智慧农场控制室智能网关管理软件设计与实现

**专业：软件工程**

**学生：张毅 指导老师：庄建**

摘要：使用智能技术的精准农业正在迅速发展，以改变传统方法，实现智能管理农场各种设备，既降低人工面临的各种困难和危险，又提高了农业生产效率。

论文主要研究基于嵌入式设备上开发一款管理农场控制室的各种智能设备的管理软件，软件开发基于Qt开发工具，C++开发语言，MQTT协议实现远程控制农场控制室的相关设备，农场设备收发数据借助Home Assistant这款国外智能家居开源平台，其主要特点是能将不同厂商的智能设备连接在统一平台，进行统一管理，很大程度方便了用户的使用。

在本次项目中，选择使用的硬件设备是树莓派，树莓派以其体积小，能耗低，计算快的特点广泛应用于物联网的各种场景。Qt具有多平台兼容的特点，本次项目选择的开发工具版本是Qt5。使用管理软件可远程接收智能设备的数据和状态，并对一些设备进行相关操作。此次项目也是将智能家居应用到智慧农场的一次小小挑战。

关键词： Qt；树莓派；Home Assistant；MQTT

**Design and Implementation of Intelligent Gateway Management Software for Intelligent Farm** **Control Room**

**Major: Software Engineering**

**Student: Zhang Yi Supervisor: Zhuang Jian**

**Abstract:** Precision agriculture using intelligent technology is developing rapidly to change traditional methods and achieve intelligent management of farm equipment, which not only reduces the difficulties and risks faced by human beings, but also improves agricultural production efficiency.

This paper mainly studies the development of a management farm control room based on embedded devices of various intelligent equipment management software, software development based on the Qt development tool, C++ development language, MQTT protocol to achieve remote control of farm control room related equipment, the farm equipment to send and receive data with the help of a foreign intelligence lives in the open source platform Home Assistant, Its main feature is that intelligent devices from different manufacturers can be connected to a unified platform for unified management, which greatly facilitates the use of users.

In this project, raspberry PI is the hardware device selected. Raspberry PI is widely used in various scenarios of the Internet of Things due to its small size, low energy consumption and fast computing. Qt has the characteristics of multi-platform compatibility, the development tool version of this project is Qt5. You can use the management software to remotely receive the data and status of intelligent devices and perform related operations on some devices. This project is a small attempt to apply smart home to smart farms.

**Key Words:** Qt; Raspberry Pie; Home Assistant; MQTT

目 录

[1 绪论 1](#_Toc99461291)

[1.1 研究背景和意义 1](#_Toc99461292)

[1.2 研究方向与现状 1](#_Toc99461293)

[1.3 本文的章节安排 2](#_Toc99461294)

[2 系统结构及功能概述 3](#_Toc99461295)

[2.1 系统结构 3](#_Toc99461296)

[2.2 功能概述 3](#_Toc99461297)

[3 项目涉及的硬件设备和软件技术 5](#_Toc99461298)

[3.1 树莓派介绍 5](#_Toc99461299)

[3.1.1 树莓派硬件结构介绍 5](#_Toc99461300)

[3.2 传感器及设备介绍 6](#_Toc99461301)

[3.2.1 传感器和智能设备 6](#_Toc99461302)

[3.2.2 智能网关 7](#_Toc99461303)

[3.3 Qt/MQTT 7](#_Toc99461304)

[3.3.1 Qt开发工具介绍 7](#_Toc99461305)

[3.3.2 MQTT通信协议 8](#_Toc99461306)

[3.4 Home Assistant平台 9](#_Toc99461307)

[4 环境搭建 10](#_Toc99461308)

[4.1 树莓派开发环境 10](#_Toc99461309)

[4.1.1 系统安装 10](#_Toc99461310)

[4.1.2 安装Qt CREATOR及MQTT库 12](#_Toc99461311)

[5 管理软件的系统设计 13](#_Toc99461312)

[5.1 软件结构 13](#_Toc99461313)

[5.2 界面设计 13](#_Toc99461314)

[5.2.1 控制室设备监测界面 13](#_Toc99461315)

[5.2.2 控制室设备管理界面 15](#_Toc99461316)

[5.2.3 实验室设备管理界面 15](#_Toc99461317)

[5.2.4 操作日志界面 16](#_Toc99461318)

[5.3 消息结构设计 17](#_Toc99461319)

[5.3.1 消息订阅 17](#_Toc99461320)

[5.3.2 消息发布 18](#_Toc99461321)

[5.4 传感器及设备结构化数据设计 18](#_Toc99461322)

[5.4.1 传感器及设备数据描述 18](#_Toc99461323)

[5.4.2 订阅数据结构 19](#_Toc99461324)

[5.4.3 发布数据结构 21](#_Toc99461325)

[6 主要功能实现 24](#_Toc99461326)

[6.1 连接MQTT服务器 24](#_Toc99461327)

[6.2 订阅数据 25](#_Toc99461328)

[6.3 数据解析 25](#_Toc99461329)

[6.4 数据显示 26](#_Toc99461330)

[6.5 发布数据 27](#_Toc99461331)

[6.6 操作日志 29](#_Toc99461332)

[6.6.1 显示操作记录 29](#_Toc99461333)

[6.6.2 查找操作记录 31](#_Toc99461334)

[6.6.3 清空操作记录 33](#_Toc99461335)

[7 测试结果 35](#_Toc99461336)

[7.1 树莓派测试 35](#_Toc99461337)

[结论 42](#_Toc99461338)

[参考文献 43](#_Toc99461339)

[致谢 44](#_Toc99461340)

1 绪论

1.1 研究背景和意义

智慧农业的智能化主要体现在一个智慧农业管理系统上，用来监测农场各类传感器数据从而制定解决方案去控制相应智能设备完成农场的自动化管理。

在社会上，智慧农业已成农业成长历程中的必要趋向。比如，农场管理员通过智慧农业系统可以远程查看农场的天气、土壤、温湿度、虫害等实时信息来进行相应的操作，确保农场农作物在一个舒适的环境中生长，整个过程不需要农场工作人员去现场进行操作，避免了现场工作可能带来的危险，同时利用智能化的决策去应对自然环境风险很明显提高了农业生产的效率，使其传统的农业生产模式转变为高效率的自动化生产模式。

在经济上，智慧农业也是智慧经济发展中的主要组成部分。对[发展中国家](https://baike.baidu.com/item/%E5%8F%91%E5%B1%95%E4%B8%AD%E5%9B%BD%E5%AE%B6/652451)来说，智慧农业便成了智慧经济发展中的核心部分，是发展中国家实现国家科技水平提高、解决经济落后、赶上一些发达国家的经济发展水平的重要途径。

从科学上讲，我国的农业生产位于世界前列，但对于我国目前的农业技术仍需继续发展以达到农业生产强国的生产水平。我国的农业依然以传统出产模式为主，在浪费大量的劳动力和物质材料的基础上，还造成了水土环境的破坏和人身安全的诸多隐患。为了降低这些风险，引入自动化设备成为必要，在物联网基础上，结合人工智能、大数据等相关技术去提高农业生产效率，降低生产风险，让智慧农业更加智能。

1.2 研究方向与现状

随着近几年来科技水平和生活水平不断地提高，大多数的人对生活上的要求不再是吃饱穿暖，而是有更高的娱乐上和精神上的满足。按照马斯洛需求理论层次，人如果满足了基本的生活需求，就会开始去追求自我价值的实现[1]。

将物联网技术应用到智慧农业中，确实能够大大地提升农作物的生产质量，进而促进了我国农业现代化的发展。随着近几年我国物联网技术的普遍应用，以及我国农业生产管理模式逐步完善，农业信息技术的应用也就越来越广泛。目前智能决策技术的提升和智能终端设备的不断推广应用，智能农业必将实现新的发展，进而促进了我国农业生产效率和生产水平的提高，间接加速了我国农业经济的发展[2]。

在国内，目前我国的农业生产依然存在生产模式陈旧、生产价值链低、劳动力短缺、农业从业者收入增长偏低等大量问题。为了推进我国农业生产力的进步和提高农业生产的效率，在2018到2019年期间，国家对于农业建设下发了相关政策，国内大量的企业开始对智慧农业进行投资布局，去加快我国农业的智能化建设，比如，蒙牛乳业实现从“草原牛”到“数字牛”、通威股份“渔光一体”、美团“AI食物农场”和阿里云“智慧养猪”等。

在国外，大多数国家也在农业建设投入了不少精力，纷纷将自动化融入农业生产中。目前农业生产走在前面的一些国家有日本、荷兰、德国等，其主要都是将信息化技术、生物技术应用到农业生产中，从而实现农业信息处理、智能控制、精准决策的新一代的农业生产模式。例如，英国对于农业研发的“除草机器人”，可以在100英亩的土地上自动完成从播种到收成的整个过程。

本文主要是对在树莓派上开发智能管理软件的环境配置和相关功能上进行说明。根据相关设备完成对应的显示和控制，对嵌入式平台的开发进行了进一步的深入学习。智能家居与智慧农业本就如出一辙，本次项目将智能家居结合到智慧农业，使用嵌入式设备完成计算机的相关工作，实现客户端远程控制智慧农场中的智能设备，目前智慧农业处于发展中，但万物互联，智能产业是必然趋势，此项目也是解决智慧农业智能化管理的一次小小的研发和应用。

