



雲南民族大學

YUNNAN MINZU UNIVERSITY

本 科 生 毕 业 论 文(设计)

基于树莓派的智能家庭健康监测系统设计

学 院： 数学与计算机科学学院

专 业： 物联网工程

姓 名： 陈玄

学 号： 21213020490120

指导教师/职称： 王武/副教授

提 交 日 期： 2025 年 4 月 12 日

教 务 处 制

云南民族大学本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的毕业论文(设计)，是本人在指导教师的指导下进行研究工作所取得的成果。除论文中已经注明引用的内容外，本论文没有抄袭、剽窃他人已经发表的研究成果。本声明的法律结果由本人承担。

论文(设计)作者签名：

日 期： 年 月 日

.....

云南民族大学毕业论文(设计)使用授权的说明

本人完全了解云南民族大学有关保留、使用毕业论文(设计)的规定，即：学校有权保留、送交论文的复印件，允许论文被查阅，学校可以公布论文(设计)的全部或部分内容，可以采用影印或其他复制手段保存论文(设计)。

（保密论文在解密后应遵守）

指导教师签名：

论文(设计)作者签名：

日 期： 年 月 日

摘要

近些年来，物联网、人工智能还有嵌入式技术都在不断发展，且发展速度较快，这便促使智能硬件以及传感器在众多领域都得到了颇为广泛的应用。本论文主要探索怎样设计出一个以树莓派为基础的智能家庭健康监测系统，以此来应对现代家庭在步入老龄化社会以及慢性病高发这样的大背景之下，对于多参数健康监测以及智能化环境管理方面所存在的诸多需求。本项研究运用了软硬件相结合的方式开展设计工作：在硬件部分，是以 STM32F205RET6 微控制器作为核心部分，由其来负责数据的采集以及设备的控制，并且配合树莓派 5 作为数据处理与显示的平台，同时将多种传感器和执行器都集成到一起；在软件部分，是利用 Qt 框架去开发用户界面，通过调用百度大模型 API 来达成语音识别以及对话的功能，依靠阿里云服务器搭建起 MySQL 数据库以及 MQTT 服务器，进而完成数据的存储与传输相关操作，而且还借助 FreeRTOS 实时操作系统来提升整个系统的响应速度，最后经过实验测试来对系统性能加以验证。测试所得到的结果显示，该系统能够对心率、血氧、体温还有空气质量等各类环境参数展开实时监测，并且可以实现智能化的环境调节，其具备不错的实用性，在可扩展性方面表现良好，还具有低功耗以及低成本等诸多特点。

关键词：智能家庭健康监测；树莓派；百度大模型；MQTT；STM32F205RET6

Abstract

In recent years, the Internet of Things, artificial intelligence and embedded technology are constantly developing, and the development speed is fast, which has prompted intelligent hardware and sensors in many fields have been widely used. This paper mainly explores how to design a smart home health monitoring system based on Raspberry PI, so as to meet the needs of multi-parameter health monitoring and intelligent environmental management in the context of modern families entering the aging society and the high incidence of chronic diseases. This research uses a combination of hardware and software to carry out the design work. In the hardware part, the STM32F205RET6 microcontroller is used as the core part, which is responsible for data collection and device control, and the Raspberry PI 5 is used as the data processing and display platform, and a variety of sensors and actuators are integrated together. In the software part, the Qt framework is used to develop the user interface, voice recognition and dialogue functions are achieved by calling Baidu big model API, MySQL database and MQTT server are built by Alibaba Cloud server, and data storage and transmission related operations are completed. In addition, the FreeRTOS real-time operating system is used to improve the response speed of the whole system, and finally the system performance is verified by experimental tests. The test results show that the system can carry out real-time monitoring of various environmental parameters such as heart rate, blood oxygen, body temperature and air quality, and can realize intelligent environmental regulation. It has good practicability, good performance in scalability, and low power consumption and low cost.

Key words: Smart Home Health Monitoring; Raspberry PI; Baidu Big model; MQTT; STM32F205RET6

目录

| | |
|--------------------------------|-----------|
| 1 绪论 | 1 |
| 1.1 研究背景和意义 | 1 |
| 1.1.1 研究背景 | 1 |
| 1.1.2 研究意义 | 2 |
| 1.2 国内外研究现状 | 3 |
| 1.3 研究内容和思路 | 4 |
| 1.3.1 研究内容 | 4 |
| 1.3.2 研究思路 | 5 |
| 1.4 论文组织结构安排 | 6 |
| 2 智能家庭健康监测系统概述 | 8 |
| 2.1 智能家庭健康监测系统的定义与组成 | 8 |
| 2.1.1 系统定义 | 8 |
| 2.1.2 系统组成 | 8 |
| 2.2 功能特点 | 9 |
| 2.3 智能家庭健康监测系统的发展现状与应用前景 | 9 |
| 2.3.1 发展现状 | 9 |
| 2.3.2 应用前景 | 9 |
| 3 智能家庭健康监测系统设计 | 11 |
| 3.1 智能家庭健康监测系统的功能需求分析 | 11 |
| 3.1.1 需求分析概述 | 11 |
| 3.1.2 外部接口需求 | 11 |
| 3.1.3 模块功能需求分析 | 12 |
| 3.2 系统结构分析 | 13 |
| 3.3 系统工作流程分析 | 14 |
| 4 智能家庭健康监测系统开发 | 15 |
| 4.1 监测/采集终端开发 | 15 |
| 4.1.1 硬件设计 | 15 |
| 4.1.2 软件设计 | 20 |
| 4.1.3 监测/控制终端程序设计 | 23 |
| 4.2 控制/显示终端开发 | 26 |

| | |
|-----------------------------|-----------|
| 4.2.1 软件移植和环境配置 | 27 |
| 4.2.2 Qt 软件程序开发 | 28 |
| 4.3 智能家庭健康监测系统服务器搭建 | 35 |
| 4.3.1 MQTT 服务器的搭建 | 35 |
| 5 智能家庭健康监测系统测试 | 36 |
| 5.1 硬件测试 | 36 |
| 5.2 软件编译 | 38 |
| 5.2.1 单片机程序编译 | 38 |
| 5.2.2 Qt 程序的编译 | 38 |
| 5.3 系统功能测试 | 39 |
| 5.3.1 系统通信功能测试 | 40 |
| 5.3.2 功能测试总结 | 43 |
| 6 总结与展望 | 44 |
| 6.1 研究总结 | 44 |
| 6.2 研究展望 | 45 |
| 参考文献 | 47 |
| 致谢 | 48 |

1 绪论

1.1 研究背景和意义

1.1.1 研究背景

根据 2020 年第七次人口普查所公布的数据来看，我国年龄在 60 岁及以上的老年人口，其总数大概达到了 2 亿 6300 万之多，这一数量在当年全国人口总数当中所占的比例为 18.7%。在这之中，年龄达到 65 岁及以上的人口，其比例是 13.50%，大约有 3500 万的规模。与之前的 10 年相比较，60 岁以及 65 岁这两个不同年龄段的人口增长率，分别达到了 5.44%和 4.63%。按照相关的预测情况而言，等到 2050 年的时候，我国将会迎来老龄化人口的高峰阶段，到那时，平均每 3 个人里面就会出现一位老年人^[1]。在步入老龄化社会这样的大背景之下，慢性病呈现出高发的态势，这就使得在日常生活当中针对健康状况进行监测这件事显得格外重要起来。家庭成员们若是能够在平常的生活里，对各项生理参数给予相应的监测的话，那么就可以在疾病刚刚出现的早期阶段，对身体所出现的异常情况以及可能存在的疾病发出预警信息，进而使得家庭生活质量得以提升^[2]。传统意义上的医疗体系，主要是依靠医院以及各类专业的医疗机构来完成疾病的诊断与治疗工作的。不过呢，当面对如今规模日益庞大的老年群体的时候，这种传统的模式就暴露出了诸多的不足，显得有些力不从心。一方面是医疗资源的分布并不均衡，另一方面呢，频繁地前往医院就医，无论是对于老年人自身而言，还是对于其所在的家庭来讲，都额外增加了经济方面以及时间方面的成本支出。所以说，把健康管理的重点逐步地朝着家庭环境这个方向去转移，凭借着在日常生活当中开展的监测工作，以此来实现对疾病的早期预警，这已然成为提升老年人生活质量、同时减轻整个社会所面临的医疗压力的一条极为重要的途径。

随着人们对健康生活愈发关注起来，家庭健康监测设备的需求也在持续增多。不管是老年人也好，年轻人也罢，都期望能够以科学的途径去知晓自身的健康情况，进而采取一些预防举措。不过呢，现有的设备存在功能较为单一的情况，很难满足现代家庭对于多参数监测以及智能化操作方面的诸多需求。物

联网（Internet of Things, IoT）技术在快速地发展，这就使得对家庭身体健康数据进行监测有了全新的办法^[3]。把传感器、无线通信以及云服务等相关技术结合到一起，物联网便可以达成多种健康数据的采集、传输以及存储等操作。在这同一时期，人工智能技术也取得了进步，如此一来，数据的分析以及智能化反馈也就变成了现实状况。AI 借助算法模型来对采集到的数据加以分析，从中识别出异常之处，对潜在的健康风险予以预测，还能为用户给出科学合理的健康管理方面的建议。基于这样的情况，将物联网技术和人工智能技术结合起来去开发一套多参数的家庭健康监测系统，不但能够契合现代家庭健康的实际需求，而且在老龄化社会当中也能够发挥出极为重要的作用。

在此背景下，本系统旨在设计并实现基于树莓派的智能家庭健康监测系统。该系统通过多种传感器采集心率、血氧饱和度、体温、噪声和空气质量等，并且借助物联网和云服务器技术将这些数据进行传输和存储。利用百度大模型分析监测的数据，提供个性化的健康建议。通过 MySQL 数据库，记录用户的个人信息^[4]。

1.1.2 研究意义

（一）现实意义

智能家庭健康监测系统针对老龄化社会以及慢性病高发的情形给出了解决办法。借助对身体各项健康数据的监测，家庭成员得以及时知晓自身的健康状况。该系统一方面能够提升家庭生活的质量，另一方面也能够切实削减因健康状况而产生的医疗费用方面的支出。

（二）社会意义

本设计也为医疗资源的优化利用提供了可能。传统医疗资源分布不均，尤其在偏远地区，优质医疗资源难以触达。而智能家庭健康监测系统通过提供的个人健康数据，能够实现初步的远程诊断支持，为基层医疗服务减轻压力^[5]。

从政策支持角度看，本系统的研究和开发也与国家“健康中国 2030”规划目标相契合，为推动国家医疗和健康服务产业的数字化转型提供了实践依据^[6]。

（三）个人意义

就我个人来讲，本项目无疑是一次能把理论和实践相结合的极为难得的机会。我准备以“基于树莓派的智能家庭健康监测系统”当作研究的目标所在，凭借对该系统进行软硬件方面的设计以及开发工作，进而把在课堂上学到的诸

如嵌入式技术、物联网通信协议、数据分析方法等等诸多知识，全部运用到实际的项目里面去。

在该项目里面，我会去完成有关数据采集、传输以及分析模块的相关设计工作，并且要将其切实实现出来。与此同时，还会对怎样借助物联网技术达成家庭健康参数的智能监测展开研究。另外，通过对云端数据存储以及 AI 健康状况分析模型的开发，我能够更进一步地去领会物联网和人工智能技术集成应用方面的情况^[7]。

1.2 国内外研究现状

智能家庭健康监测系统的研究，在国内外皆呈现出极为活跃的发展态势。就国内而言，近年来相关研究更是热度颇高，这主要得益于物联网、人工智能等技术的快速发展，以及人们对于健康管理需求的不断增长。学者们常常会着重关注怎样把这些技术切实应用到实际场景当中，例如借助可穿戴设备以及智能家居传感器来采集心率、血氧之类的数据，随后再与云技术相结合，以实现数据的传输、存储以及分析等操作。这种以实践为导向的研究在国内已然占据了主导地位^[8]。诸多团队都在全力投入开发低成本且高精度设备的工作当中，期望能够促使该系统得以进入更多的普通家庭。与此同时，学者们也在对如何应对慢性病管理、老年人监护等各种各样不同的场景展开研究。再者，国家政策方面的支持，像是“健康中国 2030”战略^[9]，同样为这项研究给予了诸多助力。不过，若要将相关技术真正推广普及至千家万户，标准化的制定以及用户的接受程度，还得花费一定时间去逐步磨合才行。

反观国外，这方面的研究起步更早，整体显得更加成熟，理论和实践的结合也更加紧密。且国外已从技术设备开发与系统集成应用转向探讨老年人对技术的接受与采用；国际研究的知识基础可分为老年技术采纳评估、健康监测技术创新和智能健康管理应用三类。后续应加强工程学和医学多领域的跨学科交叉研究，在健康监测技术与医疗产品融合、用户体验优化及长期效用验证等方面进行深入探讨^[9]。

总的来说，国内外的研究目标其实是一致的，都是想通过技术提升健康管理效率和生活质量，用的技术也大同小异，比如无线传感器、物联网、云技术、AI 等技术。只是侧重点不一样：国内更倾向于把系统做实惠、做普及，解决基

础的需求；国外则在技术的前沿性和专业化上更有优势，同时理论研究和商业化也更成熟。未来随着技术的更新迭代，像 6G、区块链和人工智能技术的发展可能会带来更多的可能性，国内外的研究者都在努力探索如何用好这些工具，设计出既智能又安全的系统，让家庭的健康监测真正地融入家庭生活中。

1.3 研究内容和思路

1.3.1 研究内容

此篇论文意在设计并打造出一个依托树莓派构建起来的智能家庭健康监测系统。其主要是想凭借现代各类技术的相互交融，达成针对人体健康方面各项指标以及室内环境状况展开实时监测、给予智能反馈并且实现便携化管理的目标。该系统把 STM32 微控制器当作是进行数据采集以及实时控制的核心设备，该设备运行着 FreeRTOS 操作系统，承担起多传感器数据的采集工作以及传输任务。同时，选用树莓派 5 来作为主控设备，在其之上运行 Linux 系统，进而开发具备 Qt 界面的用户程序，以此来实现对数据的接收、分析、显示以及设备控制等一系列功能。此系统不但能够对心率、血氧、体温之类的身体数据加以监测，而且还可以对空气质量、温湿度、噪声等环境参数有所感知，并且借助诸如风扇、雾化器这类自动控制设备来对家庭环境予以优化。另外，结合百度 AI 所给出的分析建议，以此来满足家庭健康管理方面的诸多需求。本文的研究内容主要包括：

(1) 智能家庭健康监测系统设计。首先分析系统的功能需求，包括健康数据采集、环境监测、自动反馈和用户交互等，然后选择合适的硬件设备，如 STM32F205RET6 芯片，树莓派 5，以及 MAX30102、AHT20 等传感器模块。确定好软件开发工具，如 KEIL 用于单片机的编程，Qt 和 MySQL 用于树莓派界面与数据管理；设计系统结构，采用 MQTT 协议实现 STM32 与树莓派之间的通信，并制定传感器数据上传与控制命令的协议。

(2) 智能家庭健康监测系统开发。首先搭建开发环境，在 STM32 端开发裸机驱动程序验证各个传感器的功能，并移植 FreeRTOS 实现多任务管理，通过 WiFi 模块将数据上传至 MQTT Broker；在树莓派端基于 Qt 开发登录、注册、Home 主界面、健康记录与健康报告等程序，集成 MySQL 存储数据，并通过 HTTP 协议调用百度 AI 大模型进行分析，同时实现风扇、雾化器等设备的控制。

(3) 智能家庭健康监测系统的性能评估与优化。通过实验验证系统的各项功能，如数据的采集的准确性、传输的实时性以及控制响应的可靠性，根据测试过程中发现的问题，逐一地进行改进和优化，最终构建一套稳定可靠的智能家庭健康监测方案。

1.3.2 研究思路

(1) 查阅诸多相关文献资料，具体是去查阅国内外在智能家庭健康监测系统的诸多文献，以此来确定当下的研究实际状况以及技术发展趋势。把自身的设计相关想法与之相结合，去考量让树莓派和 STM32 相互配合开展工作的可操作性，进而提出本文的关键研究内容，也就是去开发出一个能够把健康监测、环境感知以及自动反馈这些功能都囊括其中的智能系统。

(2) 硬件选型。选用树莓派 5 作为主控核心，负责数据处理和用户交互，STM32F205RET6 微控制器作为数据采集与控制设备的核心处理器。根据功能需求，选择 MAX30102 模块监测心率和血氧，GY906 探头测量体温，Y01 传感器和 DC01 红外模块分别检测空气质量和 PM2.5，AHT20 采集环境的温湿度，LM2904 感知噪声；同时使用直流电机、雾化器、舵机和蜂鸣器作为反馈设备，通过使用嘉立创工具绘制 PCB，然后焊接组装形成完整的硬件系统。

