

# 基于树莓派的智能家居设计与实现<sup>①</sup>



张美平<sup>1,2</sup>, 吴德平<sup>1</sup>, 王灿杰<sup>1</sup>, 谢玮铭<sup>1</sup>

<sup>1</sup>(福建师范大学 数学与信息学院, 福州 350117)

<sup>2</sup>(福建省网络安全与密码技术重点实验室, 福州 350007)

通讯作者: 张美平, E-mail: [mpjason@fjnu.edu.cn](mailto:mpjason@fjnu.edu.cn)

**摘要:** 随着生活水平的提高和物联网的发展, 社会对家居的智能化需求越来越迫切, 本文阐述了基于树莓派的智能家居系统的设计与实现, 通过采用树莓派为主要模块, 搭建一款满足大众需要的智能家居系统. 本系统以树莓派为主要开发平台, 并基于 Ubuntu 操作系统进行开发的一种智能家居解决方案, 其包含了语音合成、语音识别、图像识别、数据采集、AI 对话、视频监控、语音控制、语音日志等功能. 可通过语音、手机微信、APP 与机器人和传感器进行交互, 并能登录 Web 界面查看相应底层数据并对传感器进行相应控制. 系统传感器部分采用 ZigBee 通信协议, 与服务器通信采用 MQTT 通信协议, 两种通讯协议低成本, 低功耗, 节约网络资源.

**关键词:** 树莓派; 智能家居; 数据采集

引用格式: 张美平, 吴德平, 王灿杰, 谢玮铭. 基于树莓派的智能家居设计与实现. 计算机系统应用, 2019, 28(8): 109-114. <http://www.c-s-a.org.cn/1003-3254/7020.html>

## Design and Implementation of Smart Home Based on Raspberry Pie

ZHANG Mei-Ping<sup>1,2</sup>, WU De-Ping<sup>1</sup>, WANG Can-Jie<sup>1</sup>, XIE Wei-Ming<sup>1</sup>

<sup>1</sup>(College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China)

<sup>2</sup>(Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350007, China)

**Abstract:** With the improvement of living standards and the development of the Internet of Things, the demand for intelligent home is becoming more and more urgent. This paper describes the design and implementation of smart home based on raspberry pie. Through raspberry pie as the main module, we build a smart home system to meet the needs of the public. This system takes raspberry pie as the main development platform and develops a smart home solution based on Ubuntu operating system, it contains a speech synthesis, speech recognition, image recognition, data acquisition, AI dialogue, video monitoring, voice control, voice logging, etc. Interact with robots and sensors through voice, WEB, WeChat, and APP. It can interact with robots and sensors through voice and mobile WeChat, and can log in the Web interface to view the underlying data and control the sensors accordingly. The system sensor part adopts ZigBee communication protocol, and MQTT communication protocol is adopted for communication with the server. The two communication protocols are of low cost, low power consumption, and saving network resources.

**Key words:** raspberry pie; smart home; data collection

## 引言

近几年, 随着物联网技术的快速发展, 智能生活的

理念越来越被大多数人接受, 相关技术应用至各个行

业的各个方面. 现如今, 智能感知认知、多模态人机交

① 基金项目: 2018 福建师范大学“大学生创新创业训练计划”(cxll-2018097)

Foundation item: Year 2018, Innovation Training Program for Students of Fujian Normal University (cxll-2018097)

收稿时间: 2019-01-12; 修改时间: 2019-02-21; 采用时间: 2019-03-08; csa 在线出版时间: 2019-08-08

互、云计算等智能化技术不断成熟,为智能家居的发展提供了坚实的发展基础。

由于中国现阶段的家庭大多数未能实现智能化,如何以低成本、高效率地智能化是物联网行业需要考虑的问题,所以在众多种类的设备中,树莓派的关注度逐渐提升.国内对于智能家居的研究也已经开展起来.陈必东<sup>[1]</sup>等人通过将树莓派与 UNO 等开源组件结合的方式,对智能家居雏形系统进行了深入的探索与研究;郝林炜<sup>[2]</sup>等人提出了一种以树莓派与云服务器的网络监控及智能家居控制系统的研究与实际应用,该远程实时监控以及远程家居控制系统不论是从成本上,还是从其功能上都具有相当大的优势,同时还能给用户带来良好的用户体验。

