

基于树莓派的无接触智慧电梯设计

董晨乐¹, 杨延宁^{1,2}, 朱 扬¹

(1. 延安大学 物理与电子信息学院, 陕西 延安 716000;

2. 南昌理工学院, 南昌 330044)

摘要: 传统的电梯产品单一, 人梯交互方式亘古不变, 新冠疫情肆虐以来, 人们在电梯里按键时易病毒感染; 而智能化、免接触的电梯可有效阻止病毒传染; 选用以 Linux 作为核心架构的树莓派, 通过人工智能、大数据、物联网等领域的运用, 开发了一套“免接触式”电梯控制系统; 将传统需要手动点击的物理按键替换为具备语音控制、人脸检索、语音合成、NFC 近场通信, 创建了多种新型的电梯控制方案; 并将必要信息通过 OLED 进行实时显示, 继而可在桌面或移动终端进行监控或指令下发; 拥有网页、iOS 和安卓端操作软件; 测试表明, 无接触智慧电梯系统功能完善、运行稳定, 为社区、公共场所、医院等的民众构建起生命安全防线。

关键词: 新冠疫情; 免接触; 智慧电梯; NFC; 树莓派 4B; Linux

Design of Contactless Intelligent Elevator Based on Raspberry-Pi

DONG Chenle¹, YANG Yanning^{1,2}, ZHU Yang¹

(1. School of Physics and Electronic Information, Yan'an University, Yan'an 716000, China;

2. Nanchang Institute of Technology, Nanchang 330044, China)

Abstract: Traditional elevators have single interaction between people and elevators. Since the new crown epidemic has been rampant, people are prone to virus infection when they press button in elevators. An intelligent and contactless elevator can effectively stop the virus infection. Linux is taken as the core architecture of Raspberry Pi, a contactless elevator control system is proposed by using the artificial intelligence, big data, Internet of Things and other fields. Traditional physical buttons that require manual clicking are replaced by voice control, face retrieval, voice synthesis, and NFC near field communication, a variety of new elevator control solutions are built. The necessary information is displayed in real time via OLED, then can be monitored or commanded through the desktop or mobile terminal. It has the operating software for webpage, iOS and Android. The test results show that the contactless intelligent elevator system is functional and stable in operation, building a life-safety line of defense for people in communities, public places, hospitals, etc.

Keywords: COVID-19; Contactless; Intelligent Elevator; NFC; Raspberry Pi 4B; Linux

0 引言

2020 年“新冠”病毒的传播剧烈引起人们的很大关注。针对人们当前的迫切需求和生活的便利性。为防止病毒的广泛传播, 疫情期间, 出入公共电梯, 民众采取了许多方式, 比如在按键上粘上一层厚厚的纸, 使用牙膏和门钥匙或者湿巾手纸。在已建立的行业与掌握的电子技术基础上。针对上面所提到的问题, 研究并设计了一款利用人脸检测、语音识别与合成、智能化显示、NFC 近场通信等关键技术, 针对疫情防控、智能控制、电梯监控等提供一整套解决方案并设计了一套完整的控制系统。“病毒无情人有爱”, 在目前爆发严重的疫情防控特殊环境中, 制造一款智慧的自动化电梯具备非凡的意义, 面对疫情带来的无接触的潜在

需求, 该项目采用多个控制方案, 人脸面部检测和语音识别等其他诸多项目, 以取代传统采用手动按钮的电梯, 即可摆脱疫情带来的接触感染、为群众生活带来许多便利, 并且在防止病毒威胁的情况下, 还可以更好地为老人, 儿童和残疾人提供更好的服务。在该领域进行深入研究对于家庭、社区和社会民生工作的开展具有重要价值^[1-3]。

1 智慧电梯设计主要内容和工作

目前市场上有一些电梯具有语音功能, 但只支持语音朗读相关信息和播报楼层, 即当电梯到达指定楼层提醒乘客具体到达的楼层。仅在升降机外部添加一个感应模块, 并不具备实用性和便利性。结合当下行业的发展趋势及存在的问题, 通过撰写研究课题步骤及方法, 以论证所要得

收稿日期: 2022-10-23; 修回日期: 2022-11-30。

基金项目: 延安科技计划项目(2019ZCNZ-001); 陕西省能源大数据智能处理重点实验室项目(IPBED11、IPBED16); 陕西省教育厅科技计划项目(21JK0983); 延安大学疫情防控应急科研项目(ydfk060、ydfk062); 延安大学研究生教育创新计划项目(YCX2023030)。

作者简介: 董晨乐(1999-), 男, 陕西咸阳人, 硕士, 主要从事智能检测与控制方向的研究。

杨延宁(1969-), 男, 陕西延安人, 教授, 博士, 主要从事智能检测与控制方向的研究。

引用格式: 董晨乐, 杨延宁, 朱 扬. 基于树莓派的无接触智慧电梯设计[J]. 计算机测量与控制, 2023, 31(7): 128-135.

