



基于人工智能的智慧农业监控系统

作品设计方案



目录

一、概述.....	1
1.1 开发背景.....	1
1.2 应用领域.....	1
二、系统设计架构.....	1
2.1 系统架构图.....	1
2.2 产品技术流程图.....	2
2.3 数据库结构设计.....	2
2.3.1 数据库逻辑设计.....	2
2.4 运行和开发环境.....	3
三、功能模块设计.....	3
3.1 功能分析划分.....	3
四、详细设计.....	7
4.1 前端技术细节.....	7
4.2 树莓派技术细节.....	9
4.2.1 树莓派简介.....	9
4.2.2 传感器介绍.....	9
4.3 分布式技术细节.....	12
4.4 数据传输细节.....	13
4.5 人工智能实现细节.....	14
五、系统测试.....	15
5.1 测试计划.....	15
5.1.1 测试范围.....	15
5.1.2 测试环境.....	15
5.1.3 测试进度.....	15
5.2 测试用例.....	15
六、总结.....	17
七、成员组成.....	18

一、概述

1.1 开发背景

2015年国务院印发了《关于积极推进“互联网+”行动的指导意见》，将“互联网+”现代农业作为11项重点行动之一，明确提出利用互联网提升农业生产、经营、管理和服务水平，促进农业现代化水平明显提升的总体目标。2016年底，国务院先后印发《全国农业现代化规划（2016-2020年）》《“十三五”国家信息化规划》，对全面推进农业农村信息化作出总体部署。发展智慧农业已成为发展的必然路径。

中投顾问在《2016-2020年中国智慧农业深度调研及投资前景预测报告》中提到，以应用（硬件和网络平台及服务）为基础的智慧农业市场，有望从2016年的90.2亿美元（约为613亿人民币）达到2022年的184.5亿美元（约为1254亿人民币）的规模，年均复合增长率13.8%。

本产品核心在于解决目前农业监管人工成本过高，处理不及时的问题。通过树莓派收集数据上传到服务器，通过人工智能分析数据，根据温湿度，光照强度对树莓派发出不同的指令。同时用户可以通过浏览器查看当前数据和历史数据，或者通过浏览器向树莓派发出指令。

1.2 应用领域

基于人工智能的远程监控系统是一款帮助农业从业人员整合信息，远程控制，智能处理的系统，主要用在大棚，室外种植等情况。

二、系统设计架构

2.1 系统架构图

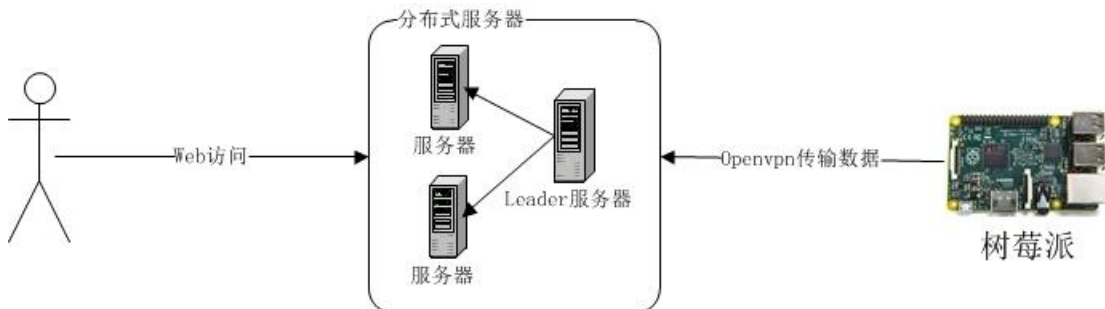


图 2-1 系统架构图

2.2 产品技术流程图

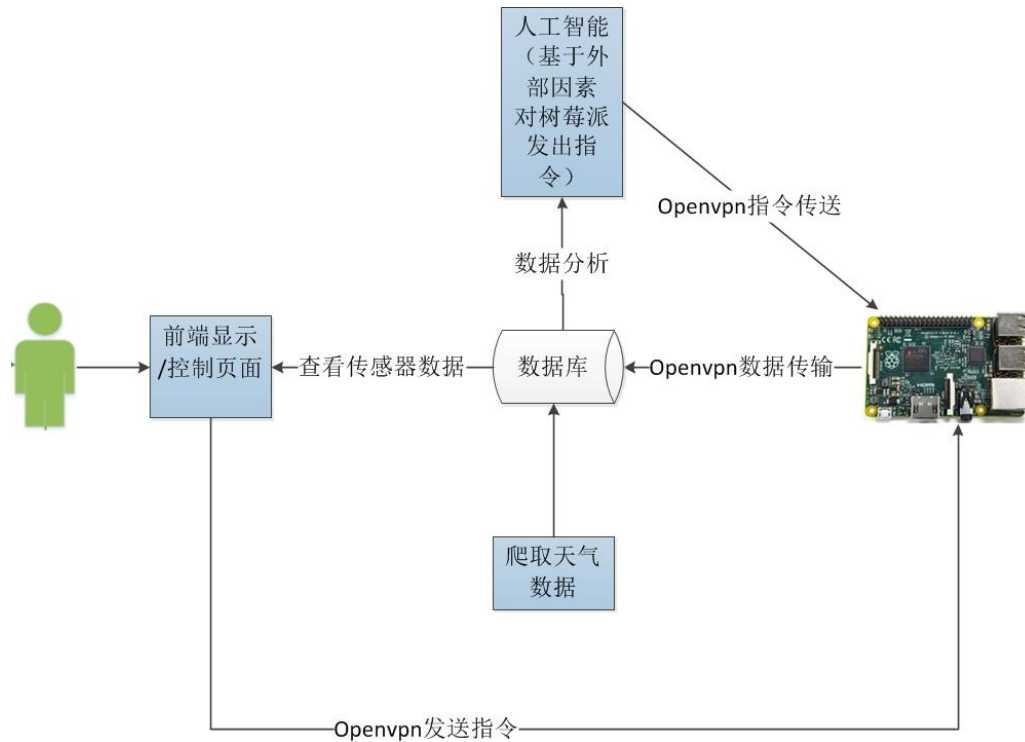


图 2-2 技术流程图

2.3 数据库结构设计

2.3.1 数据库逻辑设计

数据表 data				
代表意义	列名	数据类型	约束	备注
时间节点	Time	Datetime	Primary key	主键
温度	temperature	Double	Not null	摄氏度
湿度	humidity	Double	Not null	百分比形式
光强	light	Double	Not null	坎德拉单位
雨滴信号	Rain	Int	Not null	1 为有雨滴, 0 为无雨滴
有害气体浓度	harmgas	Double	Not null	百分比形式
人体检测信号	Person	Int	Not null	1 为有人在附近, 0 位无人在附近

