

分类号 \_\_\_\_\_

UDC \_\_\_\_\_

密级 \_\_\_\_\_

学号 6720180146



江西理工大学

# 硕士学位论文

## Thesis for Master's Degree

论文题目 家庭物联网智能控制系统

申请学位类别 工程硕士

专业名称 电子与通信工程

研究生姓名 王传强

导师姓名、职称 任克强 教授

二〇二一年五月

分类号: \_\_\_\_\_

密 级: \_\_\_\_\_

U D C: \_\_\_\_\_

学 号: 6720180546

# 江西理工大学

## 硕 士 学 位 论 文

家庭物联网智能控制系统  
Intelligent control system of home Internet of things

学 位 类 别: 工 程 硕 士

作 者 姓 名: 王传强

学 科、专 业: 电子与通信工程

研 究 方 向: 嵌入式技术与应用

指 导 教 师: 任克强

2021年 5月 25日

## 学位论文独创性声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含已获得江西理工大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并予以致谢。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

研究生签名：王传强

时间：2021年5月25日

## 学位论文版权使用授权书

本人完全了解江西理工大学关于收集、保存、使用学位论文的规定：即学校有权保留按要求提交的学位论文印刷本和电子版本，学校有权将学位论文的全部或者部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编、查阅和借阅；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版。本人允许本学位论文被查阅和借阅，同意学校向国家有关部门或机构送交论文的复印件和电子版，并通过网络向社会公众提供信息服务。

保密的学位论文在解密后适用本授权书

学位论文作者签名（手写）：王传强

导师签名（手写）：

任志强

签字日期：2021年5月25日

签字日期：2021年5月25日

第三次信息产业革命伴随着物联网的发展而兴起，在传感器和嵌入式两个技术的共同支持下，以其易于结合、容错率高、易于部署、易于增减传感器节点、执行效率和速率高等技术优势，已经应用在智能家居等诸多领域。本文将物联网引入家庭内部，设计了一个集数据采集传输、语音呼叫、移动智能监视与一体的家庭物联网系统。

基于物联网技术的发展，以设计并完善家庭内部物联网系统为根本目标，本文解释了家庭物联网监控系统所涉及的基础理论知识、国内外目前的研究现状以及相关的物联网技术原理。为了实现整个设计可以具体的应用于家庭，本文工作内容如下：

（1）数据采集传输系统：系统以 STM32F103 为核心处理器，将系统分为主节点和从节点，主从节点之间通过 LoRa 模块进行数据传输。从节点可以将采集到的各种传感器数据传输到主节点，主节点可以对数据进行分析然后可以向从节点下发控制指令，也可以通过 ESP8266 模块将数据上传至云端并将获取的数据通过改进后的 TFT 液晶屏显示。整个系统可以完成各个节点传感器数据获取、上传、监控、显示。

（2）语音呼叫系统：系统以 ESP8266 为数据传输节点，以路由器为中继，通过 UDP 通信方式。将 ESP8266 作为数据节点，所有的数据通过路由器进行交互，通过设定静态 IP 和端口号的方法确认各个节点，防止连接到路由器上的设备被随机分配地址，为了节约 ESP8266 的 IO 口，使用模拟输入的方式，使得多个按键共用一个 IO 口。经试验验证，语音呼叫系统可以实现多个节点之间的相互呼叫，能够较好的满足室内语音呼叫的需求。

（3）移动智能监视系统：系统由视频采集加无线传输组成可移动的平台。该平台由车体、旋转云台和无线视频采集组成，车体和云台部分 STM32F030 单片机对整个运动部分进行控制，控制 NRF24L01 和 ESP8266 通过 2.4G 信号和 Internet 遥控车体运动和云台自由旋转；无线视频采集系统使用高通 AR9331 路由芯片搭载 OpenWrt 路由系统，支持所有 USB 免驱 MJPEG 格式输出的摄像头，视频监控状态可通过手机或者 PC 实时查询。为了准确获取车体当前运动的位置，在软件设计中优化了定位算法，以实现对车体的实时定位。经实验验证平台运行稳定可靠，操作控制灵活简便，车体定位较准确，视频显示流畅，无卡顿现象，能够较好的满足多种应用场景下的视频监控。

本文通过对数据采集传输系统、语音呼叫系统、移动智能监控平台进行设计，最终设计并完成了可用于家庭的物联网智能控制系统。

：物联网；数据采集传输；语音呼叫；移动智能监控

## Abstract

With the development of the Internet of things, the third information industry revolution is rising. With the common support of sensor and embedded technology, it has been applied in many fields, such as smart home, because of its easy combination, high fault tolerance, easy deployment, easy to increase or decrease sensor nodes, high execution efficiency and high speed. This paper introduces the Internet of things into the home, and designs a home Internet of things system which integrates data acquisition and transmission, voice call and mobile intelligent monitoring.

Based on the development of Internet of things technology, with the fundamental goal of designing and improving the home internal Internet of things system, this paper explains the basic theoretical knowledge involved in the home Internet of things monitoring system, the current research status at home and abroad, and the relevant principles of Internet of things technology. In order to realize the specific application of the whole design in the family, the work of this paper is as follows:

(1) Data acquisition and transmission system: the system uses STM32F103 as the core processor, and the system is divided into master node and slave node. The master and slave nodes transmit data through Lora module. The slave node can transmit all kinds of sensor data collected to the master node. The master node can analyze the data, and then send control instructions to the slave node. The data can also be uploaded to the cloud through the ESP8266 module, and the obtained data can be displayed on the improved TFT LCD screen. The whole system can complete the sensor data acquisition, upload, monitoring and display of each node.

(2) Voice call system: the system uses esp8266 as data transmission node, router as relay and UDP communication mode. ESP8266 is regarded as a data node, all data are interacted through the router, and each node is confirmed by setting static IP and port number to prevent devices connected to the router from being randomly assigned addresses. In order to save the IO port of ESP8266, analog input is used to make multiple keys share one IO port. The experimental results show that the voice call system can realize the mutual call between multiple nodes, and can better meet the needs of indoor voice call.

(3) Mobile intelligent monitoring system: the system is a mobile platform composed of video acquisition and wireless transmission. The platform is composed of car body, rotating pan tilt and wireless video acquisition. STM32F103 MCU controls the whole motion part of car body and pan tilt, and controls NRF24L01 and ESP8266 to remotely control car body motion and pan tilt free



rotation through 2.4G signal and Internet; The wireless video capture system uses high pass AR9331 routing chip and openwrt routing system, supports all USB drive free MJPEG format output cameras, and the video monitoring status can be queried in real time through mobile phone or PC. In order to accurately obtain the current motion position of the car body, the positioning algorithm is optimized in the software design to realize the real-time positioning of the car body. The experimental results show that the platform is stable and reliable, flexible and simple operation and control, accurate positioning of car body, smooth video display and no stuck phenomenon, which can better meet the video monitoring requirements in a variety of application scenarios.

In this paper, through the design of data acquisition and transmission system, voice call system and mobile intelligent monitoring platform, an intelligent control system for home Internet of things is designed and completed..

**Key Words:** Internet of things; data acquisition and transmission; voice call;  
mobile intelligent monitoring

## 目 录

摘    要.....	I
Abstract.....	II
第一章 绪 论.....	1
1.1 选题背景及意义.....	1
1.2 国内外研究现状.....	1
1.3 论文主要工作 .....	3
1.4 论文组织结构.....	4
第二章 家庭物联网智能控制系统原理和设计方案.....	5
2.1 功能需求分析.....	5
2.1.1 家庭物联网智能控制系统功能需求分析.....	5
2.1.2 实现技术功能分析 .....	5
2.2 实现依据 .....	6
2.2.1 数据采集控制系统原理.....	6
2.2.2 串口驱动TFT液晶屏原理 .....	6
2.2.3 语音呼叫系统原理 .....	7
2.2.4 移动智能监控平台原理.....	7
2.3 设计方案 .....	8
2.3.1 数据采集控制系统设计方案 .....	8
2.3.2 串口驱动TFT液晶屏设计方案.....	8
2.3.3 语音呼叫系统设计方案.....	9
2.3.4 移动智能监控平台设计方案 .....	9
2.4 本章小结 .....	10
第三章 家庭物联网智能控制系统硬件设计.....	11
3.1 数据采集监控系统硬件设计 .....	11
3.1.1 系统组成.....	11
3.1.2 数据采集节点硬件设计.....	12
3.2 多通道串口驱动TFT液晶屏硬件设计 .....	13
3.2.1 STM32F429硬件电路设计 .....	14

3.2.2 MCU接口TFT液晶屏设计 .....	15
3.3 语音呼叫系统硬件设计 .....	15
3.4 移动监控平台硬件设计 .....	16
3.4.1 移动平台硬件设计 .....	16
3.4.2 无线视频监控硬件设计 .....	17
3.5 本章小结 .....	17
第四章 家庭物联网智能控制系统软件设计 .....	18
4.1 室内数据采集监控系统软件实现 .....	18
4.1.1 节点通信程序设计 .....	20
4.1.2 数据采集、控制及上传程序设计 .....	22
4.2 多通道串口驱动TFT液晶软件实现 .....	22
4.2.1 液晶显示驱动设计 .....	24
4.2.2 串口通信程序设计 .....	25
4.2.3 驱动控制接口设计 .....	26
4.3 WIFI模块组网完成语音交互软件实现 .....	27
4.3.1 驱动控制接口设计 .....	27
4.4 可移动无线视频监控平台软件实现 .....	28
4.4.1 移动平台软件设计 .....	28
4.4.2 OpenWrt系统移植 .....	30
4.4.3 定位算法 .....	31
4.5 本章小结 .....	33
第五章 家庭物联网智能控制系统应用测试 .....	34
5.1 室内采集监控系统测试 .....	34
5.2 液晶屏驱动测试 .....	35
5.3 语音呼叫测试 .....	36
5.4 移动视频监控平台测试 .....	37
5.5 本章小结 .....	38
第六章 总结与展望 .....	40
6.1 总结 .....	40
6.2 展望 .....	41



参 考 文 献.....	42
附 录A 程序清单.....	45
致 谢.....	51
攻读学位期间的研究成果 .....	52

### 1.1

通过制定协议，将物体通过传感器接入网络，在网络和物体之间通过有线和无线的方式进行数据交换，让物体拥有智能化的功能，叫做物联网。物联网的兴起需要有强大的物联网平台的支撑，对于物联网平台国内很多的企业也在纷纷进行建设，目前国内使用比较广泛的有阿里物联网平台、百度物联网平台、腾讯物联网平台等<sup>[1-2]</sup>。

物联网是在 1999 年被第一次提出，当时提出的时候概念比较简单，简而言之，就是将物体通过无线传感器设备进行连接，实现统一的、智能化的管理与识别。但是，经过这么多年的技术进步，物联网发展早已超出了简单的物体与网络相连的简单层面，而是所有的与物联网有关的都建立在一个体系平台之下，通过对设备的统一管理实现，而且操作简单方便、高效<sup>[3-5]</sup>。

最近几年，物联网发展速度十分迅速，物联网技术的使用已经涉及到了诸多领域，比如，车联网、智能交通、智能物流、智能农业、智能医疗、智能家居等众多领域。而且应用范围还在不断的扩大，这说明物联网将会是未来科技发展的趋势之一，在这一趋势下智能家居发展尤为突出。智能家居就是将物联网技术与各种信息、通信等技术结合构建出一套家庭事务管理器，在这个管理器上以提高人们生活效率为宗旨<sup>[6-7]</sup>。

目前家庭使用的部分物联网控制系统的功能比较单一，大部分都是人为发布指令被动的控制系统，无法根据采集的数据主动控制。可见目前的部分智能控制系统有自身的局限性，不适合所有家庭的使用。结合此不足，本次设计了一种基于物联网的可用于室内的物联网控制系统，通过 WIFI 和 LoRa 进行组网，实现了对数据的获取、传输、人机交互和设备控制；实现了多节点语音呼叫功能；实现了移动平台下的可移动的视频监控。

### 1.2

物联网的出现在 1999 年，但是在 6 年后的 2005 年才正式提出。从 1999 到 2005 再到今天，物联网经过 20 几年的发展在技术层面已经非常成熟，在应用层面已经被应用的到很多地方，人们万物互联设想已经走进现实。

物联网所实现的最终目标是万物互联，实现万物间和人与万物间的信息交互和传递。在这一目标下，无论是国内还是国外都在大力发展物联网产业，谷歌提出了互联

网最终将会消失，取而代之的将会是物联网的观点，中国社科院则指出，在产业规模物联网与互联网将高出至少 30 倍。而且调查研究指出，未来十年内，新增的物联网设备终端数量年复合增长率将超过 30%，说明万物互联的时代已经开启<sup>[8-9]</sup>。

### (1) 国外研究现状

近些年来在一些国家，物联网技术的发展十分迅速，物联网在国外被赋予很高的评价，他们对物联网的发展也极其重视，西方许多国家都将物联网产业列入经济发展的规划当中，并建立了很多相关的实验室进行研究，而且在大学教育也开设了物联网相关专业，培育物联网相关人才<sup>[10-13]</sup>。在国外对于物联网的项目很多的研究都放在了车联网以及农业物联网，在北美的一些国家的一些较为大型的农场内，大都采用了现代化的农业生产技术，通过物联网技术实现了对所有农产品的精准管理，从农作物的播种，生长一直到收获，运输等过程进行精准管理<sup>[14-16]</sup>。不仅如此，通过物联网技术，还可以在农作物成长中对所需的土壤、环境等外部信息进行监控。在运输过程中还可以利用 RFID 技术对每个农作物进行全程的监管，保证安全<sup>[17-18]</sup>。

### (2) 国内研究现状

我国对于物联网研究的开展非常早，就目前的发展来看我国物联网发展水平和技术已经处于世界前列。我国对传感器的研究从 1999 年就已经开始了，直到今天已经具有了从材料到技术到器件到系统到网络一个完整的链条，此链条也在不断的支撑中国物联网的发展。总的来说，我国在物联网的发展中以现有的技术和研发能力已经处于世界领先水平。例如，在北京奥运会举行期间，为保证食物质量，在整个食物的运输过程中通过电子标签对所有食品实现全程跟踪，确保种食物安全，物联网技术在北京奥运会期间得到了大规模的应用<sup>[19]</sup>。