(3) 架构选择和软件选型。系统采用 C/S 架构，树莓派作为服务端处理数据并提供操作界面，STM32 作为客户端采集并上传数据。软件开发选用跨平台的 Qt 作为树莓派端的主要技术，用于界面设计，同时集成 MySQL 数据进行数据存储和管理；STM32 端使用 KEIL 开发单片机程序，并移植 FreeRTOS 操作系统。

(4) 通信功能设计。设备间通过 WIFI 实现网络通信，采用 MQTT 协议传输数据，确保采集数据的实时性和可靠性。设计 JSON 数据格式，用于树莓派与 STM32 之间的高效通信，如传感器数据上传和设备控制指令的下发。

(5) 系统测试。通过实际测试验证系统的核心功能，包括数据采集的准确性、传输的及时性以及控制的稳定性。

1.4 论文组织结构安排

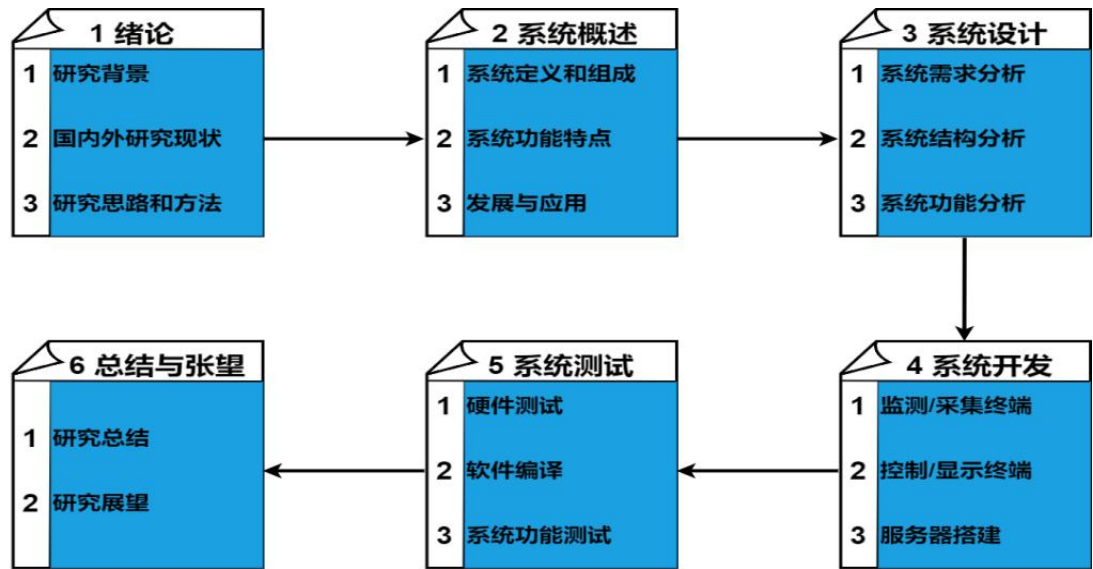


图 1.1 论文组织结构图

第一章：着眼于智能家庭健康监测当下的现实需求以及其呈现出的技术发展趋势，讲清楚此次选题的意义所在，对国内外在这方面的研究现状展开分析，进而明确以树莓派和 STM32 为基础构建智能家庭健康监测系统的相关研究内容以及具体的研究思路。

第二章：所涉及的内容为智能家庭健康监测系统的相关概述。在这部分内容里，通过查阅各类文献来对系统加以明确界定，并且细致分析其具体组成部分，进而介绍在本文当中所设计出来的健康监测以及环境控制这两方面所具备的功能特点，同时还会探讨该系统在家庭领域以及医疗领域得以应用的相关可能性。

第三章：聚焦于智能家庭健康监测系统的设计方面。在这一章当中，着重对该系统的功能需求展开分析，同时着手进行硬件以及软件结构的设计工作，进而清晰明确地界定出关于数据采集、数据传输、数据控制，还有用户交互等多个方面的功能模块。

第四章：主要围绕智能家庭健康监测系统开发展开阐述。详细说明了硬件选型以及组装方面的情况，同时对 STM32 数据采集终端、树莓派控制显示终端的相关内容介绍，还涵盖了服务器搭建的具体开发流程以及其功能是怎样实现的。

第五章：测试硬件采集精度与控制响应、软件通信与界面稳定性及整体功能表现，优化系统性能。

第六章：总结与展望。此部分主要是对基于树莓派所开展的智能家庭健康监测系统相关设计以及实现方面所取得的成果加以总结，同时针对当下仍然留存的一些问题展开分析，进而对该系统在未来技术层面能够做出的改进以及应用方面可以实现的扩展方向予以展望。

2 智能家庭健康监测系统概述

2.1 智能家庭健康监测系统的定义与组成

2.1.1 系统定义

智能家庭健康监测系统是依靠物联网技术、运用传感器并借助无线通信来构建的，它是一个能对老年人以及家庭成员的身体健康指标还有家庭环境参数予以实时采集并监测的综合系统。赵有明^[10]指出，智慧家庭健康监测系统重点关注慢性病患者和独居老人，着重通过低成本且能实时监测的方式来减轻医疗负担，进而实现家庭健康管理，与此同时还借助服务器端和客户端的沟通来传输数据，以确保医生和家属可以及时知晓健康状况。凌信航^[11]又提到，该系统一般是以微控制器作为核心，凭借 WIFI、MQTT 或者 NB-IOT 等技术把数据传输到云平台端或者移动端应用上，从而达成远程监控、异常报警以及智能控制等功能，其强调系统要集成可穿戴设备以及室内环境监测，以此来满足后疫情时代人们对于家庭健康和环境健康的需求。从整体来看，智能家庭健康监控系统有着低成本、高精确度、具备实时性以及用户友好等特征，适用于家庭健康管理、远程监控、医疗辅助等场景的应用。

2.1.2 系统组成

本文的树莓派智能家庭健康监测系统结构主要分为不同的功能模块，各功能模块的介绍如下：

- (1) 健康监测模块。包括心率、血氧、体温等传感器，利用 STM32 采集人体健康数据，并发送至树莓派显示，供用户了解自身健康情况。
- (2) 环境调节模块：包括空气质量监测传感器、温湿度传感器、噪声传感器、风扇、雾化器、舵机、继电器等执行器，实现对室内环境的智能监测和调节，给人们营造一个健康的、安全的居家环境。
- (3) 智能控制模块。利用无线通信技术 WIFI 和 MQTT 协议，实现对家庭设备的远程控制管理，用户可以通过树莓派 Qt 界面远程开关风扇、雾化器或调节舵机开关等功能。

(4) 人工智能交互模块。通过 HTTP 协议调用百度 AI 大模型，实现健康数据的分析与对话功能，生成个性化健康建议，并为用户提供咨询服务，增强系统的智能化和用户体验。

2.2 功能特点

本文介绍的家庭健康监测系统，主要具备以下几个核心功能：

(1) 健康数据与环境监测。通过心率血氧传感器、体温探头、空气质量传感器、PM2.5 传感器、噪声传感器及温湿度传感器，实现人体健康数据和室内环境数据的采集。

(2) 智能数据传输和云端互联。基于 STM32 的多种通信接口采集传感器数据，并通过 WIFI 和 MQTT 协议上传至云端，支持远程访问与数据管理。

(3) AI 分析与个性化建议。将采集的数据上传至百度 AI 模型进行分析，生成健康建议，同时支持 AI 对话功能，增强与用户的交互性。

(4) 用户友好交互设计。提供美观的登录注册界面、多功能数据显示界面以及软键盘支持，适配家庭多用户管理，操作便捷直观。

2.3 智能家庭健康监测系统的发展现状与应用前景

2.3.1 发展现状

智能家庭健康监测系统正慢慢变成家庭生活里极为重要的一个组成部分，物联网、人工智能、传感器还有云计算等相关技术也一直在不断取得进展。该系统运用微控制器、嵌入式平台、多种多样的传感器模块以及云服务器，以此来达成对人体健康参数和室内环境进行实时监测以及智能管理的目的。虽然在市场上已然出现了像智能手环、空气净化装置等不少与之相关的产品，然而那种把健康监测、环境感知、环境调控以及人工智能分析深度融合起来的家庭健康综合解决方案，当前仍旧处在高速发展的时期。凭借着低成本、高度集成化的设计方式，这一系统在家庭场景当中具备颇为可观的应用潜力，并且技术的持续更新也给系统的功能拓展带来了更多的发展可能性与可扩展性。就比如说，MQTT 协议的运用让数据传输效率有了明显的提高，而将人工智能技术融入其中则使得对数据的分析变得更加智能化了。

2.3.2 应用前景

在当下老龄化态势不断加剧，且人们对自身身体状况愈发关注的社会大环境之中，智能家庭健康监测系统于养老服务方面、健康保险领域以及智慧社区的建设进程里，都彰显出颇为可观的市场潜力。企业能够借助定制各类功能的方式，去契合不同用户所具有的多样化需求。该系统依靠着多种多样的功能、相对较低的成本以及较高的扩展性，在家庭健康管理范畴、环境优化相关事宜以及智能生活这片广阔领域，均有着较为不错的发展前景。伴随技术持续取得进步，再加上用户需求呈现出多样化的特点，这个系统是很有希望能够成为智能家居生态体系里极为重要的一个组成部分，进而为提升人们的生活质量，以及推动社会整体健康水平的提升贡献一份力量。

3 智能家庭健康监测系统设计

3.1 智能家庭健康监测系统的功能需求分析

3.1.1 需求分析概述

智能家庭健康监测系统由三个核心部分组成：

- (1) 数据采集与控制终端：基于自制的 PCB 板，集成 STM32F205RET6 及多种传感器，负责采集健康数据和环境数据、处理数据并执行反馈控制。
- (2) 数据处理与显示终端：采用树莓派 5 运行嵌入式 Linux 操作系统，通过 Qt 框架实现数据的接收、存储、分析以及可视化展示。
- (3) 云服务器：基于阿里云服务器，运行 MQTT Broker 和 MySQL 数据库，支持数据上传和存储，同时结合百度 AI 模型进行分析。
- (4) 通过这三部分的协同工作，系统能够实现家庭成员健康监测、环境优化以及智能化管理。

3.1.2 外部接口需求

(1) 用户接口需求

树莓派上的 Qt 客户端，提供图形化的界面，支持用户触摸操作。

(2) 硬件接口需求

① 自制的 STM32 开发板：

- IIC 接口：MAX30102 心率血氧传感器、GY906 体温探头、AHT20 温湿度、OLED 显示屏
- USART 接口：空气质量传感器、PM2.5 传感器、ESP8266-01s
- ADC 接口：LM2904 噪声传感器
- PWM 接口：直流电机风扇、舵机
- GPIO 接口：LED、蜂鸣器、继电器、雾化器模块

② 树莓派 5：

- USB 接口：麦克风、喇叭
- HDMI 接口：7 英寸触摸显示屏

(3) 软件接口需求

- ① 嵌入式 Linux 操作系统：树莓派 5 运行 ubuntu24.10。
- ② Qt 库：用于界面开发。
- ③ MySQL 数据库：连接阿里云服务器的 MySQL 数据库。
- ④ MQTT 客户端：连接阿里云服务器部署的 Mosquitto。
- ⑤ C/C++ 语言：STM32 和树莓派程序开发。
- ⑥ FreeRTOS 支持库：STM32 实时任务调度。

(4) 通信接口需求

- ① IIC、USART、ADC、GPIO、PWM：传感器与 STM32 开发板进行通信。
- ② WIFI：STM32 与阿里云服务器中的 MQTT Broker 进行通信。
- ③ USB：树莓派与外设通信。
- ④ MQTT 协议：STM32 与树莓派通过阿里云 MQTT Broker 通信。
- ⑤ HTTP 协议：百度 AI 模型交互。

3.1.3 模块功能需求分析

(1) 输入/输出数据

- ① 输入数据（STM32）：
 - 健康数据：心率（次/分钟）、血氧饱和度（%）、人体温度（℃）
 - 环境数据：TVOC 浓度（ppm）、CO2 浓度（ppm）、甲醛浓度（ug/m³）、PM2.5 浓度（ug/m³）、噪声水平（db）、环境温度（℃）、环境湿度（%RH）
- ② 输出数据（树莓派）：
 - 健康数据展示：心率、血氧、体温数值
 - 环境数据展示：TVOC、CO2、甲醛、PM2.5、噪声、温湿度数值
 - 设备状态：风扇、雾化器、舵机、蜂鸣器
 - AI 分析结果：健康建议、对话

(2) 功能描述

本系统的功能涵盖健康监测、环境控制和设备控制等功能，具体如下：

- ① 健康监测功能：通过 MAX30102 采集心率和血氧数据，GY906 测量体温，实时显示于 Qt 界面。

- ② 环境监测与自动调节功能：采集 TVOC、CO₂、甲醛、PM_{2.5}、噪声、温湿度数据，根据阈值自动控制。
- ③ 设备管理功能：通过 Qt 界面手动控制风扇，加湿器、舵机等的状态。
- ④ AI 健康分析功能：通过 HTTP 协议上传至百度 AI 模型，生成健康建议并支持 AI 对话。
- ⑤ 多用户管理：通过 MySQL 数据库支持多用户注册与登录，记录家庭成员信息如：姓名、年龄、性别、身高、体重，界面支持用户的切换。

3.2 系统结构分析

这套系统以物联网技术为基础，把智能家庭健康监测系统所具备的全部功能划分成了感知层、网络层以及应用层这三个不同的层次。感知层作为数据采集的终端，主要实现传感器数据的采集和设备的控制功能。网络层充当着感知层与应用层之间进行通信的桥梁，其采用的是 WIFI 技术来传输数据，并且依靠以 TCP/IP 协议栈为基础的 MQTT 协议达成高效且可靠的网络通信。至于应用层面，则是以树莓派作为硬件平台，运行基于 QT 框架所开发的 Qt 软件，该软件承担着可视化显示、与用户进行交互、生成健康数据以及发送相关控制指令等诸多职责，系统架构可参照图 3.1 所示内容：

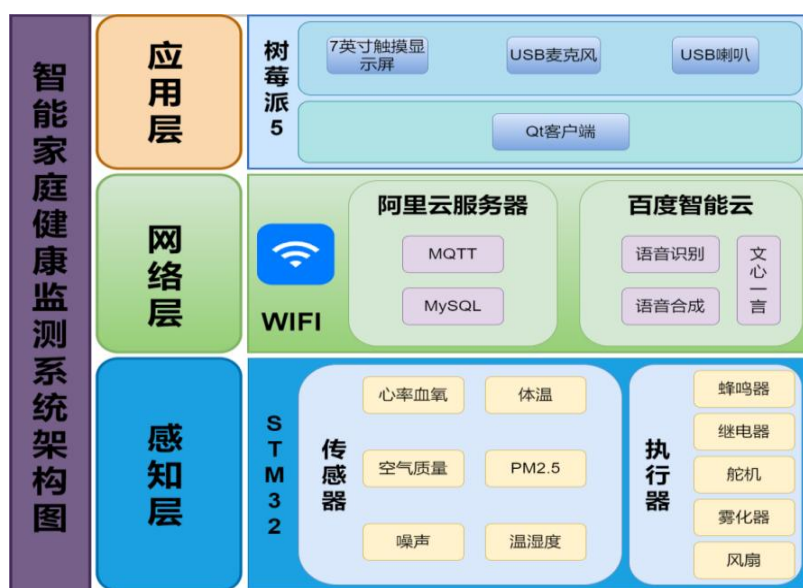


图 3.1 智能家庭健康系统架构图

3.3 系统工作流程分析

智能家庭健康监测系统主要包含三个部分，分别是控制或监测终端、客户端程序以及云服务器。控制或监测终端能够凭借串口、IIC 以及 GPIO 等通信接口来达成和各类传感器、控制器之间的数据传输任务。客户端程序是在树莓派上运行的，其借助 USB 接口把麦克风和喇叭连接起来，进而实现音频的输入与输出操作，并且还能通过 WIFI 网络或者有线网络与云服务器以及百度 AI 大模型构建起连接关系，在与百度云数据通信方面采用的是 HTTP 协议。云服务器是以阿里云平台为基础搭建起来的，在该云平台之上部署了 MQTT 服务以及 MySQL 数据库。而控制或监测终端则可通过 WIFI 模块，运用 MQTT 协议来和云服务器展开高效的数据通信。关于各个部分具体的通信结构，如图 3.2 所示：