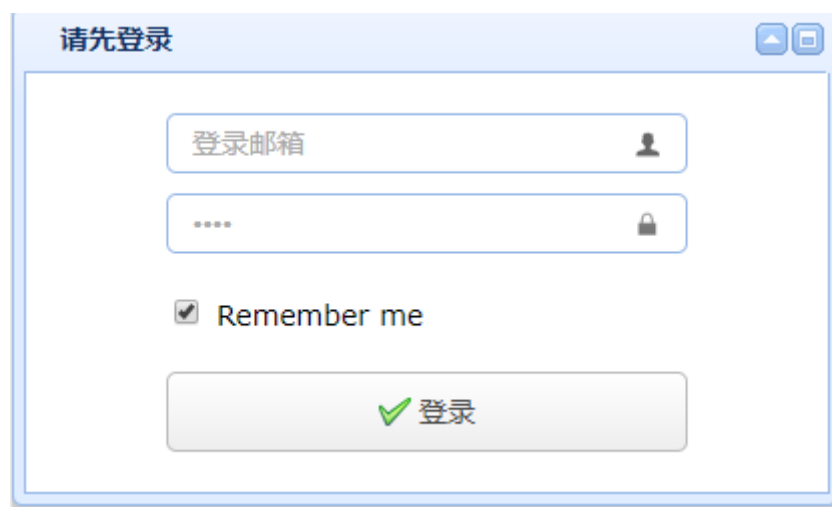
2.4 运行和开发环境

运行环境：普华服务器系统（可跨平台）树莓派

开发环境：idea idle

三、功能模块设计

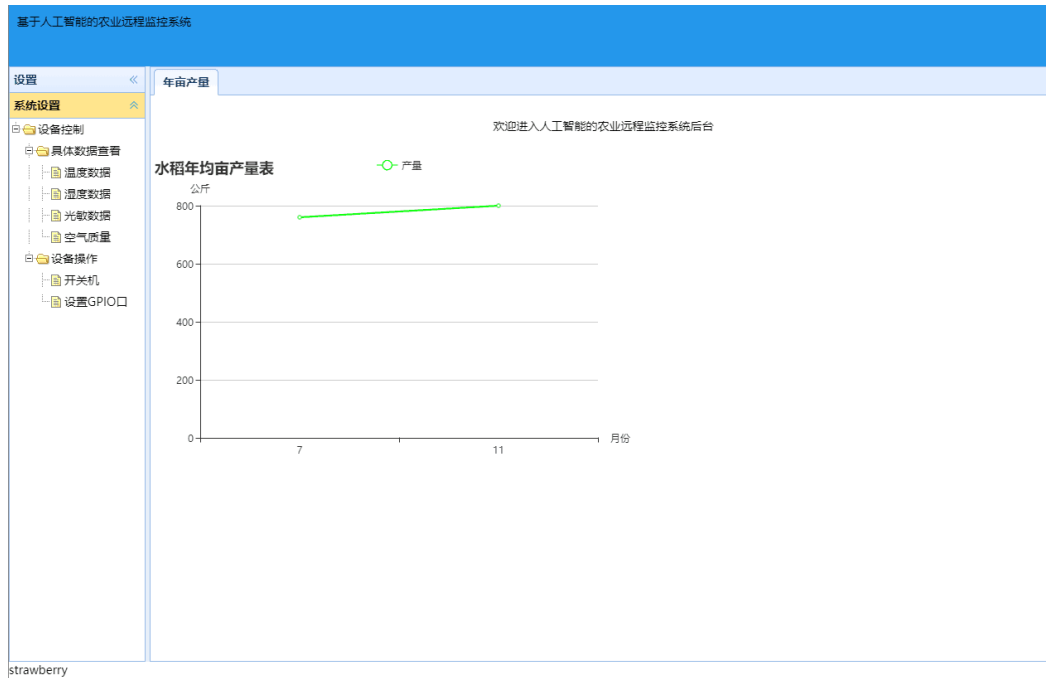
3.1 功能分析划分



用户登录界面

用户进入后台界面后点击左侧的系统设置后可以通过该后台管理平台查看树莓派采集的温度及湿度数据以及对设备进行操作

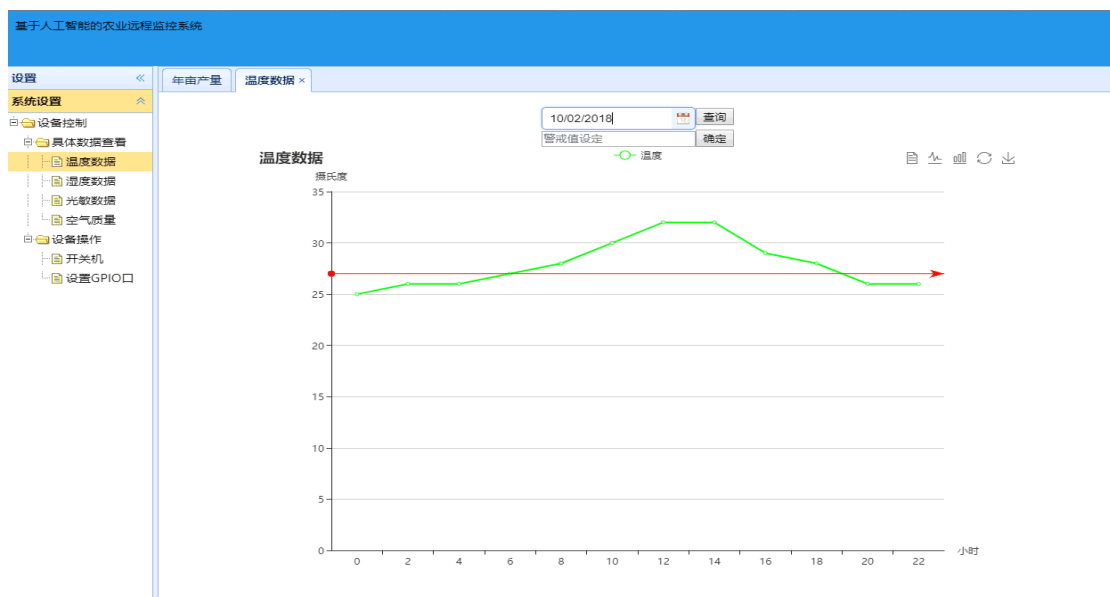
温度、湿度、空气质量等数据查看界面通过 servlet 从数据库获取数据并将数据以 json 的形式传到前端，然后通过 Echarts 实现数据可视化并可以通过 Echarts 右上角的工具栏进行样式的切换和保存数据图片。