随着树莓派的火热发展,基于树莓派的物联网 ZigBee 网关的设计<sup>[3,4]</sup>也层出不穷,但大多数的树莓派网关的设计停留在数据传递的功能应用层面,树莓派的优势没有得到明显的体现.且在机器人应用技术的推动下,国内各大企业都开启了对智能家居系统的开发.但国内市面上的智能家居平台众多,架构复杂,功耗较大,且其交互方式较为单一。

针对以上的问题,本文阐述设计一套基于树莓派(raspberry pi)智能家居系统,该系统包含了图像识别、数据采集、AI 语音对话、视频监控、语音控制、语音日志等功能.用户可通过语音、手机微信、APP 与树莓派和传感器进行交互,并能登录 web 界面查看相应底层数据并对传感器进行相应控制.树莓派的应用得到了扩展,它既作为网关来传输数据,还可以作为应用来与用户交互.具有低功耗、低成本、节约网络资源,压缩系统结构等特点。

### 1 系统整体功能

本系统主要功能模块可分为树莓派模块、用户应用模块、数据采集传输模块 3 个方面进行设计,其系统框架如图 1 所示。

**树莓派模块:** 作为系统的主体部分,拥有语音识别、语音合成、图像识别、AI 对话、微信交互、视频监控、传递数据等功能。

**数据采集传输模块(智能家居):** 利用树莓派作为网关(GW),CC2530 节点采集数据,ZigBee 组网,通过 MQTT 协议与网关交互,网关进行协议转换之后使节点接入 MQTT 服务器。

**应用终端模块:** 通过 APP、网页后台查看底层节点的环境数据,控制节点状态及进行网页视频监控功能。

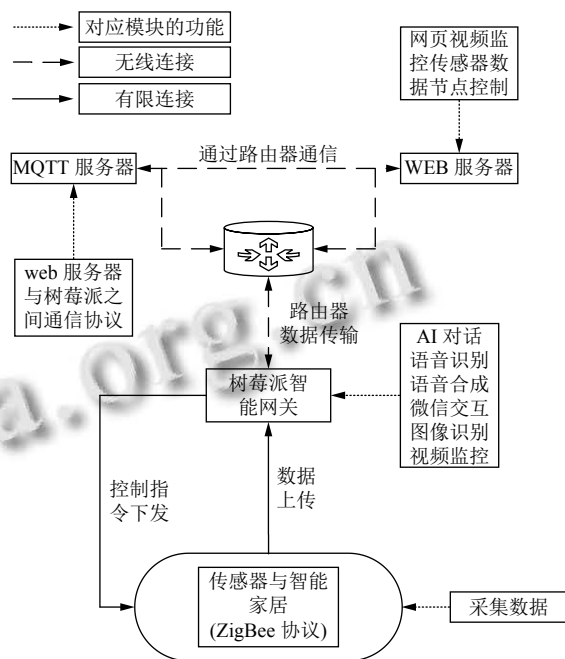


图 1 系统整体框架

### 2 系统整体架构

考虑到使用环境为家庭环境,将系统压缩为适用家庭的简易三层结构,树莓派既作为应用终端,也作为底层节点的网关使用,压缩系统体积的同时,也满足了应用需求。

#### 2.1 系统硬件

在家庭环境下部署传感器节点与执行节点,通过 ZigBee 通信协议实现数据的采集与传递的功能.感知节点由 CC2530 开发板和各类传感器或执行器构成.节点采集数据之后,以数据包的形式将数据传至树莓派网关.同理,网关对应用终端发来的指令也通过包的形式发送至执行节点.节点间与树莓派网关的通信如图 2 所示。

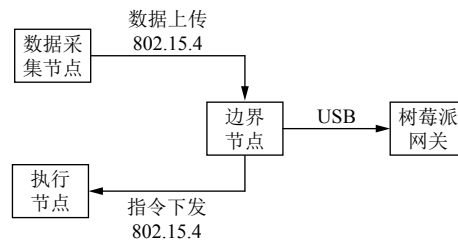


图 2 节点与树莓派网关通信

树莓派体积小, 功能强大, 兼容性强, 最主要是它支持多种计算机语言, 主流的有 C、Python, 其中 Python 最适合树莓派开发. 针对以上特性本设计中, 树莓派除了作为网关使用, 还扩展了其他交互功能, 作为智能家居系统的主体, 在树莓派上外接树莓派麦克风 (respeaker-2mic), 摄像头 (PI-camera) 和小型音箱, 为用户提供多种与系统的交互方式.

