

基于树莓派和云平台的智能灌溉系统^①



赵苏徽¹, 陈晓^{1,2}

¹(南京信息工程大学 电子与信息工程学院, 南京 210044)

²(南京信息工程大学 江苏省大气环境与装备技术协同创新中心, 南京 210044)

通信作者: 陈晓, E-mail: chenxiao@nuist.edu.cn

摘要: 针对目前农业灌溉用水浪费的情况, 提出了一种以树莓派为主控器件、云平台为监控的智能灌溉系统. 本文以农作物为灌溉对象, 分析了模糊控制算法、树莓派 4B、传感技术、HTTP 协议、MySQL 数据库和 OneNET 云平台等技术的特点, 设计了一种能够根据环境状况自动灌溉的智能灌溉系统. 系统的硬件部分能完成空气温湿度、土壤湿度的采集. 软件部分考虑云平台技术开发成本低、功能上完全可以替代传统的 Web, 而选择在云平台上开发了应用平台, 能控制灌溉并把数据上传至 OneNET 云平台, 同时将环境数据导入模糊控制器使得灌溉策略更加科学. 经过灌溉实验测试, 系统最大灌溉误差率不超过 2%, 能够很好地解决灌溉用水浪费问题, 具有广阔的应用前景.

关键词: 树莓派; 智能灌溉; 模糊控制算法; 云平台; PID; 大数据

引用格式: 赵苏徽, 陈晓. 基于树莓派和云平台的智能灌溉系统. 计算机系统应用, 2022, 31(4): 123-129. <http://www.c-s-a.org.cn/1003-3254/8456.html>

Intelligent Irrigation System Based on Raspberry Pi and Cloud Platform

ZHAO Su-Hui¹, CHEN Xiao^{1,2}

¹(School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China)

²(Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China)

Abstract: In view of the current waste of agricultural irrigation water, an intelligent irrigation system with a Raspberry Pi as the main control device and a cloud platform for monitoring is proposed. This study, with the crop as the experimental object, analyzes the characteristics of the fuzzy control algorithm, Raspberry Pi 4B, sensor technology, HTTP protocol, MySQL database, and OneNET cloud platform and designs an intelligent irrigation system that can automatically irrigate according to environmental conditions. The hardware part of the system can complete the collection of air temperature and humidity and soil humidity. In terms of the software part, considering the low development cost of cloud platform technology that can completely replace the traditional Web in function, applications are developed on the cloud platform, and they can control irrigation and upload the data to the OneNET cloud platform. Meanwhile, they can also import the environmental data into the fuzzy controller to make the irrigation strategy more scientific. In irrigation experimental tests, the maximum irrigation error of the system does not exceed 2%, which means the system can solve the problem of irrigation water waste and therefore has broad application prospects.

Key words: Raspberry Pi; intelligent irrigation; fuzzy control algorithm; cloud platform; PID; big data

① 基金项目: 江苏省 333 高层次人才培养工程 (2625)

收稿时间: 2021-07-09; 修改时间: 2021-08-13; 采用时间: 2021-08-24; csa 在线出版时间: 2022-03-22

与世界上其它国家相比,我国淡水资源总量处于前列,但是由于人口基数大使得我国人均水资源拥有量处于世界末位^[1]。许多地区缺乏水资源,当地只能通过大量开采地下水来缓解水资源危机。目前我国地下水使用量已经严重超过国际警戒线,地下水超采面积也已从 5.6 万 km² 扩大到 18 万 km²^[2]。与此同时中国大部分地区仍在使用传统的漫灌方式,这也是导致水资源危机的重要原因^[3]。从目前发展的趋势来看,尽管我国农业高效节水灌溉的总面积不断增加,但使用喷灌和微灌方式的灌溉面积仍然很小,加之对雨水利用率低,使得我国在对水资源利用的问题上与世界主要国家相比相差甚远。因此为缓解水资源短缺问题而发展先进的灌溉技术已经迫在眉睫。