strawberry

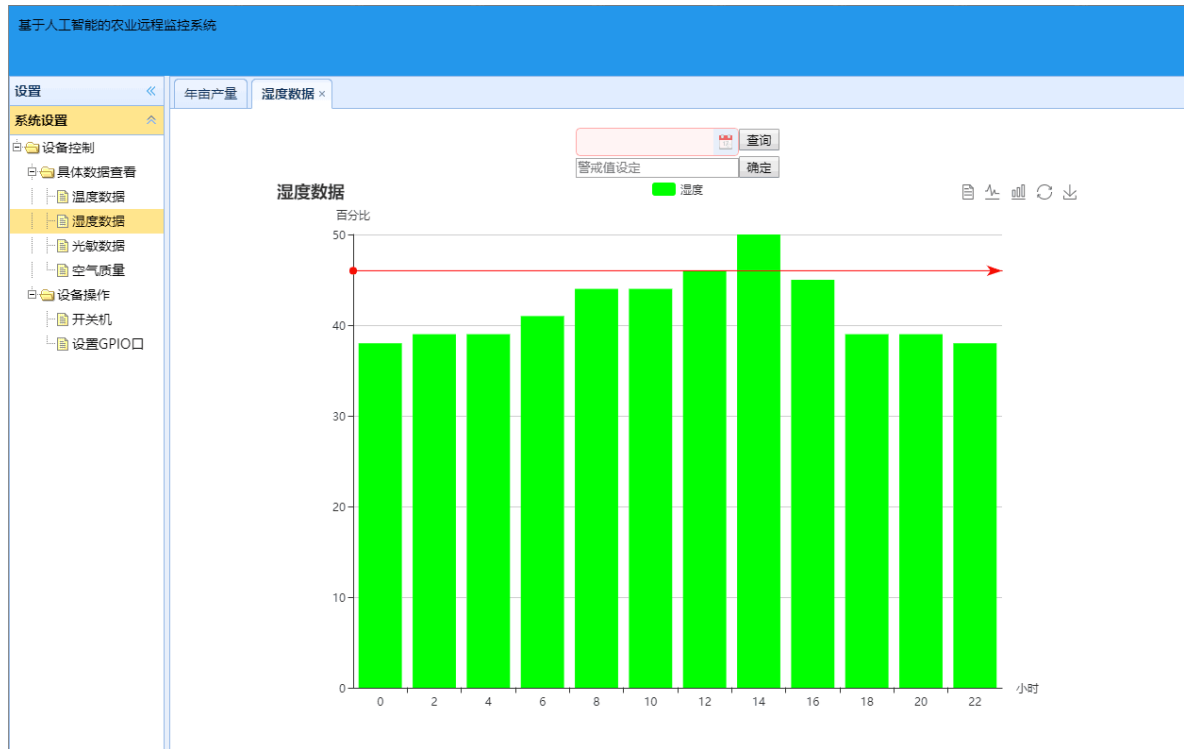
作物产量界面

通过后台分析农作物不同产量对应的温湿度，光照等数据，预测出怎么样的温湿度，光照强度搭配可以产生更高的产量。从而获得更大的收益。



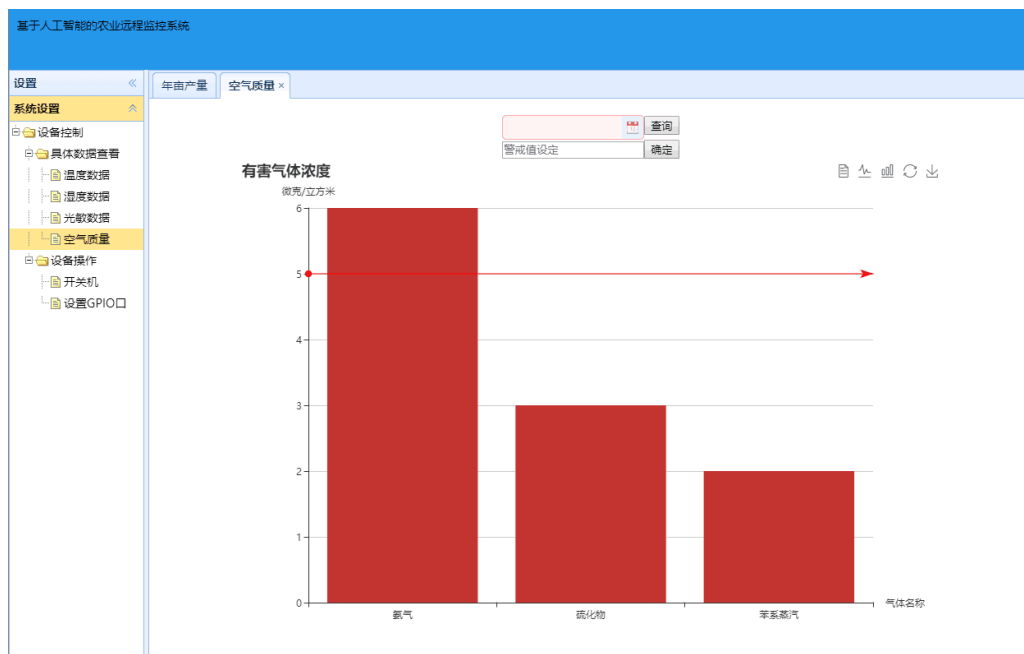
温度数据显示

该界面可以实时查看当前温度数据也可以选择日期显示历史数据。如果温度低于管理员设定温度可以自动打开大棚中的加热装置，或者手动强制打开。



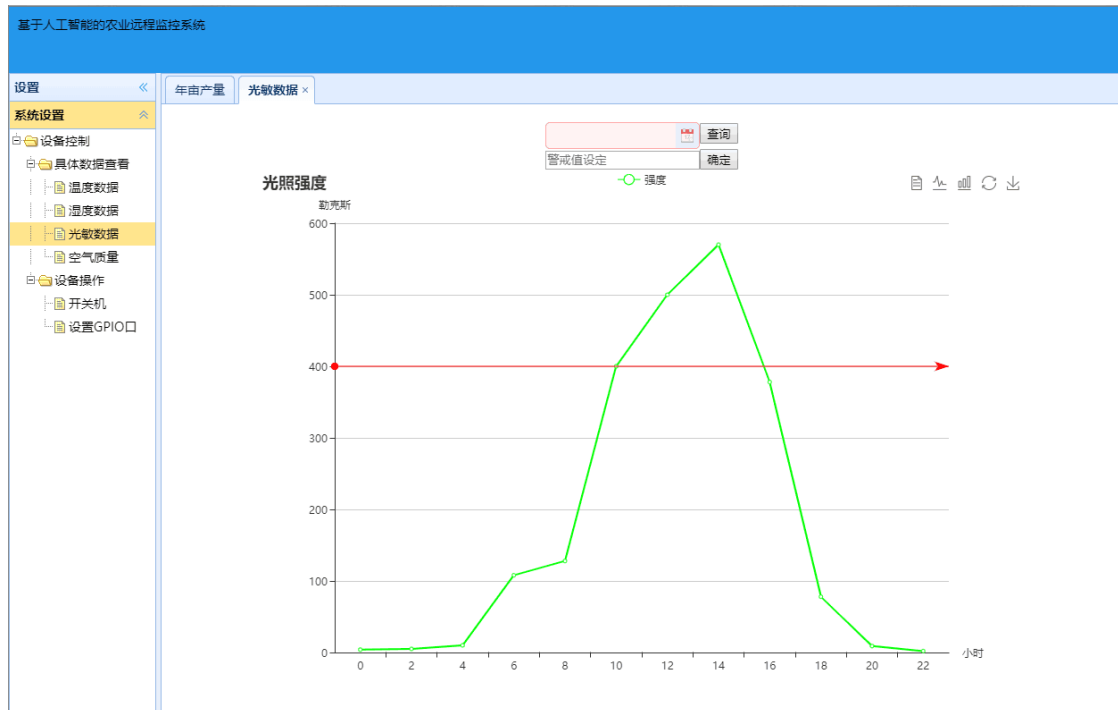
湿度数据显示

该界面可以实时查看当前湿度数据也可以选择日期显示历史数据。如果湿度低于管理员设定湿度将会自动打开滴灌系统。



空气质量数据显示

在农业生产过程中，常常存在着很多工厂所排放的污染气体，由于农户缺乏足够的技术水平，所以通过此页面可以轻松的看到所在农地的空气质量并及时进行防护。



光照情况显示

该界面可以实时查看当前光照强度数据也可以选择日期显示历史数据。如果光照强度低于管理员设定值，将会自动打开灯光系统。



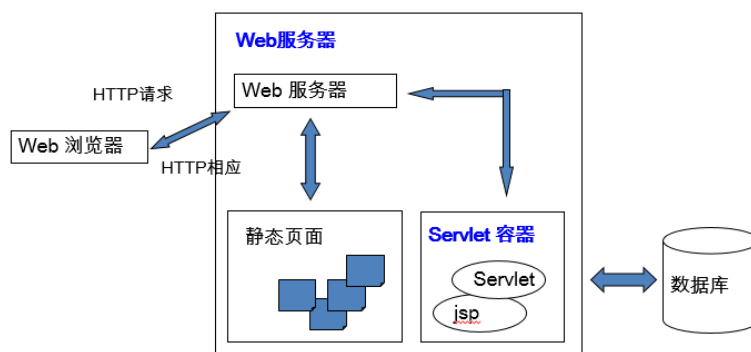