在 2009 年 8 月，“感知中国”的构想被温家宝总理提出，从此中国物联网的发展才真正意义上开始。同年 9 月份，国家工业和信息化部公开提及传感器网络，并将其上升为战略新兴产业，并大力支持物联网的发展。经过这么多年的发展，中国的物联网产业已经非常成熟，物联网也得到了广泛的应用，很多关于物联网的应用也被相继推出，更是涌现出一批优秀的物联网制造企业，比如，华为、小米、格力等。

不仅如此，中国也有一批优秀的企业在建设物联网平台。物联网平台在整个物联网建设中起着承上启下的作用，对下可以对底层设备进行管理、控制等，可以为上层应用提供接口和更多应用的开发和适配，可以简化物联网开发流程<sup>[20]</sup>。目前中国顶尖的物联网云主要有：华为云、阿里云、百度云等。

为了培养物联网优秀人才，我国的很多高校都开设了物联网专业，加强物联网人才的培养，目前已经有 500 多所高校开设了物联网课程，也展示了我国对物联网方面人才培养的决心和国家对物联网产业发展的投入。

随着 5G 技术的发展，5G 技术正在和物联网进行融合。我国的 5G 技术当前处于世界领先地位，这为我国物联网产业的发展提供了更好更快的数据通信解决方案，将 5G 技术应用到物联网中不但可以发挥 5G 的传输速率快、可靠性高、功耗低等特点，而且大幅度的提升了物联网技术，加速了物联网的发展。有人说 5G 就是为物联网而生的，而中国的 5G 技术领先全球，并且中国的 5G 技术正逐步应用在物联网，未来中国的物联网技术和产业将领先全球。

### 1.3

本文在充分学习物联网发展以及理论基础的前提下，以实现家庭内部物联网控制系统为主要目的，基于当前物联网技术进行课题研究，具体工作内容如下：

(1)系统采用 STM32F103 作为主控芯片，将 WIFI 和 LoRa 用作数据无线传输设备，并将数据显示在 TFT 屏幕上。系统需要完成多个数据节点的部署，每个从节点需要完成传感器数据的采集和对控制器件的控制，主节点需要完成数据的接收、下发、上传显示等工作，还在主节点加入了日历显示功能，以提高系统的实用性。

(2)MCU 接口的 TFT 液晶屏的使用对处理器的要求比较高，一般的处理器无法直接驱动，而且数据采集系统用到的处理器如果将 TFT 液晶屏作为显示设备，则会造成处理器资源大量被消耗，IO 口可能不够用，因此将对现有的 MCU 接口的 TFT 屏幕进行改进，利用串口的方式完成屏幕的控制。主要的方式将是外部处理器通过串口与 STM32F4 单片机进行数据交互，STM32F4 单片机根据外部发来的数据指令进行处理，从而完成屏幕控制，为了达到系统可以同时被 4 个串口驱动，引入了 uC/OS-III 操作系统，并采用 DMA 的方式实现串口数据收发。通过改进后，整个系统的显示设备得到了较大的改进，显示的内容也增加很多。

(3)语音呼叫系统：语音呼叫系统可以方便的在室内不同房间完成呼叫，可以用于有老年人的家庭，主要实现方式是通过 WIFI 模块，连接路由器进行组网，理论上一个路由器可以连接 254 个节点，但是由于路由器还要连接其他设备，因此不会达到 254 个，但也完全足够部署在家庭内部。

语音呼叫系统：整个系统以 ESP8266 模块为主要器件，可将 ESP8266 看做一个带有 WIFI 功能的处理器，以每个 ESP8266 作为一个控制节点，将每个节点连接到路由器，以

UDP 的方式进行节点之间数据传输，每个节点装有语音模块和按键控制模块，实现语音播放和控制其它节点功能。根据 IP 地址分配情况，一个路由器可分配 255 个 IP 地址，其中一个用于后台管理，其余 254 个用于连接设备，因此，一台路由器最大支持 254 个节点接入。

(4)移动视频监控平台研究：在家庭内部为了方便快捷的获取视频信息就需要安装摄像头，如果安装固定的摄像头就会带来很多不便，因此为了解决固定摄像头监控方式存在的不足，提高视频监控的范围和适应性，可通过一种可以移动的可以进行无线视频传输的监控平台实现。该平台由车体、云台和无线视频采集系统组成，车体和云台部分使用 STM32F030 单片机作为控制主单元，控制 NRF24L01 和 ESP8266 通过 2.4G 信号和 Internet 遥控车体运动和云台自由旋转；无线视频采集系统使用高通 AR9331 路由芯片搭载 OpenWrt 路由系统，可支持所有 MJPEG 格式的 USB 免驱摄像头，输出的视频状态可以使用手机或者电脑进行实时查询。为了准确获取车体当前运动的位置，在软件设计中优化了定位算法，以实现对车体的实时定位。

## 1.4

本文主要由以下六个章节组成：

第一章：绪论。首先介绍了家庭物联网智能控制系统的选题意义与背景；其次阐述了国内外对物联网技术的研究以及现状；再介绍了本文的主要研究工作；最后列出本文的组织结构。

第二章：家庭物联网智能控制系统原理和设计方案。从本次设计的理论依据到整个设计方案作出了具体的阐述。

第三章：家庭物联网智能控制系统各部分硬件设计。本章对系统与硬件相关的实现做出论述。

第四章：家庭物联网智能控制系统各部分软件设计。本章对系统与软件有关的实现做出论述。

第五章：家庭物联网智能控制系统应用测试。本章主要是论述了对整个系统的测试结果并进行分析。

第六章：总结与展望。对相关的工作进行总结归纳，展望后续物联网发展方向及前景。

## 2.1

整个家庭物联网智能控制系统需要实现以下功能：对家庭内部各个节点的数据采集以及根据数据实现相对应的控制；将采集到的数据保存、上传等；实现对每个节点的语音呼叫功能；实现可移动的视频监控平台。

### 2.1.1

针对目前部分物联网控制系统的弱点，实现一种多功能的家庭物联网控制系统，此系统需要具备多点、多种数据采集功能，并能将这些数据通过无线的方式进行汇总，还需具备控制功能，一旦有数据不在正常范围内还应做出响应，还需具有人机交互界面，为了增加系统的功能丰富性，还加入了语音呼叫功能和可移动的视频监控功能。语音功能可以实现不同节点的语音呼叫，非常适合老年人的家庭，移动视频监控功能，由车体搭载云台，云台装有摄像头，可实现巡视监控，监控视频可以通过手机或者 PC 进行网页查看。

### 2.1.2

在设计过程中，以下技术需要在设计中使用，需要掌握：

（1）单片机技术：整个设计中主控芯片主要有 STM32 和 ESP8266，主要编程软件是 Keil MDK 和 Arduino IDE，要熟练掌握两款处理器硬件外设以及如何使用外设搭建处理器平台，对于 ESP8266 需要额外掌握网络功能的使用方法。

（2）传感器技术：在数据采集和控制过程中需要用到大量的传感器，这些传感器有 IIC、SPI、单总线等不一样的通信接口，需要使用数据手册结合单片机完成对传感器的驱动。

（3）网络技术：在语音呼叫系统中，整个系统使用 UDP 通信方式，因此需要了解网络通信方式，UDP 通信原理以及通信方法，在 ESP8266 中 UDP 方式的使用以及 UDP API 函数的调用。

（4）操作系统移植：在串口驱动 TFT 液晶时使用到了 uC/OS-III 操作系统，在移动监控平台设计中使用到了 OpenWrt 操作系统，对这两款操作系统的移植需要掌握。

（5）无线通信技术：除了 ESP8266 外，在数据传送中还使用到了 LoRa 和 NRF24L01 进行无线数据传输，因此需要对无线通信技术、无线通信方式、模块驱动方式掌握。

(6) 硬件电路设计：系统实现需要硬件电路的支持，硬件电路的设计需要有模拟电子技术、数字电子技术、电路等理论知识的支持，还必须会使用硬件电路设计软件。

## 2.2

整个系统的实现，需要现有的物联网技术作为支撑，以及对各个系统所能实现的原理以及可能性分析，自己对物联网技术的了解，以及需要自身有扎实的理论知识，还需要有实践经验和快速查阅资料的能力等。

### 2.2.1

数据之间的无线传输是实现多节点远距离通信的关键。数据采集控制系统选用 LoRa 和 ESP8266 作为节点之间实现无线数据传输的介质，LoRa 即（Long Range Radio）远距离无线电，属于线性调频扩频的调制技术，并在 2012 年将此技术封装在一颗小小的芯片中，并基于 LoRa 技术开发了一整套的远距离通信方案，也是从这一年起，LoRa 开始了芯片化之路。一般在城市或者有大量建筑物阻挡的前提下，传输距离可以在 2 到 5 千米，如果在开阔的场地进行数据传输，传输距离可达 15 千米左右，并且 LoRa 可以进行组网通信，一个 LoRa 网关，可以被成千上万其他 LoRa 节点连接，本设计也是采用这种方式，可以实现多个 LoRa 节点之间相互数据通信<sup>[22]</sup>。传输速率可在几百到几万 bps 之间，传输距离越长，传输速率越慢，但是应用在室内，完全不用考虑距离会影响传输速率的问题。WIFI 模块采用的是乐鑫的 ESP8266，该模块内置 32 位 CPU，运行速度可达 160MHz，低功耗，带有 WIFI 天线，支持 WIFI 协议，使用简单方便。

### 2.2.2 TFT

在嵌入式系统中，显示器件作为人机交互界面使用十分广泛，但是对于显示性能较好、分辨率较高的 TFT 液晶屏而言，大部分低性能的单片机无法直接驱动，即使可以驱动也会消耗单片机大量的资源，使用起来具有较大的局限性，并不适合在低端单片机中推广使用。因此可以考虑将 TFT 液晶屏和处理器相结合，设计一种低端处理器可以使用串口完成屏幕驱动的方式<sup>[23]</sup>。

TFT 液晶屏大都是 RGB 接口，对于大部分被 STM32 处理器而言，并没有 RGB 接口，如果想要驱动需要将 RGB 接口转换为 MCU 接口，转换之后，使用 STM32 的 FSMC 功能进行屏幕驱动，但是在成功驱动后，会消耗大量 CPU 资源，再结合数据采集监控系统使用的屏幕也需要进行改进，因此设计了一款通过外部串口控制 TFT 液晶屏的方案，实现原理是外部串口与屏幕带有的处理器进行通信，通过通信协议以及底层固定的 API 函数，完成外部串



口与屏幕的交互。这样以来只要有串口的单片机就可以直接进行对屏幕进行控制，就算没有串口也可以通过 IO 口模拟串口进行通信，大大简化了对 TFT 屏幕控制的流程。

### 2.2.3

ESP8266 体积非常小，而且支持站点模式、热点模式和站点+热点模式，支持 802.11b/g/n WIFI 协议。不仅如此，它还具有单片机的功能，内置 10 位高精度 ADC，支持一般单片机具有的外设，如串口、IIC、SPI、PWM 等。时钟频率可以达到 160MHz，比一般的 STM32F1 要高出 2 倍多。支持低功耗模式、还提供可以二次开发的 SDK 包，也支持 AT 指令，可实现快速开发。

语音呼叫采用的是 WIFI 模块，通过 UDP 的方式进行数据传输。对于 UDP 通信，UDP 作为网络协议的一种，每个数据报都有完整的源地址和目标地址，与 TCP 相比，使用 UDP 进行数据传输效率较高，在不需要非常安全的情况下，UDP 数据传输是一个很好的选择。

在 Arduino IDE 中提供了对 ESP8266 的支持包，并在支持包中封装了大量的 API 函数，其中就包括 UDP 通信的 API，可以通过简单的 API 调用，实现 UDP 数据通信。

对于 ESP8266 如何连接到路由器，ESP8266 支持智能配网，具体可分为两种联网方式：（1）使用手机连接到 ESP8266 需要连接的网络，然后 ESP8266 通过调用相应的 API 函数，进入联网模式，手机 APP 端就会进行信号捕捉，最后完成网络的连接；（2）ESP8266 可以通过网页的方式进行联网，在 ESP8266 程序内可以写一段网页代码，在联网时 ESP8266 首先会进入 AP 模式，然后释放 WIFI 热点，通过手机登陆对应的 IP 地址，进入网页，然后在网页上输入 WIFI 名和 WIFI 密码，点击连接即可联网。

### 2.2.4

随着无线数据传输技术和传感器技术的不断革新，无线视频监控技术以其部署简单、易于组网、运行稳定等特点，已被广泛的应用在家庭、工业、农业等诸多领域。虽然无线视频监控技术在视频监控领域展现出巨大优势，但是其移动性能较差，一般是固定摄像头监控某个特定的区域，监控范围较小，难以满足一些复杂区域的监控需求<sup>[24]</sup>。

NRF24L01 模块是一个可以实现无线数据传输的模块，利用 NRF24L01 模块实现车体和遥控之间进行数据通信，通过数据传输实现了对车体的控制和对云台的控制，实现了小车在室内可全方位的移动。在配合 ESP8266 模块，也可以实现更远距离的遥控。

OpenWrt 作为一款可用于嵌入式的 Linux 操作系统，它拥有非常高性能的网络组件并具有强大的可扩展性，在路由器中经常会使用 OpenWrt。本系统实现视频访问的方式就是

在 AR9331 模块中移植 OpenWrt 操作系统，实现一个路由系统，通过此系统实现小车的视频监控。

为了使车体在运行过程中不会走偏，引入了 PID 速度调节，保证小车可以直线行走。车体定位的实现主要是依靠安装在车轮的编码器和车体的陀螺仪，然后经过一系列的算法最终可以准确的计算出小车当前的位置。

## 2.3

系统设计方案主要考虑的因素有硬件和软件。硬件方面，需要考虑所需器件的可行性，是否可满足设计要求，性价比等问题。软件方面，需要考虑软件实现的复杂度，需要明确自己在软件方面的不足，进行理论知识学习，增强实践能力。

### 2.3.1

硬件设计方案：系统控制器使用 STM32F103 单片机，属于 Cortex-M3 内核，频率可达 72MHz，可满足系统对单片机的要求。对于传感器的选择，为了适应多种数据采集监控的要求，硬件方面预留了多种传感器接口，增加了传感器的兼容性，可方便根据每个家庭不同需求进行调整。对于从节点上安装的控制器件也要考虑多方兼容性，有可能被控器件是大功率，有可能需要控制电压型器件、有可能需要控制电流型器件等。无线数据传输器件选用的是 LoRa 和 ESP8266，LoRa 负责室内各个节点与主节点之间的通信，ESP8266 负责将数据上传到物联网云平台。