出的研究成果和所要展示的形式^[4]。

该论文的目标是实现电梯系统所有模块在 Linux 开发板树莓派 4B (Raspberry-Pi 4B) 上稳定运行; 实现 Baidu_Net、Home-Kit、Home-Assistant 所有的传感器的稳定运行; 并使系统所有代码可零成本快速打包迁移和部署; 开发并完成智慧电梯实物模型一套。针对软件控制, 开发了对应的安卓 APK 软件包和 HTML5、CSS 网页控制系统。

2 基于树莓派的电梯系统架构搭建

2.1 总体结构设计

基于传统的电梯控制方案中存在的诸多的待解决问题, 急切能够完成电梯轿厢自动控制的装置。区别于传统方案, 应该采用哪种控制方案。主要提出了以下 4 种控制方式^[5]。

1) 与传统的电梯一样, 采用带有详细的楼层信息的物理开关或按键检测来控制电梯到达对应的楼层位置。

2) 在数据库提前存储用户的人脸信息和其所对应的楼层编号信息, 之后利用人脸搜索技术, 来对目标用户的人脸进行读取和检测, 将获取到当前用户对应的楼层信息来向 Raspberry-Pi 的 GPIO 口下达高低电平指令驱动步进电机带动电梯轿厢前往指定楼层。

3) 针对陌生的用户群体非本栋楼层的用户或没有提前录入到人脸信息, 可以对电梯下达语音指令来进行其设备的控制。

4) 手机用户采用 NFC 控制芯片, 电梯控制实现真正的零接触, 将指令通过自己的手机直接进行下发传达来控制 Raspberry-Pi 的 GPIO, 这里的 NFC 控制方案区别于传统的“门禁卡”用户可以直接用手机内置的 NFC 芯片来进行通信。智慧电梯总体方案设计逻辑如图 1 所示^[6]。

2.2 核心方案设计

利用局域网环境搭建起 HTML5 网页, 打开 HTTP 服务器的监控功能。使用 Linux 终端定义两个路由, 定义一个 POST 的类, 第一个路由的作用就是扔给浏览器一个 HTML 文档后再从中读取 index 数据模板, 发送至客户端服务器。针对第二个路由函数, 使用 Python 调用 Bottle 的 Run 函数方法创建一个 HTTP 服务器, 使其通过浏览器访问的界面。针对客户端 (控制界面) 引用 jQuery 和 Bootstrap 这

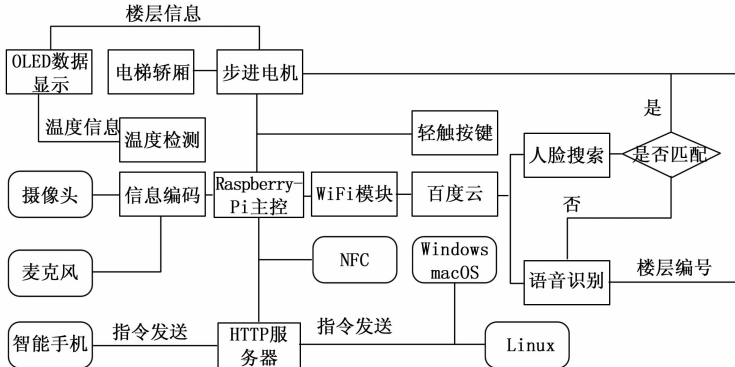


图 1 智慧电梯总体方案设计逻辑

两个前端框架, 这样就完整的实现和 Raspberry-Pi 之间的网络通信^[7]。

在基于 Python3 的开源家庭的自动化平台上。采用 HASS 类似于“智能家居”的环境部署方案, 来进行对 Raspberry-Pi 的远程控制。Hass.io 是一个专为集成操作, s 将电梯控制系统与 Hass.io 和 Docker 集成在一起, 用以简化 Home-Assistant 配置。

结合 iOS 终端的自动化和快捷指令程序进行更高复杂度的编程设计。摄像头将图像捕捉之后, 利用互联网云市场上第三方的 API 接口来完成对采集到人脸进行数据分析与处理。

3 智慧电梯系统的控制硬件选型

3.1 控制器选择

采用装载 Linux 微型操作系统 Raspberry-Pi 的开发板为主要控制器, 作为通用微型计算机, 拥有特殊的硬件性能、信号绝缘, 低功耗等特点。可拓展 USB 摄像机或帧接口摄像头。

在 Raspberry-Pi 4B 的内部, 设置的传输模块有 Wi-Fi 和蓝牙等, Raspberry-Pi 的操作方式主要有两种, 一种是图形界面操作, 一种是命令行模式。这里使用 Wi-Fi 传输协议来进行所有数据收发和处理。内部硬件设备在远程连接、Windows 通信、百度云平台连接均使用 Wi-Fi 协议, Raspberry-Pi 主控板如图 2 所示^[8]。

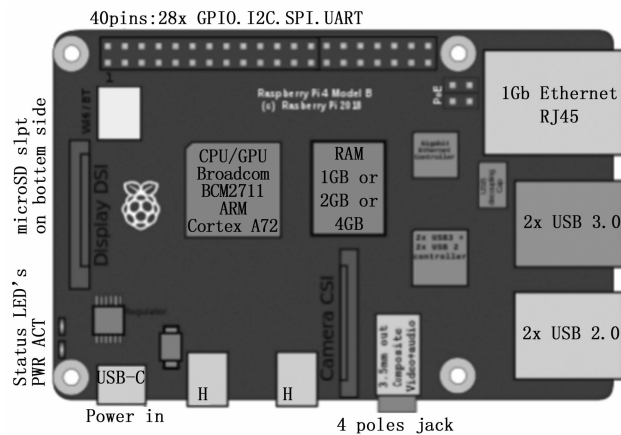


图 2 Raspberry-Pi 主控板

3.2 系统数据传输

采用步进电机的传动装置作为电梯的动力系统来连接模拟电梯轿厢。具有精度高、灵敏度高、启停迅速等特点。在与 Raspberry-Pi 的连接运行中, 步进电机负责对电梯的动力驱动。使用 USB 麦克风的最显著优势之一是即插即用, 不需要其他的驱动。Raspberry-Pi 操作系统当接入麦克风会自动检测麦克风驱动程序。在此设计中使用 Windows 系统的笔记本电脑来和实体世界进行 Samba 数据交互并将信号发送至 Linux 开发板。