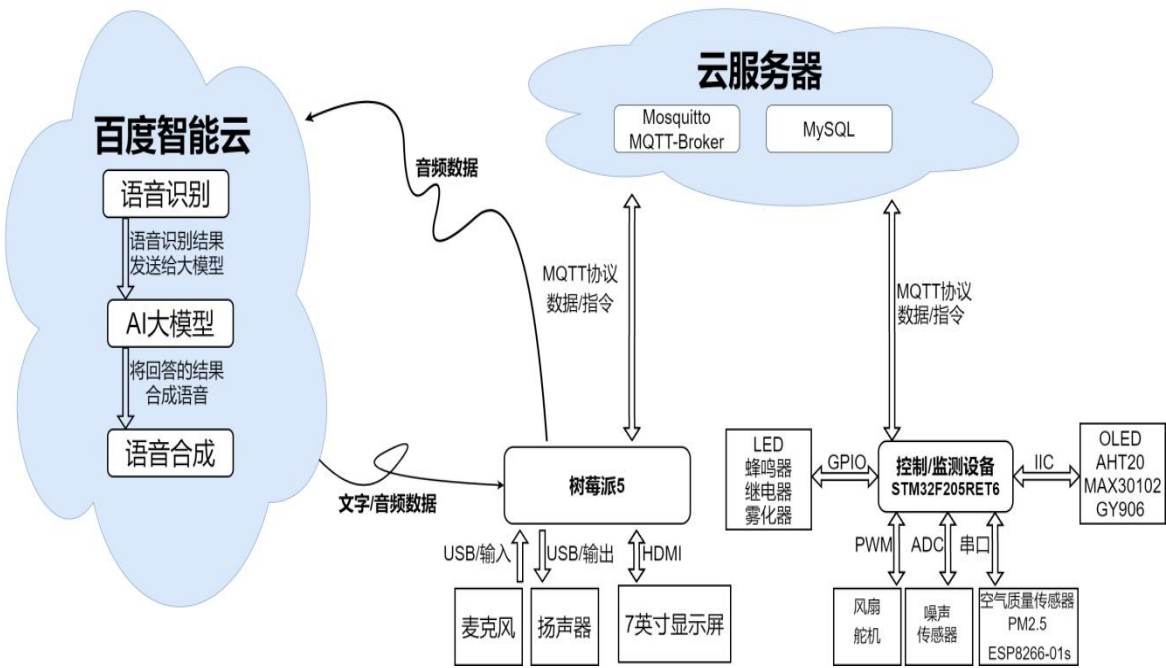


图 3.2 系统通信流程图

4 智能家庭健康监测系统开发

4.1 监测/采集终端开发

4.1.1 硬件设计

监测与控制终端是智能家庭健康监测系统当中的核心硬件，它承担着采集健康数据以及环境数据的重要职责，并且还会对采集到的数据做初步的处理工作，同时也负责对相关设备实施控制。此硬件在设计时，使用的是以自制的 PCB 板为基础来开展的，选用了 STM32F205RET6 微控制器来作为主芯片，将多种传感器和执行器有效结合起来，借助于合理的电路设计以及恰当的接口配置方式，以此来确保数据采集能够达到应有的准确性，也让设备控制具备可靠的性能。

(1) 电源部分

电源主要通过 USB 接口或电源接口输入 5V，经 AMS1117-3.3 线性稳压器输出 3.3V，为 STM32 提供 3.3V 电压，为风扇，舵机等器件提供 5V 电压。加入 10uF 与 100nF 去耦电容滤除电源的噪声。通过 CH340X 芯片将 USB 信号转为串口信号方便调试。具体原理图如图 4.1 和图 4.2 所示：

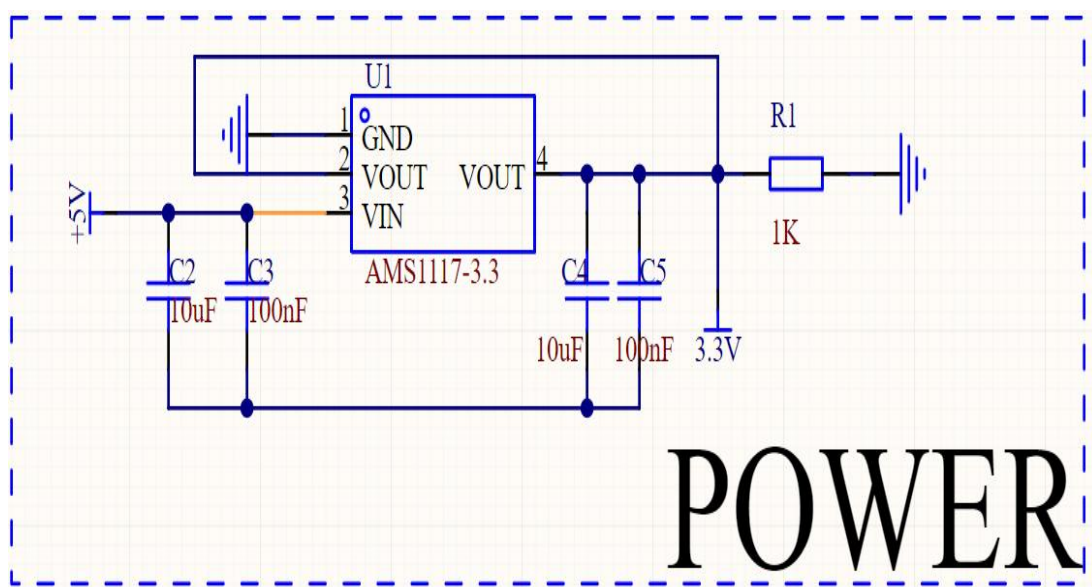


图 4.1 电源模块

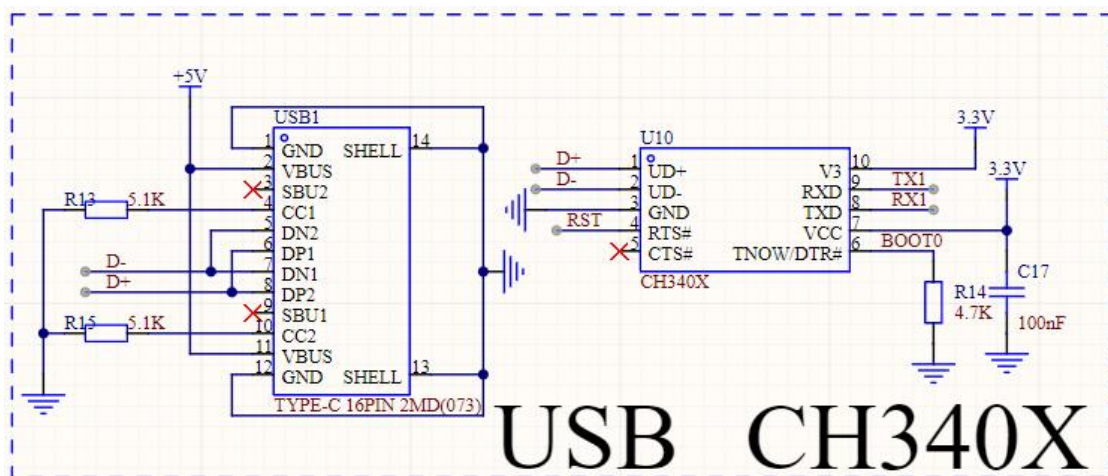


图 4.2 USB 模块

(2) STM32F205RET6 外围电路

STM32F205RET6 外围电路是本系统的核心部分，STM32F205RET6 是一款 32 位 Cortex-M3 微控制器，具有丰富的 GPIO 引脚、多通信接口和内部外设。原理图包括微控制器核心部件以及与之相连的外围电路，如电源管理电路、时钟电路、复位电路，调试下载电路等。如图 4.3 所示：

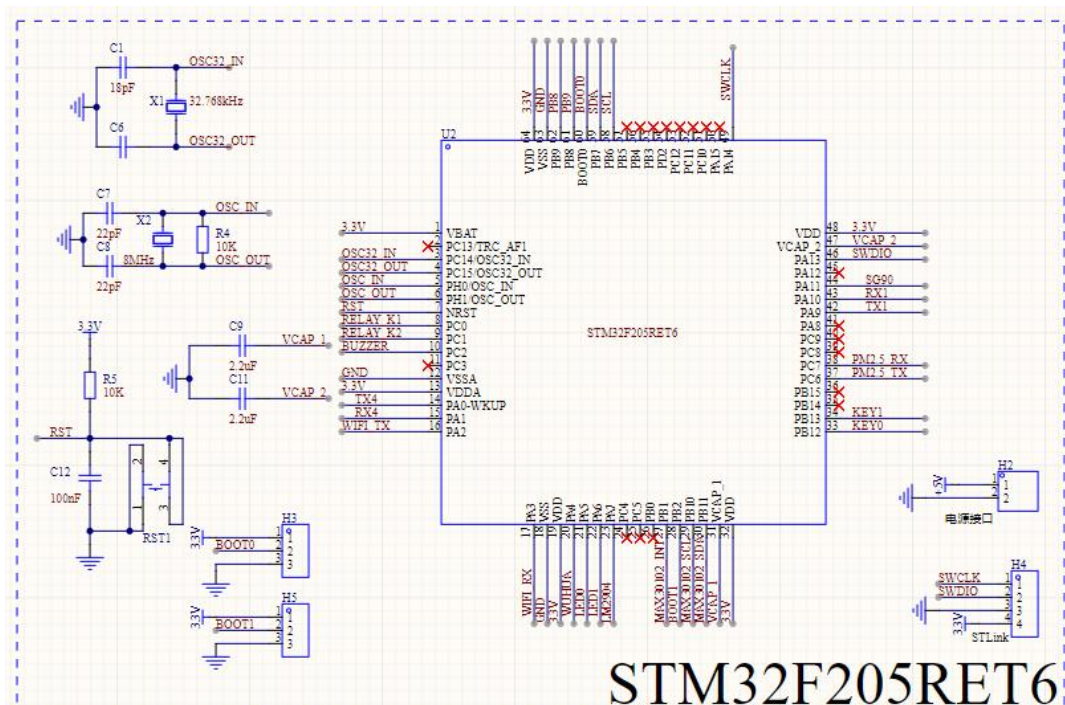


图 4.3 STM32F205RET6 外围电路

(3) 串口电路

在本设计里，串口电路发挥的主要作用是达成微控制器和外部传感器以及执行器相互之间的数据通信。该电路运用串行通信协议，凭借 UART 接口来实现数据的接收与发送。从图 4.4 能够看到，一方面有 PM2.5 传感器、空气质量传感器跟微控制器的串口连接，另一方面也展示了 WIFI 模块与微控制器的串口连接，这些都充分体现出多设备之间串口扩展以及数据交互方面的功能。

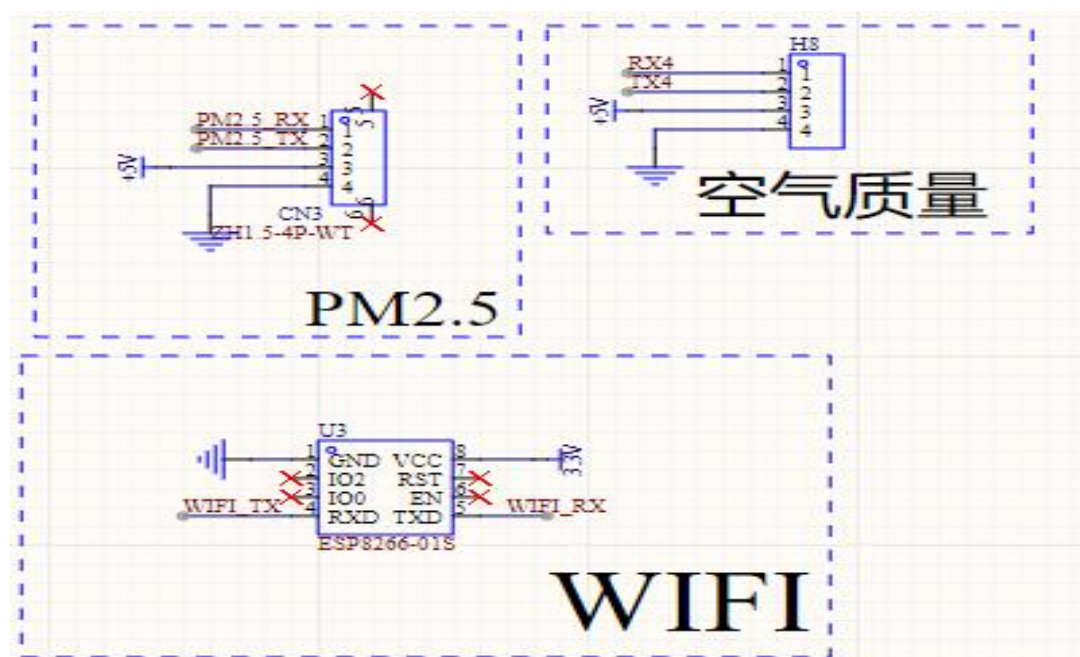


图 4.4 串口电路

(4) IIC 电路

在 IIC 电路设计环节，OLED 屏幕、AHT20 温湿度传感器还有体温传感器借助 IIC 总线来实现高效的通信。设计方面充分运用了 IIC 总线具备的多设备扩展能力，其硬件连接得以简化，并且功耗也比较低，这些优点共同促使系统的集成效率获得了提升。在电路当中，通过选用 $4.7k\Omega$ 的上拉电阻并且搭配上滤波元件，以此确保了数据通信能够具备稳定性以及可靠性。STM32 微控制器在其中充当总线的主设备，而 OLED 屏幕、AHT20 温湿度传感器以及体温传感器则是作为从设备存在，仅仅依靠 SDA 和 SCL 这两条信号线，便可以开展数据的通信工作。具体的连接方式可参照图 4.5 所示：

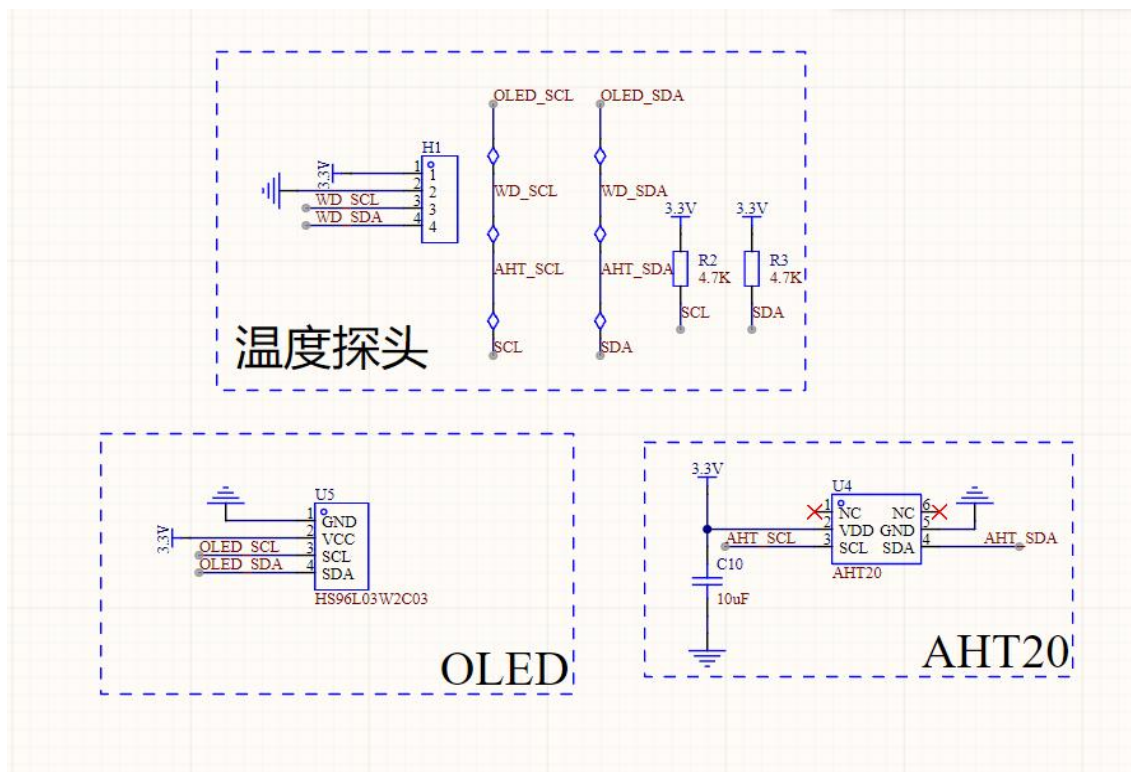


图 4.5 IIC 电路

(5) GPIO 口电路

在 GPIO 口的电路设计环节，按键、LED 指示灯以及蜂鸣器这几个部件，都是依靠 STM32 微控制器的 GPIO 引脚予以控制的。该设计充分发挥了 GPIO 接口所具备的多功能特性以及连接简便的优势，以此促使系统的交互效率得以提升。在电路当中，借助上拉电阻、限流电阻还有三极管驱动等方式，确保了信号的稳定状态以及驱动的可靠程度。STM32 微控制器扮演着控制主设备的角色，与之相对的，按键、LED 指示灯以及蜂鸣器则属于从属模块，仅仅凭借 GPIO 引脚，便能够对这些输入输出模块实施灵活有效的控制。其具体的原理图可参照图 4.6 所示内容：