在树莓派上进行对应编程, 运用 Mjpg-Streamer、MQTT 通信协议、Snowboy 唤醒、微信接入、Python 串口读写、调用百度智能 AI, tensorflow 深度学习等技术实现视频监控, 数据传输, 语音对话, 微信控制, 图像识别等功能.

### 2.2 系统软件

系统软件集成在应用终端, 提供底层数据的检测和对底层节点控制的 Web 与 APP, 应用终端与树莓派网关的网络通讯, 采用 MQTT 通信协议规范<sup>[5]</sup>. MQTT 结构由 MQTT 消息代理和 MQTT 客户端组成, 图 3 是 MQTT 结构.

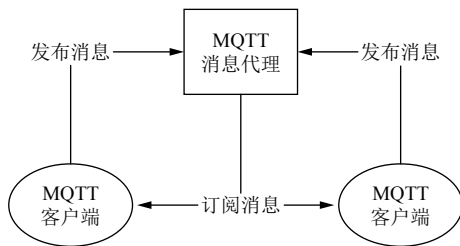


图 3 MQTT 结构

应用终端向 MQTT 服务器订阅 (subscribe) 数据主题, 树莓派网关接收到数据后将数据及其主题发布 (publish) 至 MQTT 服务器, 此时, MQTT 服务器将数据推送至订阅对应主题的应用终端. 反之, 执行指令的下发流程相同. 应用终端与树莓派网关的通讯框架如图 4 所示.

## 3 系统设计与实现

### 3.1 数据采集与传输功能模块

底层物联网设备通过 ZigBee 网络来连接, ZigBee 网络支持星型、树型和网状网络拓扑结构, 本系统采用树型结构, 树型结构在传输距离上有一定优势, 并且使用的时间会较长, 成本低.

底层数据需要约定数据上传与反向控制的格式,

方便树莓派网关和应用终端对数据的解析和对指令的生成, 具体格式与设备编号如表 1、表 2 所示.

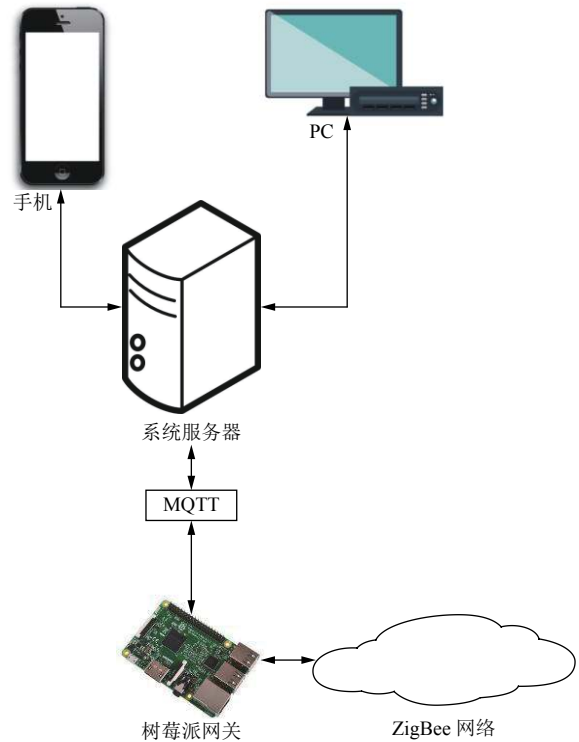


图 4 应用终端与树莓派通信

表 1 数据/指令格式

位	参数注释	实例
00	数据头	AA
01 02	设备编号	40 01
03 04	数据/指令	00 3F

表 2 系统节点编号

节点名称	设备编号
温湿度节点	0x4001
继电器节点	0x5001
烟雾节点	0x6001
关照节点	0x7001

### 3.2 树莓派功能模块

树莓派作为整个系统的主体部分, 拥有语音识别、语音合成、图像识别、AI 对话、微信交互、视频监控、传递数据等功能.