软件设计方案：使用 STM32F103 对各种传感器的控制，这需要用到 STM32 的一些外设，如 IIC、SPI、IO 等，通过硬件或者软件模拟协议的方式完成对传感器的控制，还需要对采集到的数据通过无线模块进行发送和接收。使用 ESP8266 除了需要掌握基本的外设控制外，还需要掌握一些关于网络的 API 函数。

整个方案的实施，需要在室内部署多个节点，包括多个从节点，和一个主节点。每个从节点都要承担数据的上传、接收和控制等任务，主节点负责数据的汇总、保存、上传至云平台等任务。整个系统将由大量的控制节点组成一个完整的、可覆盖整个室内的无线数据监控通信网络。

### 2.3.2 TFT

一般的 TFT 屏幕为 MCU 接口，对于高性能的 STM32 单片机具有 FSMC 功能，可以用来驱动 TFT 液晶屏，但是对于不带有 FSMC 接口或其他低性能的单片机而言，则无法直接驱动。因此，为了满足多种型号单片机对 TFT 液晶屏的使用，也为了满足数据采集控制系统

人机交互更加丰富，需要将 MCU 接口的 TFT 液晶显示屏进行改进。主要设计思路是将 STM32F429 处理器做中间桥梁，底层用于驱动 TFT 液晶屏，上层通过串口与外界进行信息交换，通过编写 API 函数，实现外部处理器对 TFT 液晶屏的控制。

STM32F429 有 8 个串口，使用其中的 4 个串口，可以实现 4 个外部处理器同时对屏幕的控制。实现同时对 4 个串口进行控制，还需引入操作系统，本系统引入了一种轻量级的嵌入式实时操作系统 uC/OS-III，对 4 个串口进行任务管理，可实现 4 个串口同时被外界处理器访问控制，增加了系统的实时性。

串口之间的通信，需要同步波特率，因此系统必须支持多种常用波特率，而且必须做到波特率可调。

### 2.3.3

要想进行多个点之间的 UDP 通信，需要将多个点连接到同一路由器上，组成一个网络，还需要知道所要通信方的 IP 地址和端口号。但是，在设备连接到路由器后，路由器会自动分配一个 IP 地址给设备，但是分配后的设备 IP 地址，我们并不知道，只能通过路由器后台进行查找，而且在每次断开连接后 IP 地址很可能会改变，因此如果 ESP8266 连接到路由器后不知道 IP 地址，将无法进行通信，如果查路由器后台，那么多 ESP8266 同时联机将很难分辨而且还耗费时间，因此，在连接前，需要给 ESP8266 固定 IP 地址，并做好记录，在后续进行节点通信的时候可以很方便的进行。在固定 IP 地址后，然后在给每个节点设置一个不一样的端口号，这样在调用 UDP 通信接口，即可完成 UDP 通信。

在理解 UDP 通信模式后，设计方案变得更加简单，只需要将可以联网的模块接入到路由器，并配置为 UDP 通信方式，并给每一个接入的模块分配好 IP 地址和对用的端口号后就可以相互通信了。

对于传输不同的数据，就要靠节点上连接的按键实现了，按键的实现使用的是模拟量采集的方式，不同按键按下，ESP8266 采集到的电压将不同，通过判断电压值，从而可发送不同的键值，可以实现多按键共用一个端口。

语音模块音频可以自己录制也可以通过语音转文字进行生成，语音支持 MP3 和 WAV 格式音频，语音录入也比较方便，将音频文件准备好，然后将语音模块通过 USB 线接入电脑，电脑端会虚拟出一个 U 盘，然后将语音文件复制进去即可。

### 2.3.4

针对传统固定摄像头监控方式的不足，本文将无线视频监控与移动平台相结合，设计并实现了一种可移动的无线视频监控平台。平台采用 STM32F030 处理器芯片处理 NRF24L01

的 2.4G 信号和 ESP8266 的 Internet 数据，实现两种方式的移动控制；使用 AR9331 路由芯片运行 OpenWrt 系统，硬件搭载 USB 摄像头实现视频信号采集，通过 WiFi 将视频信号传输到手机或者 PC 端以网页的方式进行显示；通过 PID 算法对车体运动过程中产生的方向偏差进行修正、并用坐标定位算法定位和获取车体当前的位置坐标、运行距离和当前距出发点的直线距离。

为了增加移动平台的适应性，设计了两种遥控方案，一是通过 NRF24L01 实现近距离遥控车体运动，另一种方式是通过 ESP8266 实现车体远距离的遥控，只要有网络的环境下，无论在哪里，都可以实现对车体的遥控。

对于视频监控方面，经过多方查找方案，最终选用 AR9331 搭载 Openwrt 路由方案进行解决。主要是因为：1、入门难度低，可轻松的刷入 Openwrt 路由系统；2、资料较多，在系统移植过程中，如果遇到什么问题，可以轻松查找资料进行解决；3、支持多种摄像头，支持网页查看视频，降低开发成本。因此，Openwrt 是一个很好的选择。

对于车体的定位，主要是利用编码器记录小车运行的里程，使用陀螺仪记录小车的方向，然后经过一定的数学算法，可以计算出车体当前的位置信息。使用这种方法可以较为准确的定位到小车运动期间的位置信息。

## 2.4

本章主要对家庭物联网智能控制系统的功能需求做了分析，并给出设计原理，提出整个系统设计所需要使用的技术，以及对系统方案做出论述。为后续硬件方案设计和软件方案设计做好准备。

整个系统的硬件设计将从如下几个方面进行设计：数据采集监控系统硬件设计；TFT液晶显示系统硬件设计；语音呼叫系统硬件设计；移动视频监控平台硬件设计。各部分硬件系统又包括具体的子模块、子系统，将这些子模块、子系统最终通过处理器进行结合，逐渐形成一个完整的系统。

### 3.1

数据采集监控系统硬件部分主要有从节点和主节点组成，节点通过处理器挂载一些传感器和控制器组成。主节点通过 STM32F103 处理器外部挂载了 WIFI 模块、FLASH 存储模块、时间输出模块、LoRa 模块、人机交互模块等构成；从节点通过 STM32F103 处理器外部挂载了 LoRa 模块、传感器模块、控制模块等构成。

#### 3.1.1

数据采集监控系统采用的是一对多和多对一的模式，即一个主节点可以同时控制多个从节点，一个从节点可以控制多个传感器，从节点上挂载的多个传感器采集的数据回传到这个从节点，每个从节点的数据最终回传到主节点。以这种方式，最终可实现大规模的数据节点部署<sup>[25-26]</sup>。

系统拓扑结构图如图 3.1 所示。图中可以看出整个系统分为云服务器、主控制器节点、从节点三部分，其中在硬件部分上，主控制器节点和连接到云端的设备共同组成主节点。无论是从节点还是主节点，核心部件都是一颗 STM32F103 处理器，用来控制传感器采集信息、数据指令的下发、控制数据无线传输、驱动显示器进行显示等。在从节点和主节点之间通过 LoRa 进行组网，通过配置 LoRa 模式，实现广播监听，通过 LoRa 数据传输协议和自定义格式数据收发协议，实现从节点数据准确上传到主节点，以及主节点将控制指令准确的下发到从节点。主节点外置 FLASH 存储芯片，可将重要的配置信息和数据信息进行保存。对于主控制器节点和云服务器端通过 ESP8266 进行数据交互，将来自从节点的数据通过串口的方式发送至 ESP8266，ESP8266 再将数据通过网络发送到远程云平台。

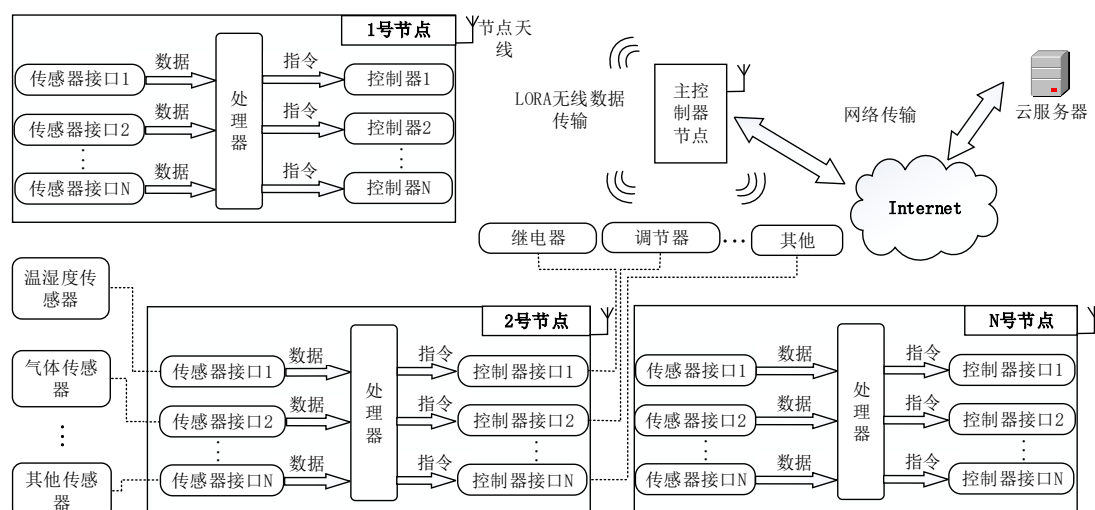


图 3.1 系统拓扑结构

### 3.1.2

数据采集系统节点硬件框图如图 3.2。从图中可以看出主控制器节点和从节点在硬件设计上具有相同部分，如 JTAG 接口用于程序下载仿真，USB 串口可用于调试也可用于与外围器件进行通信，电源电路用来给整个系统供电，其余不同的为主控制器节点外置 FLASH 芯片，用于数据存储，具有语音输出功能，用于信息提示，具有时钟芯片，可获取离线时间信息，外置的 ESP8266 也会将数据通过网络传输到网络云平台；从节点主要是外置多种传感器和控制器<sup>[27]</sup>。

**处理器：**主节点和从节点使用的处理器都是内核为 Cortex-M3 的 STM32F103，处理器运行时钟频率为 72MHz，在外设接口方面具有硬件 IIC、硬件 SPI、UART、ADC、PWM、DMA 等功能，还具有丰富的 GPIO，满足系统设计需求。

**电源：**整个数据采集控制系统，需要用到 3.3 伏和 5 伏电压，因此使用 12 伏 2A 的电源作为输入，然后通过 LM2596 稳压芯片，将电压转变到 5 伏，然后在使用 AMS1117 将 5 伏电压转换为 3.3 伏，得到系统所使用到的两种电压。

**数据传输模块：**整个系统使用到 LoRa 和 ESP8266 两种无线数据传输模块。LoRa 模块传输距离远、传输速率快、将 LoRa 部署在室内，在传输距离上完全满足需求，不用担心室内墙壁造成数据传输失败等问题。ESP8266 内置 IEEE802.11b.g.n 网络协议，在 Arduino IDE 编译器中可以添加 ESP8266 支持包，方便实现网络功能。

**传感器与控制器：**系统已经预留多种传感器接口，方便根据不同家庭需求进行增加和减少，对于一些主要的传感器已经提供程序支持包，可以即插即用。比如，可在卧室按照温湿度传感器，实时监控卧室当期温度和湿度；可以在厨房部署气敏传感器；可以将粉尘传感器、PM2.5 等与环境相关的传感器放在门外。在系统设计时选择了较为典型的传感器

进行测试，温湿度传感器选用的是 AM2302、气敏传感器选用的是 MQ-4，控制器件选择的是继电器和高速光耦。

显示设备：显示设备最初使用的是 LCD 液晶屏，此款显示屏，显示内容有限，在对 TFT 液晶屏使用串口驱动成功后，将替换最初使用的 LCD 液晶显示屏，作为系统的显示设备。

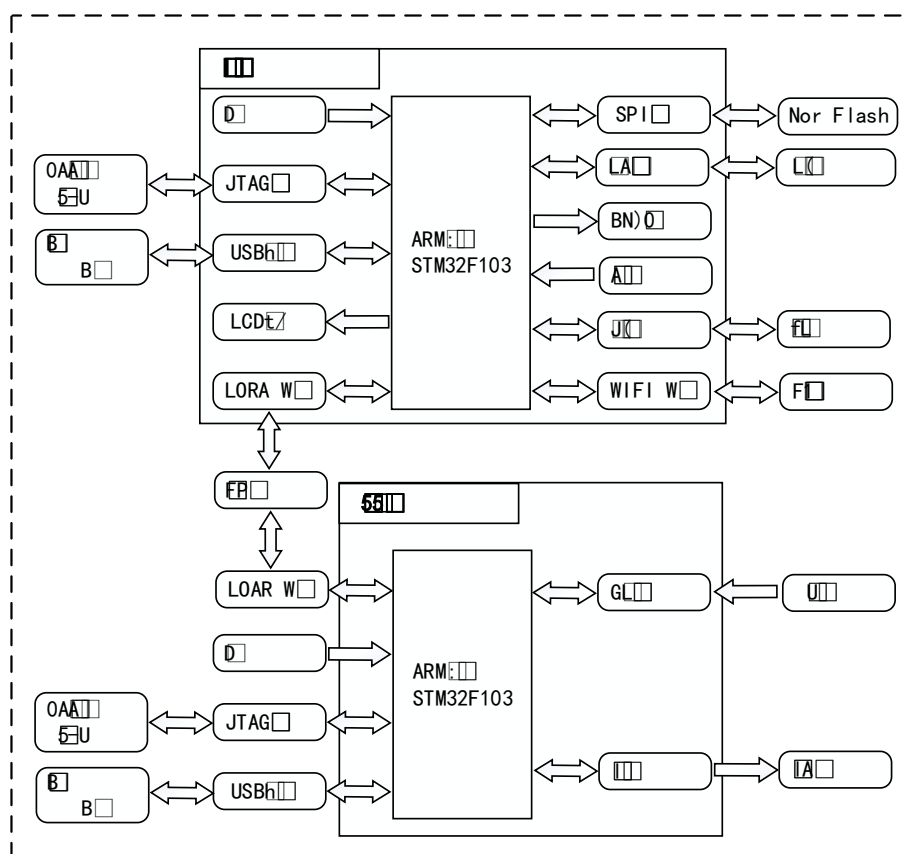


图 3.2 数据采集节点硬件框图

## 3.2 TFT

串口驱动 TFT 液晶屏硬件组成较为简单，有两部分组成：STM32F429 单片机系统和 MCU 接口的 TFT 液晶屏。整个系统的硬件框图如图 3.3 所示。在硬件部分需要设计系统供电部分、STM32F429 处理器最小系统以及 TFT 液晶屏接口、将 RGB 接口的 TFT 液晶屏转换成 MCU 接口的 TFT 液晶屏。对于系统供电已经在数据采集节点电源硬件设计中已经形成模块化的电源，可以直接对此系统进行供电，无需在对电源进行硬件设计<sup>[28-29]</sup>。