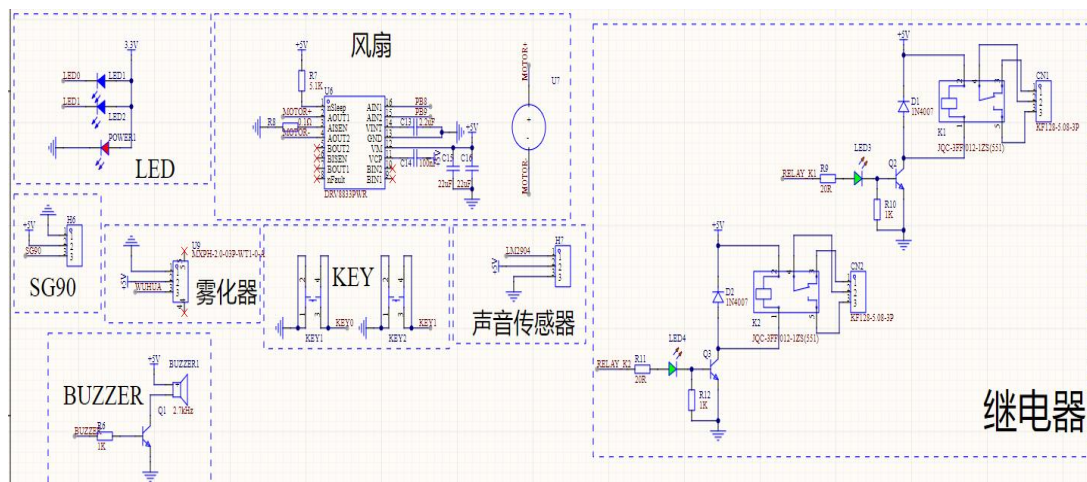


图 4.6 GPIO 口电路

(6) 硬件系统整体设计总结

此智能家居健康监测系统在硬件设计方面，将 STM32F205RET6 微控制器作为核心部件，同时融合了各式各样的传感器，由此达成了对健康数据、环境数据的及时采集、对采集到的数据进行处理以及实现设备控制等一系列功能。在设计过程中，凭借着对电路模块进行合理划分的方式，切实保证了数据采集具备较高的可靠性。该硬件系统能够应用于智能家居以及健康监测等诸多场景之中，从而为家庭用户带来便利的环境管理以及健康监测方面的服务。本系统的 3D 模型图以及实物图可参照图 4.7 与图 4.8。

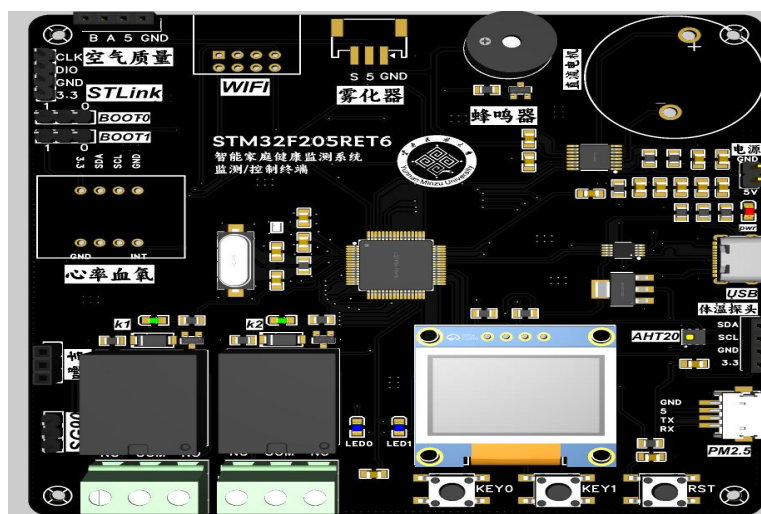


图 4.7 PCB3D 模型图

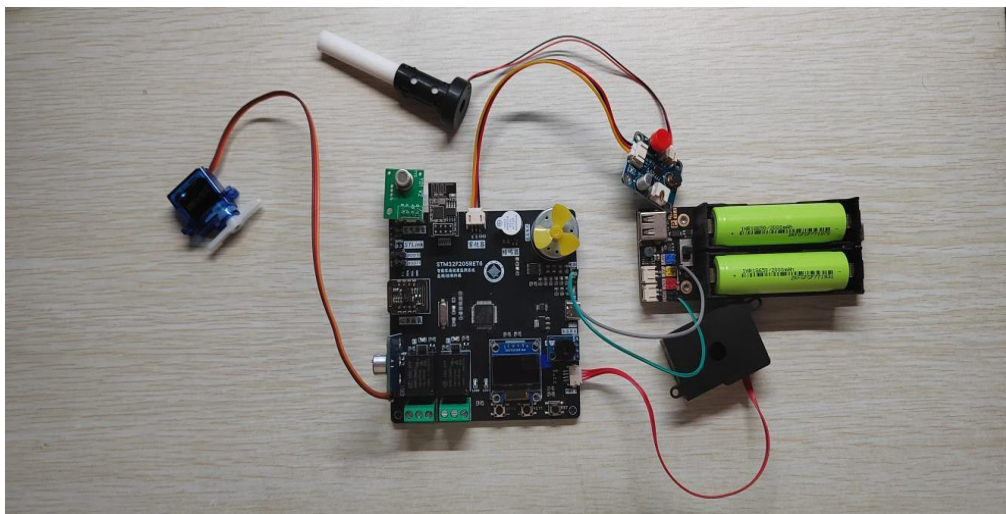


图 4.8 实物图

4.1.2 软件设计

(1) FreeRTOS 实时操作系统移植

在单片机程序的开发环节里，其开发方式大体上可划分成裸机开发以及实时操作系统开发这两种类型，这二者在具体的开发方式方面可以说各有各的特点。它们的核心差别体现为，裸机开发的整个过程显得更为直观一些，开发人员能够直接借助操作寄存器或者库函数来展开编程工作，通常是以单线程的方式来推进的，如此一来操作起来较为便捷，不过其在实时性方面表现得相对较弱。与之形成对照的是，实时操作系统的开发过程就显得稍微复杂，这需要对任务调度以及资源分配加以妥善管理才行，可它却能够有效提升系统的实时性。所以，在裸机开发的过程当中，还得针对系统去开展实时管理方面的相关工作。裸机程序是依照顺序依次执行的，而实时操作系统则是凭借任务调度来达成并行执行的。鉴于本系统牵涉到对多个传感器以及诸多设备的控制，并且还需要对相关信号展开实时的接收以及处理操作，于是我们便选取了实时操作系统的开发模式，目的在于确保系统可以及时做出响应，进而提升用户的使用体验。

当下，在单片机这一领域当中，流行着诸多不同种类的实时操作系统。就本文而言，选取了 FreeRTOS 来作为开发平台。FreeRTOS 凭借着开源且免费、开发流程较为简便等一系列优势，收获了广泛的认可。只不过，在开发的过程里面，是没办法直接去运用官方所提供的内核文件的，而是得从官方网站去下

载与之相关的一些文件，然后对其加以移植以及适配方面的操作，从而使其能够和本项目的开发板硬件环境相互匹配起来。

FreeRTOS 操作系统的移植步骤如下：

- ① 从 FreeRTOS 官网下载最新的内核源码。
- ② 在 KEIL5 中创建一个包含 STM32F205RET6 的基本配置的工程。
- ③ 将 FreeRTOS 源码中的内核文件 task.c、queue.c 等、内存管理文件 heap_4.c 以及文件 port.c 和 portmacro.h 添加到工程中。
- ④ 修改 FreeRTOSConfig.h 文件，根据项目需求来设置系统功能，比如设置任务调度方式、嘀嗒定时器的频率以及任务优先级的数量等。
- ⑤ 修改 stm32f2xx_it.c 文件，确保 SysTick 中断和 PendSV 中断处理函数适配 FreeRTOS 的调度需求。
- ⑥ 在 main.c 中编写测试任务，调用 vTaskStartScheduler 启动调度器，验证移植是否成功。测试成功后，FreeRTOS 移植完成，可直接使用系统提供的 API 函数如任务创建、信号量管理等进行后续软件的开发。

(2) 软件系统架构分析

本系统的软件架构采用分层设计，分为 APP 层、操作系统层和驱动层，运行于 STM32 监测/控制终端，确保数据采集、处理、传输和设备控制的高效实现。以下结合图 4.9 系统软件架构图进行分析各层功能及其交互。

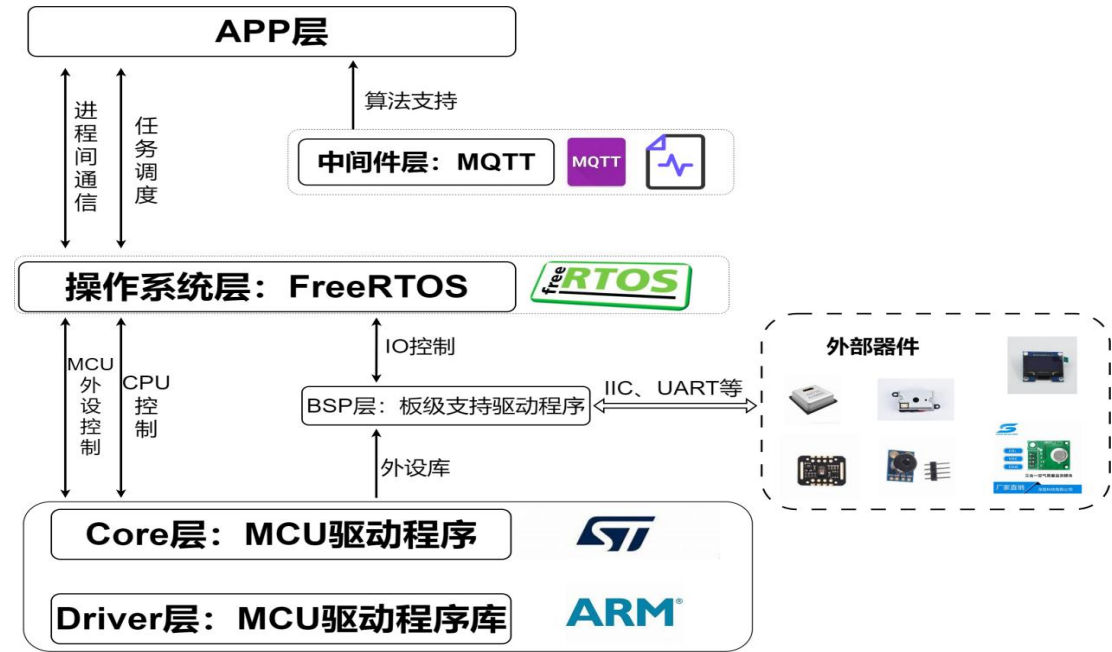


图 4.9 系统软件架构图

① APP 层

APP 层指基于 FreeRTOS 实现的任务管理与功能调度,主要包括传感器数据采集任务、数据发送任务、数据接收任务和设备控制任务:

- 数据采集任务:通过 IIC 和 UART 接口,利用 MAX30102 模块完成心率以及血氧数据的采集;依靠 GY906 探头来获取体温数据;凭借 AHT20 传感器去采集环境温湿度方面的数据;通过空气质量传感器来收集 TVOC、CO2 还有甲醛的数据;运用 DC01 传感器采集 PM2.5 的数据;另外,还借助 LM2904 传感器完成噪声数据的采集。

- 数据发送任务:利用 MQTT 协议,通过 WIFI 模块将采集到的数据发布到 MQTT Broker,实现数据上传。

- 数据接收任务:通过 MQTT 协议订阅树莓派下发的控制指令,例如设备开关命令,完成远程控制。

- 设备控制任务:根据传感器数据分析结果和收到的指令,控制执行器动作。

② 操作系统层

操作系统层基于 FreeRTOS 操作系统,负责多任务调度和资源管理:

- 任务调度:FreeRTOS 运用抢占式调度机制对多个任务予以管理,以此来保障数据采集环节、数据处理环节以及设备控制环节都能具备实时性。举例来讲,数据发送这一任务的优先级处于较高水平,其能够保证传感器所采集到的数据集合可以及时传输出去。

- 资源管理:通信信号量、消息队列等机制实现任务间的同步和通信,例如要使用消息队列将传感器采集到的数据发送给数据上传任务。

③ 驱动层

- 硬件驱动:包括 IIC、UART、ADC、PWM、TIM、GPIO 等驱动。

- 板级支持包 BSP:基于 STM32HAL 库进行开发,HAL 库提供了 STM32 的外设初始化和操作函数,例如时钟配置、GPIO 配置、NVIC 中断管理等,为上层任务提供统一的硬件访问接口。

(3) 软件系统架构总结

系统的数据流是按照从底层逐步往上层的顺序依次传递的。具体而言,驱动层会借助 IIC、UART 这类接口来采集传感器的数据,随后把采集到的数据传送到 FreeRTOS 层以便对其加以处理。FreeRTOS 层呢,依靠任务调度的方式来

完成数据的传输工作，同时还能实现任务之间的同步操作。而应用层则会依据所采集到的数据或者接收到的相关指令，去调用驱动层的接口，以此来对风扇、舵机等设备进行控制，进而达成闭环控制的效果，并且还能够通过 MQTT 协议把数据发布到云端之上。本监测/控制终端的软件架构以 FreeRTOS 为核心，结合 STM32 硬件平台，实现了多传感器数据采集、设备控制和数据传输的完整功能。分层设计和任务调度机制确保了系统的实时性和稳定性，为智能家庭健康监测提供了可靠的软件支持。