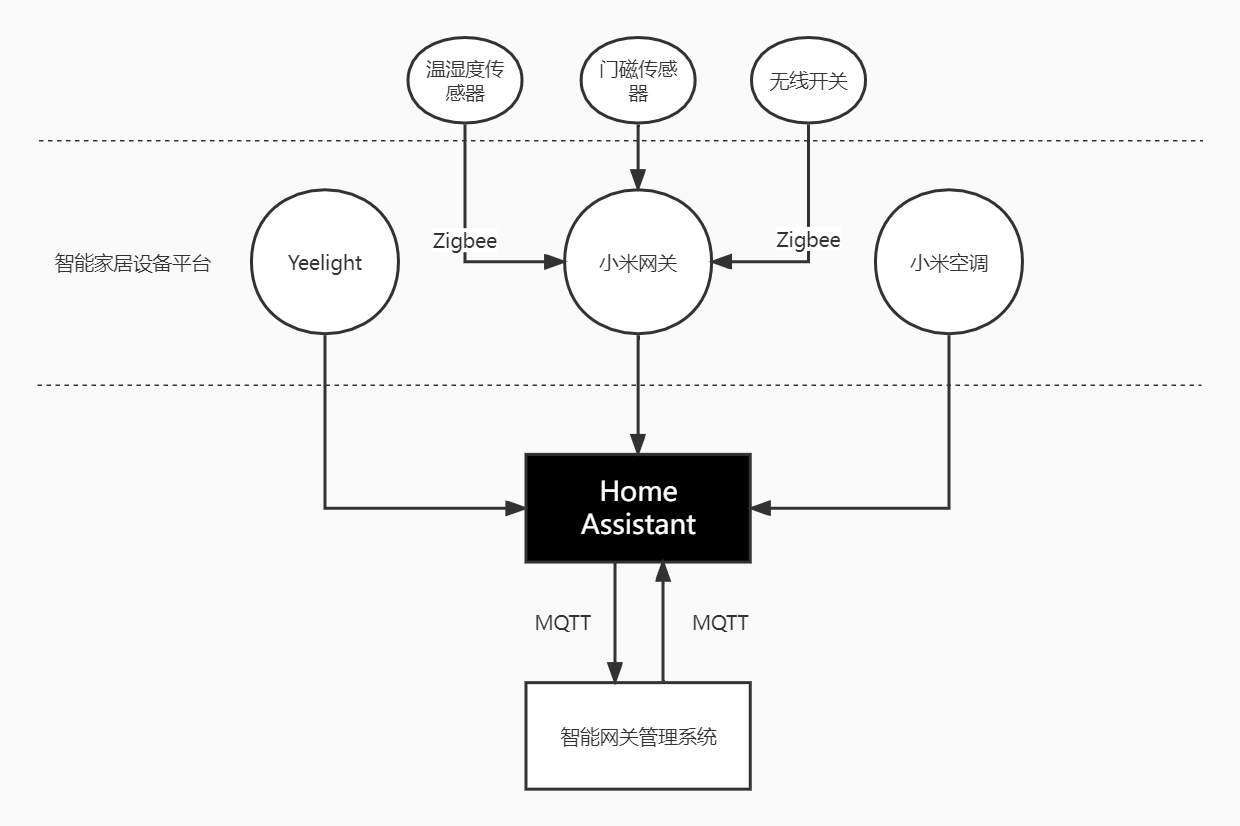
1.3 本文的章节安排

对于此次项目，本篇论文一共分为七个部分。第一章简述设计思想及本文章节概括。第二章节给出了系统总体结构图及功能概述。第三章对本次项目开发环境及通信协议进行介绍。第四章对树莓派上的开发环境的搭建和配置进行说明。第五章列出了消息通信的结构设计、界面的总体设计。六章对管理软件的主要功能模块的具体实现进行说明。第七章对智慧农场控制室智能网关管理软件进行实际测试和总结。

2 系统结构及功能概述

2.1 系统结构

本次项目是利用Home Assistant平台作为管理软件和小米传感器设备之间的桥梁，将所有的小米设备接入平台，平台通过自己的接口获取这些传感器的数据，作为发布者将数据通过MQTT协议发送消息，管理软件作为订阅者去接收消息，同时又作为发布者发送控制命令，Hass平台接收到消息后通过API向指定的设备发送命令，形成双向通信。如图2-1是管理系统的系统结构。



**图2-1 系统结构框图**

2.2 功能概述

在本次项目设计中，“智能网关管理系统”主要包括以下功能模块：

1、传感器及设备数据监测

主动收集传感器及设备相关数据，如空气温湿度数据、吸顶灯开关状态，用户可通过设备监测界面查看所有已接入设备的状态信息。

2、设备管理

对于一些可控的智能设备，如空调、吸顶灯、烟雾报警器，用户可通过设备管理界面控制一些设备的状态。

3、操作记录

用户在界面所做的操作记录到日志文件中，记录了用户在某个时刻的操作，方便用户查询操作记录。

以下是管理软件的功能描述，如表2-1。

**表2-1 功能描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | | 功能描述 | 备注 |
| 连接MQTT服务器 | | 连接到MQTT服务器，用于数据的接收和发送。 | - |
| 设备监测 | 订阅数据 | 利用MQTT库客户端订阅功能，获取传感器状态数据，用于界面显示。 | - |
| 数据解析 | 利用QJSON类对收到的数据进行解析，得到关键字段，并重新定义数据格式。 | - |
| 数据显示 | 利用Qt图形控件及方法，显示传感器数据信息。 | - |
| 设备管理 | 发送数据 | 利用MQTT库客户端发布功能，向智能网关发送控制命令。 | - |
| 操作记录 | 显示记录 | 利用QFile类读取文件数据，显示在界面。 | - |
| 查找记录 | 遍历文件数据行，并将符合条件的记录显示出来。 | - |
| 清空记录 | 用QFile::Truncate重写文件数据，清除文件数据。 | - |

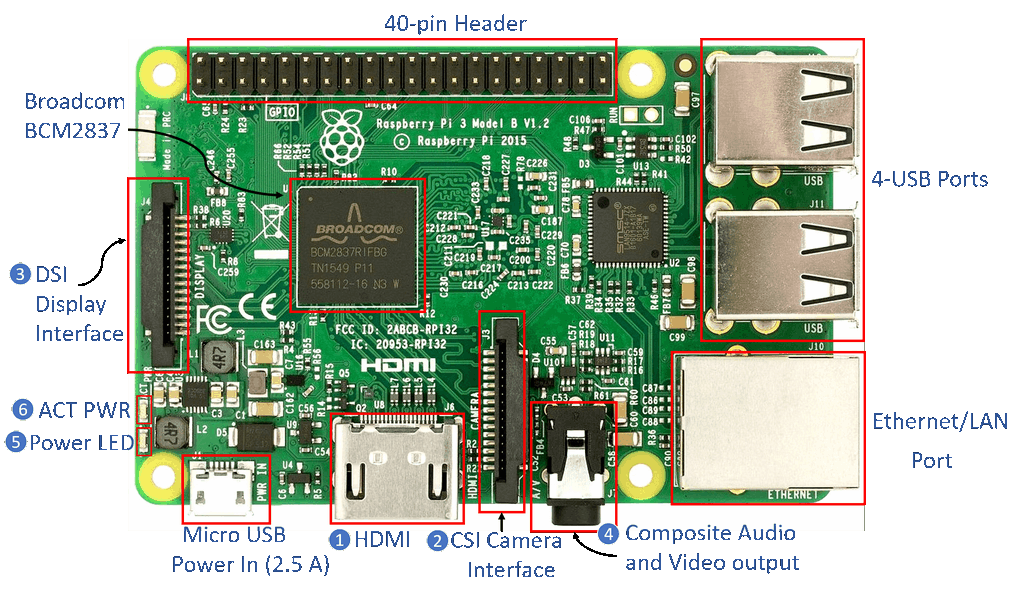
3 项目涉及的硬件设备和软件技术

3.1 树莓派介绍

3.1.1 树莓派硬件结构介绍

树莓派作为我们日常开发比较常见的一个嵌入式设备，因其能耗低，易移动等优点成为目前嵌入式开发主流的硬件设备。树莓派其实是一个操作系统+硬件结合起来的一个微型计算机，基本具备了我们生活中经常使用的PC机的核心功能，并且有一定的CPU和GPU的计算能力，带有USB、HDMI常用接口以及WIFI功能。

此次开发使用的嵌入式设备为树莓派3B型，支持在Linux和Windows不同系统下的开发，具备我们日常使用的电脑的基础功能，而且价格相对较低，提供了一个比较理想的开发平台，其微电路配备了40个GPIO 引脚，用于输入和输出[3]。对于用到的树莓派刻入的系统是树莓派目前最新系统，如图3-1为树莓派硬件结构。