国外在智能灌溉方面起步比我国要早,农业生产自动化程度高。早在上世纪 80 年代,美国的 Benami 等人^[4]开发了一个节水灌溉系统,其使用土壤湿度传感器检测土壤水分含量,将数据传给控制器,而控制器根据传感器测得的土壤相对湿度与预设值进行对比决定是否灌溉。Goap 等人^[5]基于物联网技术开发了一套智能灌溉系统,利用 ZigBee^[6]以及 WiFi 技术进行通信组网,并将传感节点的实时数据发送至远程服务器进行数据解析与存储,同时决定是否需要进行灌溉,并在网页端进行直观的显示。我国的灌溉技术虽然起步没有国外早,但现阶段的发展情况已经相差不多。刘书伦等人将物联网技术与 Android 相结合设计实现了利用 Android 平台对农业农田远程灌溉系统的控制^[7]。系统利用多种传感器组成的节点网络来实现远程监测的同时控制多个控制器节点。何江^[8]将物联网技术与树莓派相结合开发了一种智能云灌溉系统,利用了现在流行的 Web 技术,如 Python, Flask 等,实现从 Web 端进行远程控制以及数据监控,模糊算法把 PWM 作为输出控制泵的转速。

目前的智能灌溉系统大多为闭环结构,虽然闭环使整个系统流程更为简单,但由于其需要一直处于工作状态,大大增加了系统的功耗,另一方面可能会由于测量数据的微弱变化导致电机频繁地间断工作,加快了电机的使用寿命,若将灌溉条件放宽则会使系统灵敏度不够,降低了节水效果。把 PWM 作为模糊控制算法的输出,在实验室小流量的灌溉适用效果好,但要用在大规模的农田灌溉无法调节电机的 PWM,同样 PWM 如果过高则会造成灌溉时的水土流失。本文旨在开发基于树莓派和云平台的智能灌溉系统,使用空气温湿度传感器和土壤湿度传感器采集空气温湿度和土

壤湿度;采集到的数据通过 I²C 总线传输到树莓派并保存到 MySQL 数据库中;利用 Python 爬虫功能从互联网获取当地明日雨量数据并和 MySQL 数据库中保存的数据共同导入模糊控制器进行处理,计算出灌溉时长;系统根据灌溉时长控制直流蠕动泵进行自动灌溉。树莓派同时通过 HTTP 协议将数据上传到 OneNET 云平台,用户可以使用电脑端直接登录云平台或者使用设备云 APP 即可实时查看当前环境状况。系统整体为“开环”控制,通过模糊控制算法提升系统灌溉的精准度,灌溉时长作为模糊控制的输出能够适应今后大规模的农田灌溉。

1 系统整体设计

在核心控制器件选择时,对比 C51、STM32 等系列的单片机^[9-11],树莓派具有功能强大、操作方便的优点,本身既可作为下位机也可以作为上位机,所以本文选择了目前最新型号的树莓派 4B 作为核心控制器件。软件设计时考虑到云平台技术开发成本低、需求扩展灵活、数据访问的移动性强等优点,完全可以替代传统的 Web,所以并没有选择 Web 结构,而是在云平台上开发应用,降低整套系统成本。现在云平台有很多,本文使用的是中国移动的 OneNET 云平台。系统整体分为硬件和软件两大部分,其中硬件系统包括土壤湿度采集电路、空气温湿度采集电路和直流蠕动泵控制电路;软件系统包括土壤湿度采集和上传程序、空气温湿度采集和上传程序、直流蠕动泵驱动程序、气象数据获取程序、数据库设计和模糊控制器设计程序等。系统整体框图如图 1 所示。