设备控制

通过控制界面我们可以在检测仪未达到管理员设定的临界值时强制开启或关闭设备，以应对突发情况。

四、详细设计

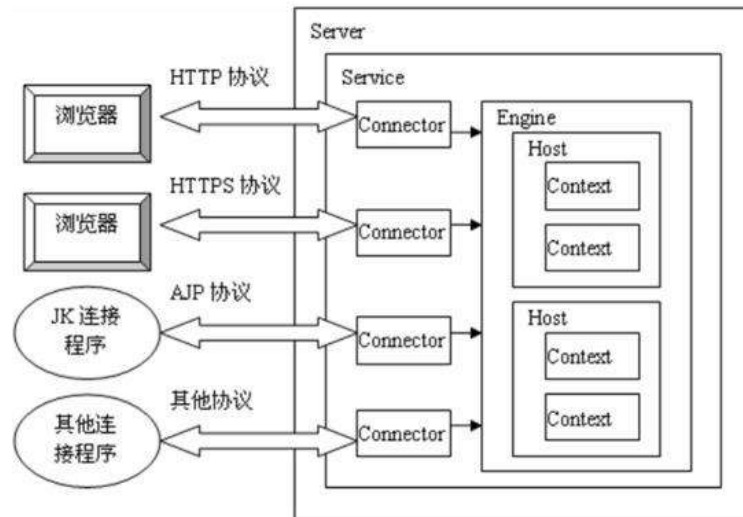
4.1 前端技术细节

JavaWeb 技术：JavaWeb 也就是 J2EE，JavaWeb 主要是使用各种 Java 企业级技术来解决相关 web 互联网领域的技术总和。而且这些技术有一个标准也就是 J2EE 规范，J2EE 规范是这样定义 J2EE 组件的：客户端应用程序和 applet 是运行在客户端的组件；Java Servlet 和 Java Server Pages (JSP) 是运行在服务器端的 Web 组件；Enterprise Java Bean (EJB)组件是运行在服务器端的业务组件。J2EE 组件和“标准的” Java 类的不同点在于：它被装配在一个 J2EE 应用中，具有固定的格式并遵守 J2EE 规范，由 J2EE 服务器对其进行管理。所以 web 包括：web 服务器和 web 客户端两部分。Java 在服务器端的应用非常的丰富，比如 Servlet, JSP 和第三方框架等等。