Samba 可以使运行 Linux 的计算系统能够参与操作其他计算机文件中，与 Microsoft Windows 进行相同的网络协议通信，并从 Windows 客户端的角度作为另一个文件系统在网络上显示。Windows 和 Linux 文件交换如图 3 所示。

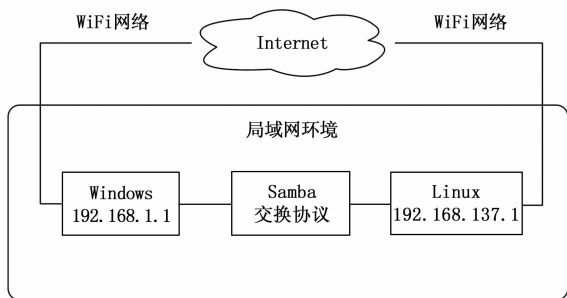


图 3 Windows 和 Linux 文件交换

3.3 系统驱动装置

使用步进电机将脉冲信号转换为普通的物理机械运动，并且在使用步进电机时系统是可以不需要额外的数据反馈，即使没有反馈系统，步进电机也可以通过精确的位置控制进行连续旋转就既可以计算出相应的转动角度。针对步进电机的运行机制和特殊基准。除非在运行中产生错误丢失转动步数，否则它的旋转的角度对的系统和程序是已知的。

由于已知电机的旋转位置，因此可以精确地控制电机的位置。这里采用步进电机的传动装置作为电梯的动力系统来连接电梯轿厢。具有精度高、灵敏度高、启动停止迅速等特点。

步进电机的驱动形式主要有两种，在本次设计中采用的是两次相控驱动：这种输出方式的主要转矩特别大并且产生的震动较小。在转换的时候可以有一个通电线圈作为动力输出，这样既可以让它的输出扭矩变大，也可以减少其震动。两次相控驱动可提供更好的运动扭矩。当两相接通时，电动机产生更大的转动扭矩。需注意的是：转矩脉动是一个待解决的问题，因为它可能引起共振和振动导致步进电机不能运行。

针对和 Raspberry-Pi 的连接，本项目的设计方案主要是采用 ULN2003 激励作为驱动器，也就是将集成放大器至于电路内部，并且不需要在电路连接中设置其他电容或电阻元件，步进电机驱动板连接如图 4 所示^[9]。

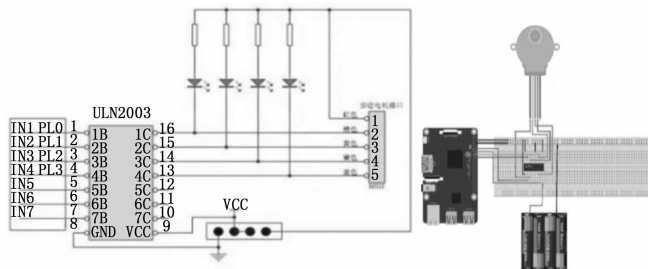


图 4 步进电机驱动板连接

4 百度云平台和 HASS 局域网监控设计

4.1 操作系统选型

树莓派 ARM 架构的卡片电脑，可以安装非常多的 Linux 发行版。Raspberry-Pi 官方的操作系统是 Raspberry-Pi OS，一共有 3 个版本，这 3 个版本都是可以兼容所有的 Raspberry-Pi 操作系统是一个免费的基于 Debian 的操作系统，也是最适合 Raspberry-Pi 设备。Raspberry-Pi OS 在操作系统中已经预装了 3 500 多个常用软件包，其具备完整的 Linux 操作系统功能，可以很轻松安装在 Raspberry-Pi 上。Raspberry-Pi 操作系统与大多数常见台式机兼容，例如 MacOS 和 Microsoft-Windows。

4.1.1 操作系统选择

访问 Raspberry-Pi China 的官方网站以下载最新版本的 Raspberry File 和 Photography 文件。刻录过程首先选择目的镜像如图 5 所示。

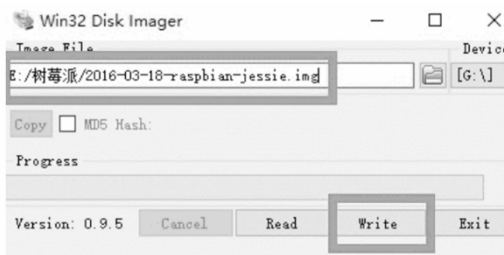


图 5 使用 Win32 进行系统烧录

当所有的软件连接配置完成之后，启动远程连接软件程序，主要有 3 种方式可以连接操作系统，这里使用 Xshell 接入到 Raspberry-Pi 操作系统中。

4.1.2 Python 编译环境的安装

Python 作为本次项目设计开发的主要编程语言，在的 Linux 终端上需直接运行此程序语言，开发环境搭建，访问官方网站 <https://www.python.org/>，下载需要的安装包和在程序设计中的相关依赖即可完成对应操作。

4.1.3 HASS-IO 远程控制系统的安装

作为基础框架，本项目利用 Docker 来部署 Home-Assistant，并且为 Home-Assistant 提供各种插件 (addons)。如此来为在后面和电梯的连接中基本控制器 (消毒系统) 提供了帮助。而 Home-Assistant 在 Hassio 中，Home-Assistant 是利用 Docker 部署起来，Hassio 和 Home-Assistant 是通过内置的 API 进行联系和沟通以连接电梯的控制模块^[10]。