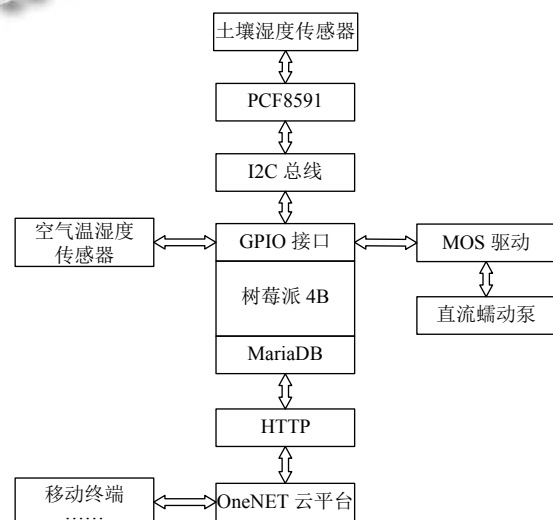


图 1 系统整体框图

2 系统硬件设计

硬件电路连接时将空气温湿度传感器 DHT11、土壤湿度传感器 YL-69 和数模转换器 PCF8591 的 VCC 端都接到树莓派的 2 号脚, GND 端都接到树莓派的 6

号脚. DHT11 的 DATA 端接到树莓派的 7 号脚, PCF8591 的 SDA 端和 SCL 端分别接到树莓派的 3 号脚和 5 号脚. MOS 管驱动的 GND 端和 PWM 端分别接到树莓派的 2 号脚和 40 号脚, 具体连接如图 2 所示.

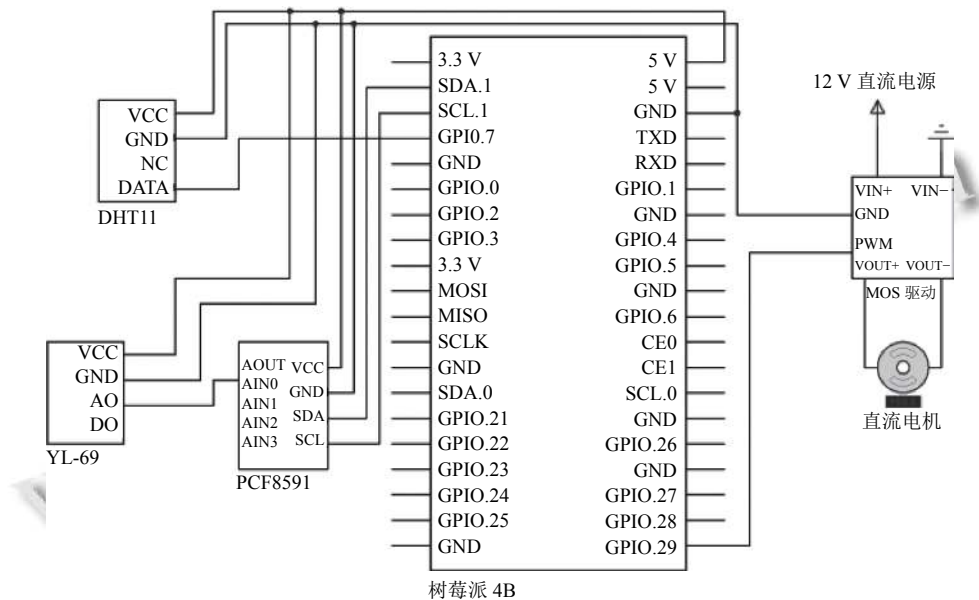


图2 智能灌溉系统整体连接图

2.1 土壤湿度采集电路设计

YL-69 土壤湿度传感器为四线制, 其采用的电源为 3.3 V~5 V, 电压比较器 LM393, DO 引脚通过和设定的阈值相比输出数字量 0 或 1, AO 引脚可以输出模拟值, 这个数值更加准确. 随着土壤湿度值变大时, 得到的模拟量值也相应的变大.

本文在使用 YL-69 土壤湿度传感器时因为要输出精确的土壤相对湿度, 所以要从 AO 引脚输出模拟量, 因此必须要有一个模数转换器将模拟信号转换为数字信号传输到树莓派. 采用的模数转换器是 PCF8591, 将 YL-69 的模拟量输出端 AO 连到 PCF8591 的模拟信号输入端 AIN0, 模数转换器将土壤湿度转换成精确的数值, 然后通过 I²C 总线传输到树莓派进行数据处理. 电路连接如图 3 所示.