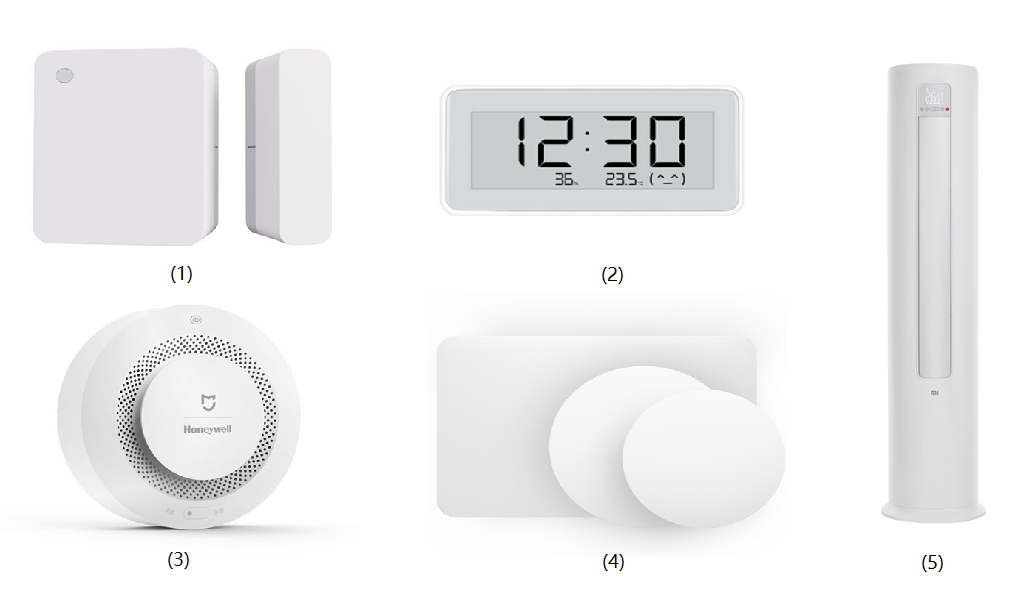
**图3-1 树莓派3B硬件结构**

有许多基于物联网的应用，如家庭自动化，智能农业，智能停车，智能教育和智能电网。使用了大量的传感器和类似树莓派这类微型计算机，融入人工智能相关技术，使传感器和执行器之间的通信使用不同的协议，如MQTT，CoAP，XMPP，REST，6LowPAN等。物联网目前主要发展为可穿戴和嵌入式两种技术，可穿戴设备具体取决于开发人员，是在自己的平台开发的，嵌入式的话主要使用树莓派、Arduino等开发平台[4]。其中Raspberry Pi价格低廉，开发成本地成为了本次项目开发选择硬件设备的主要因素。

3.2 传感器及设备介绍

3.2.1 传感器和智能设备

本次项目使用到的传感器有门窗传感器、温湿度传感器、烟雾传感器，智能设备有吸顶灯和空调。传感器及智能设备如图3-2。



**图3-2 传感器及智能设备**

⑴门窗传感器，安装在农场控制室大门和窗户等位置，通过传感器主体与磁铁的靠近与分开，实时感知门窗开关状态，从而发出状态数据，这款门窗传感器具有低功耗、安装和使用简便等特点。

(2)温湿度传感器，摆放在农场控制室桌面上，可实时监测室内温湿度，屏幕智能显示电量、数据传输、舒适度等。有摆放方便，耗能低，监测灵敏等特点。

(3)烟雾传感器，能及时检测到烟雾，并发出滴滴声和灯光闪烁，同时联动的网关也报警，能实时发送烟雾量数据出来，而且用户在手机端也能实时收到室内烟雾量的消息通知，可手动解除警报，根据实际测量，这款设备发出的警报分贝定在85分贝到92.4分贝左右。同时也可以和小米系的摄像头进行联动，配合手机推送，能更及时的发现房间异常情况，守护个人及家庭安全。

(4)吸顶灯，显色清晰明亮，使房间整体环境看起来光线明亮，颜色更真实。这款吸顶灯的优点是集成度高、厚度小。可以配合无线开关、智能网关联合使用，可定时打开和关闭吸顶灯。

(5)空调，有温度设置、风速调节、左右扫风、定时、辅热、干燥自动清洁等功能，有能耗低、使用方便等特点。此款空调发出的温度数据、风量数据、工作状态数据使其能十分方便的监测当前室内的环境状况，可以和温湿度一起使用，以便于更好的调节。

3.2.2 智能网关

采用小米智能多模网关对各种设备进行管理，所有的Wi-Fi、Zigbee智能设备都连在智能网关上面，而Wi-Fi、Zigbee、蓝牙等无线智能设备作为网关的子设备，可实现相互连接通信。通过网关接入可方便的对各种设备进行操作和控制，从而实现远程获取各种设备的状态信息和发送控制命令。以下是网关实物图，如图3-3。



**图3-3 多模智能网关**

3.3 Qt/MQTT

3.3.1 Qt开发工具介绍

Qt Creator是一个功能强大的嵌入式开发工具，是一个跨平台GUI工具包, 适用于Unix和Windows两种系统[5]。Qt Creator开发工具在Linux系统上各方面表现良好，因此成为了嵌入式设备主流开发工具，具有以下主要特点。

其一，具有跨平台性。支持Windows系列、Linux、Unix等各类系统，省掉了不少移植相关的工作，同时采用模块化设计，使其代码具有很大的弹性，能更好的在嵌入式设备上环境下运行，在一台PC机上开发，可以转移到任何其他设备上运行，不需要进行二次开发，只经过一次重新编译，便可运行。

其二，有充足的接口类。Qt配置C、C++、BOOST等以及Qt本身的库，还支持文件读取、输入输出管理、空间配置、内存管理等常用类，同时支持算法迭代、数据处理，对一些外部库，如MQTT、OpenCV等也支持。

其三，支持图形化设计。支持2D/3D图形化渲染，可以动态的拖动组件，自动生成XML文件，这与JAVA面向对象编程非常类似，极大了简化了开发者的开发过程，提高了代码的复用效率，结合MVC模式，使项目更加工程化。

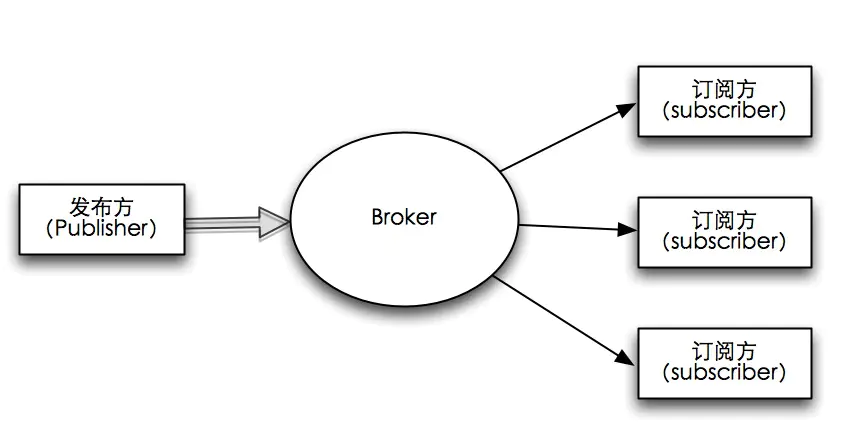
在Linux系统开发中，较常用的图形开发有OPENGUI、Qt / Embedded等，由于其自身特点，所以应用领域不一，鉴于它们在软件功能体系结构上大相径庭，从程序开发周期以及兼容性等角度考虑，便选择利用Qt来构建GUI[6]。本篇论文以智能设备管理系统设备管理界面、监测界面等设计思想去使用Qt图形设计工具来开发一款智能设备管理软件。

本次项目开发选用的软件版本是Qt4.8和Qt Creator5.2进行树莓派GUI的设计和开发，借助其可移植性的优点，只需要在一台PC机上进行开发，然后在其他机器上编译即可运行。由于本次项目使用到MQTT协议，所以还需要编译安装MQTT库，保证应用程序在PC机上能够满足管理系统的开发要求。

3.3.2 MQTT通信协议

MQTT是一个C/S模式的通讯协议，一台服务器对应多个客户端。其特点是延迟低、开销低、易于实现以及开放性。在物联网领域应用甚广，如各类传感器，智能家居，车载系统，医疗设备，一般针对嵌入式设备，在TCP/IP网络层工作。

MQTT通信协议有3种身份，分别为发布者、接收者、服务器代理。MQTT客户端与代理服务器建立连接，负责发布或者订阅相关主题的消息，而MQTT消息代理负责将收到的消息进行处理，然后将其消息转发给接收者，从而实现广播模式的消息传输[7]。如图3-4是MQTT通信模型。



**图3-4 MQTT协议通信模型**

MQTT协议主要用于计算能力要求不高、网络环境不理想、消息的可靠性较低传输的远程通讯设备的一种协议，广泛用于一些物联网嵌入式设备进行数据传输。其有以下三个主要特点：1、基于发布/订阅传输消息模式，客户端发送一类主题消息，经过MQTT代理服务器，转发到订阅此类主题的客户端，从而去实现广播模式的数据传输；2、网络连接采用TCP/IP协议，保证了消息的可靠性；3、传输速率快，由于不保证消息的服务质量，传输过程可能丢高或重复，工作在低延迟、低流量的网络环境中，所以传输速率较快。

3.4 Home Assistant平台

 Home Assistant目前已经比较的成熟了，对各种智能设备的支持比较完整，作为开源的系统，很多开发者也在一直更新插件，能接入市面上的大多数智能产品，是构建智慧空间的神器，用户可以无偿使用。支持不同品牌的智能设备连接在同一个平台，进行统一管理。Home Assistant对低版本的Python环境不支持，对于树莓派系统默认Python2.7版本Home Assistant无法兼容，需要升级Python版本。