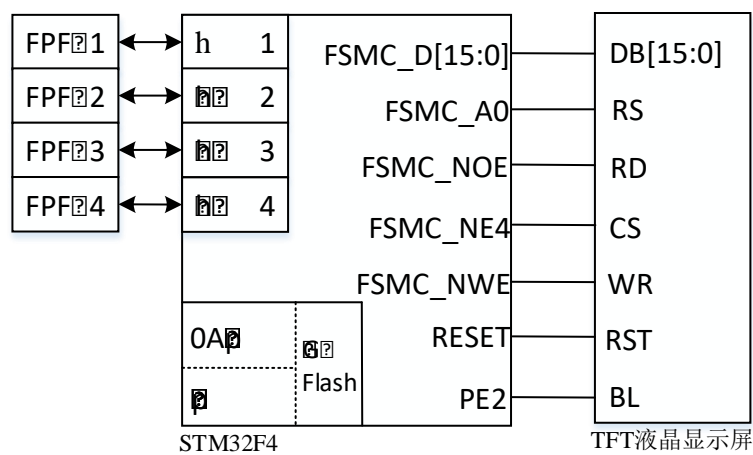


图 3.3 串口驱动 TFT 液晶屏硬件框图

3.2.1 STM32F429

图 3.4 为 STM32F429 硬件电路设计框图。从图中可以看出整个硬件电路由电源电路、程序下载电路、模式选择电路、复位电路、晶振电路、外围必要的阻容、串口通信接口和连接 TFT 液晶屏的 MCU 接口。

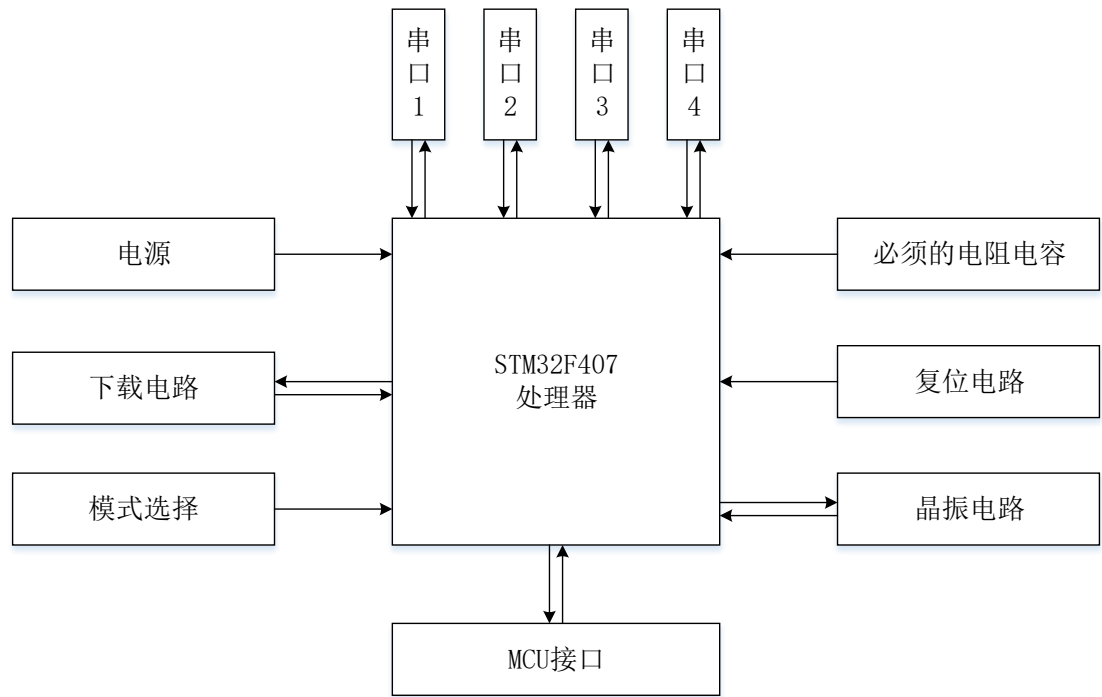


图 3.4 STM32F429 硬件框图

电源已经形成了模块化的系统，可以用前面设计好的电路进行供电；程序下载使用 ST-Link 进行下载，在系统供电的情况下，只需要 SWCLK、SWDIO、GND 三根线即可完成程序的下载；STM32F429 有三种启动模式选择，一是选择 FLASH 作为启动空间、二是选择系统存储器作为启动空间、三是选择 SRAM 作为启动空间，具体的选择方式可以通过控制外

部 BOOT 引脚电平进行；复位电路硬件是通过外部按键完成的，按键按下，给复位管脚一段时间的低电平，让单片机完成复位，通常为了使单片机稳定复位，会在按键两端并联一个电容；在系统中晶振电路可以给整个单片机系统提供稳定的时钟源，STM32F429 也可以使用内部晶振，但是内部晶振对比外部晶振在稳定性上较差。

### 3.2.2 MCU TFT

MCU 接口 TFT 液晶屏硬件框图如图 3.5 所示。在硬件设计上，RGB 接口转 MCU 接口相对简单，将 RGB 的数据接口通过小阻值的电阻串联作为 MCU 接口的数据输出口，然后在根据接口所需的不同电压进行电压转换即可。对于触摸屏，采用的是多点触控，触摸芯片驱动方式为 IIC，实现功能主要依靠软件<sup>[30-31]</sup>。

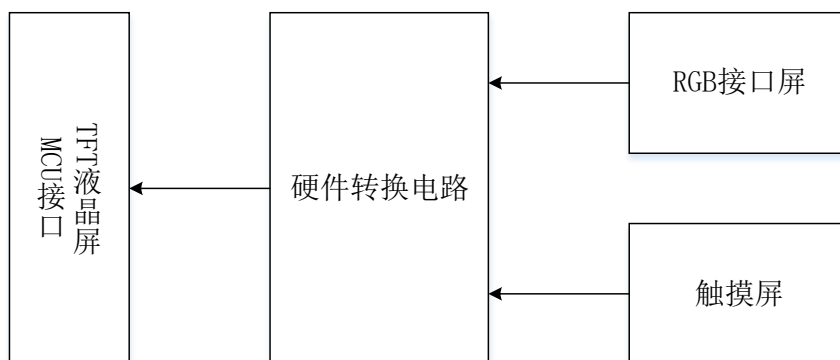


图 3.5 MCU 接口 TFT 液晶屏硬件框图

## 3.3

语音呼叫系统硬件设计相对来说较为简单，硬件外设也使用的相对较少。图 3.6 描述的为语音呼叫系统的硬件框图。图中，可以看出整个硬件框图由路由器、ESP8266 模块、语音模块和按键模块组成，路由器无需硬件设计，家庭使用的路由器即可使用；ESP8266 模块需要进行最小系统的设计，一个 ESP8266 芯片如果不做最小系统的设计则不能进行使用，最小系统电路包括模块引脚所需的上下拉电阻、下载电路、供电电路等。语音模块和 ESP8266 为串口通信，硬件方面将语音模块与 ESP8266 串口连接即可。按键模块需要用到电阻分压，不同按键串联的电阻不一样，因此不同按键按下所分得的电压也不一样，可以通过检查电压值判断对应按键是否按下。

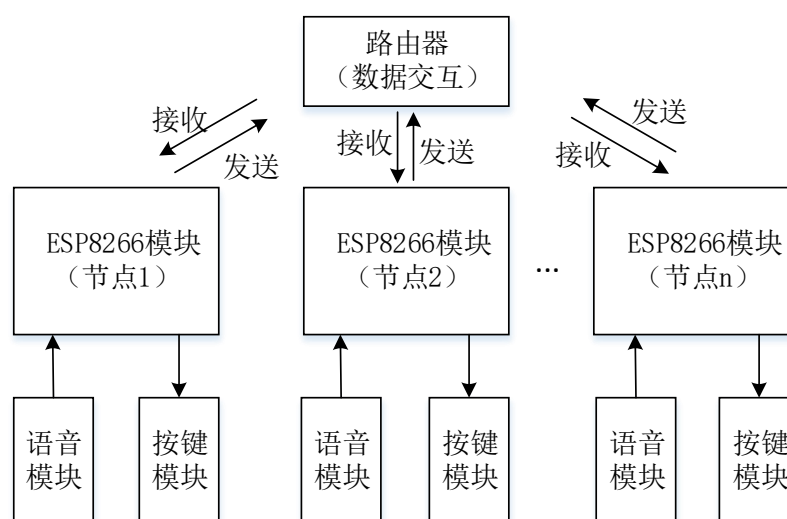


图 3.6 语音呼叫系统硬件框图

### 3.4

#### 3.4.1

移动平台的硬件部分由下面部分构成：遥控部分和车体部分。遥控部分可以通过自制遥控器和手机进行遥控。遥控器主要是以 STM32F030 为核心，外接控制摇杆、按键和 OLED 屏幕，通过 NRF24L01 与车体部分的 NRF24L01 进行通信实现对车体的控制；手机端主要是用 APP 通过网络实现与车体端的 ESP8266 通信，实现对车体的控制。车体部分硬件主要以 STM32F030 为核心，外接用于通信的 NRF24L01 和 ESP8266、用于车体移动的电机驱动芯片、用于带动摄像头旋转的舵机云台、用于测量信息的传感器、与定位有关的陀螺仪和编码器、用于显示信息的 OLED 屏幕<sup>[32-33]</sup>。图 3.7 为移动平台的硬件框图。

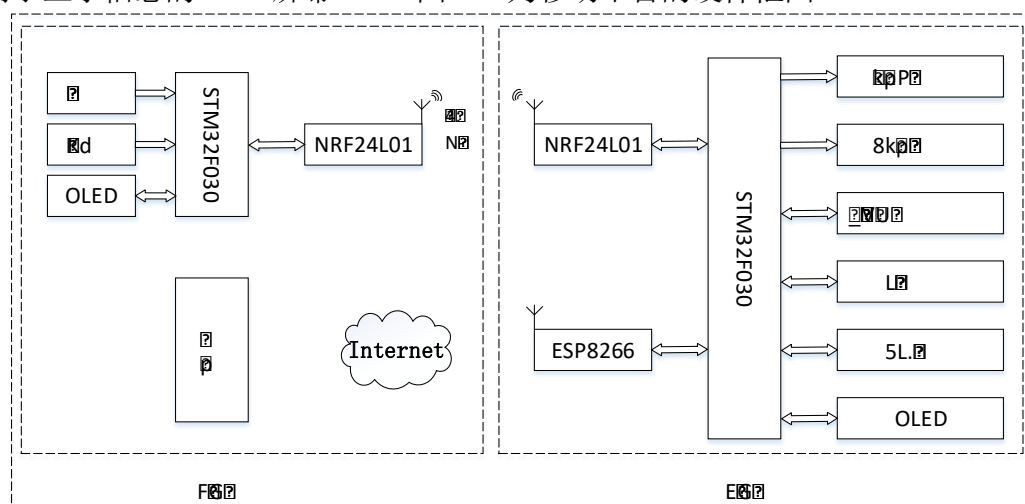


图 3.7 移动平台硬件框图

### 3.4.2

图 3.8 表示的为无线视频采集与监测端的硬件框图。主控 AR9331 路由芯片是一款低功耗、高性能的无线路由芯片，内部集成 CPU 和 WiFi 基带，最高主频高达 400MHz，最大无线数据传输速率可达 150Mbps。外围配备 8M FLASH 用于存储 OpenWrt 系统和 64M 的 DDR2 内存<sup>[34]</sup>。集成 USB2.0 接口，可外接 USB 免驱 MJPEG 格式输出的摄像头。手机或 PC 可以连接 AR9331 发出的 WiFi 信号，通过网络进行视频传输。也可以通过租赁服务器，申请动态域名实现远程互联网视频监控。预留网络接口用于系统移植。

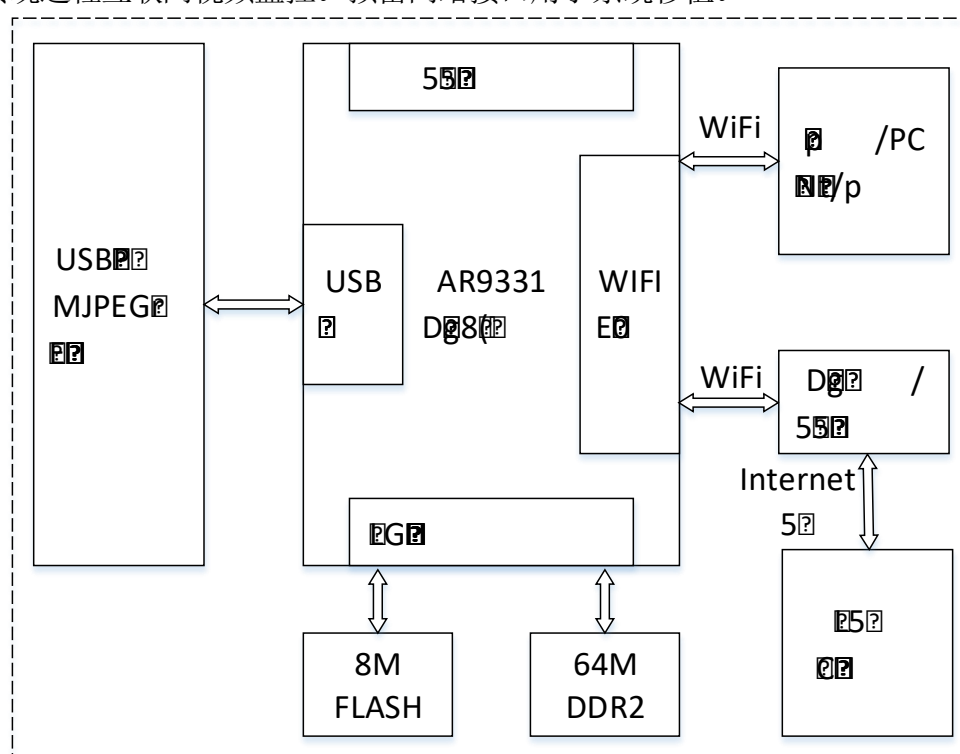


图 3.8 无线视频监控硬件框图

## 3.5

本章对整个系统的硬件部分做了分析和阐述。首先是室内数据采集监控系统的硬件介绍，给出硬件设计框图，详细说明了各个模块的硬件组成，以及设计方法；然后是对显示设备的改进硬件部分做出了详细的与分析，首先给出了整个系统的硬件框图，并对硬件部分做出了介绍，然后对不同型号的屏幕驱动做出了具体分析；然后是对语音呼叫系统硬件部分做出介绍，此硬件部分相对简单，给出了硬件设计框图；最后是对移动监控平台硬件部分做出介绍与分析，给出了整个系统的硬件设计框图，并做出分析，对移动平台硬件、无线视频监控硬件给出了具体分析。