2.2 空气温湿度采集电路设计

DHT11 传感器出厂时便集成了电阻式的感湿元件和 NTC 测量温度的元件, 内部已经包含了 8 位的单片机. 由于它具有串行单线接口, 所以体积小, 耗电少, 并且传输的信号最高可达 20 米. 它拥有优良的品质、超快的反应能力、强大的抗干扰能力和超高的性价比

让很多精准测量甚至可以说是苛刻测量的场所都选择它. DHT11 温湿度传感器共有 4 个引脚, 在使用时 NC 脚悬空, VCC 接 5V 直流电源, GND 接地, DATA 连接到树莓派的 GPIO.7 脚, 电路连接图如图 4 所示.

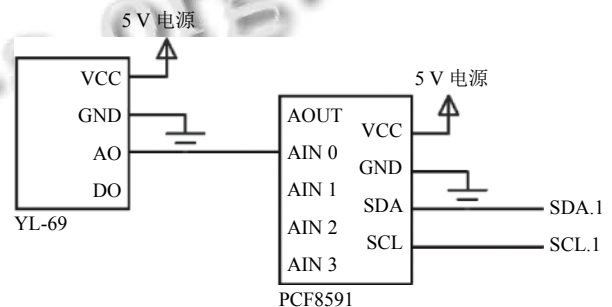


图3 土壤湿度采集电路连接

2.3 直流蠕动泵控制电路设计

本文在实验时灌溉对象是盆栽草莓, 需水量不大, 因此灌溉时采用微型蠕动泵即可. 其功率较小, 能够使用晶体管驱动, 更加适合间歇性的工作. 选用电机时, 步进电机的工作方式主要是“开环”^[12], 而且价格方面比较贵. 相比之下直流电机价格比较便宜, 并且可以通过反馈系统间接控制电机. 综合以上考虑, 本文采用直

流电机加蠕动泵的组合. 直流蠕动泵驱动电路采用 N 沟道增强型场效应管, 双 MOS 管并联结构可以使导通电阻降低一半. 与常用的三极管驱动电路 L298N 相比较, 该驱动模块少占一个输出引脚, 并且 MOS 管是电压型器件, 需要的驱动电流更小, 功耗也远低于三极管^[13]. 电路连接如图 5 所示.

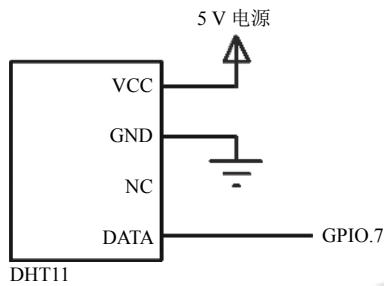


图 4 空气温湿度采集电路连接

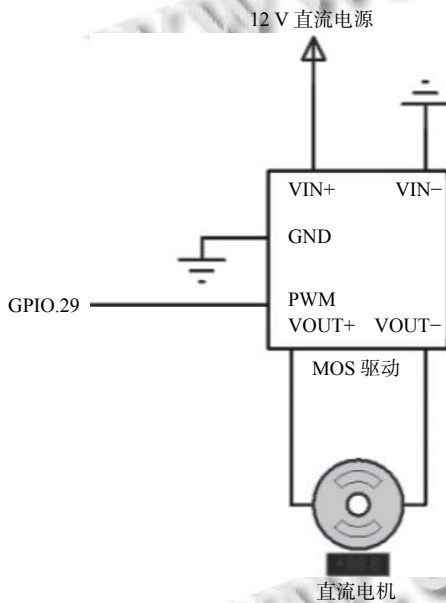


图 5 直流蠕动泵控制电路

3 系统软件设计

本灌溉系统的控制程序在树莓派 4B 的 Raspbian 系统上运行. 编程软件为 Python 3.7, 数据库为 MariaDB (MySQL 的一个分支)^[14]. 主要流程是: 首先开启 I²C 设备, 然后传感器按照设定好的时间间隔读取当前土壤湿度和空气温、湿度, 将采集到的数据传入到树莓派的 MySQL 数据库, 并上传至 OneNET 云平台. 与此同时从互联网获取明日气象数据并将 MySQL 数据库中的数据共同发送到模糊控制器, 模糊控制器将数据进