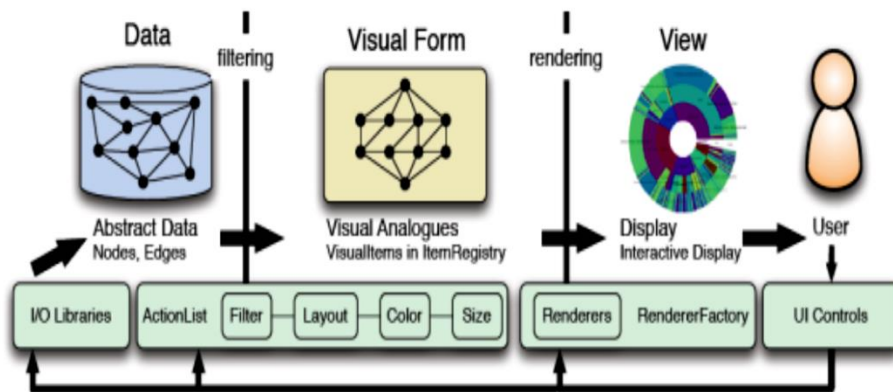
JavaWeb 技术

Tomcat 技术：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器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。Tomcat 主要组件：服务器 Server，服务 Service，连接器 Connector、容器 Container。连接器 Connector 和容器 Container 是 Tomcat 的核心。一个 Container 容器和一个或多个 Connector 组合在一起，加上其他一些支持的组件共同组成一个 Service 服务，有了 Service 服务便可以对外提供能力了，但是 Service 服务的生存需要一个环境，这个环境便是 Server，Server 组件为 Service 服务的正常使用提供了生存环境，Server 组件可以同时管理一个或多个 Service 服务。



Tomcat 技术

数据可视化技术: 数据可视化主要旨在借助于图形化手段, 清晰有效地传达与沟通信息。与信息图形、信息可视化、科学可视化以及统计图形密切相关。当前, 在研究、教学和开发领域, 数据可视化乃是一个极为活跃而又关键的方面。“数据可视化”这条术语实现了成熟的科学可视化领域与较年轻的信息可视化领域的统一。



数据可视化技术

其中本项目使用了 JavaWeb 技术搭建前端的整体框架, Servlet 实现交互式地浏览和修改数据, 生成动态 Web 内容, Ajax 通过在后台与服务器进行少量数据交换, 使网页实现异步更新。使用 JavaScript 实现的开源可视化库——ECharts, 依赖轻量级的矢量图形库 ZRender, 提供直观, 交互丰富, 可高度个性化定制的数据可视化图表。

4.2 树莓派技术细节

4.2.1 树莓派简介

Raspberry Pi(中文名为“树莓派”,简称为 RPi, (或者 RasPi / RPI), 是为学习计算机编程教育而设计), 只有信用卡大小的微型电脑, 其系统基于 Linux。随着 Windows 10 IoT 的发布, 我们也将可以用上运行 Windows 的树莓派。

自问世以来, 受众多计算机发烧友和创客的追捧, 曾经一“派”难求。别看其外表“娇小”, 内“心”却很强大, 视频、音频等功能通通皆有, 可谓是“麻雀虽小, 五脏俱全”。

它是一款基于 ARM 的微型电脑主板, 以 SD/MicroSD 卡为内存硬盘, 卡片主板周围有 1/2/4 个 USB 接口和一个 10/100 以太网接口 (A 型没有网口), 可连接键盘、鼠标和网线, 同时拥有视频模拟信号的电视输出接口和 HDMI 高清视频输出接口, 以上部件全部整合在一张仅比信用卡稍大的主板上, 具备所有 PC 的基本功能只需接通显示屏和键盘, 就能执行如电子表格、文字处理、玩游戏、播放高清视频等诸多功能。Raspberry Pi B 款只提供电路板, 无内存、电源、键盘、机箱或连线。

4.2.2 传感器介绍

本产品共使用了五种传感器, 一个摄像头, 一个 2 路 5V 继电器, 通过进行收集现场温度、湿度、光强、空气质量、雨滴信号数据并进行上传至服务器。

4.2.2.1 雨滴模块传感器

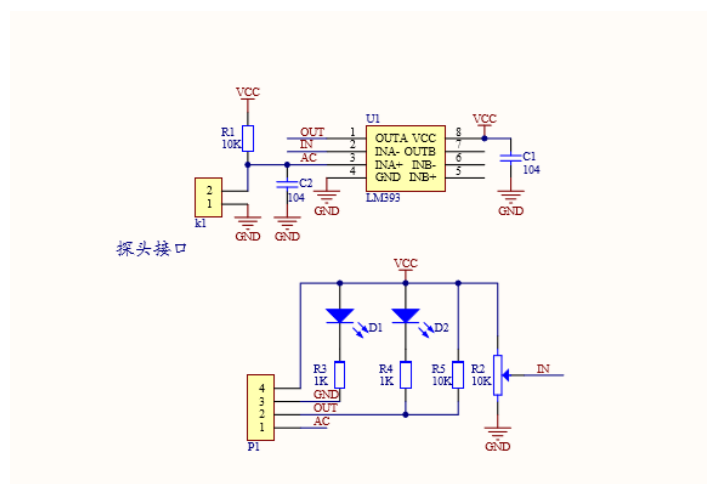


图 雨滴模块传感器原理图

雨滴传感器一般应用在机器人身上, 用于水分以及雨滴的报警传感器, 便于机器人与水的感知以及及时作出反映, 同时, 对于拥有大面积农地的农户来说, 对于天气的监测是一件麻烦的事情。而且本元器件重要的作用是给自身以报警, 做好保护自身的防护。

4.2.2.2 MQ-135 空气质量传感器——有害气体检测模块

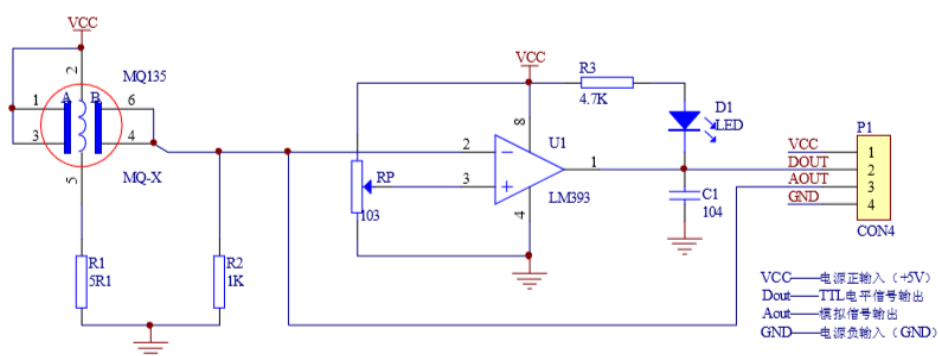


图 MQ-135 空气质量传感器原理图

MQ-135 空气质量传感器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_2)。当传感器所处环境中存在污染气体时，传感器的电导率随着空气中污染气体浓度的增大而增加。使用简单的电路可以将电导率的变化转化为与该气体浓度相对应的输出信号。