本章主要对整个系统的软件实现进行描述，软件部分主要通过以下几个方面实现：室内数据采集监控系统软件实现；多通道串口驱动 TFT 液晶显示软件实现；语音呼叫系统语音软件实现；可移动无线视频监控平台实现。通过对各个模块进行软件设计，进而实现软件功能，最终实现整个家庭物联网智能控制系统的软件设计。

## 4.1

室内数据采集系统在软件方面主要需要对主节点、从节点和网络数据收发进行设计。在主节点和从节点软件设计上主要使用 Keil5 MKD 进行编程开发设计，对于网络功能的实现主要使用 Arduino IDE 编程，在结合网络云平台实现远程数据采集显示端的搭建。在调试方面，Keil5 MKD 自带调试仿真功能，在软件设计过程中使用仿真工具节约了软件开发时间<sup>[35]</sup>。

图 4.1 为室内数据采集系统的软件流程图。从图中可以看出，当系统上电，首先进行一系列的软件初始化，完成相关的软件、系统配置，然后网络端将进入联网模式，寻找 WIFI 信号，如果找到的 WIFI 信息之前已经连接过，则直接连接，如果所有 WIFI 信息没有连接过，则系统进入自动配网模式，可以使用手机完成网络连接。然后在主节点和从节点之间为了确保通信正常，会首先进行通信测试，在测试通信正常后，从节点就开始将从各个传感器采集的数据开始上传，主节点在接收到数据后进行分析判断，如果分析后的数据不在设定的范围内，主节点将下发控制指令到对应的从节点，从节点接收控制指令完成控制。主节点也会将此次异常的数据进行存储，方便用户查看。主节点也会将接收到的数据上传到云服务器，方便用户在远程进行监控，也会将数据显示在本地的屏幕上。通过进行软件设计最后整个系统可以实现从节点将所属传感器采集的数据上传到主节点，主节点根据接收的数据下发控制指令，并将数据在本地显示端显示，也会将接收到的数据上传到远程的物联网云平台。

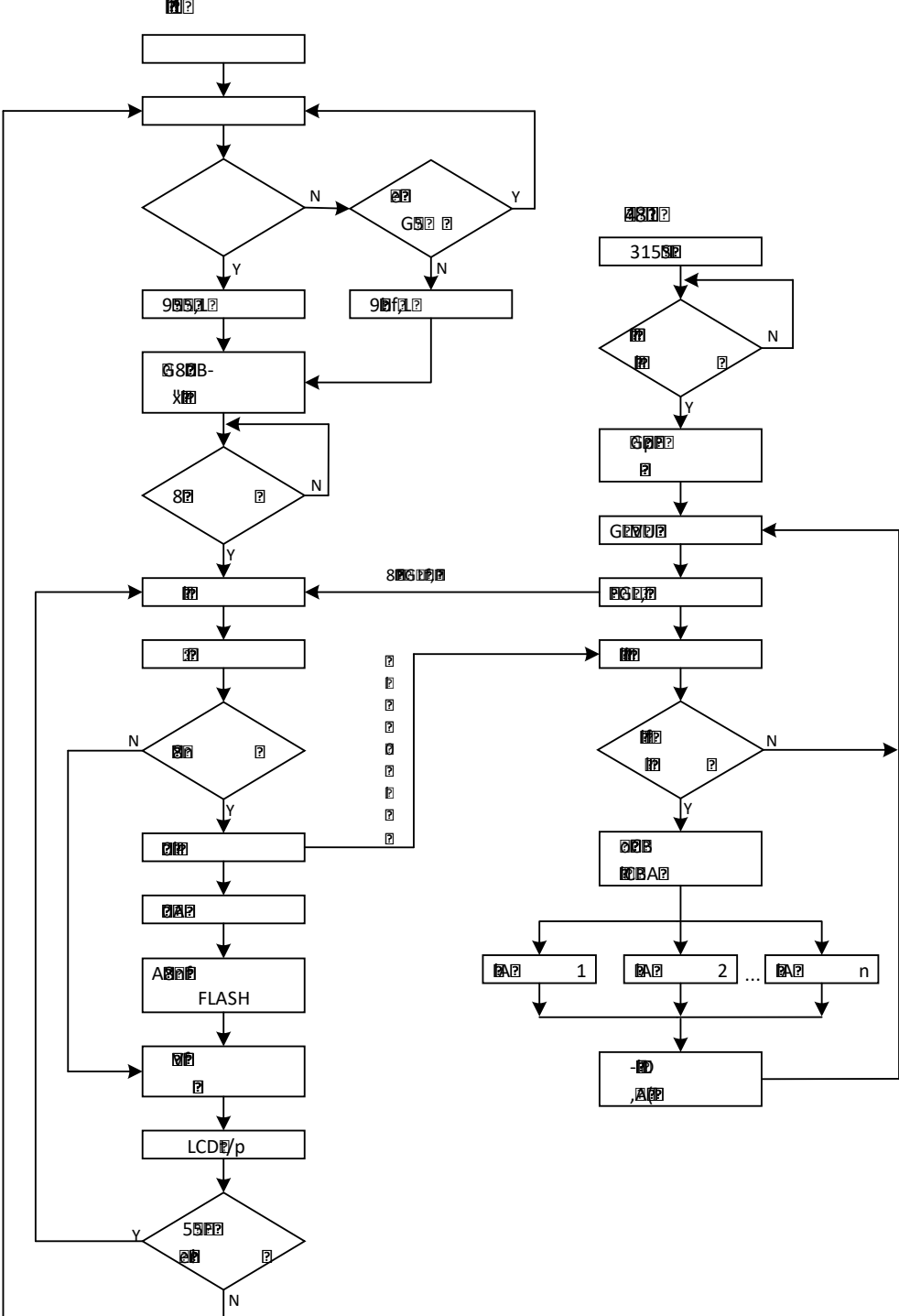


图 4.1 系统软件流程图

### 4.1.1

根据 LoRa 设置模式，将 LoRa 模块地址设置为 0xFFFF，可以使得 LoRa 模块工作在广播监听模式，广播监听模式下地址为 0xFFFF 的 LoRa 模块可以与任何与他同信道和同速率的其他 LoRa 模块相互通信。因此可将主节点地址设置为 0xFFFF，使其工作在广播监听模式，这样其余的从节点就可以和主节点之间进行相互通信。图 4.2 为整个系统所有节点的对应关系，图中 A 点为主节点，其余为从节点<sup>[36]</sup>。

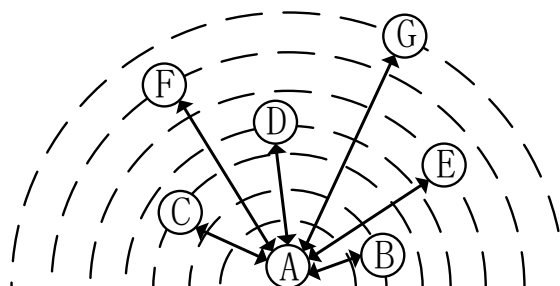
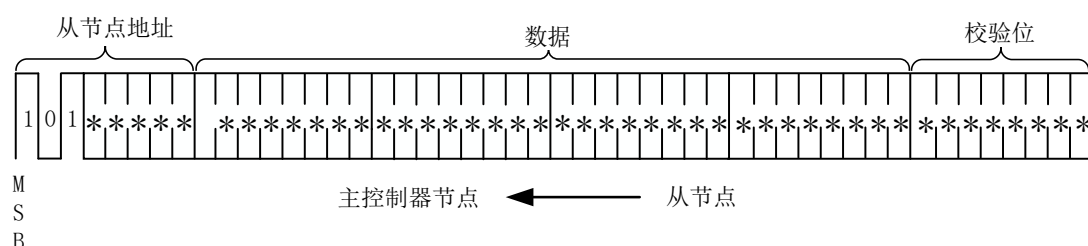
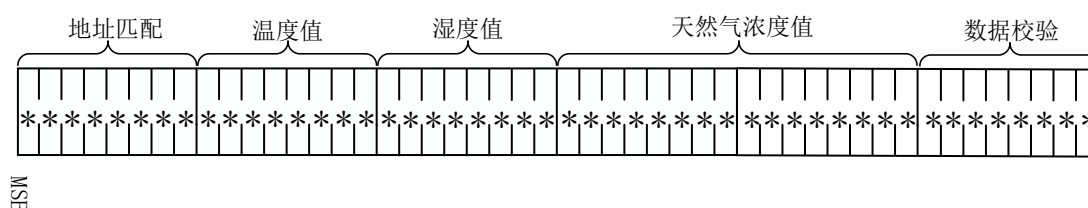


图 4.2 主从节点分布图

LoRa 工作在广播监听模式下，但是当所有从节点将数据上报到主节点时，并不能判断是哪一个是哪个从节点的数据，主节点也不能准确下发指令到某个从节点执行控制指令。因此为了解决上述在多节点通信中存在的不足，系统在原有的数据通信基础上增加了新的数据通信格式，解决了上述存在的不足。图 4.3 的数据格式图为新增加的数据传输格式。从图中可以看出，数据传输格式中包括从节点地址和主控制器节点地址，在原本的通信中是不包括这两个地址的，为了区分不同的节点，在新的数据通信中加入了这两个地址，目的就是让主节点区分不同的从节点发来的数据并作出对各个从节点精准的控制<sup>[37-38]</sup>。

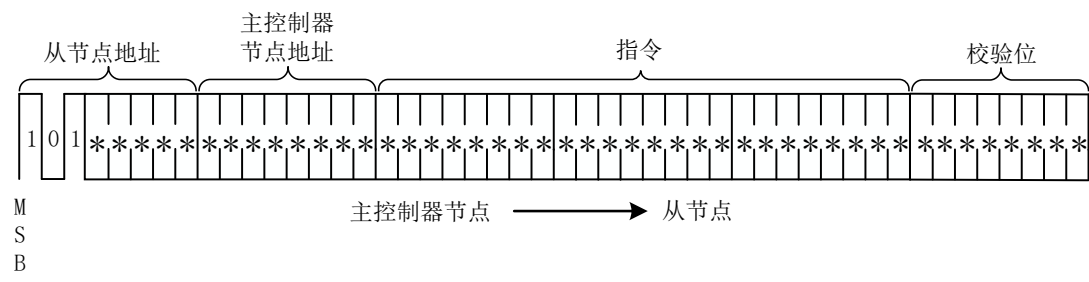


(a) 主节点接收数据

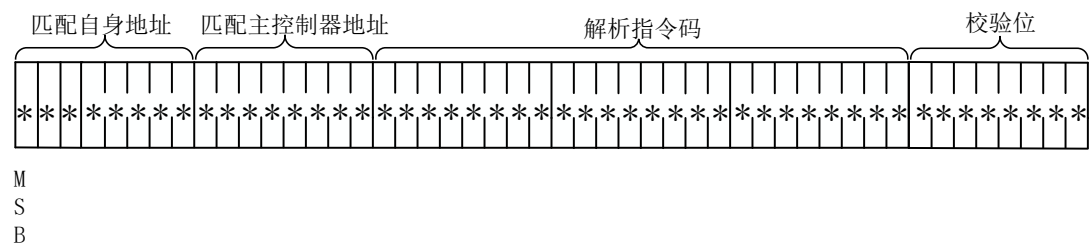


(b) 主节点解析数据





(c)主节点下发控制指令



(d)从节点解析指令

图 4.3 数据传输格式

图 4.3（a）和图 4.3（b）为主节点对从节点发来的数据进行接收和根据数据指令格式对数据进行解析，发来的数据格式如表 4.1 所示，整个数据长度为 48 位，从低位开始前 8 位为检验位，判断数据是否正确，然后是 16 位的天然气浓度值，8 位的湿度值和 7 位温度值加 1 位温度正负标志位，最后是 8 位是接收的从节点地址。图 4.3（c）和图 4.3（d）为主节点发送控制指令到从节点格式和从节点进行数据解析格式。通过对从节点设置不同的地址编号，实现了控制不同的从节点功能，通过这个通信格式，理论上可以控制 255 个设备，如果使用节点超过 255 个设备，可以通过增加主从节点地址位数的方式增加设备节点，最大可以增加到 65534 个设备。地址与设备对应编号如表 4.2 所示。

表 4.1 数据格式

二进制位(bit)	意义
40-47	从节点地址
39	温度正负标志位
32-38	温度
24-31	湿度
8-23	天然气浓度
0-7	校验位

表 4.2 地址与设备对应编号

设备地址编码	受控设备
0x0000	0 号受控设备
0x0001	1 号受控设备
.....	.....
0xFFFFE	65534 号受控设备

LoRa 模块本身带有测忙机制, 在发送数据前, 首先会检测接收节点是否处于忙状态, 如果忙则进行等待。但是 LoRa 模块存在这种机制并不能确保大量节点在收发数据的时候不产生冲突, 因此为了尽量降低或者避免冲突, 在数据发送时让节点自动产生一个 0~50 毫秒的随机延时, 延时后再进行数据发送。实现原理是根据假如设置 LoRa 模块传输速率为 19200bps, 将一个 64bit 的数据传输完需要 3.33ms, 因此可以再 0 至 50 之间设置随机产生数据间隔为 5 进行延时, 即主节点对每个从节点下发数据间隔为 5ms。0 到 50 这个范围可以根据节点数量进行动态调整。

