

基于 Arduino 与图形化编程的智能家居温度控制系统设计

赵佳桐

山东电力高等专科学校 山东省济南市 250000

DOI: 10.32629/ems.v8i7.21202

[摘 要] 本文为解决传统家居温度控制设备缺乏远程调控手段、无法定时调节温度、智能化成本高等问题,提出一种基于图形化编程的智能家居温度控制系统设计方案。本方案选用 Arduino Mega 2560 为核心控制板,集成了温湿度传感器、无线通信模块和红外发射模块,构建温控设备的硬件感知层;在 linkboy 图形化编程平台上进行硬件的运行逻辑设计,大幅降低用户的开发与使用门槛;通过腾讯巴法云平台实现控制指令的中转与设备物联胡同,并利用 MIT APP Inventor 2 可视化开发平台构建 Android 客户端的手机 APP。按上述方案组建系统后,对该智能家居温控系统的功能进行了验证。结果表明,系统可实现室内温度的实时监测与空调、地暖等设备的远程控制,且支持分时段的温控策略配置,为智能家居系统的快速开发与用户个人定制提供了新思路。

[关键词] 智能家居; 温度控制; Arduino; 物联网; 图形化编程; 低代码开发

1. 引言

随着物联网行业的快速发展与工业自动控制技术的进步,智能家居已成为广大家庭用户提升居住品质、优化家用电器使用体验的重要手段^[1]。根据《2025 年智能家居行业研究报告》预测,中国的智能家居市场规模将持续扩大,用户需求将朝着更加智能化、便捷化、精细化、整体化的方向发展。其中,家居温度控制作为调节室内环境、营造舒适家居氛围的关键环节,一直是智能家居升级改造的重点^[2]。

然而,当前传统的室内温度调节设备(如空调、地暖、空气加湿器等)普遍存在以下问题。首先,各类温控设备的操作高度依赖物理遥控器或设备本体的按钮,无法实现便捷的远程控制;其次,传统温控设备缺乏温度定时调节功能,用户需频繁重复调节;此外,大多数温控设备缺乏简单有效的云端互联机制和配套硬件设备,难以实现远程控制。

基于 Arduino 的开源硬件平台受益于其成本低、可扩展性强、社区编程资源丰富等优势,在民用物联网及个人物联网编程开发领域得到了广泛的应用^[3]。然而,传统 Arduino 硬件板的开发依赖 C/C++ 代码编程,其代码逻辑复杂、可视化性差,对非专业个人开发者而言,仍存在一定的学习和开发门槛,用户自主修改代码的难度较大。近年来,图形化编程技术通过可视化、模块化的方式构建程序逻辑,有效降低了嵌入式系统的开发难度^[4]。同时,在手机 APP 移动端也涌现出了大量的可视化编程平台,能够较为简洁地实现 APP 用户界面和编程逻辑的开发。

综上所述,本文提出一种基于图形化编程的智能家居温度控制系统,采用 linkboy 图形化编程工具进行 Arduino 硬件板的控制逻辑开发,结合 MIT APP Inventor 2 进行移动端的用户界面 UI 与代码开发,通过巴法云平台实现设备的云端互联与远程控制,构建出一套可视化、模块化、易扩展的智能家

居温度控制系统方案。

2. 系统总体架构设计

2.1 架构概述

本文提出的智能家居温控系统,采用典型的物联网设备三层架构,即硬件感知层、物联网络层和设备应用层。其中,硬件感知层选用 Arduino Mega 2560 为核心控制硬件板,增加 DHT11 温湿度传感器模块,采集室内环境的温湿度并用于反馈调节,同时连接 TSAL6200 红外发射模块,用于控制空调等需要红外信号遥控的家用电气设备,并通过继电器模块控制地暖等简单插拔家用电器的电源通断。在物联网络层,则是借助 ESP8266 Wi-Fi 通信模块,基于 TCP/IP 通讯协议接入选定的物联网平台,即腾讯巴法云平台,以云平台为中转,实现硬件感知层的各类数据上传以及设备应用层的用户调控指令下发。而设备应用层需要基于 MIT APP Inventor 2 可视化开发平台,构建 Android 移动端的应用程序,为用户提供简明快捷的室内环境实时监测与智能家居设备的分时段调控策略配置界面。

2.2 技术选型分析

主控板选用 Arduino Mega 2560,其具有 54 路数字 I/O 接口(含 15 路 PWM 输出)和 16 路模拟输入接口,可满足多传感器并行接入需求,主频 16 MHz,闪存 256 KB,能够支撑复杂的控制逻辑^[5]。相比 Arduino Uno, Mega 2560 的丰富接口为后续功能扩展预留了充足空间。

通信模块采用 ESP8266-01S,该模块集成 TCP/IP 协议栈,支持 AP/Station/AP+Station 三种工作模式,在本设计中配置为 AP+Station 模式,既可在必要时作为热点进行本地调试,又能连接家庭 Wi-Fi 路由器接入互联网^[6]。其串口通信波特率设置为 115200,与主控板通过软串口或硬串口进行数据交换。