4.1.4 百度云 SDK 的本地化安装

此 SDK 作为 Python 的一个第三方附属库，用以连接操作系统和百云的 API 数据通信数据的收发。并定义了许多方便调用的子函数。无论是在 Linux 还是在其他操作系统平台，有了 Python 编译环境之后，便可以使用 shell 命令一键安装。

4.2 百度 API 云平台

结合身份 ID 识别、人脸对比、活体检测等多项组合能力。其功能主要包含：人脸注册、人脸识别验证、人脸搜

索验证。使用百度云平台来处理系统采集到的语音信息和捕获的人脸数据。

百度云 API 技术的云端调用时, 官方提供有许多程序命令的 DEMO, 系统使用 Python 的 JSON 函数包去请求百度云服务器等待其参数结果的返回。API 的接口程序中, 本质都是调用服务器从此接口返回的结果后对数据集处理。

4.3 HASS 数据信息平台

作为连接所有 Raspberry-Pi 设备的监视平台, Hass.io 是一个简单的操作服务器和一个简单的家庭类帮助系统。在操作系统内安装相对应的 OS 软件和集成 Docker 环境使用 Home-Assistant 去简化系统使用并且更加便于移植。通过 HASS 服务分发 Home-Assistant 云服务。Hass.io 使用 Python 创建类似于 Docker Manager 的程序以及在此处运行的控件, 以升级或降级该服务环境。通过 Home-Assistant 的开源自动化平台。可以跟踪和控制电梯中的所有设备, 并实现自动化控制。并完美运行在 Raspberry-Pi。^[11-12]

4.4 电梯系统检测和监控

智慧电梯系统的网络环境配置中, Raspberry-Pi 的监控方案采用的是网络 IP 摄像头的视频流上传。在 Android、iOS、Windows 上等各种平台都可以完成配置。利用 Android 平台的网络 IP 摄像头可以完成实时动作监控等操作。

当设备接入到此局域网中, 在另一台设备安装 Cam Monitor 通信协议, 支持在两个设备之间的双向音频传输。Windows 和 macOS 等都可以兼容。通过局域网的连接方式即可接入到 HASS 的 Home-Assistant 中, 即可以在整个局域网中进行摄像头画面实时共享^[13-14]。

电梯系统支持动作检测功能, 并调整云服务器的设置以将其启用。激活运动检测模式, 用于环境网络警报。技术原理采用检测间隔帧变化的方法。目标传输和目标分配最常用的方法是关键帧的分割方法。使用像素时间差来移除图像进行的两个或 3 个相邻正方形之间的图像运动部分。首先, 将视频流截取成细微的“帧流片段”, 然后对比前后两帧的图像位置, 动作阈值检测如图 6 所示。

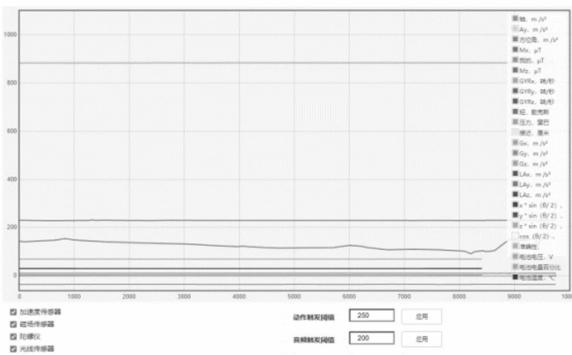


图 6 动作阈值检测

5 人梯交互与前端网页控制部署方法

5.1 HTTP 前端网页控制

无论是在 Android 的软件程序交互和 PC 或 Mac 的交互

逻辑设计, 系统采用的是统一 API 请求, 使用 Python 为主要设计语言, 采用 Flask 框架模型作为后端部署程序。网页前端设计语言采用 HTML5、CSS、JavaScript 联合许多种 CDN 库, 主要包含有 jQuery 版的 JS、Bootstrap 开发环境等。

为适应用户不同的设备终端, 借助以上框架设计了一套网页 APP, 可以完美兼容 Android 和 iOS 等操作系统平台。通过 HTML5 网页渲染技术, 使用 Python 程序中的 POST 请求给出一个服务器的 IP 的请求接口 (8080), 系统创建了两个 HTML5 网页客户前端, 其 HTTP 交互逻辑如图 7 所示^[15]。

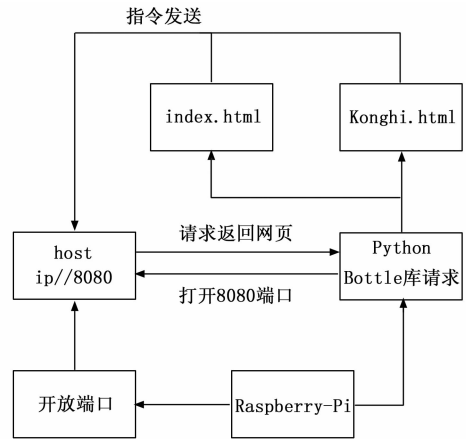


图 7 HTTP 交互流程

在 Raspberry-Pi 上运行 Python 指定程序将接口释放, 此时会生成两个 HTML 前端网页文件, 一个是针对电梯的控制端 (192.168.137.84:8080/cmd), 二是针对 IP 摄像头的控制端 (192.168.137.84:8080/cmd2)。使用 Fusion-App 将网页和 Home-Assistant 加载到“智慧电梯”安卓 APP 中。既可以使用图形化界面对电梯下达指令控制。安卓软件界面如图 8 所示。



图 8 安卓软件界面

5.2 后端框架 Python Flask

HTML 后端交互和数据传输的程序框架选择上, 采用 Python 的 Flask 架构。Flask 框架非常适用在小型网站, 也适用于开发各种 Web 服务的 API, 在大型网站开发, Flask