Home Assistant可运行在树莓派硬件上，支持大部分操作系统，其最大的优点是可实现不同厂商智能设备连接在此平台上。将Hass.io移植到已配置好的树莓派系统中，通过小米网关接收小米的烟雾报警器、门磁传感器、无线开关，空调、吸顶灯等数据信息，这些数据信号都可以与Hass.io系统进行交互[8]。所以，便可以通过一些网络交换设备去使用Home Assistant平台在同一局域网对智能家居进行管理。

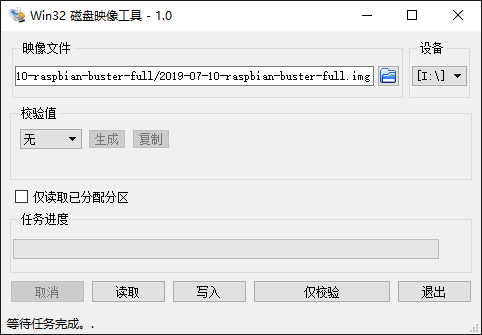
4 环境搭建

4.1 树莓派开发环境

4.1.1 系统安装

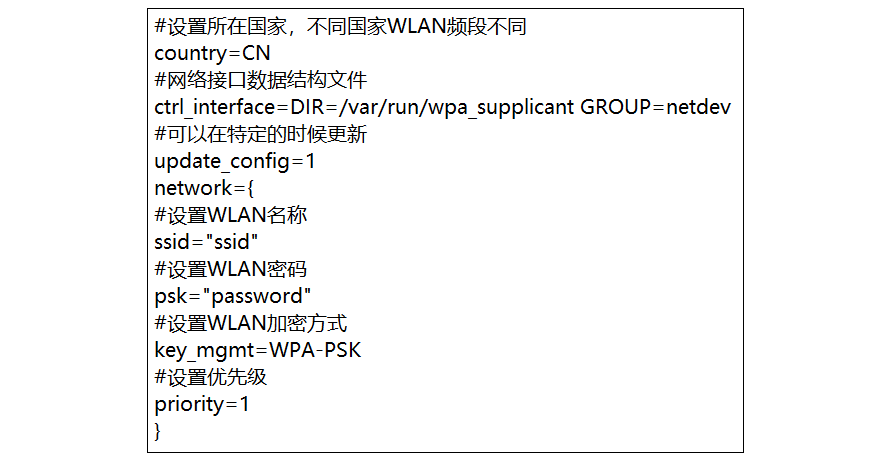
首先准备树莓派3B、16G+MicroSD卡、官方系统映像文件、SD格式化工具（可以用windows自带格式化）、磁盘映像工具（win32diskimager）、远程桌面控制软件（VNC viewer）、远程连接工具（putty），使用Windows自带的格式化工具或者其它工具，只需要将MicroSD卡格式化成FAT32格式即可，簇大小可以参考SD Card Formatter设置为32kb。

格式化完MicroSD卡后，打开软件Win32DiskImager，点击映像文件后的小图标，选择后缀名为.IMG的Raspbian系统映像文件后点击确定，然后从设备里选择MicroSD卡所在的盘符，选择好设备后点击写入，弹出来的警告同意即可，等待进度条跑完，系统就刻录成功了。如图4-1为磁盘映像工具刻录界面。



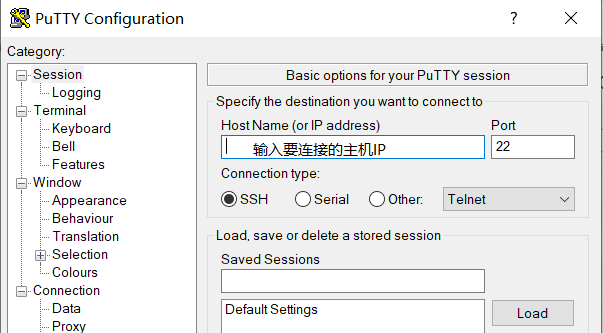
**图4-1 烧录映像文件**

系统刻录完成后，打开“此电脑”，会发现多了一个名为boot的磁盘。配置WLAN信息需要在boot目录下新建一个名为wpa\_supplicant.conf的文件（注意后缀名），打开后在文件中输入以下信息，如图4-2为文件WIFI配置信息。



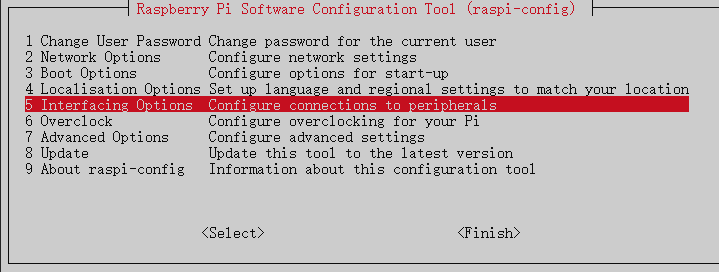
**图4-2 WIFI配置信息**

开启SSH功能只需在boot目录下新建一个名为ssh的空白文件即可。打开putty工具，在对应位置输入树莓派IP和端口号进行连接。如图4-3为putty连接工具界面。



**图4-3 SSH连接树莓派**

在树莓派安装图形界面，并使用VNC进行远程连接树莓派。首先需要在树莓派开启VNC，输入命令sudo raspi-config进入管理界面。选择选项Interfacing Options，进入接口选项，再选择VNC，进入VNC设置，最后选择YES即可成功开启VNC。然后打开VNC软件进行连接。如图4-4为VNC设置界面。



**图4-4 config界面**

4.1.2 安装Qt CREATOR及MQTT库

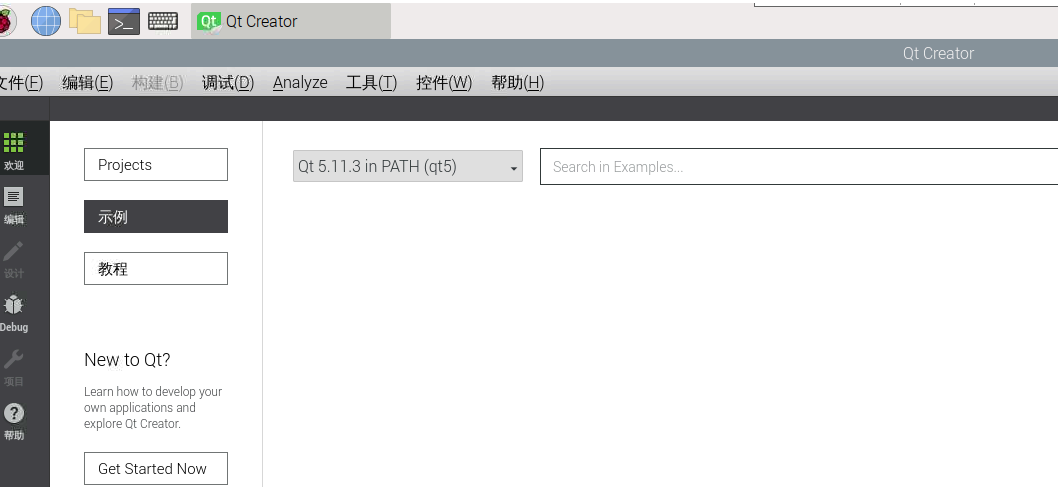
首先安装Qt5开发工具，安装步骤如下：

(1)更新：sudo apt-get update && sudo apt-get upgrade

(2)安装依赖库：sudo apt-get install qt5-default

(3)安装Qt creactor：sudo apt-get install qtcreator

安装完成后，点击树莓派主菜单可打开Qt CREATOR开发软件，打开Qt界面如图4-5所示。



**图4-5 Qt开发工具**

由于项目需要用到MQTT协议，所以需要导入MQTT库，导入步骤如下：

(1)下载qmqtt库：git clone <https://code.qt.io/qt/qtmqtt.git>

(2)安装库：sudo apt install qtbase5-private-dev

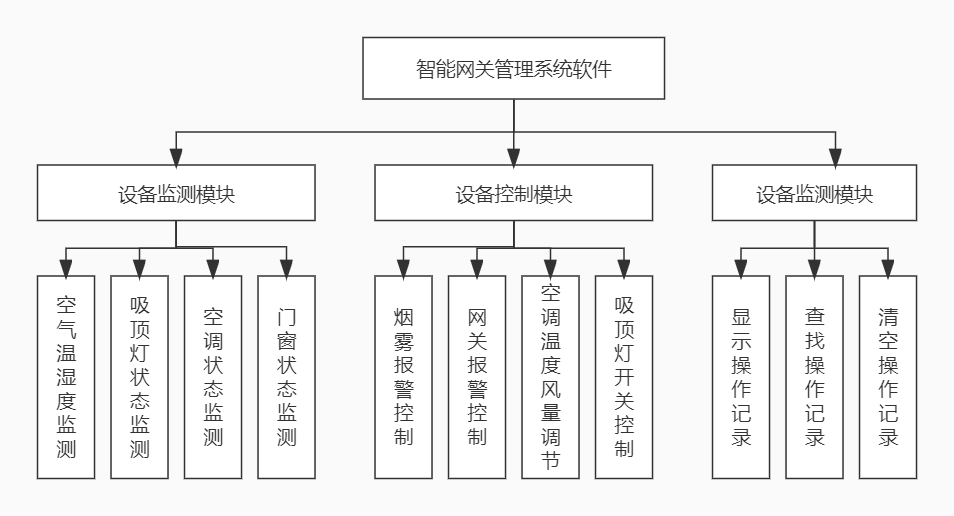
(3)编译安装MQTT源码：cd qtmqtt && qmake && make -j4 && sudo make install