4.1.3 监测/控制终端程序设计

(1) 程序设计流程图

① 传感器数据采集流程如图 4.10 所示。

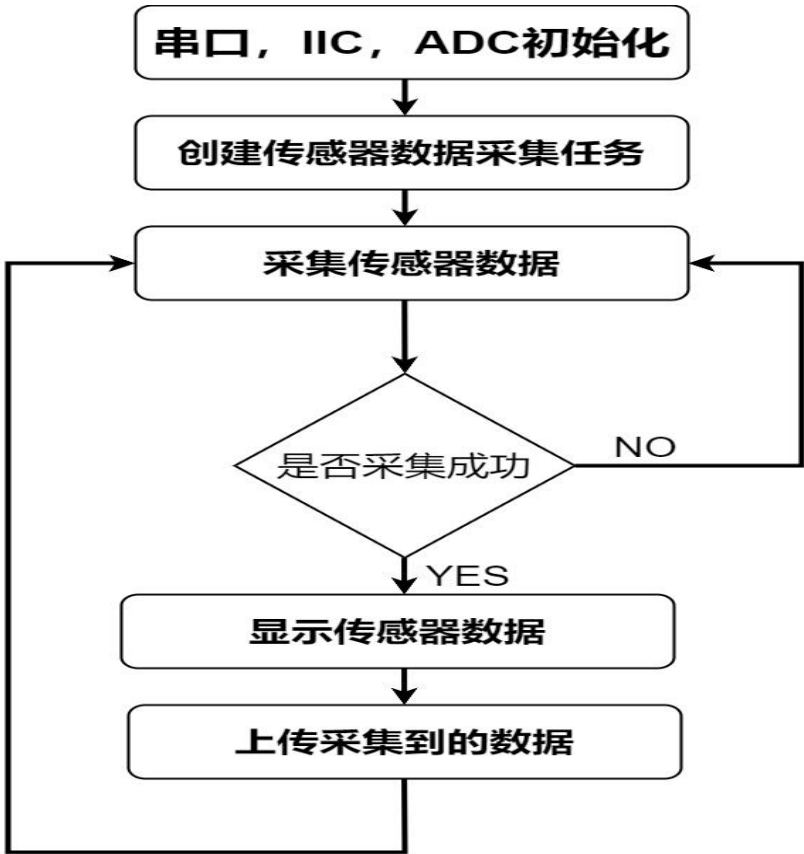


图 4.10 传感器数据采集图

② 设备控制实现流程如图 4.11 所示。

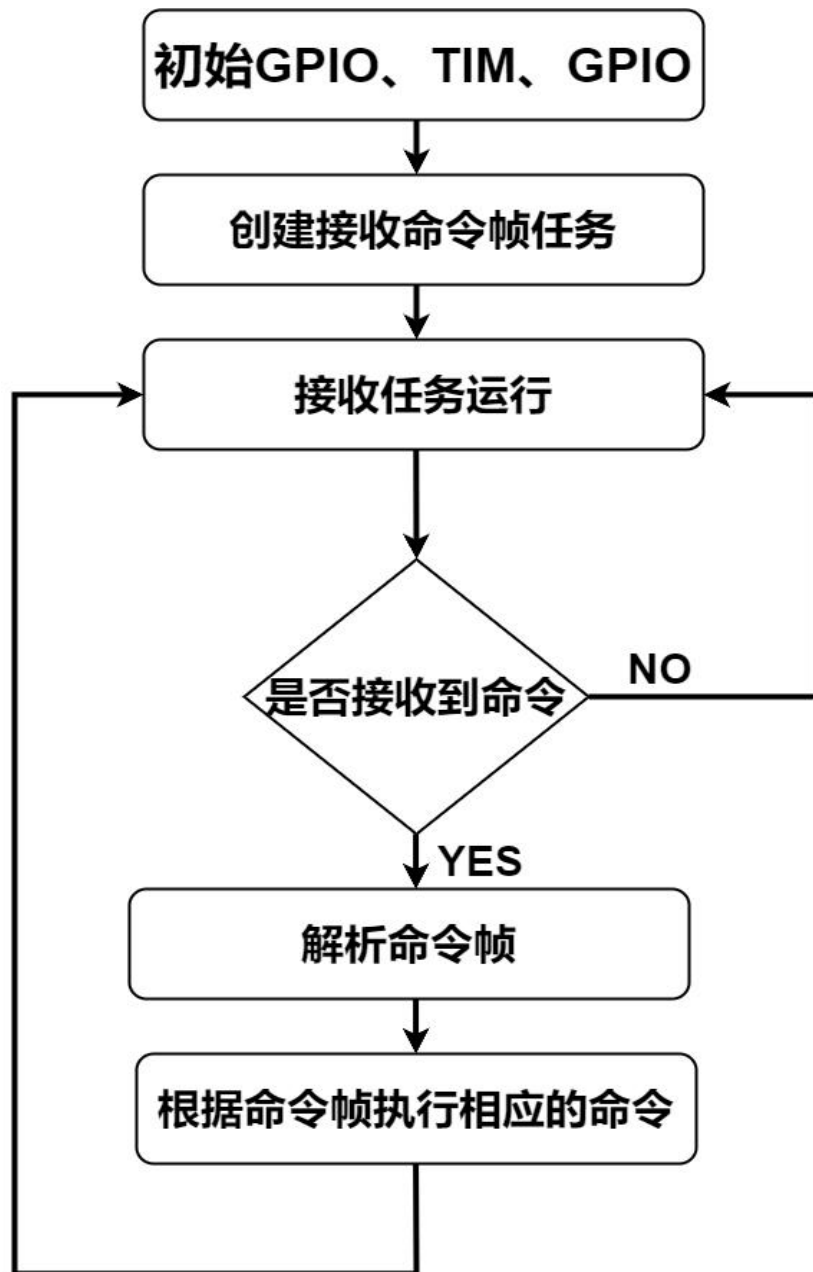


图 4.11 设备控制图

③ MQTT 通信流程如图 4.12 所示。

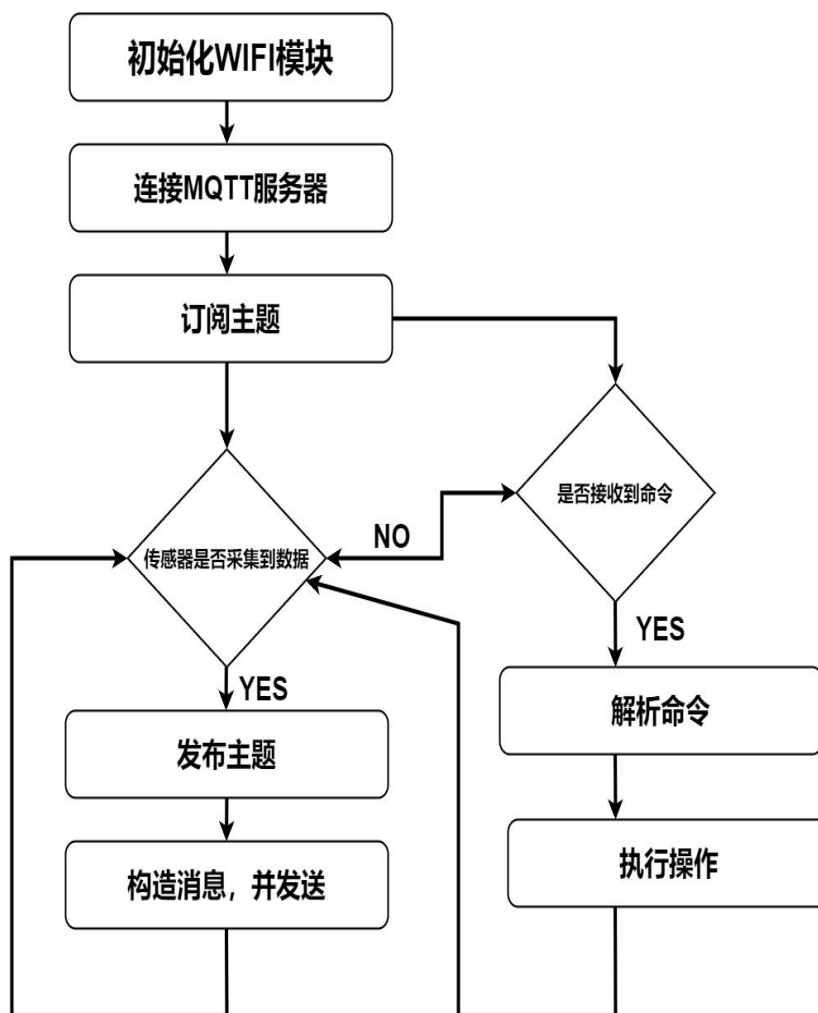


图 4.12 MQTT 通信流程图

(2) 部分编码实现

在着手开发智能家庭健康监测系统之际，我采用了依靠 FreeRTOS 实时操作系统的开发路径。通过创建诸多任务并加以合理调度的方式，以此达成系统的核心功能。该系统旨在凭借多任务同步运行这一模式，有效完成诸如传感器数据采集、数据传输、设备操控以及状态提示等一系列功能，并且借助信号量和消息队列来解决任务间的同步与通信等问题。接下来便是关于实现过程的详尽阐释：

- ① 传感器数据采集任务：包括温湿度监测任务、空气质量监测、噪声监测、体温监测、心率血氧监测，负责从各自传感器采集数据。
- ② 数据通信任务：包括数据发送任务和数据接收任务，负责通过 MQTT 协议与服务器进行数据交互。

为了促使系统能够高效地运转起来，我依照任务所具有的实时性方面的要求以及资源被占用的实际情况，对不同优先级的堆栈深度进行了相应设置。举例来讲，把数据发送方面的任务、负责数据接收的任务以及针对心率血氧展开监测的任务，都设定成了相对较高的优先级，目的在于确保数据可以及时地完成传输工作，同时也能对那些关键的健康数据实现实时性的采集操作。

4.2 控制/显示终端开发

在控制/显示终端的开发过程中，选用树莓派 5 作为处理器，使其成为整个系统当中负责核心管理以及交互的平台，该平台还一并汇聚了数据显示、设备操控还有用户交互等诸多功能。在开发进程之中，为了提升系统所具备的灵活性以及用户的使用体验，采用区分登录账号的方式去实现对家庭成员的管理。首先得进行 Ubuntu24.10 操作系统的移植工作。随后呢，要把 Qt 程序运行时所需的 Qt 客户端、MySQL 驱动、MQTT 库以及其他软件的 ARM 版本库也都一一移植过来，等这些都完成以后，还得去对接外设接口，像麦克风、显示屏之类的，只有这样，最终才可以保证 Qt 程序能够在树莓派 5 上顺利地运行起来。树莓派 5 的相关资源情况可参照图 4.13 所示内容：

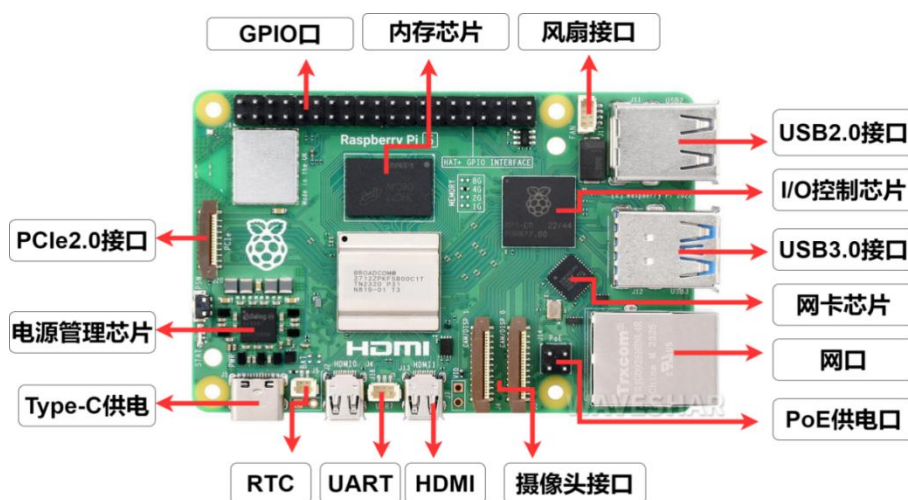


图 4.13 树莓派外设接口图

4.2.1 软件移植和环境配置

因为要在树莓派 5 中运行 Qt 图形界面程序,我们需要进行相关程序的移植,才能保证图形界面的正常运行。

(1) Ubuntu24.10 系统

树莓派能够运行许多类型的操作系统,像树莓派本身自带的 RaspberryPiOS 操作系统,除此之外,还有诸如 Apertis、AlpineLinux、Ubuntu 等其他操作系统。在此次设计当中,我把 Ubuntu 24.10 选定为树莓派 5 的操作系统。之所以选择 Ubuntu,是因为它具备丰富多样的软件生态,并且有着庞大的社区给予支持。借助 APT 包管理器,可以很轻松地安装 Qt 开发框架、MySQL 客户端这类工具,如此一来搭建开发环境就变得更为简便了。同时,Ubuntu 还能够提供完备的桌面环境,以此来对 Qt 图形界面的运行予以支持,而且它针对 ARM 架构的优化效果也比较不错,兼容性与稳定性二者兼具。再者,其活跃的社区以及技术支持对于调试和优化工作来说也是极为便利的。具体之所以选用 Ubuntu 24.10,是由于它是在 2024 年 10 月发布的最新版本,集成了最新的 Linux 内核以及软件包,能够充分地将树莓派 5 的硬件性能发挥出来,还能支持最新的 Qt 版本。这不但满足了智能家庭健康监测系统对于实时性、可视化以及交互性的高标准要求,而且还能让我在开发的过程当中去体验前沿的技术,进而为未来的优化工作打下一定的基础。下面便是相关的移植步骤:

- ① 准备一张 32G 的内存卡以及读卡器,下载树莓派镜像烧录器。
- ② 打开镜像烧录器,选择树莓派 5,选择 ubuntu24.10 操作系统下载至内存卡中。
- ③ 等待下载完成,将内存卡插入树莓派 5 中。
- ④ 启动树莓派,只要能进行 ubuntu 的桌面则移植成功。

(2) Qt5.15.13 版本移植步骤如下:

- ① `sudo apt-get install qtbase5-dev qtbase5-dev-tools qtchooser` 安装默认的 Qt 库。
- ② `sudo apt-get install qtcreator` 安装 QtCreator。
- ③ `sudo apt-get install build-essential` `sudo apt-get install libgl1-mesa-dev` `sudo apt install libfmt-dev` `sudo apt install liblog4cplus-dev` `sudo apt install libqt5sql5 libqt5sql5-psql libqt5sql5-mysql` 安装 Qt 程序运行的依赖。
- ④ 测试 Qt 程序,只要能运行程序就代表 Qt 移植完成。

(3) MQTT 的 Qt 版本移植

由于程序的开发需要支持 MQTT 协议但是 Qt 的官方库中并没有 MQTT 库，所以需要自己进行移植。

- ① 进入 GitHub 下载对应版本的 MQTT 库的源码。
- ② 用 Qt 打开然后编译即可。
- ③ 将编译好的库文件移植到 Qt 的相关目录下。

到目前为止，有关树莓派 5 的全部移植工作都已经顺利完成了，这为后续 Qt 应用程序的开发打下了基础。在嵌入式开发领域当中，往往得在性能更为强劲的 PC 端运用交叉编译工具，把相关代码编译成目标设备能够执行的文件，毕竟不少嵌入式设备所拥有的计算资源是很有限的，其内存以及处理器的性能都没办法支撑直接去运行那种复杂的开发环境以及编译流程。不过呢，树莓派 5 的内存状况和处理器性能都还算不错，所以我们完全可以直接就在树莓派上安装 Qt 开发环境，并且利用其原生编译功能去直接搞定程序的开发以及构建工作，就不需要借助交叉编译工具来开展交叉编译操作。这样的方式使得开发流程得以简化，开发效率也得到了提升，同时还充分地发挥了树莓派 5 的高性能优势，有力地确保了开发环境的稳定性以及程序运行时的流畅性，给智能家庭健康监测系统的图形界面开发带来了便利。

4.2.2 Qt 软件程序开发

(1) 软件功能分析

确定控制/显示终端负责的功能，进行 Qt 程序的设计。目前本 Qt 程序拟实现传感器数据的显示、设备的远程控制、健康分析、语音助手和用户管理等功能。传感器数据显示和设备远程控制的实现，终端连接到 MQTT 服务器，订阅 STM32 发布的主题，进行传感器数据的解析，并在 Qt 界面上进行实时显示。设备的远程控制通过 Qt 发布的主题，STM32 进行订阅，就可以实现对设备的远程控制。健康数据分析通过 HTTP 请求上传至百度 AI，AI 返回分析结果后，终端解析并显示在 Qt 界面上，提供健康评估报告和优化建议。语音识别功能实现人机交互。终端通过麦克风录音，并向百度语音识别 API 发送 HTTP 请求，获取识别结果。用户管理功能支持账号注册、登录及健康数据存储，所有用户信息存储于 MySQL 数据库，支持多用户管理。

在将各项功能都确定之后，就要着手去设计主程序的 UI 界面。其中，Home 界面主要承担着实时数据显示以及设备控制方面的职责，Report 界面的功能侧重于 AI 健康分析以及语音互动这块内容，而 User 界面则负责对用户实施管理方面的相关事宜。最终，依照上述所规划好的功能步骤来开展编码以及测试方面的工作，由此达成软件开发的最终完成。