在语音识别与合成中, 为了避免树莓派对无用的信息做出反应, 利用 Snowboy 平台训练树莓派的唤醒词, 做到先唤醒, 再反应. 树莓派的语音识别<sup>[6]</sup>与语音合成调用了百度 AI 平台的 Deep CNN+LSTM+CT 架构

和 Deep Voice<sup>[7]</sup>, 利用 TensorFlow 深度学习实现对图像的数据识别。

AI 对话功能主要利用图灵机器人平台的 DeepQA 系统. 为了方便与系统交互, 树莓派实现了微信接入的功能, 利用 itchat 可以实现对微信的调用。

视频监控功能是智能家居安全保护功能的主要体现, 有以下两种方案:

(1) 在树莓派上搭建 MJPG-streamer 视频流服务器, 就可以很方便地将树莓派摄像头所采集的视频流输出到网页服务器插件, 则用户便能直接从浏览器中观察到树莓派所拍摄到的景物。

(2) 利用 ffmpeg, ffmpeg<sup>[8]</sup>是一套可以用来记录、转换数字音频、视频, 并能将其转化为流的开源计算机程序. 鉴于树莓派没有被映射到公网中就不能从公网访问的局限性, 使用 ffmpeg 将视频流推至 rtmp 服务器, 从而突破树莓派视频监控只能在局域网下访问的限制. 其视频流向为: 树莓派摄像头->ffmpeg->rtmp->nginx->VLC media player.

这里采用第二种方案, 此方案支持公网通信访问。

除了以上功能, 树莓派作为底层节点的网关, 通过 Python 串口读写实现对环境数据的接收和控制指令的发送。

### 3.3 应用终端模块

应用终端是提供针对底层数据的检测和控制节点的网页管理系统和 APP。

为了使代码能够便捷复用, Web 端使用 MVC 框架进行开发, 流程如图 5 所示, 通 MQTT 协议对传输上来的数据进行相应的处理并显示, 并下发控制指令到节点。

Android 端调用 web 服务器提供出来的接口, 获取数据, 并通过调用 web 服务器的接口发送对应的控制命令。

## 4 运行效果演示

我们首先模拟一个家庭环境, 部署数个传感器、执行器节点, 接上灯泡模拟电器, 在笔记本上搭建 php、mosquitto 服务器运行环境, 场景模拟如图 6。

在上述环境下进行语音测试。

通过对树莓派进行关键词唤醒, 并说出相应的控制命令, 检查语音识别是否正确, 是否根据指令进行正确的操作, 结果如图 7 所示。

用微信对树莓派进行控制, 检查控制指令是否正确传达, 是否根据指令进行正确的操作, 如图 8、9。

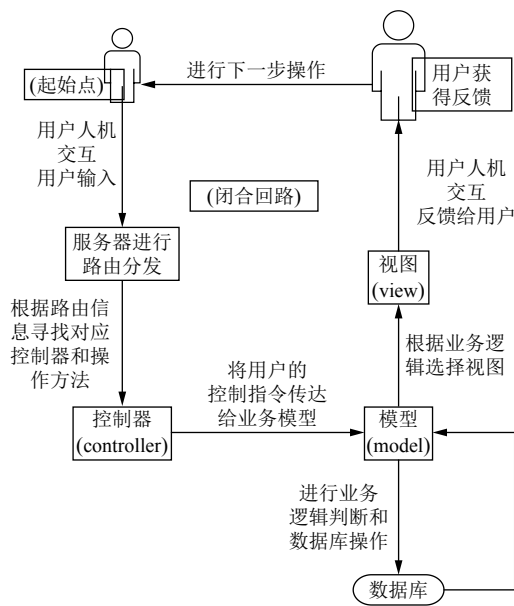


图 5 MVC 框架图



图 6 场景模拟图

性能指标 测试标号	说话内容	机器识别后 进行的操作	小派反应时间	总体效果
1	你好	调用智能 AI, 并语音合成	4 s	成功
2	现在的 时间	调用智能 AI, 并语音合成	3.5 s	成功
3	福州 天气	调用智能 AI, 并语音合成	4 s	成功
4	拍照	调用摄像头拍照, 并发至邮箱	8 s	成功
5	播放 音乐	播放音乐	3 s	成功
6	停止 播放	调用智能 AI, 并语音合成		语音识别错误
7	停止 播放	停止播放音乐	2 s	成功
8	开灯	向 MQTT 服务器 发布控制命令	3 s	成功
9	关灯	向 MQTT 服务器 发布控制命令	5 s	成功
10	这是 什么	拍照, 并进行图像 识别, 并语音合成	9 s	成功