导入完成后，可在项目工程中引入MQTT库。

5 管理软件的系统设计

5.1 软件结构

智能网关管理系统总共分为设备监测模块、设备控制模块、操作日志模块三大模块，其中设备监测模块用来获取传感器及设备的状态和数据，如吸顶灯的开关，空调的风量和温度等；设备控制模块用来向网关发送控制命令，如关闭烟雾报警，调节空调温度，打开吸顶灯等；操作日志模块用来记录用户的每一次操作，如空调升高一度，智能网关取消报警等。如图5-1是管理系统的软件框架结构。



**图5-1 软件框架结构图**

5.2 界面设计

5.2.1 控制室设备监测界面

系统运行后自动连接MQTT服务器，订阅主题消息，实时更新智能传感器数据和状态。定时监听网关信息，通过接收到网关发送过来的数据并解析，然后更换相应控件的内容。显示各类传感器数据和状态，未收到数据显示默认值。界面已有的设备有门窗传感器、温湿度传感器、空调、智能网关、吸顶灯、烟雾报警器。如图5-2是控制室设备监测界面。



**图5-2 控制室设备监测界面**

⑴门窗传感器，开/关门或者开/关窗时，门窗传感器会发送状态信息改变的命令，然后通过网关发送，客户端接收到消息后改变控件内容，状态显示已打开/已关闭。

⑵温湿度传感器，实时检测室内温湿度，若有变化，网关会将变化后的温湿度发送给客户端，然后客户端更新温湿度。

⑶空调，空调目前的温度和风量值将实时通过网关发送到客户端。客户端进行实时显示。

⑷智能网关，智能网关目前的状态是警戒还是无警戒，警戒状态下网关会一只响动，直到解除警戒为止。

⑸吸顶灯，目前吸顶灯的状态开/关会通过网关发送到客户端，然后实时显示吸顶灯状态。

⑹烟雾报警器，检测房间烟雾量，超过一定量时烟雾报警器会发出报警声，可手动控制解除报警。

5.2.2 控制室设备管理界面

对接入平台的设备进行控制。点击相应按钮或者推动滑动条来改变其状态和状态数据，并及时向网关发送状态信息，网关作相应的控制并更新状态。目前界面已有的设备有网关、空调、吸顶灯，其中空调可手动输入或拉动滚动条设置温度。以下是控制室设备管理界面，如图5-3。



**图5-3 控制室设备管理界面**

5.2.3 实验室设备管理界面

对在实验室接入平台的设备进行控制和显示。点击相应按钮改变其状态，并及时向网关发送状态信息，网关作相应的控制并更新状态。目前界面已有的设备有网关、门窗，能及时更新门窗的开关状态。以下是实验室的设备管理界面，如图5-4。



**图5-4 实验室设备管理界面**

5.2.4 操作日志界面

对用户在管理系统界面进行的一些操作进行记录。上半部分的文本框会实时显示用户做了哪些操作，点击显示历史记录会显示用户的所有操作，点击清空会清除所有操作数据，下半部分文本框用来显示用户查询的操作记录，用户输入关键词可以查找相关操作记录。以下是操作日志界面，如图5-5。



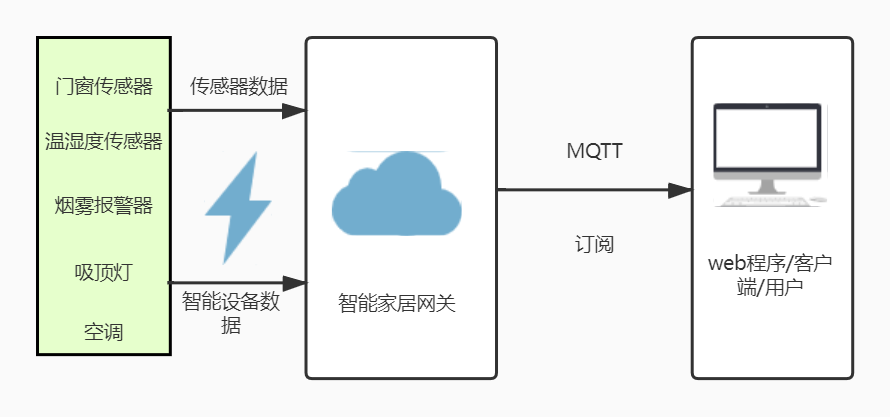
**图5-5 操作日志界面**

5.3 消息结构设计

5.3.1 消息订阅

本次项目中消息传输的数据格式均为JSON格式。JSON是1999年12月基于JavaScript编程语言的标准ECMA-262第三版的子集，它是一种轻量级的数据交换格式，因其简洁的表达和高效的传输及封装解析逐渐发展为主流的网络数据传输交换格式之一[9]。JSON最大的特点是其语法简单，适用性广，而且进行数据解析不需要那些复杂的函数和方法，鉴于它的语法十分简短轻量，所以去执行数据响应的延时也大大降低。

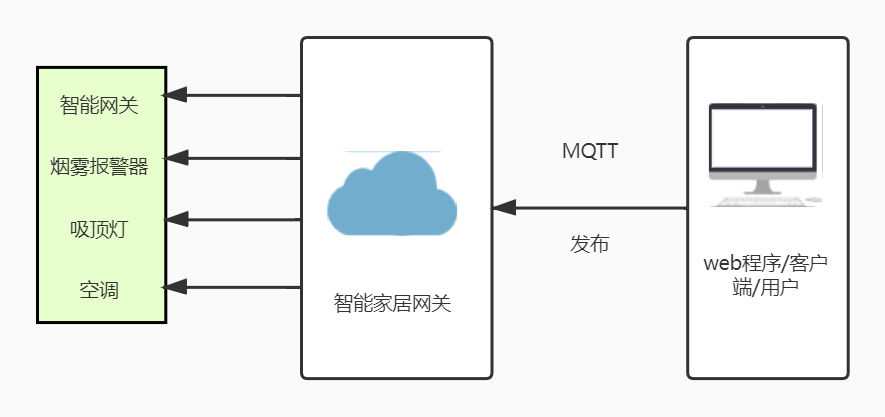
当客户端收到网关发送的相应主题的消息后，进行数据解析，然后显示到相应控件中，实现实时显示。如图5-6是消息订阅的流程。



**图5-6 消息订阅**

5.3.2 消息发布

用户想要控制某台设备的状态时，先对设备发送指令，网关接收后，对相应设备进行控制。如图5-7是消息发布的流程。



**图5-7 消息发布**

5.4 传感器及设备结构化数据设计

5.4.1 传感器及设备数据描述

为了方便各类传感器消息通信及编码，使数据结构统一，故对其进行数据标识，以下是各类传感器及设备的相关数据，如表5-1是智能设备型号及数量描述，表5-2为智能家居摆放位置描述。

**表5-1** **智能家居传感器及控制器描述表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 设备名 | deviceID | typeID | 安装位置 | 数量 |
| 门窗传感器 | 10001、10002、10003、10004、10005、10006、10007 | 501 | 控制室 | 7 |
| 门窗传感器 | 1001、1002、1003、1004、1005 | 501 | 实验室 | 5 |
| 温湿度传感器 | 20001、20002 | 502 | 控制室 | 2 |
| 烟雾报警器 | 30001、30002 | 503 | 控制室 | 2 |
| 无线开关 | 40001 | 601 | 控制室 | 1 |
| 吸顶灯 | 50001、50002 | 602 | 控制室 | 2 |
| 空调 | 60001 | 603 | 控制室 | 1 |
| 智能网关 | 70001、7001 | 604 | 控制室、实验室 | 2 |

**表5-2 智能家居房间及位置表**

|  |  |  |  |
| --- | --- | --- | --- |
| roomID | 说明 | positionID | 说明 |
| 1 | 农场控制室 | 1 | 大门 |
| 2 | 窗1（进门右边） |
| 3 | 窗2（进门右边） |
| 4 | 窗3（进门右边） |
| 5 | 窗4（进门右边） |
| 6 | 窗5（进门右边） |
| 7 | 窗6（进门右边） |
| 8 | 进门第1个 |
| 9 | 进门第2个 |
| 2 | 大棚 | — | — |
| 3 | 智能物联实验室 | — | — |

5.4.2 订阅数据结构

消息订阅主要是用户对接入的设备发送一些控制命令，网关接收到数据后进行分析处理，然后去控制相应设备的数值或状态，这下面以空调、门窗传感器数据为例：

(1)空调状态数据

消息类别：haSensorData

主题：jcsf/hadata

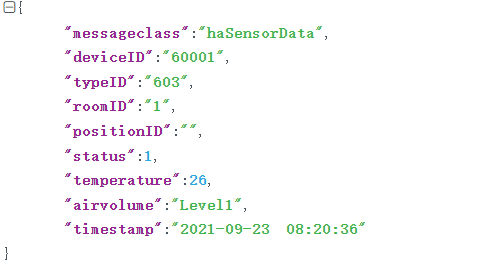
发布者：农场智能家居网关

订阅者：WEB程序/移动APP/智能网关客户端

Json字符串：

{"messageclass":"haSensorData","deviceID":"60001","typeID":"603","roomID":"1","positionID":"","status":1,"temperature":26.0,"airvolume":"Level1","timestamp":"2021-09-23 08:20:36"}