(2) 程序设计流程图

① 登录/注册功能流程如图 4.14 所示。

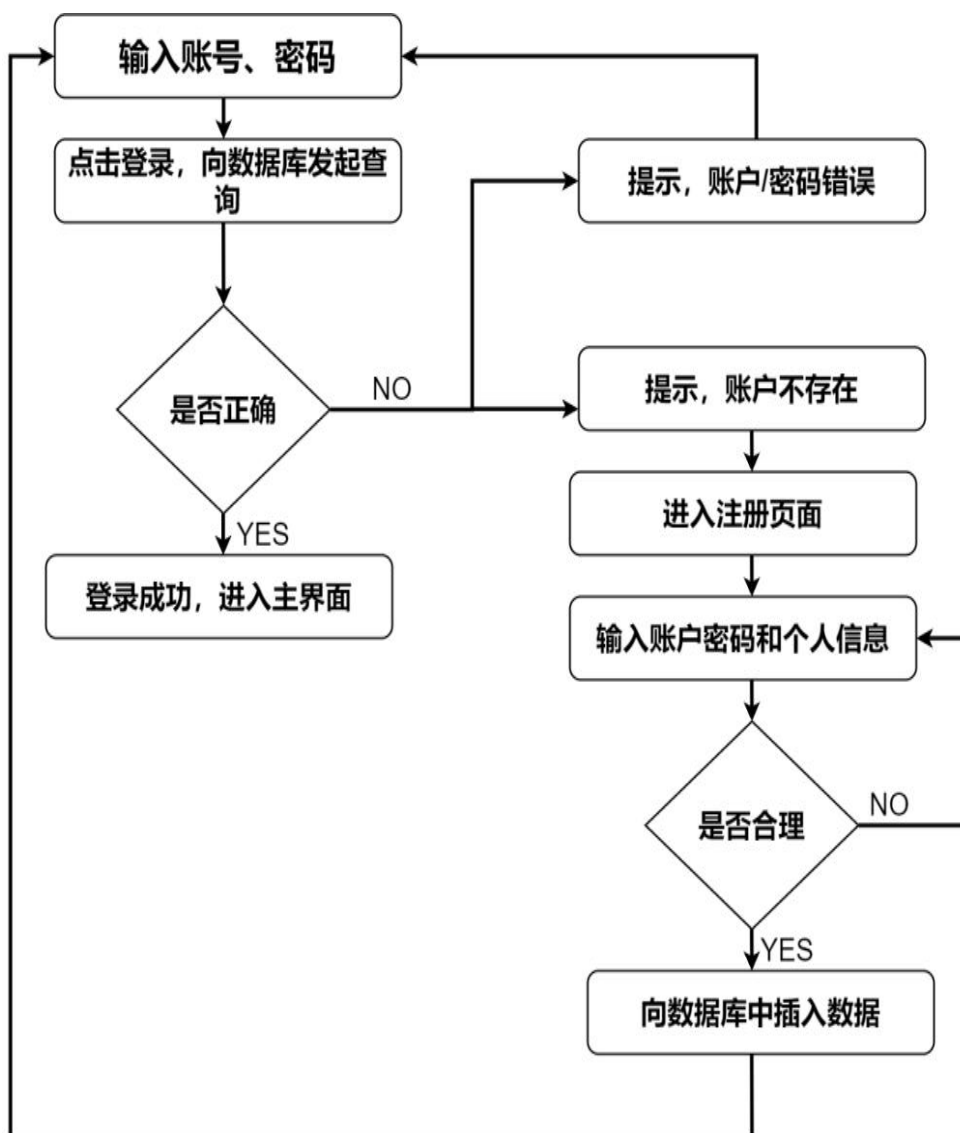


图 4.14 登录/注册界面功能流程图

② 传感器数据显示和设备控制功能如图 4.15 所示。

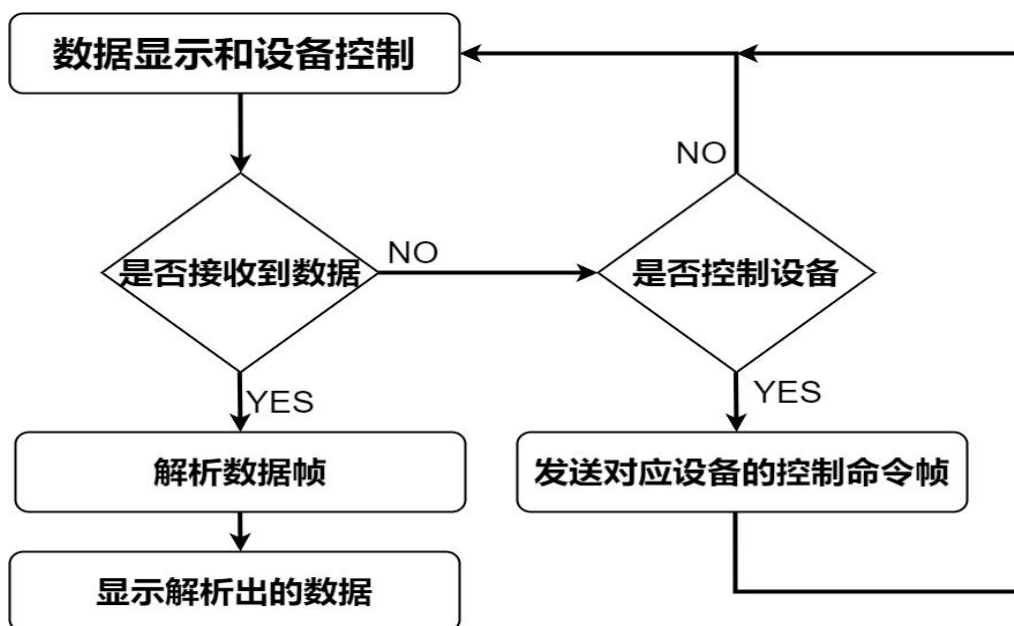


图 4.15 数据显示和控制流程图

③ 语音对话和建议功能流程如图 4.16 所示。

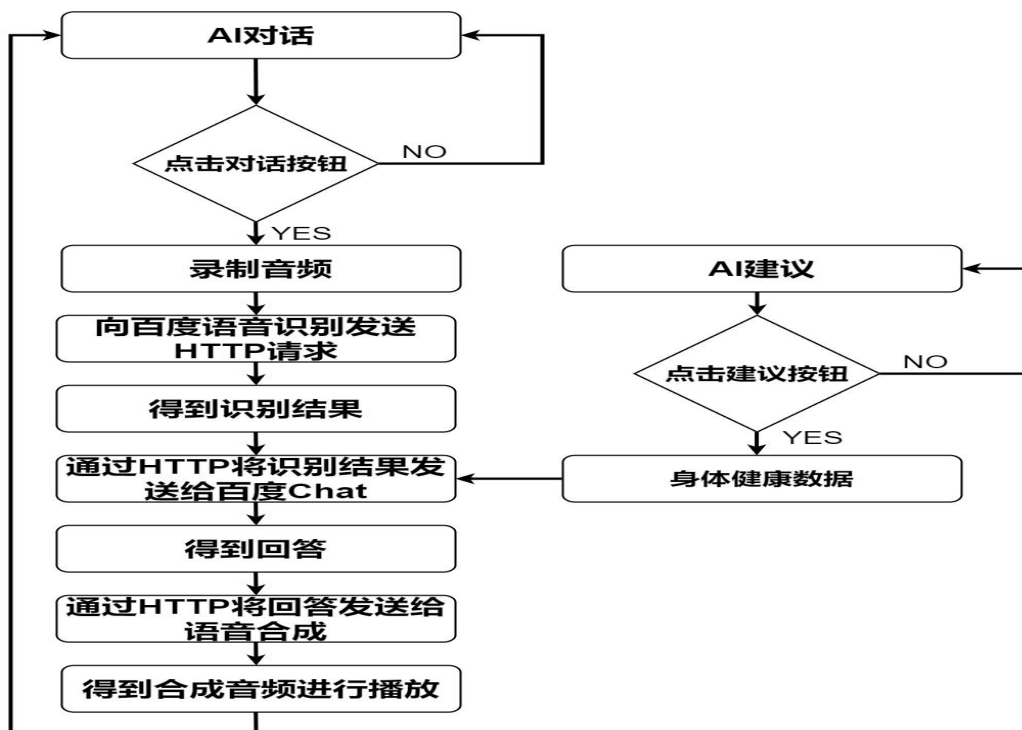


图 4.16 语音对话和建议功能流程图

(3) 程序 UI 界面设计

Qt 界面程序由自主设计实现，具体功能展示如下所示。

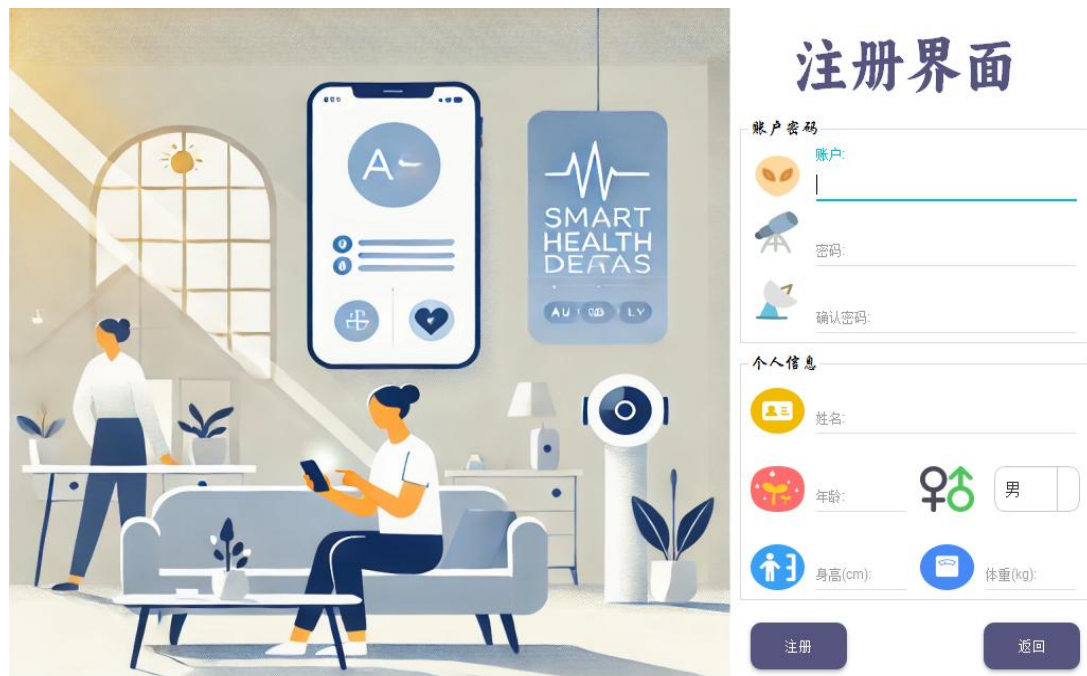


图 4.17 登录界面



图 4.18 注册界面



图 4.19 Home 界面



图 4.20 报告界面

| | username | name | age | sex | height | weight |
|---|----------------|------|-----|-----|--------|--------|
| 1 | 1581908489 | 李二 | 23 | 男 | 172 | 70 |
| 2 | 21213020490120 | 陈玄 | 22 | 男 | 171 | 63 |
| 3 | 21213020490125 | 陈哈哈 | 45 | 男 | 172 | 63 |

图 4.21 User 界面

(4) 部分编码实现

控制/显示终端的核心代码为 MQTT 通信，MySQL 数据库的插入、查询，百度 AI 的 HTTP 请求等，编程语言使用 C++ 进行编程。主要展示该部分：

```

class mqtt : public QWidget
{
    Q_OBJECT
    enum MessageQoS{
        m_QoS0,
        m_QoS1,
        m_QoS2
    };
public:
    explicit mqtt(QWidget *parent = nullptr);
    ~mqtt();
    // 连接 mqtt 服务器
    void mqtt_connect(void);

    void Subscribe(const QString &topic,int qos);//订阅消息
    int Publish(const QString &topic,const QString &message,int qos);//发布消息
    QString get_mqttValue(QJsonDocument JsonDocument, QString Key);/*分离 json 数据
*/

```

```

class health_report : public QWidget
{
    Q_OBJECT
public:
    explicit health_report(QWidget *parent = nullptr);
    ~health_report();
    /*获取百度 chat 和 语音识别的 token*/
    void baidu_http_get_token(void);
public slots:
    /*请求结束 返回结果解析*/
    void http_finished(QNetworkReply *Reply);
    /*将音频上传到百度云识别*/
    void baidu_Audio_Send(void);
    /*将语音识别完成的文字发送给 Chat*/
    void baidu_Chat_Send(QString result);
    /*将百度 Chat 的回答上传 进行语音合成*/
    void baidu_AudioOut_Send(QString text);
    /*暂停录音*/
    void on_autoStopAudio(void);
private slots:
    /*开始对话按钮*/
    void on_pushButton_AiChat_clicked();
    /*停止对话按钮*/
    void on_pushButton_stopChat_clicked();
    /*健康建议按钮*/
    void on_pushButton_AiReport_clicked();
private:
    Ui::health_report *ui;
    // get 请求返回的 token
    QString Audio_access_token = "";
    QString Chat_access_token = "";
    QNetworkAccessManager* baidu_Audio = nullptr;
    QNetworkAccessManager* baidu_Chat = nullptr;
    // 音频格式
    QAudioFormat Audio_Format;
    QAudioInput *Audio_in = nullptr;
    // 音频文件处理
    QFile destinationFile;
    QFile *sendFile;

```


4.3 智能家庭健康监测系统服务器搭建

本系统所使用的服务器是由作者购买的阿里云服务器，在该服务器上部署了 MQTT 服务和 MySQL 数据库服务。下面将对各个部分做详细说明。

4.3.1 MQTT 服务器的搭建

(1) MQTT 简介

MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输协议) 是一种采用发布/订阅 (publish/subscribe) 模式的轻量级通信协议，基于 TCP/IP 协议构建。由 IBM 于 1999 年发布，专门针对物联网设备开发，是一种低开销、低带宽占用的即时通信协议。它的设计思想简单、开放、规范，易于实现，这些特点非常适合物联网等通信场景。

MQTT 服务器通常被称为 MQTT Broker，本文选择使用 Mosquitto 版本，由 Eclipse 开源社区开发，其功能足以应对大多数使用场景。接下来，我们将在服务器上部署一个属于自己的 MQTT 服务器和 MySQL 数据库。

(2) 部署步骤

步骤一：登录阿里云服务器后输入指令安装 Mosquitto MQTT 服务器。

步骤二：安装完成后，启动 Mosquitto 服务，启动后系统将运行 Mosquitto 的 MQTT Broker。

步骤三：然后设置 Mosquitto 的账户名称和密码。重新启动 Mosquitto 服务器。

(3) MySQL 数据库部署

对于 MySQL 服务的安装，可以直接在服务器上通过 Linux 命令进行部署。安装步骤如下：

步骤一：运行 `sudo apt-get update` 命令更新系统软件包列表。

步骤二：通过 `sudo apt-get install mysql-server` 命令安装 MySQL 服务。

步骤三：执行 `sudo mysql_secure_installation` 命令进行初始安全配置。

步骤四：定位到 MySQL 安装目录下的配置文件 `mysqld.conf`，调整相关参数以支持远程连接访问。

步骤五：从远程主机尝试连接，验证连接是否成功。

如果远程主机能够顺利访问服务器上的数据库，则说明配置已完成。因为本系统的数据库比较简单，只有用户表存储家庭成员的信息，所以就不进行赘述。

5 智能家庭健康监测系统测试

5.1 硬件测试

在硬件测试环节，要检测供电状况是不是正常，就是要查看是否有稳定的 5 V 电压输出以及是否存在稳定的 3.3V 电压输出等；同时，也要看晶振是否正常的产生振荡；除此之外，还要对电路板上是不是存在虚焊等焊接方面的质量问题加以检查，以此来保障硬件具备应有的可靠性与稳定性。实物电路板可参照图 5.1 所示内容。