行处理后得到灌溉时长, 控制程序根据灌溉时长控制蠕动泵工作, 整体流程图如图 6 所示.

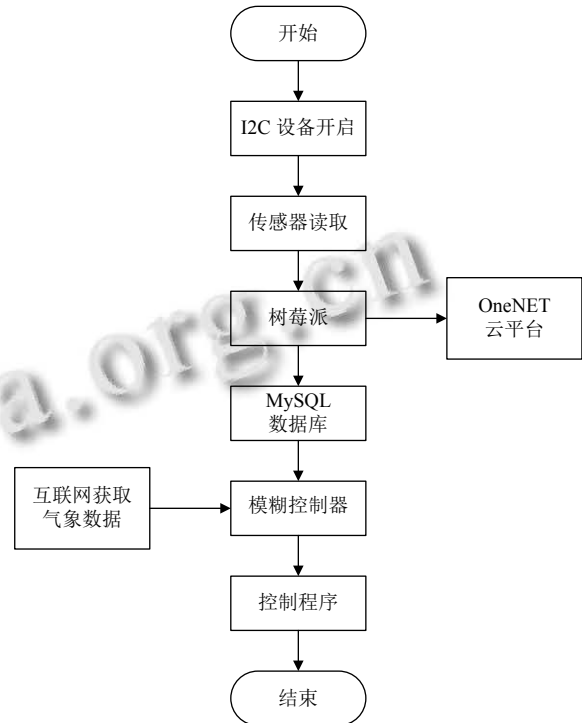


图 6 软件系统流程

3.1 空气温湿度采集功能设计

采集空气温湿度使用的传感器为 DHT11, 采集时间间隔为 5 分钟, 每次传输的是一个 40 位二进制数, 其中前 16 位是湿度数值的整数和小数, 中间 16 位是温度数值的整数和小数, 最后 8 位是校验位. 校验准则为前 32 位所得数据相加应为最后 8 位校验位的数值, 当校验成功时, 说明采集结果正确, 可以使用这个数值; 当校验失败时, 说明采集结果错误 (可能是程序错误, 也可能是传感器问题), 此次采集数据将被丢弃.

土壤湿度采集流程与空气温湿度采集流程大体一致, 只是多一步模数转换, 采集流程在这不再赘述.

3.2 直流蠕动泵驱动功能设计

本文控制灌溉时将 PWM 设置为固定值, 在得到模糊控制器计算出的灌溉时长 T 后通过 MOS 管驱动直流电机进而使蠕动泵工作, 让程序时延 T s 后即可控制直流蠕动泵工作.

3.3 数据库设计

(1) 数据库需求分析

由于本灌溉系统需要保存的数据仅有各个时刻传

感器采集到的数据,所以仅需要一张传感器表来存储这些数据,包括土壤湿度、空气湿度、空气温度、时间和一个为了区分记录的id.

(2) 数据库概念结构设计

通过对智能灌溉系统的需求进行分析,本系统需要一个实体属性模型,即传感器实体,其实体属性图如图7所示.

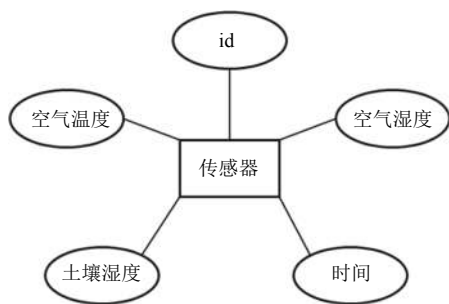


图7 传感器实体属性

(3) 数据库逻辑结构设计

根据上述需求分析和概念结构设计已经抽象出实体的属性,建立传感器表见表1.

