包括三大部分：精灵技能市场、硬件开放平台、行业解决方案，全面赋能智能家居、新制造、新零售、酒店、航空等服务场景

天猫精灵智能家居技能授权及网关配置：

7. 结论与鸣谢

7.1 项目结论

物联网最重要的是数据获取而并非设备控制。经过长期的发展，物联网获取到的数据将是大量的，具有价值的。通过大数据的算法对数据进行数据分析和数据挖掘，为人类决策提供科学的建议，为人工智能的发展提供数据支撑。在未来，物联网将是当今的互联网，是一个或不可缺的基础智能服务。物联网是大数据重要来源，大数据为物联网数据分析，挖掘提供支撑；物联网为云计算提供应用领域，云计算为物联网提供海量数据存储能力和计算能力；云计算为大数据提供用技术基础，大数据为云计算提供用武之地。

物联网的普及是一个漫长的征程，不仅需要产品的迭代琪，也需　　要考虑用户的接受程度，因为物联网的数据涉及到用户隐私，涉及到用户的方方面面。当数据保存处理不安全时，甚至涉及到用户的人身安全。所以，物联网的发展空间很大，但需要考虑的方面也很多，物联网的进步与普及任重道远。

7.2 参考文献

a. MQTT 官方文档
b. 树莓派官方文档
c. 百度 DuerOS 官方文档
d. 阿里 AliGenic 官方文档
e. NodeMCU 官方文档
f. OAUTH 官方文档
g. Raspberry Pi 官方文档
h. Arduino 官方文档
i. ESP8266 官方文档
j. JSON 官方文档
k. MySQL 官方文档
l. Tomca 官方文档
m. EMQTT 官方文档
n. Ajax 官方文档

7.3 关于开源

本项目完全开源，代码托管于 GitHub
https://github.com/huangdayu/yibuwulianwang