开发灵活。修相比 Django 同样作为普遍的 Web 服务框架，Flask 框架具有更多的功能和交互性可以通过高度定制服务来完成更灵活的服务设计。选用 Flask 作为主框架，可以更好地避免程序代码过多。在 HTML5 设计中，设定了 6 个 API 服务用作于硬件开关的接入。这里的 Web 服务网关接口，作为 Web 服务器和应用程序之间的通用接口，选用 Python 的 Flask 框架，确保了不同的 Web 服务器类型可以相互通信。

在电梯的控制环节中，现在人们周围许多的设备都具有 NFC 近场通信功能，例如可以用手机代替公交卡、银行卡甚至身份证等。本项目将这一技术应用到针对电梯的控制中去，即利用 NFC 的识别返回值或检测 POST 的命令请求来完成电梯的控制指令下发^[16]。

5.3 NFC 近场控制

在电梯的控制环节中，现在人们周围许多的设备都具有 NFC 近场通信功能，例如可以用手机代替公交卡、银行卡甚至身份证等。本项目将这一技术应用到针对电梯的控制中去，即利用 NFC 的识别返回值或检测 POST 的命令请求来完成电梯的控制指令下发。

NFC 作为一种（近场通信）无线通信协议，主要是在两个电子设备之间 4 厘米或更少的距离内进行协议交换。连接准备、速度快。也可以引导许多针对设备的功能性操作。例如钥匙卡、明文身份标识卡，并且它们也可以用于各种支付系统或信用卡等。

在 NFC 的芯片选取过程中，本着低成本、高稳定性、易连接等主要特点。最终确认并选用 ISO 14443-3A 的 NXP 技术标准，在连接技术方案上采用的是 NFC，并支持反复擦写读取。在 Android 和 iOS 的操作系统中，实现由两种方案设计。针对 Android 方案，利用 NFC-Tools 写入工具如图 9 所示，将在 index 的 HTML5 网页中的接口写入到 NFC 芯片内存中去^[17]。

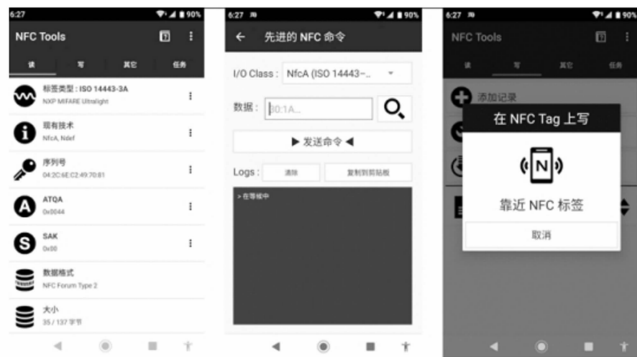


图 9 NFC-Tools 写入工具

针对 iOS 操作系统，利用每一张 NFC 的身份标识码，iOS 操作系统对一张卡片完成检测后，会返回一个指定的芯片标识类型。分析到特定的标识符后，再通过快捷指令的 SSH 脚本去请求指定好的 Python 程序这样就完成了相应的

请求。iOS 操作系统可视化编程界面如图 10 所示。



图 10 iOS 操作系统可视化编程界面

5.4 人脸识别控制

人脸搜索技术是由 Google 开发的搜索服务，该服务的目的是用户可以从已上传到 Internet 的图像中搜索该人的所有图像。在电梯总体控制方案中，人脸对比技术是其本次设计任务设的主线，采用百度云 API 服务调用处理。

利用摄像头将图像捕捉之后，利用互联网云市场上第三方的 API 接口来完成对采集到人脸进行数据分析与处理。这里采用的是百度云平台开放的 AI 处理工具百度云—人脸搜索。在人脸数据库中搜索人脸图像，返回数据库中和人脸组和最匹配的人脸信息关键值，它可以提供有人脸注册，更新，删除和请求用户信息等 11 种界面功能，并支持多级数据库管理界面。

为便于后期系统测试，系统关闭了活体检测。在 API 的返回信息中，如果对代码进行设置不同的控制度，就会得到不同相对应的质量控制阈值，若服务器的返回值不符合阈值的要求，即会向控制台发送并提醒错误信息。HTTP 请求后既可以得到 JSON 返回结果信息又可以达到和 Python Flask 框架的交互。返回示例用 User_id 来判断用户姓名，以此控制步进电机以驱动电梯^[18]。

5.5 语音识别控制

5.5.1 语音识别

系统录音方案采用的是第三方的网络平台设备（Windows 操作系统的录音模块 PY）。使用 ssh 局域网通信协议传输到 Linux 的文档目录下，Windows 和 Raspberry-Pi 交互逻辑如图 11 所示。

使用主控制器（Raspberry-Pi）的数据控制介质进行数据收集，短语音识别技术在语音转文本的过程中，30 秒的语音长度大多数情况下几秒钟即可完成。因此完全可以将其在应用在控制和输入场景。

一是通过 REST-API 接入，在每个 API 的接口调用中，都会有专门的 HTTP 接口，用以上传编码后的完整录音文件。二是使用本地 SDK 接入并支持超过 60 秒的实时长语音识别。同时也支持上传完整录音文件。支持离线唤醒词、命令词、通用语义解析等功能。在这里选用的是 REST-API