表1 传感器表

序号	字段名称	字段类型	大小	允许为空	主键	备注
1	id	char	16	否	是	区分记录
2	soilhum	float	4	否	否	土壤湿度
3	airtem	float	4	否	否	空气温度
4	airhum	float	4	否	否	空气湿度
5	time	datetime	8	否	否	时间

(4) 数据库连接

数据库创建完成后,利用Python的PyMySQL库与数据库建立连接并把数据导入.

3.4 模糊控制器设计

数据库连接本系统采用的模糊逻辑系统为经典的Mamdani型模糊逻辑系统^[15].系统共有两个输入数据分别为Eh(土壤湿度误差)和Ech(误差变化率),一个输出为T(灌溉时长).输入的语言变量都为5个{NB、NS、Z、PS、PB},论域为[-4,4];输出的语言变量也为5个{Z、S、M、B、B+},论域为[0,4].实际论域到模糊论域的量化因子为:

$$k_i = \frac{y_i}{x_i}, k_i > 0 \tag{1}$$

其中, y_i 为模糊论域上限, x_i 为实际论域上限.

模糊论域P到实际论域Q的比率因子为:

$$k_j = \frac{q}{p}, k_j > 0 \tag{2}$$

其中, q 为实际论域上限, p 为模糊论域上限.

本文的量化因子和比率因子分别为:

$$\begin{cases} k_{pe} = \frac{4}{20} = 0.2 \\ k_{pec} = \frac{4}{15} \approx 0.27 \\ k_u = \frac{200}{4} = 50 \end{cases} \tag{3}$$

在进行模糊推理之前,首先必须确定隶属度函数, E_h 、 E_c 和T全都采用三角形隶属度函数^[16].依据一线人员的工作经验和专家知识建立起模糊逻辑规则后进行赋值即可得到模糊规则响应表^[17],后续只需要执行相应的查表工作即可得到科学的灌溉时长.模糊响应规则表见表2.

表2 模糊响应规则表

T	Eh								
	-4	-3	-2	-1	0	1	2	3	4
-4	4	3.5	3	2.5	2	1	1	1	0
-3	3.5	3.5	2.5	2.5	1.5	0.5	0.5	0.5	0
-2	3	2.5	2	1.5	1	0	0	0	0
-1	2.5	2.5	1.5	1.5	0.5	0	0	0	0
Ech	0	2	1.5	1	0.5	0	0	0	0
1	1.5	1.5	0.5	0.5	0	0	0	0	0
2	1	1	0.5	0	0	0	0	0	0
3	0.5	0.5	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0

此表在逆模糊化时采用“最大隶属度法”,得到精确的输出.

灌溉时主要考虑土壤湿度的影响,但也不能忽略空气湿度的影响,经过多次测量,本地空气温度在[15, 35]之间,空气湿度在[50, 90]之间,定义由空气温度决定的灌溉时长为 T_a ,由空气湿度决定的灌溉时长为 T_b ,模糊控制器得到的灌溉时长为 T_c .其中:

$$\begin{cases} T_a = \frac{airtem - 25}{10} \times 200 \\ T_b = \frac{70 - airhum}{10} \times 200 \end{cases}$$

在空气温度小于25℃,空气湿度大于70%RH时灌溉时长都为0.经过综合考虑,空气温度、空气湿度占比和模糊量的权重占比分别为:0.1、0.2和0.7.最终得到系统灌溉时长 $T=T_a+T_b+T_c$.

3.5 OneNET云平台监控设计

首先注册一个云平台账号,然后在开发者中心创

建产品树莓派 4B, 设备接入协议选择 HTTP. 创建完产品后继续添加设备并为 OneNET 监测的环境变量添加相应的数据流模板. 树莓派上传数据都是通过数据流

传入, 可以设置数据流相关参数. 最后新建应用并添加应用 LOGO. 在应用中可以设置显示温湿度的折线图、柱状图等. OneNET 云平台监控图如图 8 所示.