MQ-135 传感器对氨气，硫化物、苯系蒸汽的灵敏度很高，对烟雾和其他有害的检测也很理想，在农业生产过程中，常常存在着很多工厂所排放的污染气体，由于农户缺乏足够的技术水平，所以通过此传感器可以在 web 端轻松的看到所在农地的空气质量并及时进行防护。

4.2.2.3 2 路 5V 高电平出发继电器

高电平出发是指信号输入端与地之间有电压的触发方式，可以理解为信号输入端与 VCC 正极短路触发的一种方式。

使用继电器的意义在于控制更多的大型电器，例如让农户头痛的大面积农药喷洒以及对农地进行洒水控制，这都需要继电器来进行控制，

4.2.2.4 光敏电阻传感器

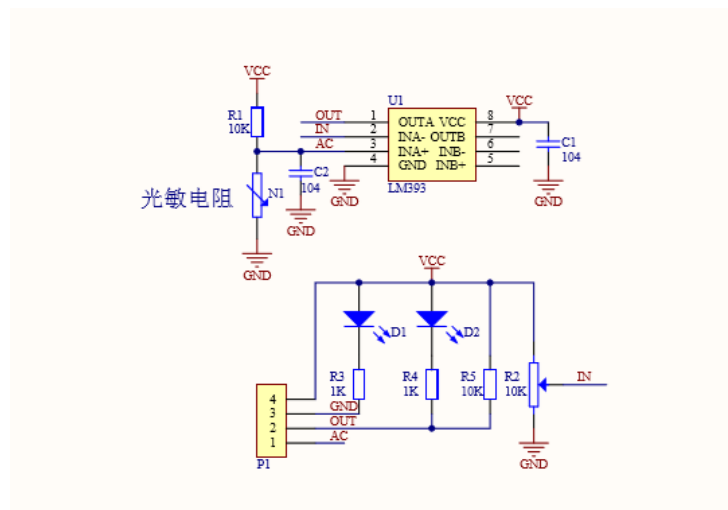


图 光敏电阻传感器原理图

光敏电阻传感器主要用于光线亮度检测，光线亮度传感，重要的应用就是智能车的寻光模块，对环境光线十分敏感，一般用来检测周围环境光强。在本应用中最大的作用就是精确的得知每天的太阳升起的时间并且自控制启动日常工作流水程序。

4.2.2.5 HC-SR501 人体红外感应电子模块

HC-SR501 是基于红外线技术的自动控制模块，采用 LH1778 探头设计，灵敏度高，可靠性强，低电压工作模式，广泛应用于各类自动感应电气设备。主要功能为全自动感应人体存在，当有人进入其感应范围则输出高电平，人离开则关闭高电平输出低电平。作为报警系统的基本元器件，用于保护自身不被攻击。

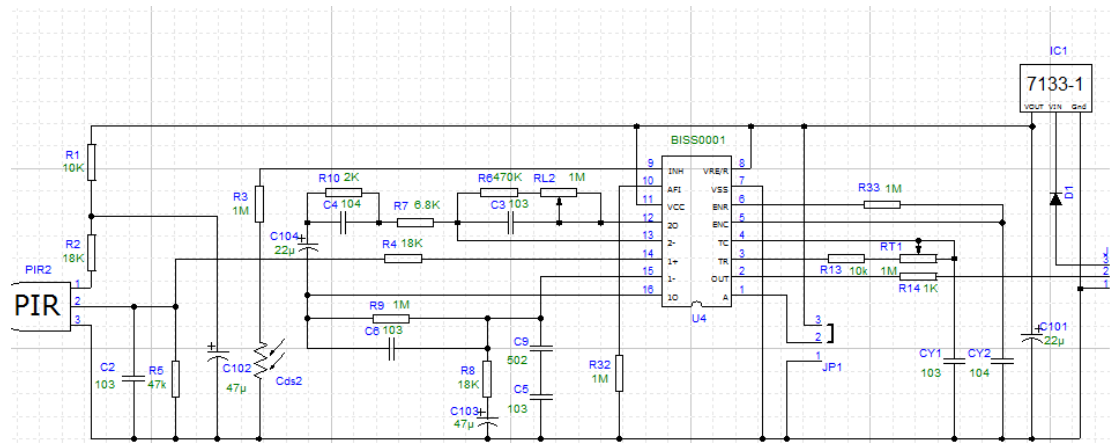


图 HC-SR501 人体红外感应电子模块原理图

4.2.2.6 DHT11 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与卓越的长期稳定性。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储存在 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，信号传输距离可达 20 米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。产品为 4 针单排引脚封装。