#### 4.1.2

系统预留多种传感器和控制器接口, 可根据用户需求进行调整。为了测试系统可行性, 程序设计阶段使用了较为典型的传感器和控制器进行测试, 有 AM2302、MQ-4、继电器和光耦等器件。再处理器对传感器进行数据采集时, 使用定时器中断的方式, 让处理器不断循环采集处理数据, 然后汇总并将数据上传至主节点。通过数据接收中断, 不断检查主节点发送来的指令, 一旦有指令进来, 立即进入中断, 完成控制<sup>[39-40]</sup>。

对于云上设备使用的是百度物联网云平台。再使用前首先需要在百度物联网云平台创建项目, 进行相关配置, 云平台采用 MQTT 通信协议, 再配置好项目后, 可以先用 MQTT 在线测试工具进行测试, 通信正常后可进行操作。对 ESP8266 的编程, 使用 Arduino IDE, 编译器提供了 ESP8266 的 MQTT 协议包, 只需要对相关的 API 进行调用可以轻松接入云平台。数据上传成功后就可以再云平台上创建数据仪表盘, 以数据可视化的方式显示, 然后就可以将界面发布出去, 会生成一个 Url, 通过这个链接就可实现远程数据查看。

## 4.2 TFT

在驱动 TFT 液晶屏中, 为了实现可多个通道同时使用, 使用 uC/OS-III 操作系统, 实现对屏幕的控制。uC/OS 操作系统易于移植到多种处理器上, 而且也可以根据设计需求进行大小裁剪。在 uC/OS-III 中可以在 RAM 空间足够的情况下, 可创建大量的任务。本系统在 uC/OS 的基础上, 创建多个实时任务, 完成对屏幕的驱动<sup>[41-43]</sup>。

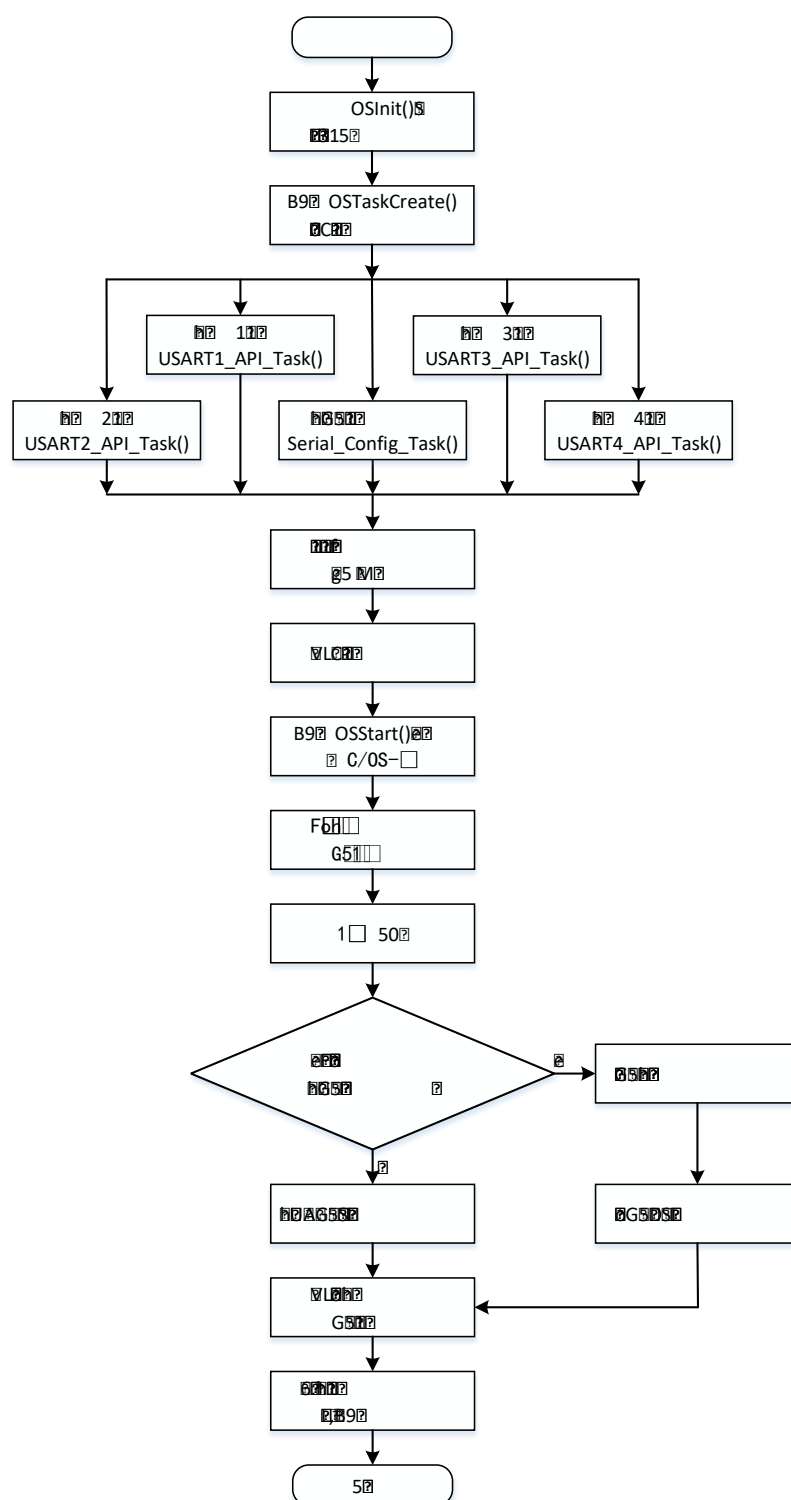


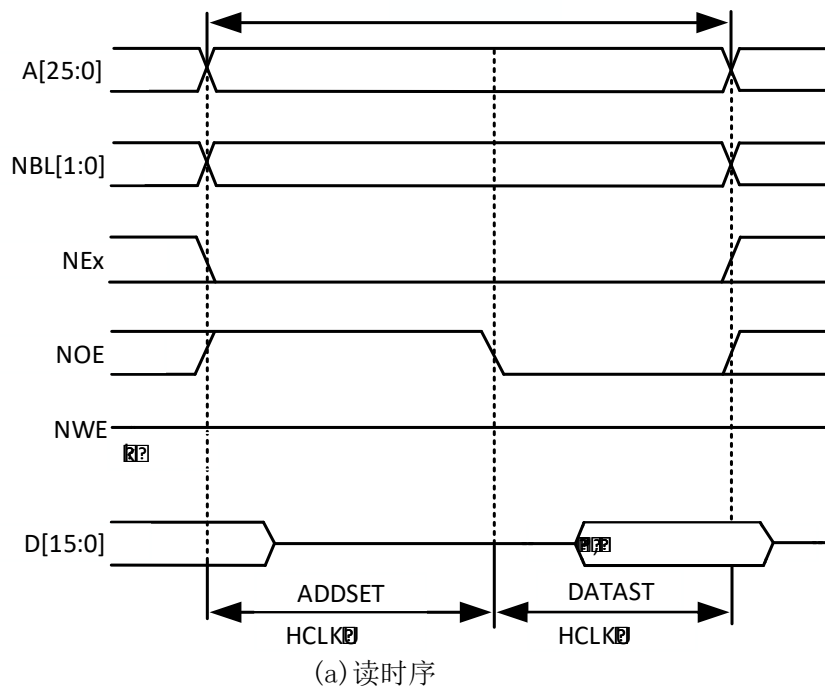
图 4.4 多通道串口驱动 TFT 液晶软件流程图

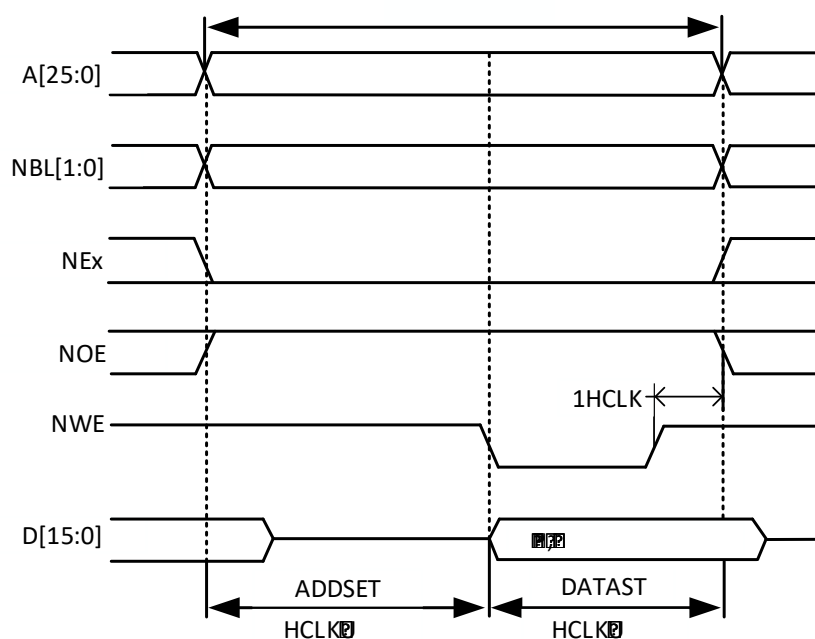
图 4.4 软件流程图表示的是多通道串口驱动 TFT 液晶软件执行过程。系统开始运行，首先会对整个系统进行初始化，初始化完成后创建起始任务，起始任务内包括 5 个任务，

其中一个串口配置任务和 4 个串口控制任务。在任务优先级设定方面，串口配置任务的优先级最高，由于串口配置任务的优先级最高，此任务将会被优先执行，执行后完成串口的一系列初始化，执行完串口配置任务后，将此任务删除，然后程序将会在其他任务之间进行调度。

### 4.2.1

系统所支持的屏幕类型为 MCU 接口的，驱动方式为 8080 时序的 TFT 液晶屏。对于同是 8080 时序但是不同型号、尺寸的屏幕，都可以使用 STM32F429 的 FSMC 接口进行驱动，区别主要在于初始换控制指令和 ID 号不同。只需要通过识别不同屏幕的 ID 号，提供相对应的初始换即可，其余一系列的读写函数都可兼容。8080 时序的 TFT 液晶屏外部数据通信接口有：16 根数据线、数据命令选择端、读写使能端、片选信号端、复位和背光端。使用 STM32F429 的 FSMC 的模式 A 可完成控制<sup>[44-45]</sup>。图 4.6 位模式 A 的读写时序图，图 4.7 为 8080 接口时序图。对比时序图，可以看出 8080 接口时序并没有地址线，而是有一个数据命令选择引脚，用来控制写入的是数据还是命令，这样可以将数据命令选择端接到其中的一跟地址线，通过对地址的写 0 和 1 的操作，即可完成数据和命令选择。这样就可以使用 FSMC 接口模拟 8080 时序，实现对 TFT 液晶屏的显示驱动。





(b) 写时序

图 4.5 FSMC 模式 A 读写时序

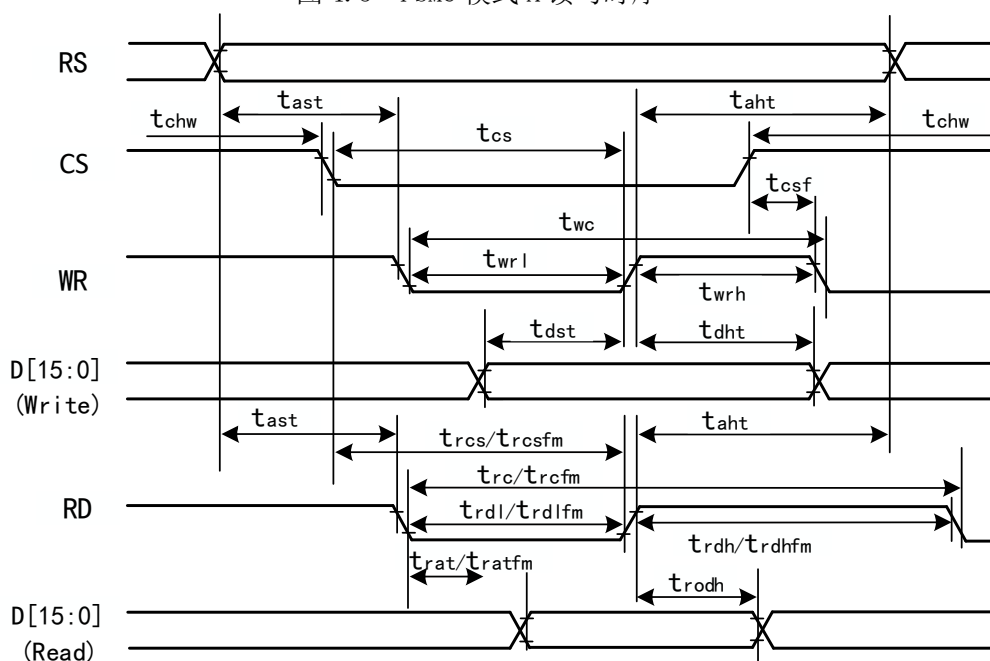


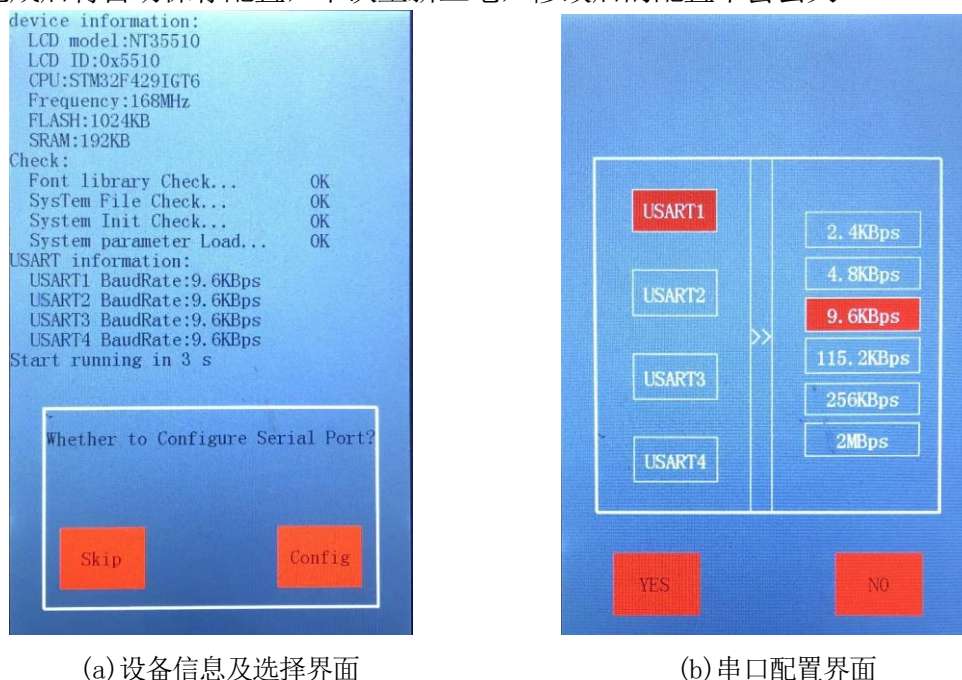
图 4.6 16 位并口 8080 接口时序

## 4.2.2

串口是外部处理器操作屏幕的接口，外部单片机控制屏幕的数据、指令都是通过串口接收并控制。整个系统共应用到了 4 个串口进行屏幕控制，并采用 DMA 的方式使串口收发