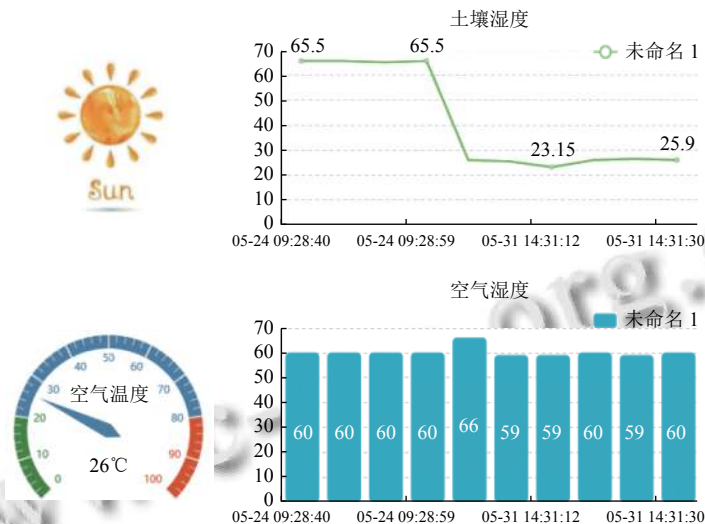


图 8 云平台监控图

用户可以通过电脑端直接登录云平台或者使用手机、平板通过设备云 APP 访问云平台即可完成数据监控, 其网络架构如图 9 所示.

不超过 2%, 能够有效地达到节水的效果, 系统完全实现了自动灌溉, 可以释放大量的劳动力, 也能够加快我国农业从经验种植到科学种植的进程.

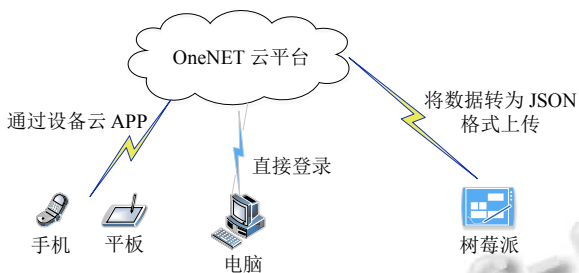


图 9 系统网络架构

4 系统测试

系统测试是检验智能灌溉系统的灵敏性和准确性. 如种植草莓平均的温度大约在 15-25℃ 范围内, 处于生长期的草莓最适宜的土壤湿度为 70%RH 左右. 将直流蠕动泵的 PWM 设为固定值 60, 环境温度为 22℃, 测试系统在不同土壤湿度环境下的灌溉时长. 为了模拟更多土壤湿度的情况, 用人工加水来提高土壤湿度. 在灌溉测试时用秒表记录灌溉时长与模糊控制响应表所得的灌溉时长进行对比. 测试结果见表 3.

由表 3 中数据可以看出系统灌溉时长的误差最大

表 3 系统灌溉结果

环境条件	土壤湿度 (%RH)	理论灌溉时长 (s)	实际灌溉时长 (s)
1	48	200	200
2	53	170	168
3	58	120	121
4	60	100	98
5	64	60	60
6	69	10	10
7	72	0	0
8	85	0	0

5 总结

本文设计了一种智能灌溉系统, 能够根据当前环境状况自动计算出灌溉时长. 系统采用了树莓派 4B 作为核心控制器件, 监控方面在云平台上开发应用取代传统的 Web 端, 大大降低开发成本, 进而降低了整套系统的成本. 系统经过初步测试, 灌溉时长的最大偏差不超过 2%, 且反应迅速, 具有很好的准确性和灵敏性, 在今后大规模的农田灌溉领域有广阔的应用前景. 后续可以利用智能算法^[18-24]研究不同农作物对化肥的需求, 将灌溉系统与施肥系统相结合, 加快农业自动化进程.