以下是空调订阅消息的Json数据格式，如图5-8。



**图5-8 订阅空调json数据**

数据说明：Messageclass表示消息的类别。deviceID和typeID分别为设备号和设备类型号，roomID为房间号，positionID为房间内的位置，status为空调状态，0-关闭，1-打开。temperature为房间温度，16-31度。airvolume为空调风量值，有level1-level7风量等级级以及Auto自动状态。timestamp为时间戳。

(2)门窗状态数据

消息类别：haSensorData

主题：jcsf/hadata

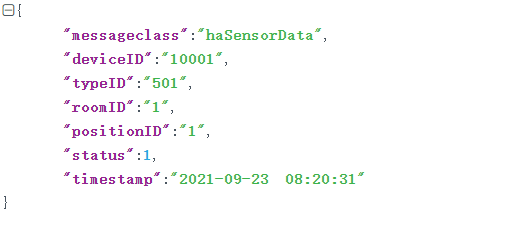
发布者：农场智能家居网关

订阅者：WEB程序/移动APP/智能网关客户端

Json字符串：

{"messageclass":"haSensorData","deviceID":"10001","typeID":"501","roomID":"1","positionID":"1","status":1,"timestamp":"2021-09-23 08:20:31"}

以下是门窗订阅消息的Json数据格式，如图5-9。



**图5-9 订阅门窗json数据**

数据说明：Messageclass表示消息的类别。deviceID和typeID分别为设备号和设备类型号，roomID为房间号，positionID为房间内的位置，status为门窗状态，0-关闭，1-打开。timestamp为时间戳。

5.4.3 发布数据结构

消息发布主要是用户对接入的设备发送一些控制命令，网关接收到数据后进行分析处理，然后去控制相应设备的数值或状态，这里以吸顶灯、智能网关为例：

(1)吸顶灯控制

消息类别：haDeviceControl

主题：jcsf/hacontrol

发布者：WEB程序/移动APP/农场智能家居网关客户端

订阅者：农场智能家居网关

Json字符串：

{"messageclass":"haDeviceControl","deviceID":"50001","typeID":"602","roomID":"1","positionID":"8","action":1}

以下是控制吸顶灯的Json数据格式，如图5-10。



**图5-10 控制吸顶灯json数据**

数据说明：messageclass表示消息的类别，deviceID和typeID分别为设备号和设备类型号，roomID为房间号，positionID为房间内的位置，action为吸顶灯状态，0-关闭，1-打开。

(2)智能网关控制

消息类别：haDeviceControl

主题：jcsf/hacontrol

发布者：WEB程序/移动APP/农场智能家居网关客户端

订阅者：农场智能家居网关

Json字符串：

{"messageclass":"haDeviceControl","deviceID":"70001","typeID":"604","roomID":"1","positionID":"","action":1}

以下是控制智能网关的Json数据格式，如图5-11。



**图5-11 控制智能网关json数据**

数据说明：messageclass表示消息的类别，deviceID和typeID分别为设备号和设备类型号，roomID为房间号，positionID为房间内的位置，action为智能网关状态，0-无警戒，1-警戒。

6 主要功能实现

6.1 连接MQTT服务器

MQTT通信过程中只有一个服务器，其他设备称为客户端，服务器建立在固定IP的主机上，客户端通过与服务器建立连接进行通信。可以用Qt提供的MQTT库创建客户端，它的创建和参数设置由new QMqtt Client实现，通过connectToHost()方法去连接代理服务器[10]。本次项目树莓派作为客户端，连接农场MQTT服务器，连接流程如图6-1。



**图6-1 连接MQTT服务器流程**

连接MQTT服务器主要代码如下：

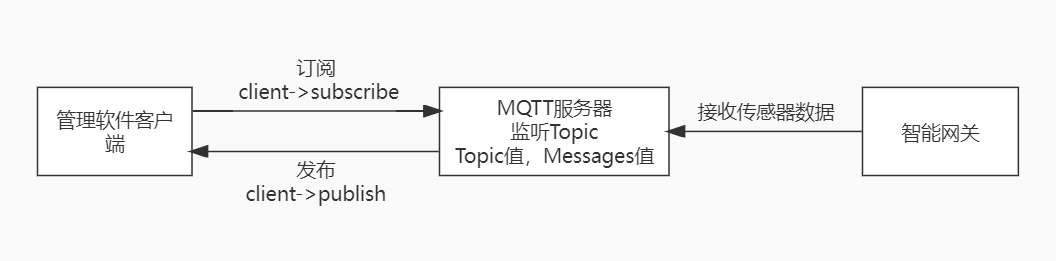
**代码6-1 连接MQTT服务器代码**

|  |
| --- |
| *void MainWindow::clientMqtt()*  *{*  *client = new QMQTT::Client(); // 初始化QMQTT客户指针*  *connect(client, SIGNAL(received(QMQTT::Message)),this, SLOT(onMQTT\_Received(QMQTT::Message)));*  *QHostAddress host("IP"); // 代理服务器IP地址（此处省略）*  *quint16 port = 1883; // 代理服务器端口*  *client->setKeepAlive(60); // 心跳*  *client->setHost(host); // 设置EMQ代理服务器*  *client->setPort(port); // 设置EMQ代理服务器端口*  *client->cleanSession();// 清除缓存*  *client->setVersion(QMQTT::MQTTVersion::V3\_1\_1);// 设置版本*  *client->connectToHost(); // 连接EMQ代理服务器*  *}* |

以上函数是Qt官方库连接MQTT服务器的固定格式，只需改变主机地址和端口号，利用信号和槽机制建立连接。

6.2 订阅数据

由于发布端发送消息是通过主题发送出去的，消息属于广播形式，接收端如果要收到消息，也需要订阅相应主题，由于传感器数据是实时更新发送出来的，所以接收端需要循环监听相应主题的数据变化。订阅消息流程如图6-2。



**图6-2 订阅数据流程**

此处用一个定时器函数来循环订阅主题，发布端和接收端建立通信定时订阅主题jcsf/hada接收从网关发送过来的传感器数据。以下是定时订阅消息主题代码：

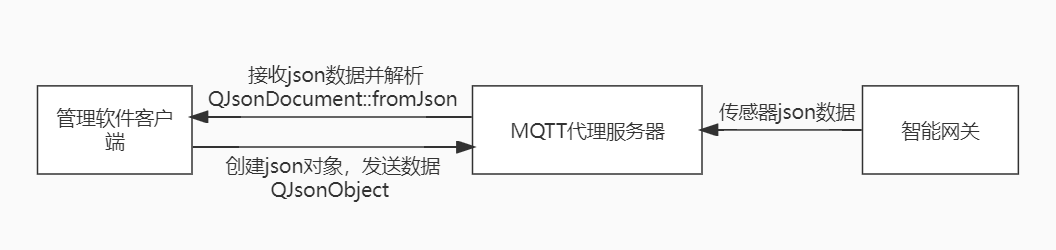
**代码6-2 定时订阅主题代码**

|  |
| --- |
| *Qtimer::singleShot(500, this, [=](){*  *client->subscribe("jcsf/hadata",0); // 定时订阅jcha/iotdata主题*  *});* |

以上定时函数以500毫秒为间隔进行一次消息订阅，能及时的获取传感器最新的状态数据。

6.3 数据解析

接收到的数据就是一段无序的字符串，我们需要将字符串格式化为json对象，以便于对数据中的特定字段和值进行提取。数据解析流程如图6-3。



**图6-3 解析数据流程**

解析JSON数据主要代码如下：

**代码6-3 解析JSON对象代码**

|  |
| --- |
| *QByteArray arr = message.payload();*  *qDebug() <<"rcive: " << message.topic() << ":" <<arr; //控制台输出*  *QString strline(arr);*  *//解析*  *QJsonParseError parseError; // 解析错误输出实例对象*  *QJsonDocument doc=QJsonDocument::fromJson(arr,&parseError);*  *if(parseError.error!=QJsonParseError::NoError)*  *{*  *qDebug()<<"转换错误";*  *return -1;*  *}*  *//获取传感器数据对象*  *QJsonObject obj=doc.object();* |

对于接收到的数据有些字段和值是不变的，我们只关注是哪类传感器以及哪些数据发生了变化，所以需要对接收到的json数据对象进行处理，取出关键字段，需要重新创建一个json对象，然后将数据存入对象中，便于发布消息数据。其主要代码如下：