数据。利用 DMA 方式可以减轻 CPU 负担，数据发送接收无需 CPU 参与，本质是将数据进行地址空间的复制<sup>[46-47]</sup>。

图 4.7 为整个设计的系统配置界面。其中图 4.7 (a) 为设备信息及选择界面，如中可以看出，系统列出了一些屏幕信息和处理器信息，并对一些初始化做了检查，也可以看出当前 4 个串口的波特率都为 9600bps。如果想更改配置，可通过触摸屏幕进行修改，当然也可以选择跳过。图 4.7 (b) 是进入串口配置界面，在此界面可以修改 4 个串口的波特率，修改完成后将自动保存配置，下次重新上电，修改后的配置不会丢失。



(a) 设备信息及选择界面

(b) 串口配置界面

图 4.7 系统配置界面

### 4.2.3

外接单片机在真正控制屏幕前，需要先进行发送设置数据，这些设置数据包括串口号信息、选择显示图片还是字符、数字和汉字。如果是图片数据，则先获取图片要显示在屏幕的坐标和图片大小，然后再接收图片数据，以点阵的方式进行显示。如果是字符类型的数据，先会确定字符的大小、字符长度和需要显示的坐标位置以及字体的颜色和屏幕背景色，然后再进行数据转换成字库中字模地址，读取字模数据。图 4.8 的流程图为系统的一些数据接口的调用步骤。

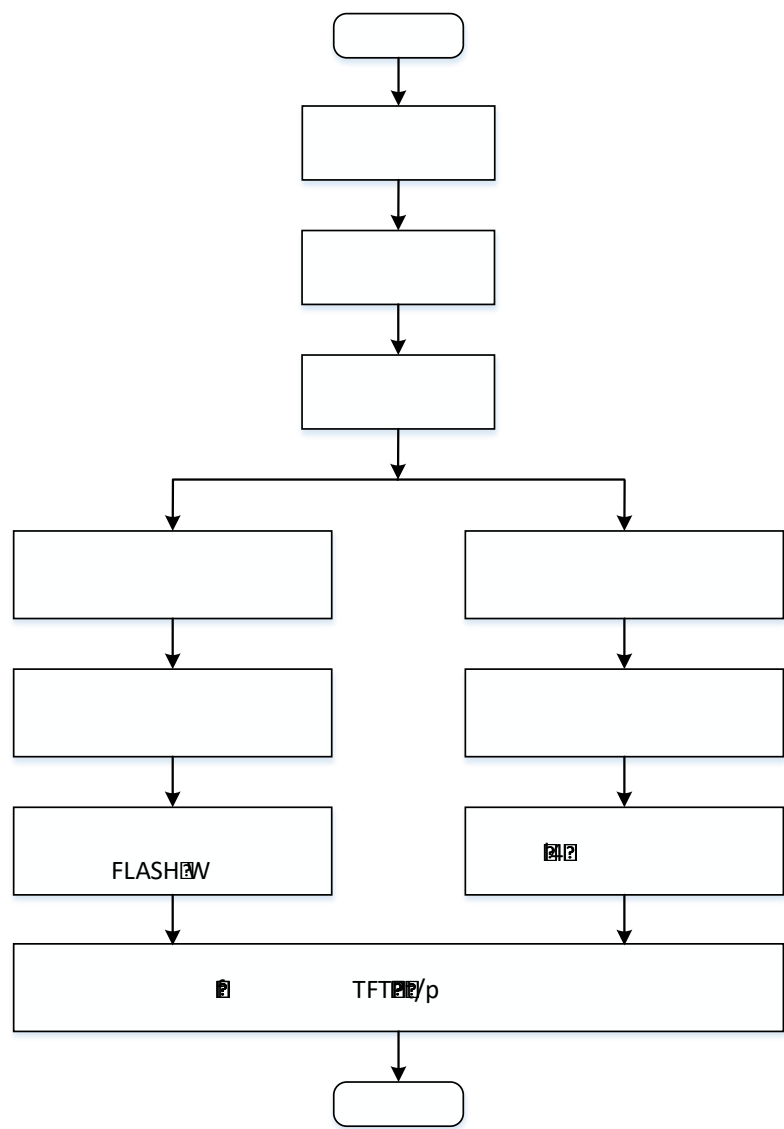


图 4.8 驱动接口调用流程图

4.3 WIFI

语音传输实际上并不是发送语音信号，而是通过控制指令，控制某一个节点事先录入的语音文件进行播放。它是通过 WIFI 模块连接家庭路由器，将路由器作为数据传输中继，实现各个节点的相互通信<sup>[48~49]</sup>。

4.3.1

首先，每个节点都有多个独立按键，用来对不同的其他节点进行呼叫。按键的实现，通过 AD 采集的方式实现，这样可以节约大量 IO 口，也防止在节点过多的情况下，造成 IO



口数量不够。实现原理如图 4.9 所示，通过按键按下时会根据电阻分压的不同产生不同的电压值进行判断： $V_{out} = R_x / (R_f + R_x)$ 。

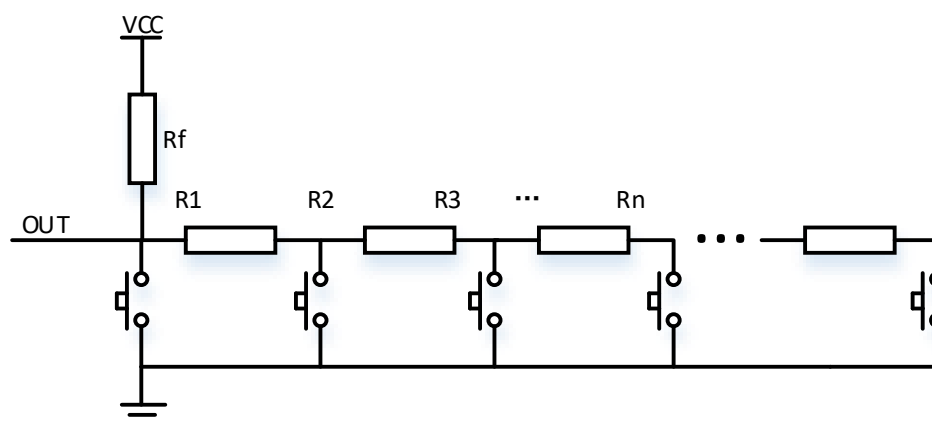


图 4.9 按键实现 AD 采集图

**数据发送：**当想要呼叫不同节点时候，只需要按下对应节点按键，以及选择需要呼叫的语音信号，数据将自动发送出去，当相应的节点收到后，将会进行反馈，并语音提示以收到。

**数据接收：**当节点接收到另外一个节点发送来的数据后，为了确保数据安全，首先需要比对发送节点的 IP 地址和端口号，比对正确后才对数据进行处理。无论是发送还是接收到的数据都需要通过路由器进行数据交换，虽然通过路由器进行数据交换，但在普通室内经过测试，数据传输基本没有任何延迟，运行较为稳定。

## 4.4

### 4.4.1

图 4.10 表示的是移动平台软件流程图。系统上电，各模块自动完成初始化配置，然后 ESP8266 进行联网与 STM32 通过串口建立通信。在车体移动前通过陀螺仪获取 Z 轴的角度，确定车体当前与虚拟坐标 x 轴的夹角，设置起始坐标  $(x_0, y_0) = (0, 0)$ 。云台调整初始位置到摄像头对准正前方，安装在车体的温湿度传感器开始获取数据并在车体 OLED 显示，为了测试本设计在车体安装了 AM2302 温湿度传感器，并预留其他传感器安装口，可根据需求增加传感器。等待遥控端发出对车体的控制指令，对于手机 APP 端，可以与车体进行双向通信，在手机端对车体控制的同时，手机端也可以接收车体传回来的传感器数据和位置信息等，传感器获取的数据会实时上传至手机端，对于位置信息，会在每次车体 Z 轴角度发生变化或停止移动时上传。对于 NRF 遥控端：两个 NRF24L01 之间只能进行单向通

信，一个作为发送端，一个作为接收端，在遥控端对车体和云台进行控制的时候，遥控端无法进行数据接收，因此需要对模式进行切换。切换方式是通过 NRF 遥控端的按键实现，当想要获取传感器数据和当前位置信息时，只需要按下 NRF 遥控端的切换按键，NRF 遥控端会发送切换指令到车体接收端，遥控端发送完指令后就会切换为接收模式，等待车体发回来的数据，当车体端接收到切换指令后，会切换到发送模式，并将数据发送出去，等待数据发送完成，车体端自动切换回接收模式，当遥控端接收完信息后自动切换为发送模式。

[50-52]

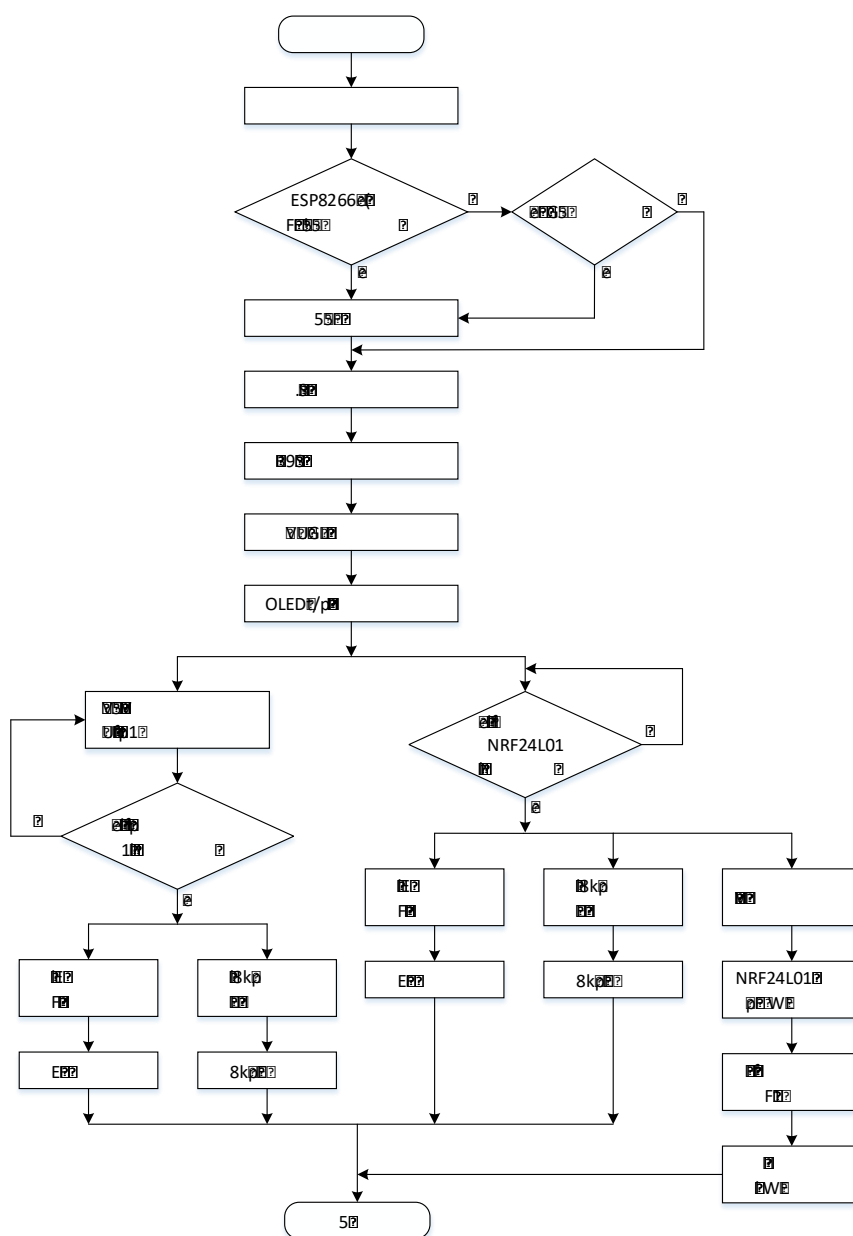


图 4.10 移动平台软件流程图

### 4.4.2 OpenWrt

OpenWrt 是 Linux 的一个嵌入式发行版本，常用于路由器。它是一个高度自动化和模块化的嵌入式 Linux 系统，其网络组件丰富、扩展性能很好<sup>[53-54]</sup>。本设计将 OpenWrt 路由系统移植到 AR9331 系统板，实现 AR9331 路由器功能。

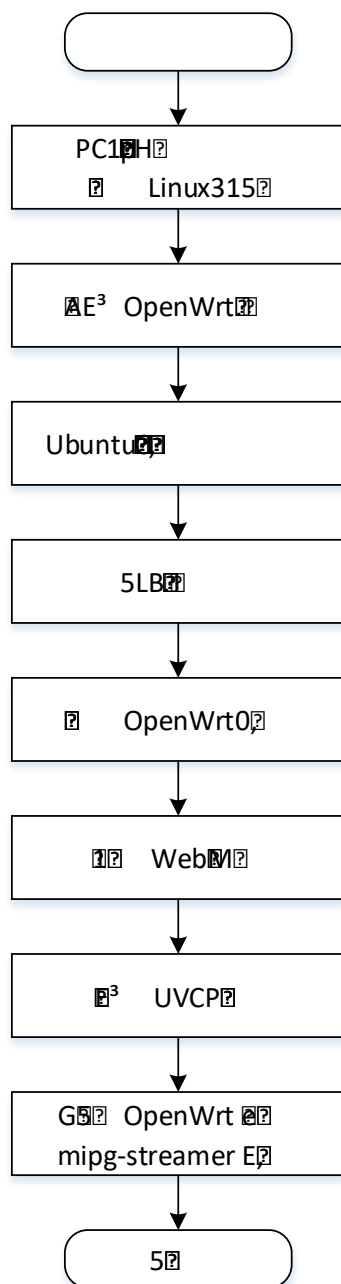


图 4.11 移动平台软件流程图

OpenWrt 系统移植流程如图 4.11 所示。对于一个完整的嵌入式操作系统，一般是由 U-Boot、内核、文件系统和应用程序组成，系统首先运行的是 U-Boot，通过 U-Boot 引导