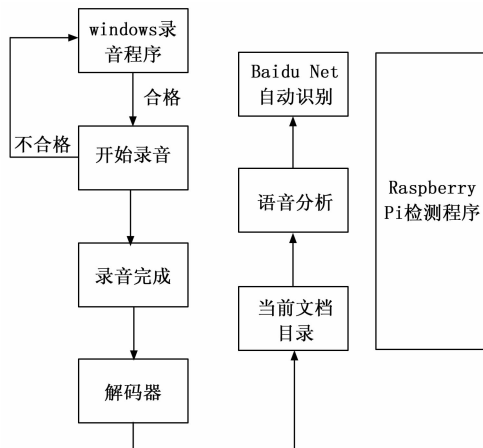


图 11 Windows 和 Raspberry-Pi 交互逻辑

的方案接入, 此方案在价格和速度上具有很大的优势, 而 SDK 所附有的其他功能不是本次项目所必需的, API 请求类型如表 1 所示。

表 1 API 请求类型

信息添加	
接口类型	REST 方式提供的 HTTP 接口
接口限制	录音文件时长不超过 60 秒
支持格式	PCM、WAV、AMR、M4A
识别请求	POST 方式通过 JSON 和 RAW
请求地址	http://vop.baidu.com/server_API

5.5.2 语音合成

语音转换 (TTS) 系统将普通语言文本转换为语音文件, 语音系统使用百度云 AI 平台。短语音模式的短语音识别模型能够以类似人类语音进行通信, 适用于电话音频记录, 语音搜索, 语音通信等其他活动。具有 5 种知识形式: 普通话, 英语, 四川语言等^[19]。其中, Android, iOS 和 Linux SDK 可以支持最多 60 秒的实时现场语音, 语音属性如表 2 所示。

表 2 语音属性列表

属性	类型
多语言	中文、英文、中英混读合成
语速	支持多种参数配置

5.6 系统辅助控制

借助 Home-Assistant 平台, 在电梯内置通风扇和紫外线消毒等, 将以上硬件设备依托 Home-Assistant 云平台, 接入到系统的控制端, 它可以监听在局域网设备环境中的所有设备, 并可以针对自动化控件, 进行指令下发和接收。

在此项目中, HASS 安装在 Home-Assistant 中的 Raspberry-Pi 硬件端口和电梯的主控上, 用于数据交互。可以使用 Raspberry-Pi 的官方 Raspberry 操作系统来安装 HASS 或 Hassbian 系统。它可以很高程度的简化与 Home-

Assistant 相关的插件和依赖项的安装和配置。

通过不同的通信协议, 连接到不同的网络设备, 并且可以在 Web 界面上编辑修改、移动端有 iOS 应用程序和 Android 应用程序, 便于用户更简洁的进行交互使用。例如 RGB 灯的设备, 也可以支持大量处理操作。

5.6.1 OLED 数据显示模块

为了显示电梯控制系统的具体对应楼层的信息, 系统需要实时显示楼层位置信息以及最新的告警消息。该系统使用 Python 向 SSD1306-OLED 模块来发送指令, 并使用完整的 IIC 和 SPI 兼容驱动程序。SSD1306 作为屏幕的驱动器, OLED 本身分辨率高, 单个像易素定位, 且像素传感器能够旋转, 屏幕明亮且功耗非常低。SS1306 上安装的包括控制器、RAM 显示器和晶体振荡器, 从而减少外部设备和功耗, LED 显示流程如图 12 所示。

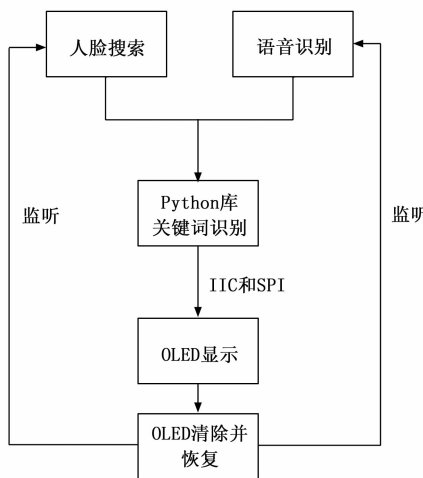


图 12 OLED 显示流程

5.6.2 Home-Kit 接入

系统中的 HASS 安装在 Home-Assistant 中的 Raspberry-Pi 硬件端口和电梯的主控上用于数据交互。可以使用 Raspberry-Pi 官方 Raspberry 操作系统来安装 HASS 或 Hassbian 系统。其可以高程度的简化与 Home-Assistant 相关的插件和依赖文件包的安装和配置。

它的通信协议机制包含例如: WeMo 协议、ZigBee 网关和 Wi-Fi 通信。允许移动设备和电话在远距离共享数据以创建个人局域网。借助 Home-Kit 平台, 将所有设备组合在一个应用程序中, 程序将文本信息发送至服务器, 服务器解析完成并关联相对应的指令信息进行返回。HomeBridge 作为连接桥梁是使用 Node.js 开发的应用连接程序框架, 可以直接运行在家庭的局域网络, 并在 iOS 设备和其他连接桥梁之间提供开放 API^[20]。

5.6.3 手机语音助手

实时语音控制可在任意时刻完成对设备的指令下发控制, 时延低、响应快、内存占用小和受众面广。采用的 iOS 终端中集成的 Siri 语音助手, 在连接家庭 APP 的同时, 在快捷指令配置好 ssh 的 Raspberry 的 Python 程序, 即可完

成在 Raspberry-Pi 和 iPhone 之间的快速操作。如图 13 为快捷指令的配置界面。



图 13 快捷指令的配置界面

针对前端技术，使用 Siri 技术的目标是处理用户请求并提供最佳结果。所做的主要任务就是通过分析用户的数据输入用户的数据类型来选择正确的技术类型进行回应并处理^[21]。