3. 硬件系统设计

3.1 硬件组成与接口配置

系统硬件主要由四大功能模块组成：主控模块、传感模

块、通信模块和执行模块。各模块与 Arduino Mega 2560 的引脚连接关系如表 1 所示。

表 1 硬件模块接口连接表

模块	引脚/功能	Arduino 引脚	说明
ESP8266	TX	RX1 (Pin 19)	串口接收
	RX	TX1 (Pin 18)	串口发送
	VCC	3.3 V	电源供电
	GND	GND	接地
DHT11	DATA	Pin 2	单总线数据
	VCC	5 V	电源供电
TSAL6200	SIG	Pin 9	PWM 信号输出
	VCC	5 V	电源供电
继电器模块	IN	Pin 7	开关控制信号

ESP8266 模块工作电压为 3.3 V，需通过稳压电路与 Arduino 的 5 V 主电源隔离，避免电压不匹配导致模块损坏。DHT11 传感器采用单总线通信协议，数据引脚需外接 4.7 kΩ 上拉电阻以保证信号稳定性^[7]。TSAL6200 红外发射二极管工作波长 940 nm，与市面上主流空调遥控器的红外载波频率 38 kHz 兼容。

3.2 图形化编程实现

区别于传统代码开发，本系统硬件逻辑采用 linkboy 图形化编程平台设计。linkboy 是由中科院计算技术研究所开发的开源免费编程仿真平台，支持“所见即所得”的硬件连线界面与面向对象的事件触发编程模式^[8]。

在 linkboy 图形化编程平台的主界面中，程序结构由“初始化设置”和“循环”两大模块构成，分别对应软件内的 setup 和 loop 函数。其中，初始化设置模块需要完成多个初始化参数的设置，例如通讯串口的波特率设置为 9600 bps、ESP8266 网络配置需要确定设备的 SSID、密码，同时要将巴法云服务器的地址输入编程软件，同时对 DHT11 传感器的初始参数进行初始化设置。而在循环模块中，需要设置系统中每 120 秒触发一次温湿度采集，同时将数据封装为 JSON 格式，再经 ESP8266 通讯模块上传至巴法云平台，同时持续监听通讯串口的缓冲区，解析来自云平台的其他控制指令。

linkboy 平台的核心优势在于其完备的软硬件一体仿真技术。通过合理的仿真手段，能够准确模拟硬件设备在实际运行时的信号传输与触发流程，同时也可以模拟通讯模块的通讯过程，能够为开发者节省大量的硬件试错成本^[9]。同时，其平台自主研发的 cx 语言编译器也能够帮助用户将简洁的图形化编程逻辑自动转换为 Arduino 系列硬件开发板能够识别的 C/C++ 代码，降低了用户的编程门槛，有效促进从产生

想法到完成产品的快速开发过程。

4. 软件系统设计

4.1 云平台通信协议

巴法云平台采用前后端分离的系统架构，同时支持高并发的异步通信，适用于进行物联网设备间的通讯和控制^[10]。系统采用 TCP 协议的长连接方式接入互联网，设备与 APP 间通过 Topic 机制进行信息通讯。进行巴法云平台设置时的关键通信协议格式如下：

- （1）订阅主题的通信协议：cmd=1&uid=<用户私钥>&topic=temp\r\n
- （2）发布消息的通信协议：cmd=2&uid=<用户私钥>&topic=con&msg=<控制指令>\r\n
- （3）维持设备信息传输活跃的 keep 周期信号：cmd=0&msg=keep\r\n（30 秒周期）

其中，用户私钥的 UID 需要在巴法云控制台注册后获取，主要用于云平台所连接的物联网设备的身份认证；主题名 temp 用于实时更新温度数据并由设备上传至原设备，con 用于空调类室内温度控制设备的控制指令发送，而 dinuan 指令则用于地暖类设备的控制指令发送。

4.2 移动应用 APP 的开发

移动端的 APP 应用主要基于 MIT APP Inventor 2 软件平台开发。该平台由麻省理工学院的媒体实验室维护，采用纯图形化的编程逻辑，适合非专业开发者快速构建 Android 端的手机应用^[11]。

MIT APP Inventor 2 的界面设计采用垂直滚动式的布局，自上而下可以分为三个功能区：

- （1）实时监测区：包含顶部标签组件，用于实时显示室内的温度；

(2) 空调控制区: 主要包含风向、风速及温度共三个列表选择框, 采用水平布局排列, 用户可通过三个选择框设置空调的控制指令;

(3) 时段策略区: 以表格布局, 呈现工作日与周末时段共 12 个时段 (6:00-8:00、8:00-12:00、12:00-14:00、14:00-17:00、17:00-21:00、21:00-6:00) 的温度设定选项, 并且也支持空调与地暖的独立配置。