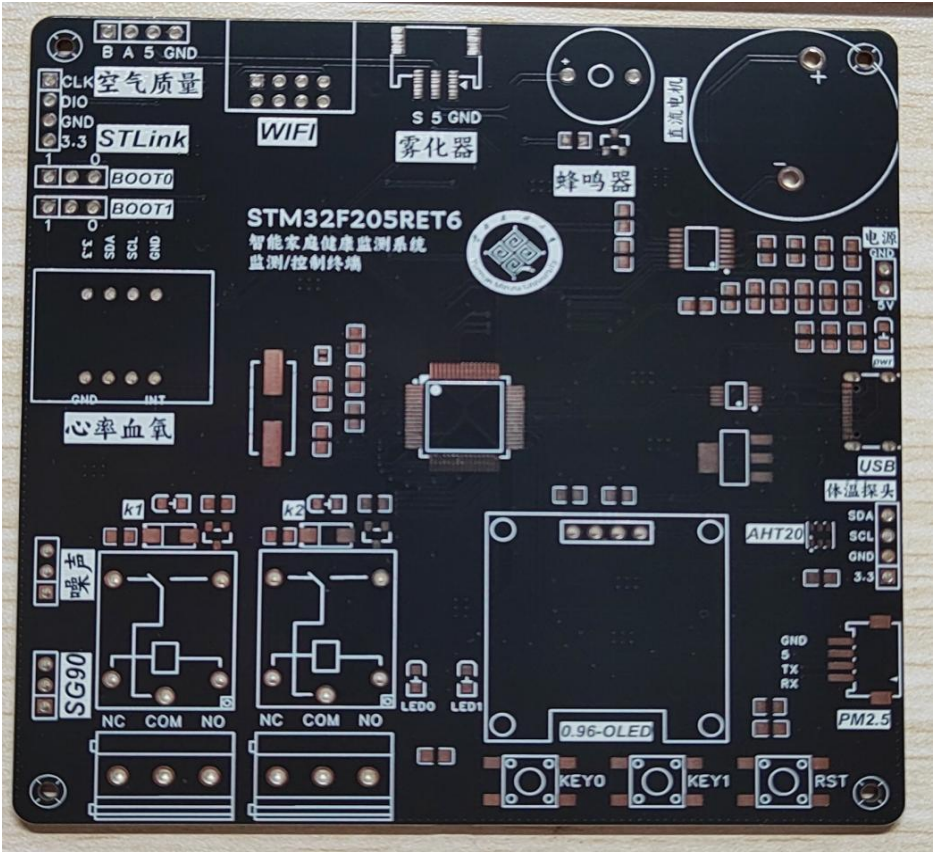


图 5.1 电路板图

- (1) 硬件电路焊接完成后进行电压测试，确保整个系统的供电正常。如图 5.2 和图 5.3 所示。

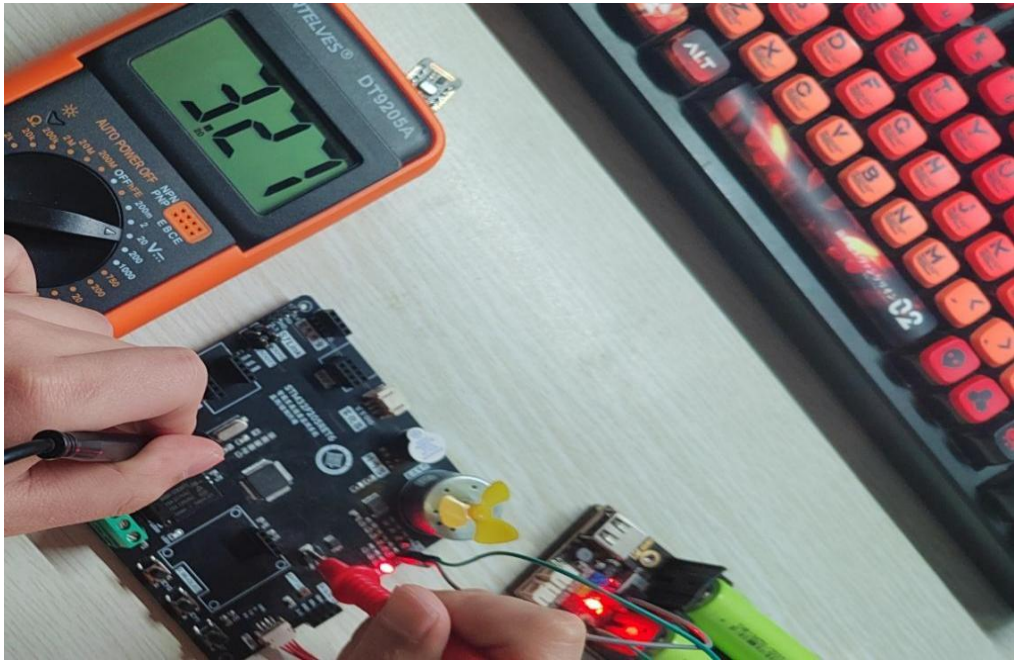


图 5.2 3.3V 电压测试

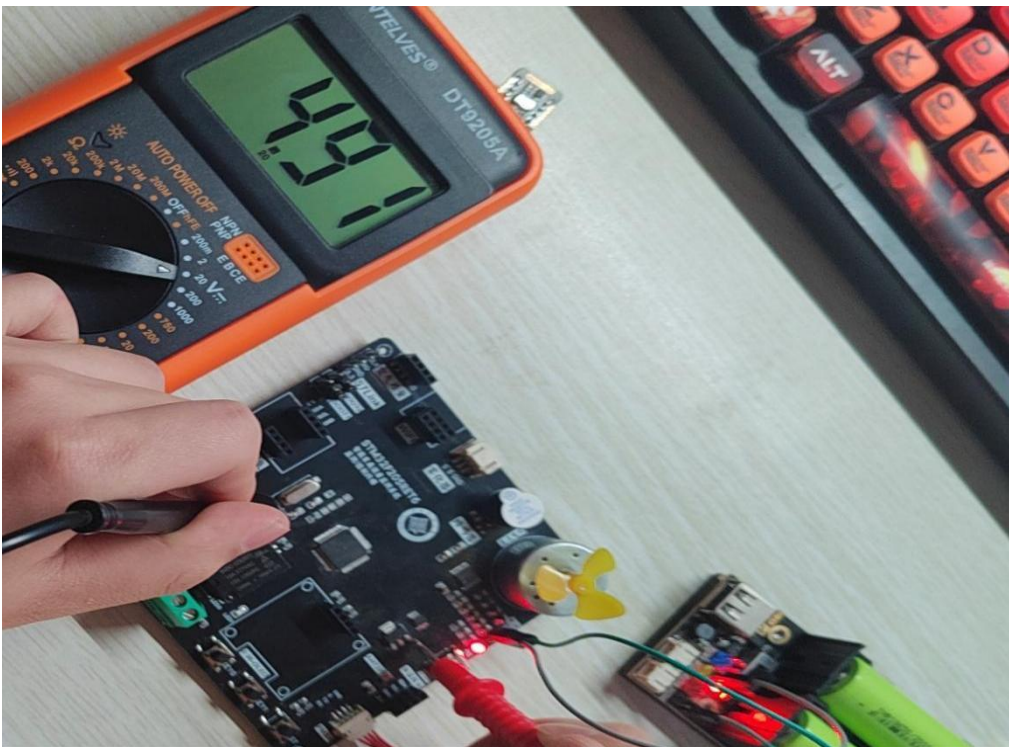


图 5.3 5V 电压测试

- (2) 本系统采用的是外部 8MHz 的晶振，要确保外部晶振启振，验证如图 5.4 所示为 8MHz。

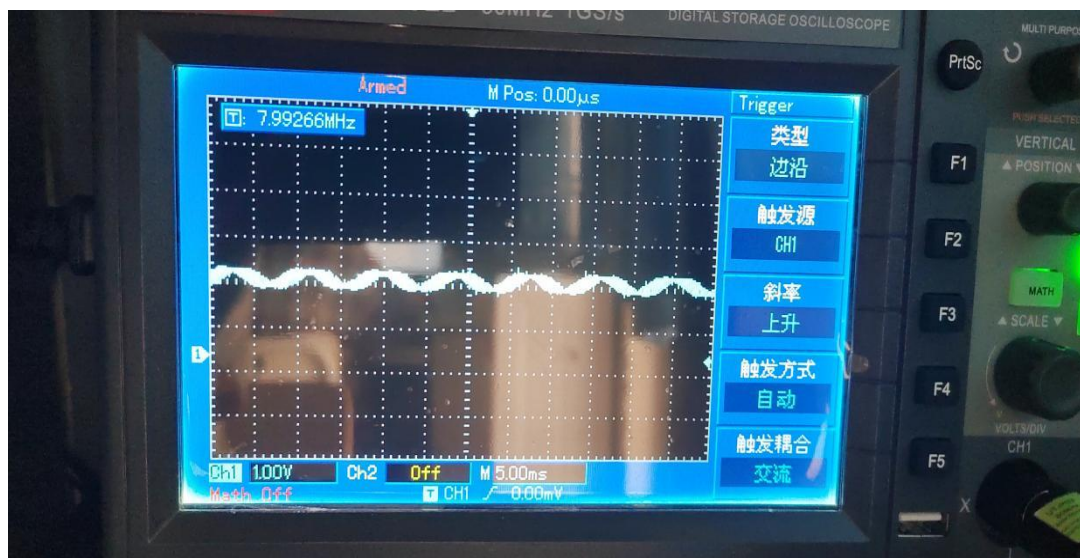


图 5.4 外部晶振测试

5.2 软件编译

5.2.1 单片机程序编译

在单片机程序开发工作当中，使用的是 KEIL5 以及 STM32CubeMX 这两款工具。其中，STM32CubeMX 主要承担起生成初始化代码框架的任务，至于具体的 STM32 程序编写工作，则是以 HAL 库作为基础来开展的，如此一来，开发的流程能够得到简化，并且代码在移植的时候也会变得更为轻松容易。当程序编写完成之后，就需要利用 KEIL5 自身所带的集成开发环境来开展编译工作了，通过编译可以对代码的语法以及逻辑方面是否正确进行细致检查。在编译顺利通过之后，再借助 STLink 调试工具把程序下载到单片机里面，以此来保证程序可以准确无误地加载起来并且正常运行。

5.2.2 Qt 程序的编译

Qt 程序的开发起初是在 Windows 系统当中开展的，借助 Qt Creator 集成开发环境来编写代码。处于 Windows 环境之下时，我凭借它所具备的便捷调试工具以及图形化界面，针对程序的各项功能展开测试与调整方面的工作，从而保

证代码的逻辑是正确无误的，界面能够实现流畅运行，而且各项功能均可以正常运转起来。在测试顺利通过之后，要把完整的源代码使用文件传输工具传送到树莓派 5 设备之上。在树莓派 5 的 Linux 环境里，首先得安装那些必不可少的 Qt 开发库以及编译工具，接着针对源码做出适配性方面的调整，随后再运用命令行或者相关工具来进行编译操作，进而生成能够在树莓派上运行的可执行文件。在编译完成之后，便可以直接在树莓派 5 上运行该程序了，对其在目标平台上所呈现出的性能以及稳定性加以测试，以此确保程序可以正常开展工作。本系统从开发一直到部署的过程，将 Windows 的开发便捷特性与树莓派的硬件优势充分地结合在了一起。

5.3 服务器搭建测试

此系统选用阿里云轻量应用服务器来充当服务器平台，是经由阿里云官方网站去购买然后激活的。借助 SSH 协议来远程登录这一服务器，进而完成 Mosquitto 服务器以及 MySQL 数据库的安装与配置相关事宜。对于 MQTT 服务器的连接测试工作，是利用 MQTTfx 来完成的；而 MySQL 数据库的连接测试，则是通过 Navicat 来实现的。

- (1) 通过服务器向 MQTTfx 发送“helloworld”消息，以测试服务器是否正常启动并能够成功接收和发送数据。

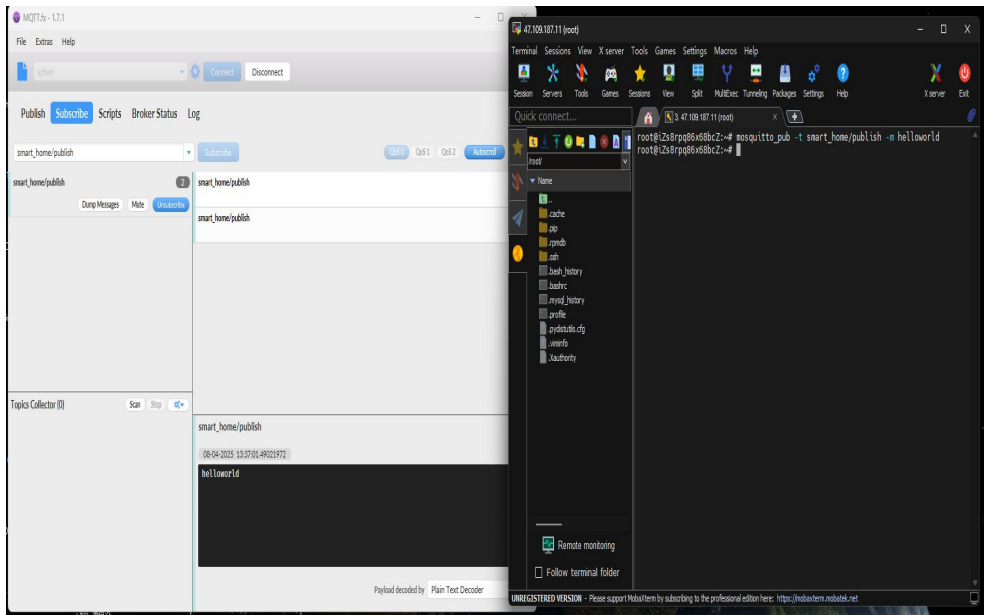


图 5.5 MQTT 服务连接测试

(2) 服务器部署好 MySQL 数据库之后，就可以通过服务器的 IP 地址和 3306 端口对数据库进行操作。使用 Navicat 软件测试数据库是否正常启动。