图 7 语音测试表



图8 微信测试对话



图9 微信测试拍照

访问网页服务器,检查传感器数据是否正常显示,是否能对家庭电器进行控制,是否能正常开启监控及看到监控画面,效果如如图10、图11所示;Android端采集数据界面展示如图12所示.

传感器ID	采集数据	时间
DC01	25.0020	2018-05-25 19:00
DC01	25.0020	2018-05-25 19:00
DC01	24.0003	2018-05-25 18:58
DC01	24.0003	2018-05-25 18:58
DC01	25.0020	2018-05-25 18:58
DC01	25.0020	2018-05-25 18:58
DC01	24.0003	2018-05-25 18:58
DC01	24.0003	2018-05-25 18:57
DC01	24.0003	2018-05-25 18:57

图10 Web显示数据界面

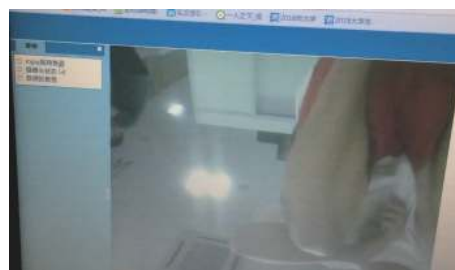


图11 反向控制、视频监控界面



图12 Android界面效果

### 5 总结

本文提出了一种智能家居系统的设计方案,该方

案主要对基础的智能家居进行完善和系统简化. 主要是在树莓派模块进行系统压缩和功能扩展, 增加更多的智能应用功能, 让树莓派作为数据传输的网关的同时, 还能拥有更加智能的应用终端功能.

本文所述的系统总体有3个大的功能模块: 数据采集传输模块、树莓派模块、应用终端模块. 本文中, 感知节点通过 ZigBee 组网, 将环境数据上传到树莓派网关, 树莓派网关将数据发布到 MQTT 服务器, 订阅对应主题的应用终端模块便能接收到对应数据. 树莓派模块以树莓派作为主控, 外接硬件设备作为系统与外界的音频信息进行交互的途径. 由此, 开发出本文所述, 简约的, 低功耗, 低成本, 节约网络资源的智能家居系统.

#### 参考文献

- 1 陈必东, 谭美娟. 基于 RaspberryPi+Arduino 智能家居雏形系统的探索与研究. 智慧工厂, 2016, (6): 49–54, 63.
- 2 郝林炜, 梁颖. 基于树莓派+云服务器的网络监控及家居控制系统的研究与实际应用. 物联网技术, 2016, 6(9): 45–47, 50.
- 3 陈詮, 毕忠勤, 刘大明. 基于树莓派的物联网 ZigBee 网关的设计与实现. 上海电力学院学报, 2016, 32(6): 599–602. [doi: 10.3969/j.issn.1006-4729.2016.06.020]
- 4 王梁栋. 基于树莓派的智能家居温度远程监控系统的设计. 信息通信, 2018, (5): 65–66. [doi: 10.3969/j.issn.1673-1131.2018.05.027]
- 5 方霞. 基于 MQTT 协议的农业物联网消息推送系统. 计算机技术与发展, 2018, 28(9): 168–171. [doi: 10.3969/j.issn.1673-629X.2018.09.034]
- 6 Amodei D, Anubhai R, Battenberg E, *et al.* Deep speech 2: End-to-end speech recognition in English and Mandarin. <https://arxiv.org/abs/1512.02595>.
- 7 Arik SO, Chrzanowski M, Coates A, *et al.* Deep voice: Real-time neural text-to-speech. <https://arxiv.org/abs/1702.07825>.
- 8 胡国强, 周兆永, 信朝霞. 基于 SRS 的开源直播系统的设计与实现. 现代电子技术, 2016, 39(16): 36–39, 43.