在安卓 APP 的逻辑设计中, 先通过 ClientSocketAI2Ex1 组件建立稳定的 TCP 连接, 再利用计时器组件实现 keep 周期信号的定时发送。同时, 针对外界的信息交互, 当手机端接收到主题 temp 的消息时, 能够使用文本分割指令, 提取温度数值并更新界面显示。而在点击上传按钮时, 系统根据当前的时段, 自动匹配预设参数以及用户发送的 con/dinuan 指令控制参数, 识别对应的控制指令时段并发送至对应主题。

5. 物联网系统的测试与验证

5.1 硬件仿真测试

在硬件仿真层面, 主要考虑在 linkboy 软件下的硬件设备运行状态和信息传输的稳定性。在 linkboy 的硬件仿真环境中, 采用 RGB 全彩 LED 来模拟空调遥控器的红外信号输出, 建立颜色编码与空调控制指令的映射关系。其中, 红色的分量对应温度调节的相关指令, 绿色分量对应风向调节的相关指令, 蓝色分量对应制冷/制热模式与风速组合的相关调节指令, 共可模拟 192 种控制状态。经过实际的仿真运行表明, 逻辑判断与信号生成时序准确, 满足对应的载波调制要求。

5.2 系统联调测试

实际测试环境为 90 m² 住宅, 将主控板置于客厅中央, 同时硬件板的 DHT11 温湿度采集传感器悬置于距地面 1.5m 处, 模拟真实的空调运行环境。

测试结果表明:

(1) 数据采集层面: 主控板的温度采样周期为 120 秒, 物联网设备与云平台间的通信延迟小于 500 ms, 同时物联网设备与云平台、手机 app 间能够连续通讯 72 小时, 无丢包现象;

(2) 远程控制层面: 在 APP 端下发指令至设备响应完成, 整个流程的平均延迟小于 2 秒, 且采用红外遥控控制空调的成功率大于 95%;

(3) 温控精度层面: 通过对比住宅内的温湿度采集传感器显示与设定的温度, 设定温度与实际温度偏差小于 $\pm 1^{\circ}\text{C}$, 满足基本的温控精度要求;

(4) 稳定性层面: 如果巴法云平台的系统意外断网后, ESP8266 通讯模块的自动重连机制可在 10 秒内恢复物联网设

备与云平台间的连接。

6. 结论

本文设计并实现了一种基于图形化编程的智能家居温度控制系统。以 Arduino 为硬件开发板, 在 linkboy 平台上实现了对硬件感知层的图形化编程; 以巴法云平台为载体, 实现物联网设备与用户调控指令的双向传输与实时交互; 并在 MIT APP Inventor 2 平台上进行集成, 实现对安卓 app 的图形化开发和用户界面设计, 从而完成设备应用层的搭建。基于图形化编程的智能家居温度控制系统不仅能够显著降低智能家居系统的开发门槛, 还实现了环境温湿度实时监测、智能家居设备远程调控及用户自定义配置家居环境方案, 整体系统具备较高的稳定性与可扩展性。未来, 该系统可以继续集成人脸识别、语音控制、ai 云平台优化等功能, 持续优化用户的家居体验。

[参考文献]

- [1]刘云浩. 物联网导论[M]. 北京: 科学出版社, 2017: 15-28.
- [2]中国智能家居产业联盟. 2020 年中国智能家居行业研究报告[R]. 北京: 中国智能家居产业联盟, 2020.
- [3]Banzi M, Shiloh M. Make: Getting Started with Arduino[M]. 3rd ed. Sebastopol: Maker Media, 2014: 32-45.
- [4]陈清. Linkboy 图形化编程在职业技术教学的优势分析[J]. 科技与创新, 2019(12): 156-158.
- [5]Arduino AG. Arduino Mega 2560 Rev3 Technical Specifications[EB/OL]. (2020-06-15) [2024-04-07]. <https://docs.arduino.cc/hardware/mega-2560>.
- [6]Espressif Systems. ESP8266 Technical Reference[EB/OL]. Version 1.4, 2020.
- [7]广州奥松电子有限公司. DHT11 温湿度传感器技术手册[Z]. 2018.
- [8]王强, 张伟. LinkBoy: 软硬件集成的电子积木创意平台[J]. 计算机教育, 2015(14): 112-116.
- [9]linkboy 开发团队. linkboy 图形化编程平台技术白皮书[EB/OL]. (2023-05-20) [2024-04-07]. <http://www.linkboy.com.cn/>.
- [10]巴法云物联网平台. TCP 创客云接入文档[EB/OL]. (2022-03-25) [2024-04-07]. <https://cloud.bemfa.com/>.
- [11]Abelson H, Spertus E, Wolber D. App Inventor: Create Your Own Android Apps[M]. Sebastopol: O'Reilly Media, 2021: 45-62.