参考文献

- 1 刘玉明. 我国水资源现状及高效节水型农业发展对策. 农业科技与信息, 2020, (16): 80–81, 83.
- 2 韩军桥. 水资源开发利用及保护策略探讨. 科技创新与应用, 2020, (2): 117–118.
- 3 贾咏霖, 屈忠义, 丁艳宏, 等. 不同灌溉方式下施用生物炭对土壤水盐运移规律及玉米水分利用效率的影响. 灌溉排水学报, 2020, 39(8): 44–51.
- 4 Benami A, Ofen A. Irrigation engineering. Sprinkler, trickle, surface irrigation principles and agricultural practices. *Agricultural Water Management*, 1983, 9(3): 263–264.
- 5 Goap A, Sharma D, Shukla AK, *et al.* An IoT based smart irrigation management system using machine learning and open source technologies. *Computers and Electronics in Agriculture*, 2018, 155: 41–49.
- 6 卢明乔, 陈晓. 基于 ZigBee 传感网的农业智能照明系统. 湖北农业科学, 2014, 53(23): 5858–5860.
- 7 刘书伦, 冯高峰, 贾宝华. 基于物联网 Android 平台的远程智能节水灌溉系统. 农机化研究, 2015, 37(6): 217–220.
- 8 何江. 一种智能云灌溉系统. 节水灌溉, 2017, (3): 97–99.
- 9 邢玉品, 陈晓. 基于 TDC-GP22 的超声波测风仪设计. 现代电子技术, 2018, 41(21): 128–131, 136.
- 10 杨应鹏, 陈晓. 基于蓝牙的可穿戴式脉搏波形采集系统设计. 激光杂志, 2018, 39(6): 74–77.
- 11 荣百川, 陈晓. 基于蓝牙的智能防摔监测系统. 激光杂志, 2019, 40(7): 32–34.
- 12 郑雪钦, 高镛源. 基于正弦阈值滤波的步进电机转子位置检测. 控制工程, 2018, 25(3): 408–412.
- 13 戴贻康, 焦运良, 范晶. H 桥式电路驱动无刷直流电机的设计. 信息技术与网络安全, 2019, 38(8): 58–63.
- 14 黎忠东. MariaDB 数据库不运行故障. 网络安全和信息化, 2019, (4): 151–153.
- 15 Mahmoud MS. *Fuzzy Control, Estimation and Diagnosis*. Cham: Springer, 2018.
- 16 Klawonn F, Kruse R, Winkler R. Fuzzy clustering: More than just fuzzification. *Fuzzy Sets and Systems*, 2015, 281: 272–279.
- 17 王晓晶, 刘美珍, 陈帅, 等. BP 神经网络预测模糊控制液压马达性能研究. 控制工程, 2020, 27(8): 1394–1400.
- 18 杨瑶, 陈晓. 基于神经网络的说话人识别实验设计. 实验室研究与探索, 2020, 39(9): 38–41, 50.
- 19 陈霞, 陈晓, 邹胜男. 基于正态检验的室内定位算法. 激光杂志, 2017, 38(3): 41–45.
- 20 任晓晔, 陈晓, 郭妍. 基于 Fluent 和 LSTM 神经网络的超声波测风仪阴影效应补偿研究. 计算机应用与软件, 2019, 36(7): 89–98.
- 21 邹胜男, 陈晓, 陈霞. 基于改进的 UKF 算法的室内测距定位. 激光杂志, 2017, 38(4): 61–65.
- 22 孙超文, 陈晓. 基于多尺度特征融合反投影网络的图像超分辨率重建. 自动化学报, 2021, 47(7): 1689–1700.
- 23 郭妍, 陈晓, 任晓晔. 一种优化随机森林模型的室内定位方法. 激光杂志, 2018, 39(10): 70–74.
- 24 朱娟, 陈晓. 改进胶囊网络的有序重叠手写数字识别方法. 激光杂志, 2019, 40(7): 43–46.