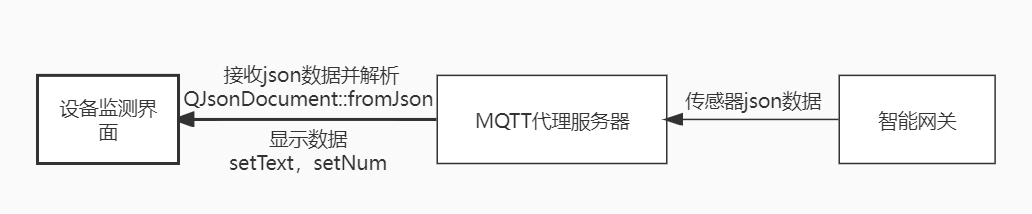
**代码6-4 创建JSON对象代码**

|  |
| --- |
| *//创建解析后的数据*  *QJsonObject sub;*  *//存入输出sub对象*  *sub.insert("messageclass",QJsonValue(messageclass));*  *sub.insert("deviceID",QJsonValue(deviceID));*  *sub.insert("sensorDevID",QJsonValue(sensorDevID));*  *sub.insert("room",QJsonValue(room));*  *sub.insert("position",QJsonValue(position));*  *sub.insert("timestamp",QJsonValue(timestamp));* |

以上是将格式化对象中的数据提取出来，删除有些不要的字段，重新创建一个对象存入这些数据。

6.4 数据显示

解析接收到的json数据，并将其更新到对应界面控件中，去更新界面显示。数据显示大多使用文字描述型控件，通过文字颜色、背景等方式，显示模块的实时状态和数据变化[11]。以温湿度传感器为例，更新控件内容流程如图6-4。



**图6-4 显示数据流程**

更新控件内容主要代码：

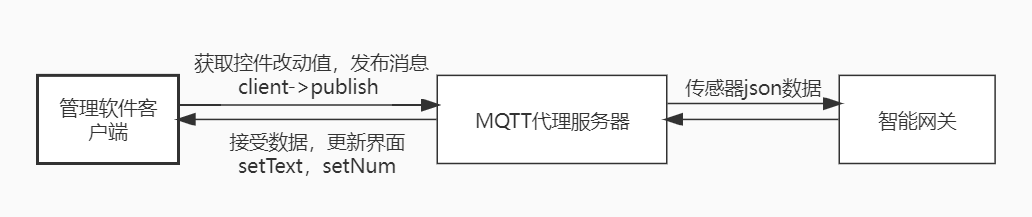
**代码6-5 解析JSON数据代码**

|  |
| --- |
| *//温湿度传感器（温度）数据*  *if(obj["deviceID"] == "20001")*  *{*  *QJsonValue temperature=obj.value("temperature"); // 获取温度值*  *sub.insert("temperature",QJsonValue(temperature)); // 创建温度值对象*  *if(!temperature.isNull())*  *{*  *double t = temperature.toDouble();*  *ui->label\_11->setNum(t); // 显示温度*  *}*  *}* |

以上是对温湿度传感器中的温度数据进行解析，根据数据类型，在相应界面的控件中进行更新。

6.5 发布数据

客户端通过点击按钮或者拉动滑动条去改变相应设备的状态，当用户执行某一操作时，实际上是获取当前数值向网关发送消息，网关获取到消息后再进行处理去改变对应的设备。下面以空调温度值为例，用户通过拉动滑动条变化值去更新温度，发布数据流程如图6-5。



**图6-5 发布数据流程**

使用垂直滑动条快速的调节空调的温度，滑动滑动条可以将空调温度调到温度限定值内的任意温度。滑动条改变温度主要代码如下：

**代码6-6 发布空调数据代码**

|  |
| --- |
| *//空调温度值*  *void MainWindow::on\_verticalSlider\_valueChanged(int value)*  *{*  *int t = value;// 获取滑动条的值*  *QString t1 = QString::number(t,10);// 数据格式转换*  *ui->lineEdit->setText(t1);// 更新控件数据*  *QString py5 = QString("{\"messageclass\":\"haDeviceControl\",\"deviceID\":\"60001\",\"typeID\":\"603\",\"roomID\":\"1\",\"positionID\":\"\",\"action\":1,\"temperature\":%1}").arg(t);*  *QMQTT::Message msg5(0,"jcsf/hacontrol",py5.toLatin1());*  *client->publish(msg5);// 发送消息*  *}* |

用户在界面可点击加减按钮升高和降低温度，点击一次增加或减少1度。按钮调节空调温度代码如下：

**代码6-7 发布空调数据代码**

|  |
| --- |
| *void MainWindow::on\_pushButton\_4\_clicked(){*  *QString py11("{\"messageclass\":\"haDeviceControl\",\"deviceID\":\"60001\",\"typeID\":\"603\",\"roomID\":\"1\",\"positionID\":\"\",\"action\":1,\"ud\":\"up\"}");*  *QMQTT::Message msg11(0,"jcsf/hacontrol",py11.toLatin1());*  *client->publish(msg11); // 发布数据*  *}* |

水平滑动条是调节风量的，由于滑动条的值只能是数字类型，所以需要转换成相应字符串进行显示。其主要代码如下：

**代码6-8 调节空调风量代码**

|  |
| --- |
| *void MainWindow::on\_horizontalSlider\_valueChanged(int value)*  *{*  *int a = value; // 获取风量值*  *QString a1 = QString::number(a,10); // 格式转换*  *ui->lineEdit\_2->setText(a1); // 显示风量*  *QString a2;*  *if(a == 1){*  *a2 = "Level1"; // 变换数据类型*  *}*  *}* |

6.6 操作日志

6.6.1 显示操作记录

用日志文件去记录用户操作，用户每操作一次，然后向文件中写入一行，通过访问文件内容，去显示操作记录。

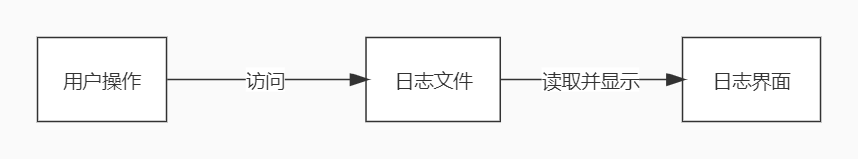
首先获取当前操作的时间，实现代码如下：

**代码6-9 获取当前时间代码**

|  |
| --- |
| *//获取当前时间*  *QString MainWindow::get\_time()*  *{*  *QDateTime curDateTime=QDateTime::currentDateTime(); // 实例化对象*  *QString date = curDateTime.toString("yyyy-MM-dd hh:mm:ss");*  *qDebug() << "date:"<<date;*  *return date; // 返回时间*  *}* |

返回的日期格式如：2022-01-01 09:29:13。

客户端访问日志文件流程如下，如图6-6。



**图6-6 显示操作记录流程**

将用户操作记录和返回的时间组合成新的字符串，然后写入文本文件，并在界面实时显示操作记录，实现的代码如下：

**代码6-10 写入文件代码**

|  |
| --- |
| *//写入内容*  *void MainWindow::content\_change(QString str)*  *{*  *QString d = get\_time(); // 获取时间*  *QString strl = str+"，"+d+"；"+"\n"; // 连接字符串*  *QByteArray bytes = strl.toUtf8();*  *ui->logEdit->append(bytes); // 更新界面*  *QFile out("./log.txt"); // 定义日志文件名*  *if(out.open(QIODevice::Append | QIODevice::Text)){ // 打开日志文件*  *out.write(bytes); // 写入数据*  *}*  *out.close(); // 关闭文件*  *}* |

以上函数实现了获取用户操作记录写入到日志文件，以追加的方式以行的格式进行添加，在界面的相应按钮处每点击一次，就在调用一次此函数，例如空调降低1度，实现代码如下：

**代码6-11 操作记录代码**

|  |
| --- |
| *//空调减1*  *void MainWindow::on\_* *ControlRoomAirConditioning\_down()*  *{*  *QString str = "控制室空调降低了1度";*  *content\_change(str);*  *}* |

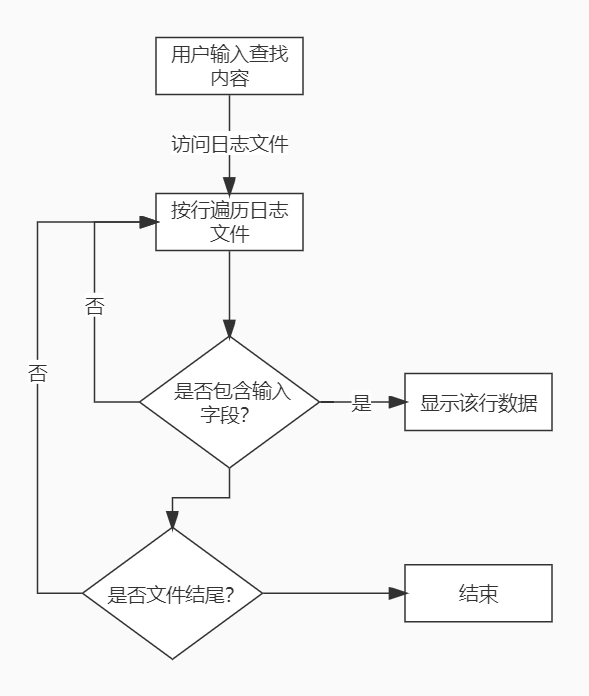
当用户在界面点击降低温度时，调用以上函数，实现将操作记录写入到日志文件中，并在界面显示，如图6-7。



**图6-7 显示操作记录**

6.6.2 查找操作记录

首先当用户需要去查找某一设备在在某个时刻做了哪些状态的改变，就需要实现一个设备操作日志查找功能，按照关键词查找日志文件中的相关内容并显示出来，用户查找记录日志流程如图6-8。