6 实验结果与分析

6.1 电梯部分功能的运行方法

系统人脸和语音控制逻辑：当行人步行到电梯旁，电梯的红外检测摄像头先对人脸进行定位提取关键点信息并上传至 BAIDU 或本地数据库记录的人脸服务器进行比对分析，由主控制器 Raspberry 计算后给出是否匹配的结果，若通过匹配将数据库记录的楼层信息反馈至驱动装置控制电梯轿厢完成运行到指定楼层。若检测到的人脸信息不在系统或 BAIDU 的人脸数据库中，会立即启动语音检测提示用户语音给出口层信息，计算之后反馈至驱动装置。

NFC 和 HTML 网页控制逻辑：可以部署在货运电梯或医院，实现使用手机触碰电梯内置的 NFC 感应芯片或操作人员使用特定的 APP 对电梯进行监控或控制。

6.2 系统总体调试和测试

系统调试利用接近实际客观的环境条件，对系统的硬件和系统软件进行分步测试，然后对整个系统进行整体调试并收集记录测试数据。确保系统硬件和软件的整体可靠性。除去必要的软件开发、维护软件和操作系统、同时也是确定软件系统是否完整测试的重要依据。在软件正式应用前，提前测试各项指标去发现在使用过程中的错误和不合理之处，将发现的问题进行纠错并完善系统，其主要测试范围包括：响应速度、功能、软件的兼容性和性能指标。软件测试分类如表 3 所示。

1) 所有系统设计其实都是针对时间延时进行的但是在整体系统的设计中，这个延时时间是无法精确到毫秒测量。

最后的解决方法是，每次程序开始运行打印的当前时间，精确到毫秒，结束运行后打印当前时间。两者时间之差就是程序的运行时间。

2) 在对文件上传速度的测量中，有一个是百度云服务器的时间，另一个是本地控制器的时间，两者的时间差值在 8 小时。所有百度云对图片的处理时间是无法确定的。对实验数据进行改进，确定本地上传数据后的时间相对于服务器结果返回的时间差，那么就可以得到数据处理时间和上传时间总和。

3) 在语音处理的函数选择上，百度云服务和许多云服务厂商的数学模型完全是不同的。造成对比数据量大，成本高，没有办法逐个进行数据测试。最后为了保证系统的整体统一，就全部采用白云的服务器并针对延时进行测试和数据收集。

表 3 系统运行的软件架构见表

分类	软件及版本
运行设备	Raspberry-Pi 4B 1GB
运行架构	ARM 微型主板
运行系统	Linux 发行版 Pi
数据接口	Baidu API、Apple HOMEKIT

结合上述的测试数据，既要保证上传速度另外还要保证传输的准确性与识别的准确度。最终实验得出系统的总体设计延迟如表 4 所示。

表 4 总体设计延迟

图像距离	图像占比	上传速度	图像大小	检测用时	检测速度	语音长度	上传速度	检测耗时
35 cm	0.48	0.19 S	0.9 M	0.33 S	0.62 S	3.4 S	0.51 S	0.7 S

1) 对整体的系统硬件设计方案进行周密的论证和实物测试。确定每一个模块的设计要求和方案，最终完成实物电梯模型一套。

2) 对整体系统的电路系统重新设计新的 PCB 电路板和基础硬件布局、连接和焊接。

3) 开发了完整的 HTML 网页、对应 Android 操作软件、整体的 Python 代码。

6.3 未来展望

通过对设计整体的改进，本文“智慧电梯”主要有三点，可以在功能和易用性方面进行优化。

1) 针对人脸检索技术，这一点仅做的是少量人脸比对，并没有考虑在密集人群中的识别度和准确度，这在产品升级中是非常关键的一点。后续通过算法升级和图像分割等技术等完成更多的功能。

2) 在整体的控制系统上，因控制终端的算力限制，不能在本地完成数据计算和数据比对，如果增强算力会导致成本的上升，这一点要针对具体的用途完成相应的调整。

3) OLED 的楼层显示上，不能快速实时地更新显示当前的楼层位置，在高层电梯和安全性上尤为重要。后续可

以通过每个楼层设计传感器来实现。

4) 针对无线终端的控制, 后期应制作轻量化应用程序, 在一个楼层内完成在多个电梯之间的交互通信, 使用 ESP8266 模块来降低使用成本。

7 结束语

在全球智能化背景的今天, 随着日益增长的社会基础设施要求的提高。本文针对基于树莓派的无接触智慧电梯设计系统进行分析研究, 分析了现有的民用电梯的结构和优缺点, 此设计的研究结果阐明了智慧电梯在日常生活的关键性作用和设计意义, 解决了民生和疫情工作中的实际问题。基于 Linux 的发行版 Raspberry-Pi 操作软件的整套电梯控制系统其包含功能多达数十项可供选择, 结合当下最普遍的手机操作、人脸监控、语音控制等 4 项功能对其在许多方面的应用, 最后论述了其实际价值、意义和作用, 对下一步的研究做了分析和展望。

参考文献:

- [1] 郭明月, 苏红青. 新冠肺炎疫情常态化下病房楼电梯运行管控研究——以南京市 7 家公立医院为例 [J]. 中国医院建筑与装备, 2022, 23 (8): 94-96.
- [2] 黄竹岩, 张鑫. AI 语音电梯不仅仅是“防疫电梯”[N]. 北京青年报日报, 2020-03-09.
- [3] 李强强, 高红霞. 抗击疫情成都这群“电梯人”全力保障电梯运行 [N]. 人民日报, 2020-01-30.
- [4] 汪鑫, 于浩, 张雨婷, 等. 基于单片机的智能电梯系统的设计与研究 [J]. 物联网技术, 2022, 12 (9): 68-69.
- [5] 尹恒, 刘庭勇, 彭诗琦. 非接触电梯按键控制系统 [J]. 信息技术与信息化, 2022 (3): 32-35.
- [6] 杨乐晨, 刘忠富, 贺薪诺, 等. 面向疫情的非接触电梯操控系统设计 [J]. 自动化与仪表, 2022, 37 (4): 103-108.
- [7] 刘一镜, 徐畅. 基于 HomeAssistant 和 MQTT 的控制系统研究 [J]. 信息通信, 2020 (9): 51-54.
- [8] 李文胜. 基于树莓派的嵌入式 Linux 开发教学探索 [J]. 电子

(上接第 127 页)

- [9] 唐翠微, 蒲东清. 工业机器人低合金钢焊缝自适应焊接控制方法 [J]. 机械设计与制造, 2022 (3): 45-49, 54.
- [10] 李铭磊, 孙涛, 刘海英. 基于雷塞控制系统的五轴焊接机器人的轨迹规划 [J]. 机床与液压, 2021, 49 (3): 30-34.
- [11] 杨成超. 基于 Matlab 的 UR5 机器人相贯焊接模型运动学分析和轨迹规划 [J]. 机械传动, 2021, 45 (12): 68-73.
- [12] 王超, 张文辉, 江洁, 等. 基于 Matlab/Adams 的工业焊接机器人运动学分析与控制联合仿真 [J]. 中国工程机械学报, 2020, 18 (6): 504-509.
- [13] 陈亚, 史钊亮, 高锦宏, 等. 基于 STM32+FPGA 的六自由度机器人运动控制器设计 [J]. 机械设计与制造, 2020 (4): 240-243.
- [14] 刘志成, 郑莉芳, 王旭. 基于混合神经振荡器的爬行四足机器人运动协调控制 [J]. 兵工学报, 2020, 41 (11): 2303-2312.

技术与软件工程, 2014 (9): 219-220.

- [9] 张建军, 李沙沙, 米兆文, 等. 基于 STM32 的力觉反馈装置及用于主从遥操作实现 [J]. 计算机测量与控制, 2021, 29 (12): 93-97.
- [10] 董晨乐, 张富春, 杨延宁. 超低成本家庭物联网方案的设计与实现 [J]. 物联网技术, 2020, 10 (12): 53-57.
- [11] 刘东, 蒋刚, 留沧海. 基于树莓派的六足机器人目标跟踪系统研究 [J]. 传感器与微系统, 2022, 41 (1): 65-68.
- [12] 王涛. 基于即时通讯软件的智能服务机器人平台的设计与实现 [D]. 武汉: 华中科技大学, 2010.
- [13] 冯志辉. 使用树莓派实现网络监控系统 [J]. 电子技术与软件工程, 2015 (5): 85.
- [14] 李琳杰, 赵伟博, 齐锴亮, 等. 基于阿里云的智能大棚远程监控系统研究 [J]. 自动化与仪表, 2021, 36 (1): 28-30.
- [15] PATCHAVA V, KANDALA H B, BABU P R. A smart home automation technique with raspberry pi using iot [C] //2015 International conference on smart sensors and systems (ICSSS). IEEE, 2015: 1-4.
- [16] 黄浩. 浅述利用 Python+Flask+ECharts 设计实现医疗数据可视化大屏展示 [J]. 数字技术与应用, 2022, 40 (9): 200-202.
- [17] 邓志云, 路红伟, 赵云峰, 等. 基于 NFC 的飞机起落架胎压在线检测装置设计 [J]. 计算机测量与控制, 2019, 27 (4): 41-45.
- [18] 郭伟洁, 植凯吉, 白瑀皓, 等. 基于身份证和人脸双重识别技术的智能门禁系统设计 [J]. 计算机测量与控制, 2021, 29 (2): 222-228.
- [19] 高胜杰. 基于 Android 的智能语音助理的设计与实现 [D]. 上海: 复旦大学, 2013.
- [20] 齐保良, 王旭, 蔺玉璞, 等. 一种具有语音交互功能的停车诱导系统设计 [J]. 计算机测量与控制, 2019, 27 (6): 167-171.
- [21] 张万良. 基于 ARM 处理器的智能家居综合实训教学平台设计 [D]. 成都: 西南交通大学, 2013.
- [15] 安嘉强, 汤晓华, 刘美莲, 等. 基于 ARM-MCX514 的机器人—数控协同系统运动控制器设计 [J]. 机床与液压, 2020, 48 (3): 58-62.
- [16] 董豪, 杨静, 李少波, 等. 基于深度强化学习的机器人运动控制研究进展 [J]. 控制与决策, 2022, 37 (2): 278-292.
- [17] 庄衡衡, 丁飞, 章华涛, 等. 五自由度机器人运动控制与空间位姿仿真系统 [J]. 中国测试, 2021, 47 (11): 14-20.
- [18] 王文林, 杨前明, 孙超. 双机器人系统运动协调仿真与实验研究 [J]. 组合机床与自动化加工技术, 2020 (4): 42-46, 50.
- [19] 陈光荣, 郭盛, 侯博文, 等. 基于扩展雅可比矩阵的冗余液压驱动四足机器人运动控制 [J]. 控制理论与应用, 2021, 38 (2): 213-223.
- [20] 王琪, 闵华松. 双臂机器人的协调控制算法综述 [J]. 计算机工程与应用, 2021, 57 (1): 1-16.