在农场传感器中最重要的传感器就是温湿度传感器，通过不同的温湿度给农户最真实的现场数据，以得出比较科学并且足够准确的农业推断。

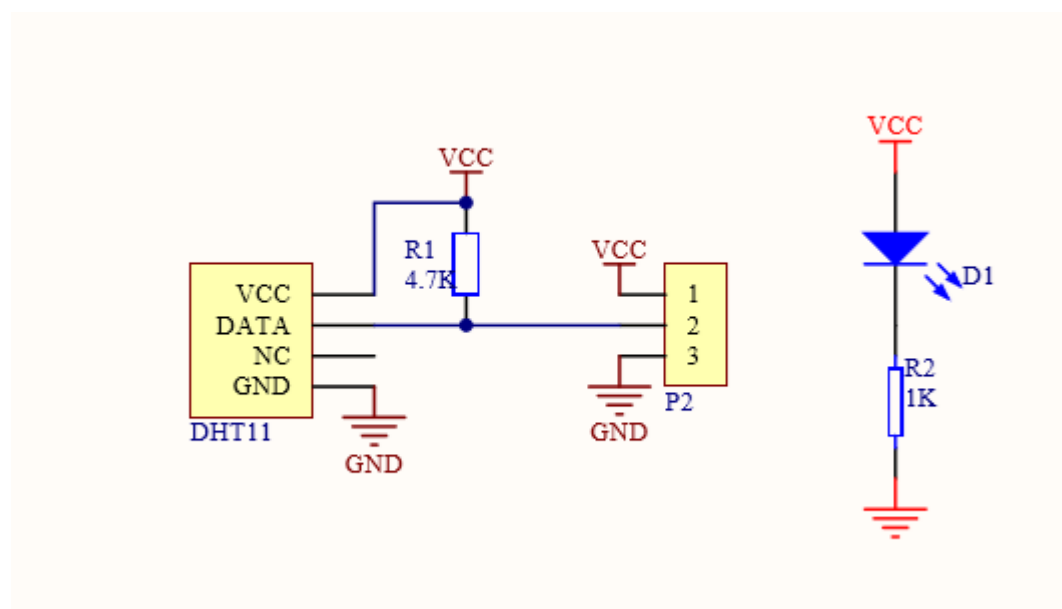


图 DHT11 传感器模块原理图

4.2.2.7 500 万像素摄像头

主要用于拍摄现场情况，或者将视频信号通过网络传输出去，便于用户在远程监测农地现场情况。

4.3 分布式技术细节

Zookeeper 是一个分布式的，开放源码的分布式应用程序协调服务，是 Google 的 Chubby 的一个开源的实现，是 Hadoop 和 Hbase 的重要组件，它是一个为分布式应用提供一致性服务的软件，提供的功能包括配置维护，域名服务，分布式同步，组服务等。zooKeeper 包含一个简单的原语集，提供 Java 和 C 的接口。Zookeeper 作为一个消息中间件，可以提

供协调服务，同时，作用于分布式系统，可以发挥其优势，为大数据提供服务。

分布式系统：很多计算机组成一个整体，一个整体一致对外并且处理同一个请求，其内部的每台计算机都可以相互通信，(rest/rpc)，客户端到服务端的一次请求到相应结束会经历多台计算机，通过分布式技术的运用，可以大大提高用户端请求的负载量，从而提高其运行效率，同时分布式系统有主备之分，即使主机当机，也不会影响整体的运行，保证数据传输的稳定性，zookeeper 可以将主备数据实时同步，保证高并发性下的数据的一致性，同时数据也是按照顺序分批入库，便于管理，zookeeper 还具有原子性，所谓原子性是事务要么成功要么失败，不会部分成功而部分失败，而且具有单一视图，即客户端连接集群中的任一 zk 节点，数据都是一致的。Zookeeper 还具有可靠性，每次对 zk 的操作都会保存在服务端，可以查询到每次的操作以及进行回滚等操作。

Zookeeper 类似于一个树性结构，也可以理解为 Linux 或 Unix 的文件目录，每一个节点称之为 znode，可以有子节点，也可以有数据，节点分为永久节点以及临时节点，永久节点可以实现持久化，临时节点在 zookeeper 的 section 失效后 10s 左右会丢失，同时，每个 zookeeper 节点都有自己的版本号，可以通过命令行显示节点信息，每当节点数据发生改变，会增加版本号，对于后续的操作提供便利。

Zookeeper 拥有自己的四字命令，可以通过其和服务器进行交互，方便运维人员进行监控。

通过配置三台虚拟机，将 zooKeeper 集群搭建完毕

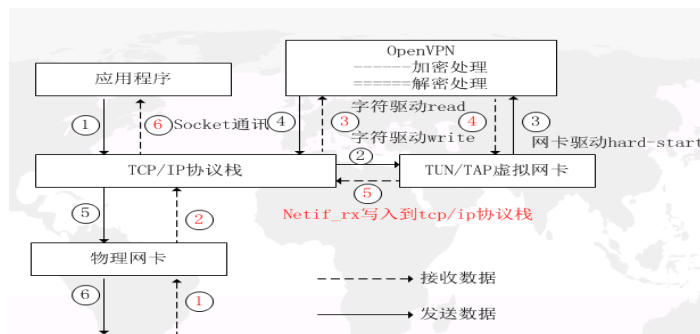
```
server.1=192.168.43.228:2888:3888
Server.2=192.168.43.229:2888:3888
server.3=192.168.43.230:2888:3888
```

通过配置文件实现对 leader 以及 slaves 的确定，zookeeper 的好处在于如果 leader 当机，其余的 slave 会通过 zookeeper 的选举机制立刻选出一个新的 leader 来进行数据的处理，同时，当旧的 leader 重新启动运行，并不会取代当前的 leader 从而保证数据传输的稳定性，通过设置 watch 对节点进行监控，通过 acl 设置权限，保证数据不会被误修改。

4.4 数据传输细节

为了确保数据通道传输的安全性，项目使用了 Openvpn 搭建 vpn 服务，确保公网下的数据传输安全。

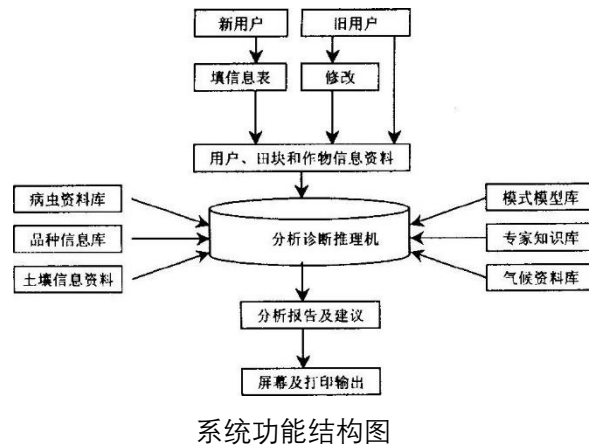
OpenVPN 是一个用于创建虚拟专用网络加密通道的软件包，它大量使用了 OpenSSL 加密库中的 SSLv3/TLSv1 协议函数库。通过在服务器上增加一个虚拟网卡，从而让数据通过这个虚拟隧道进行加密传输。



项目中使用 ccd 配置为服务器和客户机配置指定 ip,通过 management 配置对客户端进

行实时查看和管理,同时启用 comp-lzo 压缩降低数据传输量减轻隧道负担,使用 AES-128-CBC 加密算法,选择 SHA256 做为 HMAC 消息摘要算法为数据传输提供安全保障。