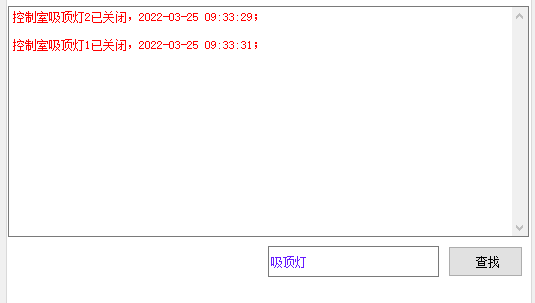
**图6-8 查找操作记录流程**

当用户输入查找内容后，直接访问操作日志文件，判断是否存在用户查找的内容字段，方便查看指定设备的相关记录，实现代码如下：

**代码6-12 查找记录代码**

|  |
| --- |
| *//查找内容*  *void MainWindow::on\_* *find\_record ()*  *{*  *QString s = ui->lineEdit\_3->text(); // 获取用户输入信息*  *QFile file("./log.txt");*  *if (file.open(QIODevice::ReadOnly | QIODevice::Text))*  *{*  *while (!file.atEnd()) // 遍历日志文件*  *{*  *QByteArray line = file.readLine();*  *QString str(line);*  *if(str.contains(s,Qt::CaseSensitive)) // 判定输入信息是否存在*  *{*  *ui->logEdit\_2->append(str);*  *}*  *}*  *file.close(); // 关闭文件*  *}*  *}* |

用户可在输入框输入要查找的记录关键词去查找操作记录，根据用户在输入框内输入要查找的内容，在日志文件中遍历查找包含此段内容的字符行，然后显示在界面。界面显示如图6-9所示。



**图6-9 显示查找记录**

6.6.3 清空操作记录

清除用户所有的历史操作记录，并清除界面显示，清除文件内容用了QFile类自带QIODevice::Truncate方法，以重写的方式打开文件，清除界面内容用了textEdit自带clear()方法，实现代码如下：

**代码6-13 清除记录代码**

|  |
| --- |
| *//清空历史*  *void MainWindow::on\_pushButton\_2\_clicked()*  *{*  *QFile fileModify("./log.txt"); // 日志文件名*  *if (!fileModify.open(QIODevice::WriteOnly | QIODevice::Text | QFile::Truncate))*  *{*  *qDebug()<<"文件清空失败";*  *}*  *fileModify.close(); // 关闭文件*  *ui->logEdit->clear();*  *}* |

以上函数实现了清空所有文件数据以及界面显示内容，用户点击清空按钮可以达到清除所有历史数据效果。

7 测试结果

7.1 树莓派测试

打开树莓派，运行程序，自动连接到农场MQTT服务器。首先查看设备监测界面，没有数据的状态如下，如图7-1。



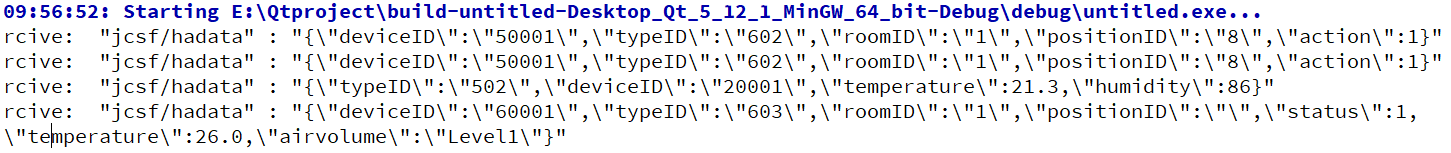
**图7-1 未收到数据界面显示**

点击相应设备开关，打开控制室的温湿度传感器、空调、吸顶灯等设备。设备状态如下，如图7-2。



**图7-2 设备状态**

同时从网关接收到的传感器数据如下，如图7-3。



**图7-3 控制台收到的传感器数据**

设备监测界面更新数据及状态如下，如图7-4。



**图7-4 收到数据界面显示**

可以看到界面内容数据与设备的实际数据完全符合，然后测试控制界面。

以下是吸顶灯空调都关闭，未收到数据的设备控制界面，如图7-5。



**图7-5 未收到数据界面显示**

当收到传感器或者设备传过来的数据时，界面更新如下，如图7-6。



**图7-6 收到数据界面显示**

拉动滚动条，将温度由28度降低到24度，吸顶灯1关闭，界面显示如下，如图7-7。



**图7-7 改变状态界面显示**

相应的设备状态也发生了改变，能成功控制农场控制室的设备，以下是设备的状态，如图7-8。



**图7-8 设备状态**

通过测试，本次项目的各个功能正常，可实时显示传感器数据和状态，可准确的控制接入平台的所有可控设备，功能实现较完善。而界面也从最开始的较为简陋的调试页面，逐步优化成操控方便、较为美观的界面。

结论

本次项目从设计开发到论文完成经过半年多的时间，经过这段时间的坚持与不懈，也顺利的完成了项目预期的各项功能，在这次的毕业设计中，借助于学校农场已有的设备，来获取各种传感器数据，作为此次开发的基本环境，然后根据指导老师安排的项目方向踏上了我的开发历程。

此次开发我运用了Linux，Python，计算机网络和Qt等相关知识，并学习了Home Assistant开源智能家居平台的开发文档，在树莓派系统上搭建了Hass.io服务平台，此平台可在局域网内自动发现设备，目前所有的智能设备都已连接到这个平台上，通过这个平台获取传感器发出的数据，而管理软件则是通过MQTT协议获取数据，最终控制都是通过Home Assistant平台进行的。

通过此次项目的实践，不管是智能家居还是智慧农场，目的都是实现用户远程管理相关设备，智慧农场不仅仅有风机、喷淋水阀、遮阳网等这些外部设备，农场控制室也得有一些内部的智能设备，将智能家居应用到智慧农场，本身就不矛盾，为了推动智慧农场的智能化管理，所以本文对智慧农业的发展也是有一定的实际意义。

在开发的过程中，还是有一些不足之处，由于本次开发在网上也获取不到相关的教程可以参考，几乎没人往这方面去做研究，自然而然遇到了很多问题解决不了，其中最大的问题就是连接智能设备必须先在Home Assistant平台上进行连接，否则管理软件是无法添加新的智能设备的，显得这个管理软件十分的被动了，而本次的管理软件客户端也并不是很完善，无法动态的在界面添加组件，所以本次开发尚不成熟，未达到一个预期的理想效果，仅仅只是完成了其核心功能的基础模块。但是就本次研究的结果来看，还是有很大的收获的，每完成一个小功能都是前进了一步，对于本次的项目，后续是可以继续研发然后逐步完善还未完成的功能，且研究前景也是紧贴时代的，经过5G技术，甚至以后的6G，万物互联已成必然趋势。

参考文献

[1]顾天威,王玮,李晶,陈雪.基于物联网技术的智能家居发展趋势探究[J].包装与设计,2021(04):116-117.

[2]王莹.物联网在智慧农业中的现状及发展趋势研究[J].技术与市场,2022,29(01):111+113.

[3]阳秋光,成建宏.基于树莓派的智能灯光控制系统设计[J].电脑知识与技术,2018,14(29):220-221+223.DOI:10.14004/j.cnki.ckt.2018.3457.

[4]Shilpa H Baria,et al.. Personal and Intelligent Home Assistant to Control Devices Using Raspberry Pi[J]. International Journal of Computing and Digital Systems,2017,6(4).

[5]李全虎.交互界面开发工具-Qt[J].中国科技信息,2005(05):33.

[6]沈炜,王晓聪.基于Qt的嵌入式图形界面的研究和应用[J].工业控制计算机,2016,29(01):101-102+104.

[7]于海飞,张爱军.基于MQTT的多协议物联网网关设计与实现[J].国外电子测量技术,2019,38(11):45-51.DOI:10.19652/j.cnki.femt.1901629.

[8]李春光,何中胜,何思楷,张淞强,季亚香.基于Home Assistant的智能家居系统研究[J].电子世界,2019(15):53-54+57.DOI:10.19353/j.cnki.dzsj.2019.15.022.

[9]刘家雨,王永生,刘爱东.基于JSON的防空武器系统应用适配器[J].兵工自动化,2021,40(07):1-4.

[10]贾贝贝,康明才.基于Qt的嵌入式系统文件传输上位机软件设计[J].电子设计工程,2022,30(03):122-125+130.DOI:10.14022/j.issn1674-6236.2022.03.027.

[11] 邵刚,谢世琪,段国宁.基于Qt编写的可配置的监控系统[J].工业控制计算机,2022,35(03):69-70.

致谢

历时半年多的时间得以将项目和论文顺利完成，本论文的工作是在庄建老师的悉心指导下完成的。

项目作为论文的基础，在做项目到论文完成期间，首先感谢我的论文指导老师庄建老师，他严谨治学的态度和孜孜不倦的教导，从选题，定题，项目开发到最后论文定稿，一直认真负责给我们每一位同学细致的指导，本篇论文的完成，无不凝聚着庄建老师大量的心血。

然后感谢这篇论文所涉及到的各位学者。本文引用了数位学者的研究文献，如果没有各位学者的研究文献给与我的帮助，我将很难完成本篇论文的写作。

最后，感谢曾经教育和帮助我的所有老师，祝老师们桃李天下，春晖四方。也感谢母校成都锦城学院的教育之恩，祝愿母校成都锦城学院更加辉煌，再续篇章！