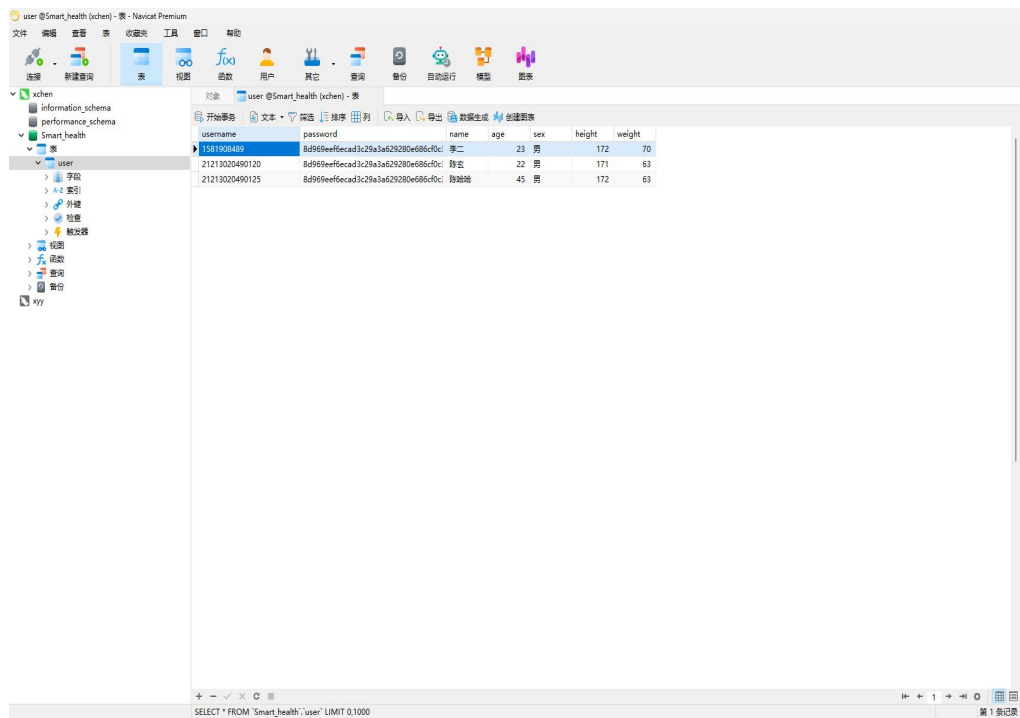


图 5.6 MySQL 数据库连接测试

至此，服务器搭建的功能已经全部完成，可以支持数据的正常通信功能。MQTT 服务器和 MySQL 数据库的部署经过测试验证，确认了数据的传输和存储的稳定性。

5.4 系统功能测试

5.4.1 系统通信功能测试

(1) 监测/采集终端数据上传测试

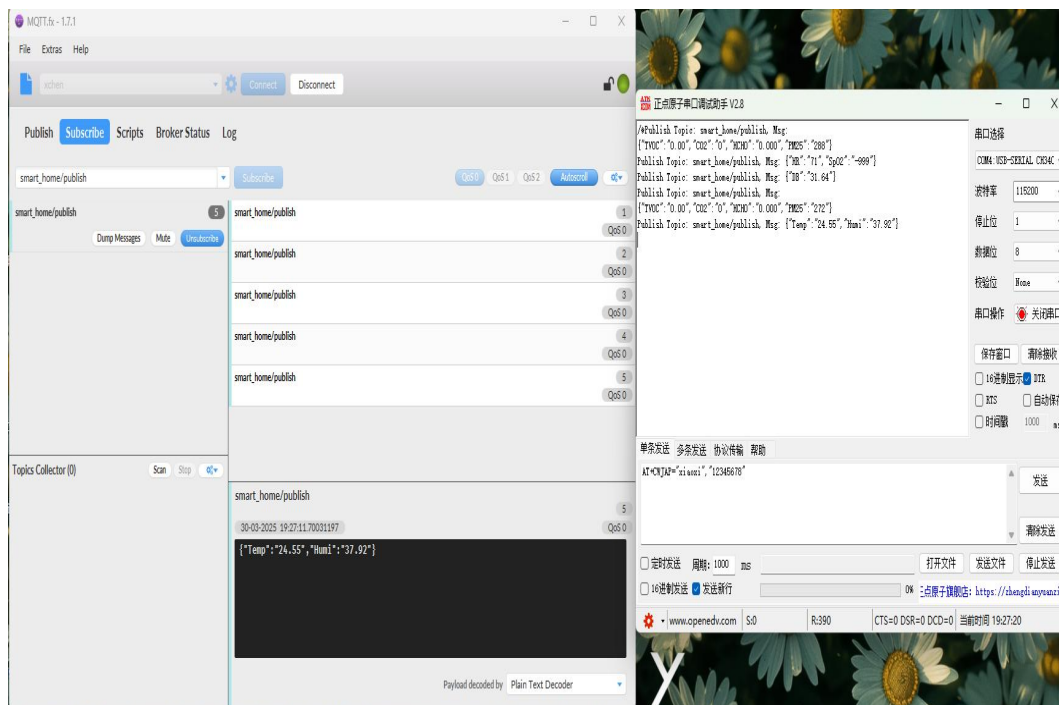


图 5.7 数据上传测试

(2) 控制/显示终端指令下发测试

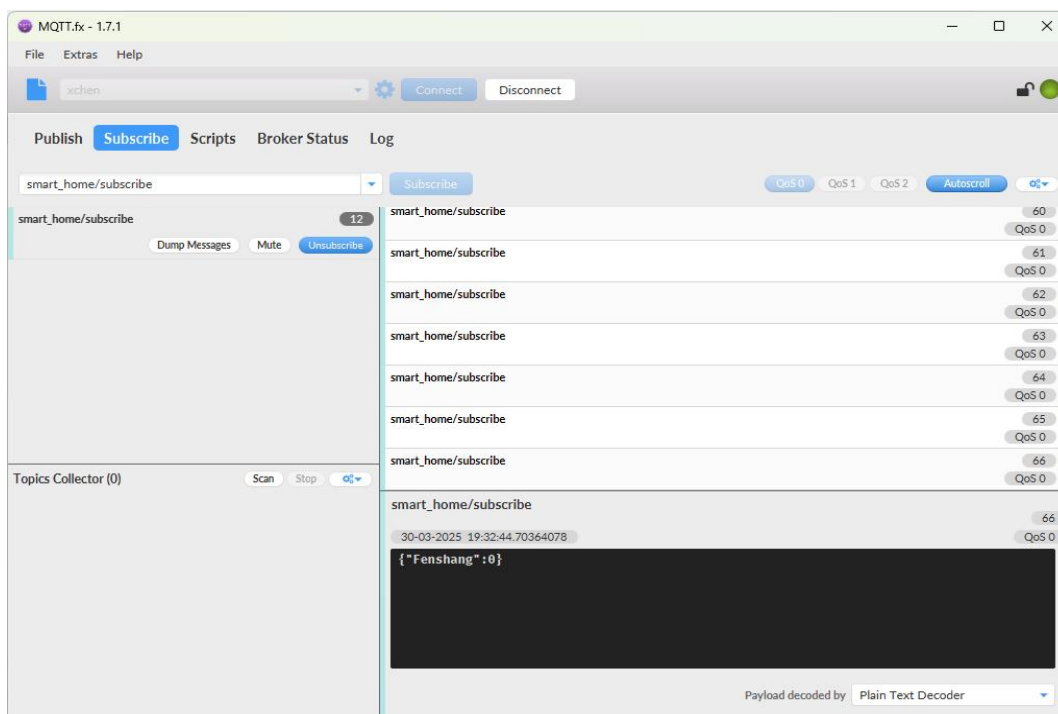


图 5.8 指令下发测试

(3) 百度智能云 HTTP 功能测试

Qt 程序的运行需首先获取百度智能云的语音识别 Token 以及文心一言的 Token，只有成功获取这些 Token 后，才能正常调用百度智能云的相关功能。

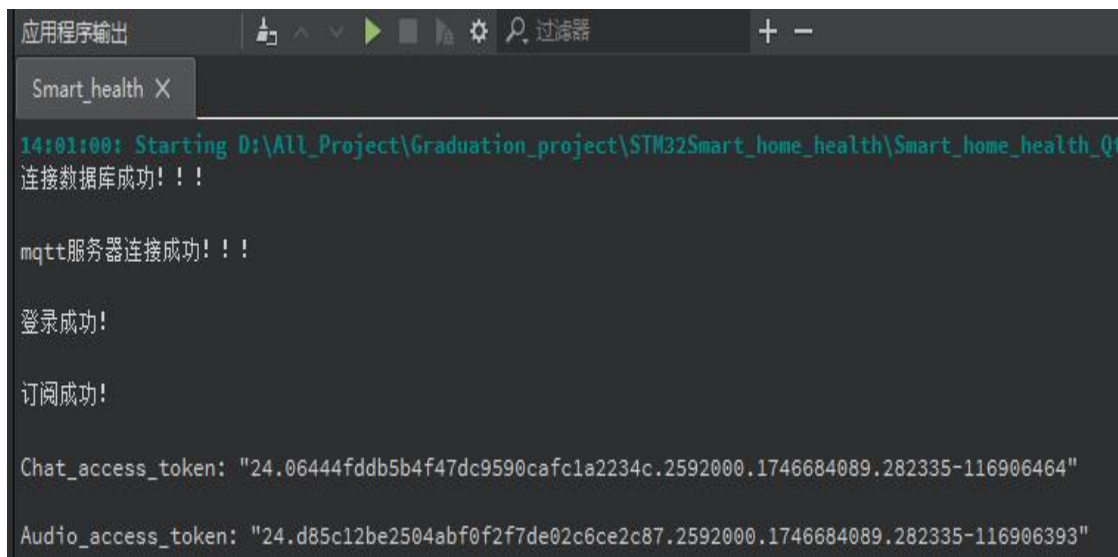


图 5.9 百度智能云功能测试

(4) 系统总体测试

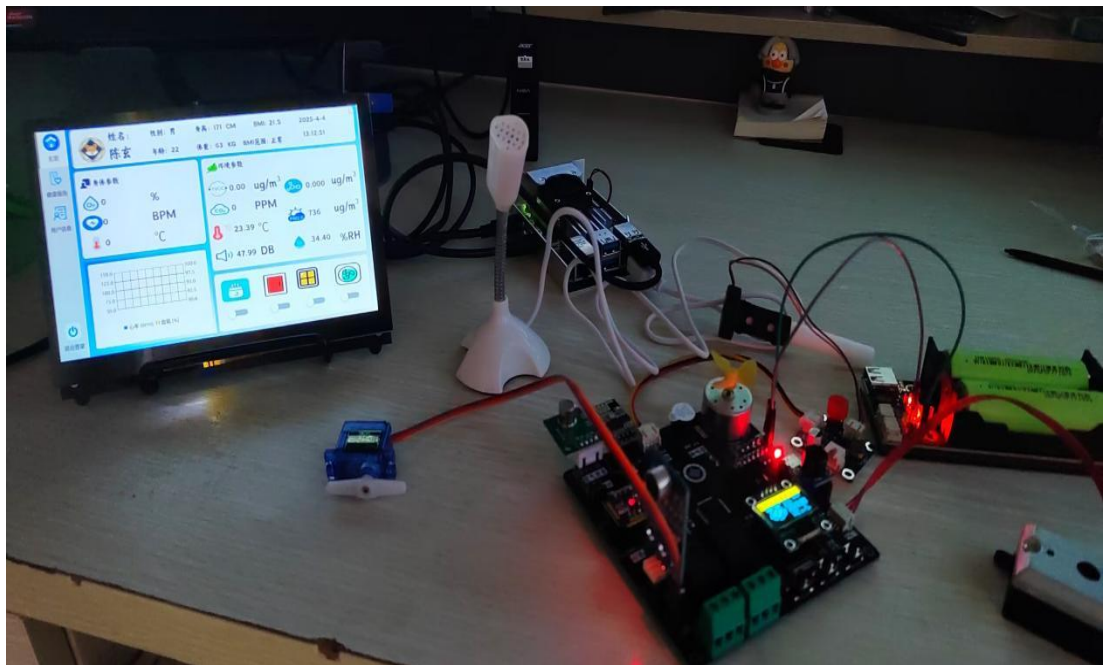


图 5.10 系统总体测试

5.4.2 功能测试总结

在针对监测及采集终端数据所开展的上传测试环节当中，借助图 5.7 清晰地呈现出了整个测试的具体过程以及最终所取得的结果，有力地证实了终端设备确实能够把采集而来的数据上传至相关系统之中，从而保障了数据传输的准确性和可靠性。在涉及控制以及显示终端指令下发的测试工作里，就如同图 5.8 所展示的那样，针对指令从控制端下发一直到显示终端的实际执行状况进行了验证，进而明确了指令不但被准确无误地接收了，而且还得以顺利地执行下去，如此一来，通信链路所具备的响应性也得到了有效的验证。在针对百度智能云展开的连接测试过程中，正如图 5.9 所表明的那样，充分说明了系统是能够正常获取百度 Token 的，这也就意味着系统能够以正常的状态和百度智能云实现通信交互。在针对系统开展的总体测试环节，就像图 5.10 呈现的一样，系统的界面已然成功地移植到了树莓派的 7 英寸显示屏之上，并且还能够正常接收到由 STM32 所发送的数据信息。

至此，基于树莓派的智能家庭健康监测系统的测试全部完成，所有的功能都已经达到了预期的效果。

6 总结与展望

6.1 研究总结

至此，基于树莓派的智能家庭健康监测系统的研究任务已顺利完成，在研究设计的过程中克服了多项困难与挑战，最终成功地开发出了功能比较完善的硬件与软件系统，总体基本达到了预期的目标。总结如下：

(1) 深入剖析“智能家庭健康监测”概念并提出个人见解

此项研究先是去深入了解国内外学者有关智能家庭健康监测系统的诸多理论，同时也对相关的实践案例加以探究，并且还结合当下老龄化社会已然形成以及健康管理需求日益凸显的这样一种大背景，进而形成了针对该系统的一些看法。智能家庭健康监测并非仅仅是对生理数据做简单的采集，它应当成为一个能够把环境感知、智能反馈以及用户交互等多方面融合起来的生态系统。在本文当中着重强调的是，该系统务必要以家庭作为核心所在，要能够为用户给予预防性的健康管理方面的服务，而绝不能仅仅停留在被动监测这样一个层面之上。

(2) 基于实际需求设计系统功能与架构并完成设备选型

在弄清楚家庭成员针对健康监测所存在的各类不同需求状况之后，本项研究进一步明确了系统需要具备的核心功能所在，这里面涵盖了多参数的健康监测方面的功能、环境数据监测功能以及由 AI 所给出的有关健康方面的建议内容。基于此，我专门设计出了一套系统架构，该架构是由感知层、网络层以及应用层共同组合而成的。其中，使用 STM32F205RET6 微控制器来负责完成高效的数据采集工作以及实时的控制任务，把树莓派 5 当作是和用户进行交互活动的核心枢纽，并且搭配上各种各样的传感器以及执行器。与此同时，再与 FreeRTOS、Qt、MQTT 和 MySQL 等一系列的技术相互结合起来，如此这般，便完成了对于软硬件的选型工作，从而切实确保整个系统在运行的时候既能够保持较高的效率，又可以比较容易地实现扩展功能。

(3) 搭建开发环境并实现系统全流程开发

本研究在树莓派 5 上搭建了嵌入式 Linux 开发环境，使用 Ubuntu 24.10 支持 Qt 图形界面的运行；在 STM32 端移植 FreeRTOS，开发了多任务并发的程序。

同时，通过阿里云服务器部署 MQTT 通信服务和 MySQL 数据库，完成了从数据采集到云端管理的全过程。系统的开发注重模块化设计与软硬件协同，成功完成了一个功能完整的智能家庭健康监测平台。

(4) 系统测试与性能优化

在一步步的测试环节里，本系统切实确认了硬件电路供电方面的稳定性状况，也明确了传感器采集数据所具备的准确性程度。与此同时，还针对软件功能持续不断地展开调试工作，并且对其进行完善处理，由此成功解决了在开发进程当中所察觉到的若干潜在问题。测试所获结果清晰地显示出，系统在对健康数据予以监测时所呈现出的实时性特点、在进行设备控制时所展现出来的反应速度情况，以及在用户界面展开操作时的实际表现，这几个方面均有着较为不错的呈现。经过一回又一回的调整举措与改进措施的施行，系统自身的稳定性已然得到了颇为显著的提升。

6.2 研究展望

虽然本系统在功能实现和性能验证方面取得了较好的成果，但系统还有一些局限性，未来的研究和开发可以从以下几个方向继续地进行改进：

(1) 功能的扩展与精准化

目前系统着重对心率、血氧、体温以及部分环境参数展开监测。在未来的发展进程中，其能够进一步整合诸多其他健康指标，像是血压、血糖以及睡眠状况的监测等，以此来契合不同人群的各类需求。与此同时，借助对传感器算法以及数据处理技术加以改良，能够促使传感器的监控精准程度得到更进一步的提升，并且还能够在用户进行操作之时，尽可能地降低对环境或者数据所产生的干扰情况。

(2) 硬件的小型化与能效优化

目前系统的硬件设计集成度已经不错，但系统的体积和功耗还可以再进行优化。未来还可以使用更先进的芯片技术和低功耗设计，开发更加小型化、更加便携的设备，提高系统的易用性和在家庭部署中的便携性。

(3) 智能化与本地化处理能力提升

目前系统是依靠百度云 AI 来开展健康分析相关工作的。在未来的优化当中，可以部署本地大模型，将一部分的数据处理以及分析方面的任务安排在本地上来

完成。如此一来，既能减少对网络的依靠程度，又能够让数据传输过程中所产生的延迟得以降低，与此同时，还可以充分保障数据的安全性。并且，通过与深度学习模型相互结合起来，依据用户长期积累下来的数据，针对健康趋势展开预测工作，同时对相关风险进行评估，进而促使智能化的水平获得有效提升。

(4) 安全性和隐私保护的强化

系统借助 MQTT 以及 HTTP 协议来传输数据，其通信方面有着较好的稳定性，不过在安全性上尚存在不足。往后可以增添端到端加密、用户认证以及数据匿名化这些措施，以此来保障用户健康数据的隐私安全，使得系统于实际运用当中能够更具可信度。

(5) 应用场景的多元化

现在主要是用来监测家庭人员的健康，后面可以延伸到养老院、社区医疗或者是远程医疗等。与医疗系统联网，可以提供数据共享、远程支持。还可以开发手机 APP，用户可以随时查看自己的身体情况。

参考文献

- [1] 耿新,邢鹏飞,王立群.人口老龄化视域下的适老化居家体系设计研究[J].设计,2024,37(19): 90-93.DOI:10.20055/j.cnki.1003-0069.002182.
- [2] 刘浩源,张慧颖.基于物联网技术的人体健康数据检测系统设计[J].吉林化工学报,2023,40(03):42-47.DOI:10.16039/j.cnki.cn22-1249.2023.03.009.
- [3] 梁峻阁,宋怡然,孙杨帆,等.基于可穿戴与可植入技术的人体健康物联网研究进展[J].物联网学报,2023,7(02):26-34.
- [4] 韩改宁,李永锋,高伊腾.基于嵌入式 Qt 下的 MySQL 数据库设计与开发[J].微型电脑应用,2020,36(05):25-27.
- [5] 汤晓.健康中国背景下家庭医生服务能力建设研究[D].上海工程技术大学,2020.DOI:10.27715/d.cnki.gshgj.2020.000094.
- [6] 顾彦.《“健康中国 2030”规划纲要》发布看健康中国如何“三步走”?[J].中国战略新兴产业,2017,(21):80-81.DOI:10.19474/j.cnki.10-1156/f.001171.
- [7] 牛帅,张弛,董振华,等.WM8978 音频处理器的百度 AI 语音识别[J].单片机与嵌入式系统应用,2023,23(02):42-44.
- [8] 凌信航,王航蜀,吴俊.基于物联网的人体健康及家庭环境监测系统[J].电子器件,2022,45(05):1272-1278.
- [9] 尚宪茹,刘子建,胡志刚,等.国内外老年智能健康穿戴产品研究态势分析[J].丝绸,2025,62(02):82-93.
- [10] 赵有明,汪涛,张庆哲,等.基于互联网和传感器的家庭健康监护系统[J].中国科技论文在线精品论文,2020,13(04):491-496.
- [11] 凌信航,王航蜀,吴俊.基于物联网的人体健康及家庭环境监测系统[J].电子器件,2022,45(05):1272-1278.

致谢

时光荏苒，四年的大学生活转瞬即逝，完成这篇毕业论文的过程让我收获颇多，也让我深刻体会到自己并非孤身一人前行。在论文的撰写和系统的设计开发过程中，我得到了许多人的帮助和支持。在这里请接受我诚挚地致谢。

首先，我要衷心感谢我的指导教师王武副教授。从论文选题到系统设计，再到最后的修改完善，王老师始终给予我耐心地指导和专业的建议，同时还要感谢数学与计算机科学学院的各位老师。无论是课堂上的悉心教导，还是课后的答疑解惑，你们的教诲为我打下了扎实的专业基础，使我能够在物联网工程领域中不断探索和实践。此外，还要感谢我身边所有的朋友和同学，你们是我大学生活中最重要的组成部分。在四年的大学生活里，不管是生活还是学习，你们都给了我很多的支持和帮助。

我还要感谢我的家人。无论我身处何地，遇到怎样的困难，家人始终是我最坚实的后盾。你们的理解和支持让我能够心无旁骛地投入学习中。

未来还有很长的路要走，但这段经历让我更加坚定了自己的方向。再次向所有帮助过我的人表示由衷的感谢！

陈玄

2025 年 3 月