4.5 人工智能实现细节



BP 神经网络是一种前馈型网络,由输入层、隐含层和输出层构成。它的特点是信号由输入层单向传输到输出层,同一层神经元之间不传递信息,每个神经元与邻层所有神经元相连,连结权重用 W_{ij} 表示,各神经元的作用函数为 Sigmoid 函数。构建 BP 网络模型通常包括其输入输出、作用函数、误差计算和自学习四部分。各模型如下:

(1) 节点输出模型

隐节点输出模型:

$$O_j = f(\sum W_{ij} \times X_i - \theta_j) \quad (1)$$

输出节点输出模型:

$$Y_k = f(\sum T_{jk} \times O_j - \theta_k) \quad (2)$$

f 为非线性作用函数; θ 为神经单元阈值。

(2) 作用函数模型

作用函数是反映下层输入对上层节点刺激脉冲强度的函数又称刺激函数,一般取为 (0, 1) 内连续取值 Sigmoid 函数:

$$f(x) = 1 / (1 + e^{-x}) \quad (3)$$

(3) 误差计算模型

误差计算模型是反映神经网络期望输出与计算输出之间误差大小的函数:

$$E_p = 1/2 \times \sum (t_{pi} - o_{pi})^2 \quad (4)$$

t_{pi} - i 节点的期望输出值; o_{pi} - i 节点计算输出值。

(4) 自学习模型神经网络的学习过程,即连接下层节点和上层节点之间的权重矩阵 W_{ij} 的设定和误差修正过程。

$$\Delta W_{ij}(n+1) = h \times \Phi_i \times O_j + a \times \Delta W_{ij}(n) \quad (5)$$

h -学习因子; Φ_i -输出节点 i 的计算误差; O_j -输出节点 j 的计算输出; a -

动量因子。

通过反复运用以上(3)、(4)两个模型,直至网络输出与期望输出间的误差满足一定的要求。

五、系统测试

5.1 测试计划

5.1.1 测试范围

1. 用户界面测试: 主要测试系统的功能, 操作性, 人机对话, 系统界面, 安全性等。主要参考对象为业主用户
2. 功能测试: 主要测试系统是否实现预计结果, 此测试为软件的基本测试, 主要参考对象为管理人员, 开发人员。
3. 服务器测试: 主要测试分布式服务器之间的连接与同步。

5.1.2 测试环境

服务器操作系统: 普华操作系统 4.0 (服务器版)

数据收集端: 树莓派

Web 服务器: Tomcat7.0

5.1.3 测试进度

任务	开始时间	结束时间	工作时间
界面测试	2018.10.01	2018.10.05	5
数据收集测试	2018.10.04	2018.10.20	17
Openvpn 连接测试	2018.10.6	2018.10.10	5
分布式测试	2018.10.10	2018.10.27	18
功能对接测试	2018.10.27	2018.10.30	4
人工智能测试	2018.10.15	进行中	

5.2 测试用例

测试方法: 因果图法

用例编号	用例名称	操作说明	预期结果	实际结果
1	登录	1.在登录页面输入正确的账号密码 2.点击登录	显示监控系统首页	显示监控系统首页
2	登录	1.在登录页面输入错误的账号密码 2.点击登录	提示账号或密码错误	提示账号或密码错误
3	温度显示	1.进入首页点击温度数据	显示实时温度数据	显示实时温度数据
4	温度显示	1.进入首页点击温度数据 2.选择日期	显示选择时间内的温度数据	显示选择时间内的温度数据
5	产量数据	1.进入首页点击产量数据 2.选择日期	显示选择年份产量	显示选择年份产量
6	湿度数据	1.进入首页点击湿度数据	显示当前湿度	显示当前湿度
7	湿度数据	1.进入首页点击产量数据 2.选择日期	显示选择时间湿度	显示选择时间湿度
8	光照数据	1.进入首页点击光照数据	显示当前光照强度	显示当前光照强度
9	光照数据	1.进入首页点击光照数据 2.选择日期	显示选择时间内光照强度	显示选择时间内光照强度
10	空气质量数据	1.进入首页点击空气质量	显示当前空气质量	显示当前空气质量
11	空气质量数据	1.进入首页点击空气质量 2.选择日期	显示选择时间内空气质量	显示选择时间内空气质量
12	温控系统	1.进入首页点击设备操作 2. 点击设置 GPIO口 3.选择温控系统点击关闭	温控系统指示灯灭	温控系统指示灯灭
13	温控系统	1.进入首页点击设备操作 2. 点击设置 GPIO口 3.选择温控系统点	温控系统指示灯亮	温控系统指示灯亮

		击开启		
14	湿度控制系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择湿度控制系统点击关闭	滴灌系统指示灯灭	滴灌系统指示灯灭
15	湿度控制系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择湿度控制系统点击开启	滴灌系统指示灯亮	滴灌系统指示灯亮
16	光照控制系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择亮度控制系统点击关闭	光照系统指示灯灭	光照系统指示灯灭
17	光照控制系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择亮度控制系统点击开启	光照系统指示灯亮	光照系统指示灯亮
18	空气净化系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择空气净化系统点击关闭	空气净化系统指示灯灭	空气净化系统指示灯灭
19	空气净化系统	1.进入首页点击设备操作 2. 点击设置 GPIO 口 3.选择空气净化系统点击开启	空气净化系统指示灯亮	空气净化系统指示灯亮

六、总结

通过本次大赛，小组全体以及导师无不倾其所有，从项目之初的规划，需求

分析，到之后的分工以及最后实现，所有人都付出了很多，同时，每位成员也都从各自的模块中学到了许多东西，像如何用树莓派通过控制芯片达到对数据的实时监控，通过 openVPN 实现数据的加密以及解密实现数据的安全保障，通过 bp 算法实现深度学习使得其可以智能分析种何种作物收益最大，通过对 zookeeper 分布式集群的搭建，对 zk 的选举机制以及整个分布式的应用有了更深层次的了解，通过大赛去学习去探索新的知识，丰富自己的人生阅历，充实自己。

七、成员组成

姓名	联系方式 (电话)
曹黔峰	18379452273
李子健	15004716512
陈镇东	13361618521
胡育诚	13065135186