启动内核，内核启动后，挂载文件系统，运行应用程序。路由器一般都有个 Web 界面进行路由器设置，OpenWrt 的 Web 界面叫 luic，可通过命令进行安装 Web 界面<sup>[55-56]</sup>。要想使 OpenWrt 支持免驱的 USB 摄像头，必须要在系统上加载 UVC 驱动程序，可以通过 make menuconfig 命令进入菜单进行配置。mjpg-streamer 是一款可用于视频数据采集、视频格式转换和视频传输的服务器软件，将此软件配置进去既可实现 Web 下的视频监控。所有项配置完成后，插入摄像头，启动系统，就可以用手机或者 PC 连接系统发出的 WiFi 信号，连接成功后，通过浏览器登录 192.168.1.1:8080/?action=stream 便可以查看视频。也可以通过外网登录服务器生产的外网访问地址查看视频监控信息<sup>[57-58]</sup>。

### 4.4.3

车体定位的实现主要是依靠和车轮同轴的编码器和车体的陀螺仪。为了保证定位准确，在算法中加入 PID 控制中的 P 控制，进行修正车体运动过程中产生的方向偏差： $\text{Turn} = -\text{Basic} * K_p$ 。（Turn：修正值，Basic：角度偏差， $K_p$ ：比例系数）。

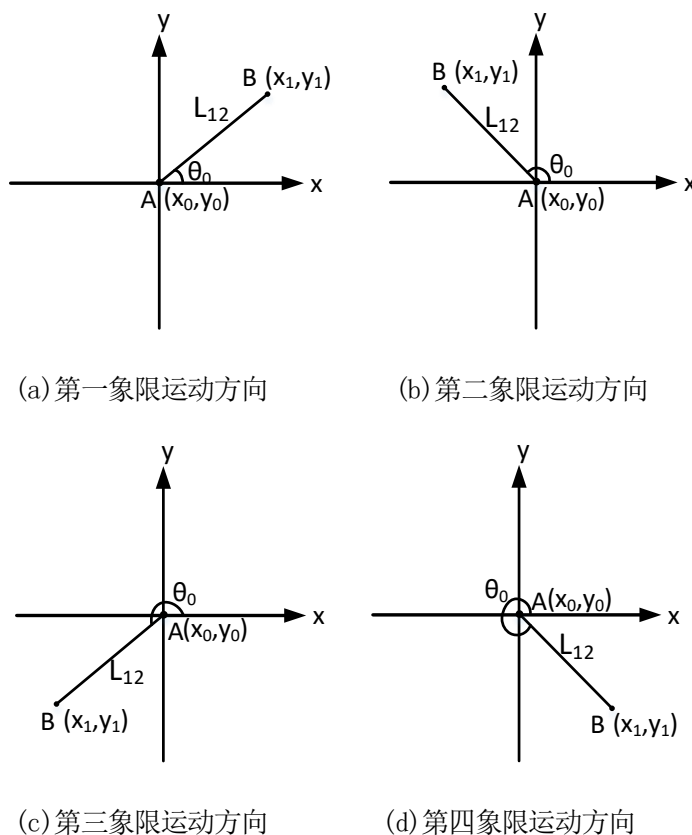


图 4.12 车体初始运动

车体初始运动方向如图 4.12 所示。利用陀螺仪获取车体运动前的初始角度  $\theta_0$ ，初始点记做 A，初始坐标为  $(x_0, y_0) = (0, 0)$ 。在车体第一次转向时运动到 B 点，坐标为  $(x_1, y_1)$ ，利用编码器测得 AB 两点距离为  $L_{12}$ 。B 点坐标计算式如式(4.1)所示。

$$(x_1, y_1) = \begin{cases} (L_{12} \cdot \cos \theta_0, L_{12} \cdot \sin \theta_0) & 0 \leq \theta_0 \leq \frac{\pi}{2} \\ (-L_{12} \cdot \cos \theta_0, L_{12} \cdot \sin \theta_0) & \frac{\pi}{2} \leq \theta_0 \leq \pi \\ (-L_{12} \cdot \cos \theta_0, -L_{12} \cdot \sin \theta_0) & \pi \leq \theta_0 \leq \frac{3\pi}{2} \\ (L_{12} \cdot \cos \theta_0, -L_{12} \cdot \sin \theta_0) & \frac{3\pi}{2} \leq \theta_0 \leq 2\pi \end{cases} \quad (4.1)$$

车体在 B 点转向后继续运动到 C 点，坐标为  $(x_2, y_2)$ ，坐标图如图 4.13 所示。

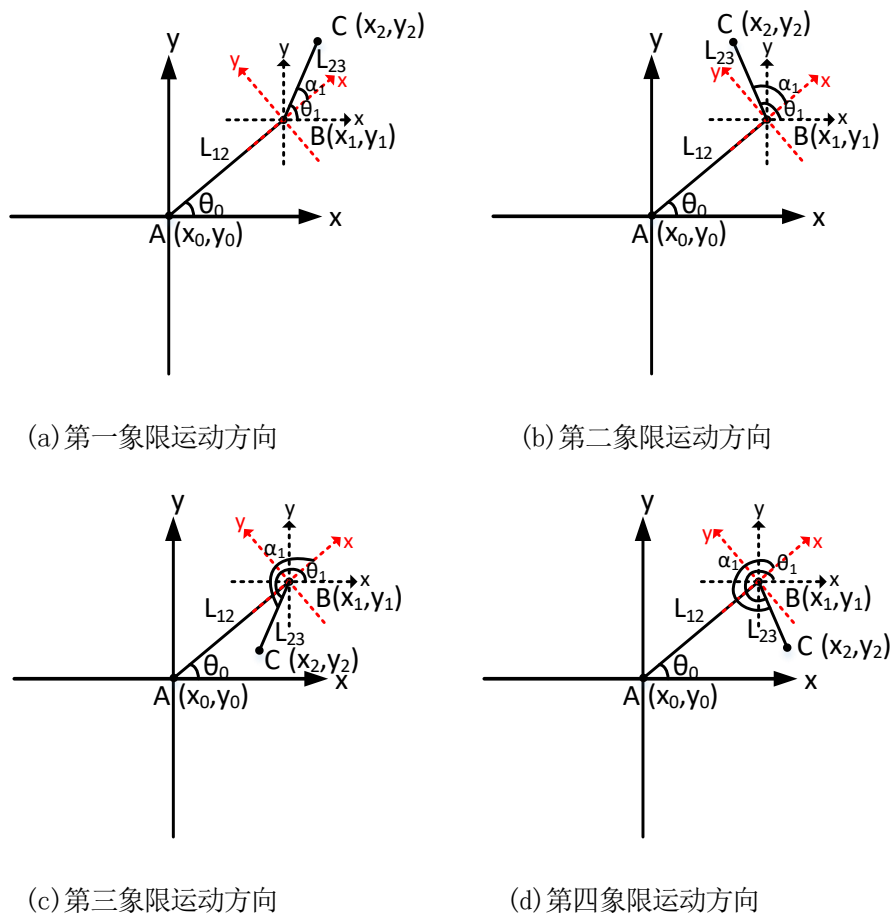


图 4.13 第一次转向后运动

图 4.12 中红色虚线坐标表示车体转向前方向坐标，与转向后的车体夹角为  $\alpha_1$ ，黑色虚线坐标表示算法纠正后的车体方向坐标，与转向后的车体夹角为  $\theta_1$ ，BC 点的距离为  $L_{23}$ 。C 点坐标计算式如式(4.2)、式(4.3)所示。

$$\theta_1 = \theta_0 + \alpha_1 \quad (4.2)$$

$$(x_2, y_2) = \begin{cases} (x_1 + L_{23} \cdot \cos \theta_1, y_1 + L_{23} \cdot \sin \theta_1) & 0 \leq \theta_1 \leq \frac{\pi}{2} \\ (x_1 - L_{23} \cdot \cos \theta_1, y_1 + L_{23} \cdot \sin \theta_1) & \frac{\pi}{2} \leq \theta_1 \leq \pi \\ (x_1 - L_{23} \cdot \cos \theta_1, y_1 - L_{23} \cdot \sin \theta_1) & \pi \leq \theta_1 \leq \frac{3\pi}{2} \\ (x_1 + L_{23} \cdot \cos \theta_1, y_1 - L_{23} \cdot \sin \theta_1) & \frac{3\pi}{2} \leq \theta_1 \leq 2\pi \end{cases} \quad (4.3)$$

同理：车体第  $n$  次转向前或者停止运动时（通过获取编码器的速度值可以判断车体是否处于运动状态）的坐标  $(x_n, y_n)$ ，坐标  $(x_n, y_n)$  的计算如下：

$$(x_n, y_n) = \begin{cases} (x_{n-1} + L_{n(n-1)} \cdot \cos \theta_{n-1}, y_{n-1} + L_{n(n-1)} \cdot \sin \theta_{n-1}) & 0 \leq \theta_{n-1} \leq \frac{\pi}{2} \\ (x_{n-1} - L_{n(n-1)} \cdot \cos \theta_{n-1}, y_{n-1} + L_{n(n-1)} \cdot \sin \theta_{n-1}) & \frac{\pi}{2} \leq \theta_{n-1} \leq \pi \\ (x_{n-1} - L_{n(n-1)} \cdot \cos \theta_{n-1}, y_{n-1} - L_{n(n-1)} \cdot \sin \theta_{n-1}) & \pi \leq \theta_{n-1} \leq \frac{3\pi}{2} \\ (x_{n-1} + L_{n(n-1)} \cdot \cos \theta_{n-1}, y_{n-1} - L_{n(n-1)} \cdot \sin \theta_{n-1}) & \frac{3\pi}{2} \leq \theta_{n-1} \leq 2\pi \end{cases} \quad (4.4)$$

经过上述算法，不仅可以定位出车体坐标点，距离初始位置距离，还可以计算出车体运动轨迹。

## 4.5

本章主要是对整个系统各部分系统的软件、算法进行分析。首先分析的是室内数据采集一些软件和算法，实现了通过在不同的节点安装不同的传感器进行数据采集、监控、上传。并可将数据通过 ESP8266 传输到物联网平台，完成了了整个系统的第一步。然后分析了改进后的屏幕显示，屏幕改进前，主要使用的是 LCD 屏幕，缺点是屏幕小，显示内容有限，改进后使用的是 TFT 液晶屏，在屏幕上加入了 STM32F4 进行控制，实现了外部处理器可以通过串口，并在遵循一定的协议下对屏幕进行控制，大大简化了 TFT 屏幕对处理器资源的消耗，减少了代码的编写量等。然后介绍了语音呼叫系统的实现，此系统主要是通过将路由器作为整个系统的中继，通过 UDP 模式进行数据传输，实现了各个节点之间的双向通信。最后对移动视频监控软件、算法进行了分析，实现了视频监控、近距离和远距离遥控、车体定位等功能。

5.1

图 5.1 为室内采集监控系统运行界面。整个界面分为三页，第一页如（a）图所示，主要显示了基本的监控信息，如温度、湿度、网络状况和天然气浓度。（b）图主要是时间日期等信息，也包括了农历日期信息。（c）图是菜单设置界面，主要是对一些功能进行设置，设置方式可通过板载按键或者红外遥控器进行。三个显示页面的翻页模式可以设置为隔一段时间自动翻页，也可以通过按键手动翻页。



图5.1 系统运行界面

为了测试 LoRa 模块数据发送的实时性，在距离 50 米以上并存在遮挡物的情况下测试一个 64bit 的数据收发时间，并通过逻辑分析仪将时间进行分析，图 5.2 为逻辑分析仪分析后的结果图。图中发送状态为逻辑电平高，接收状态为逻辑电平低。图中共有 4 个通道，通道一为主节点，其余为从节点通道。图中可以看出，主节点开始接收从节点的数据然后下发控制指令到从节点，整个时间持续小于 0.35 秒，可以较快的速度完成数据的发送和接收。

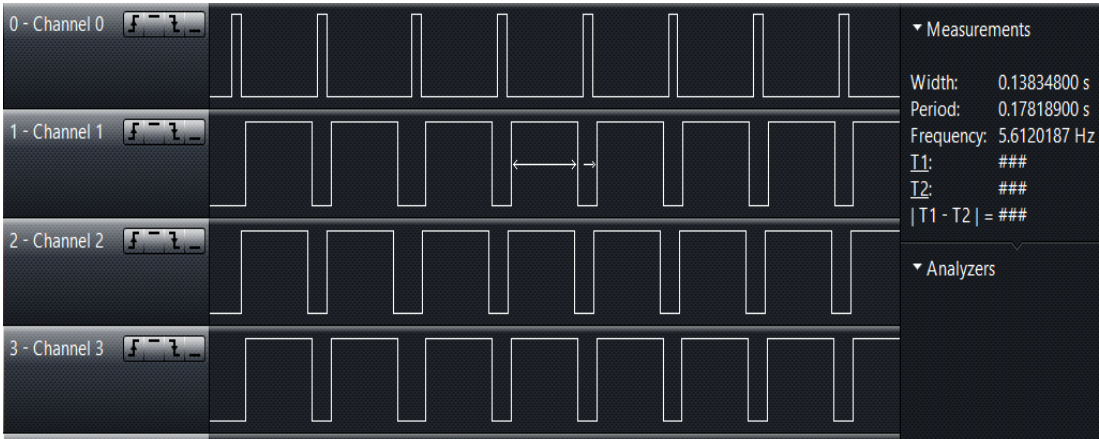


图5.2 数据传输时间采集图

不仅如此，主节点将采集后的数据后实时上传到百度云，可以通过生成的url进行访问自己创建的可视化界面，查看实时数据。百度云实时显示的图形化数据如图5.3所示。

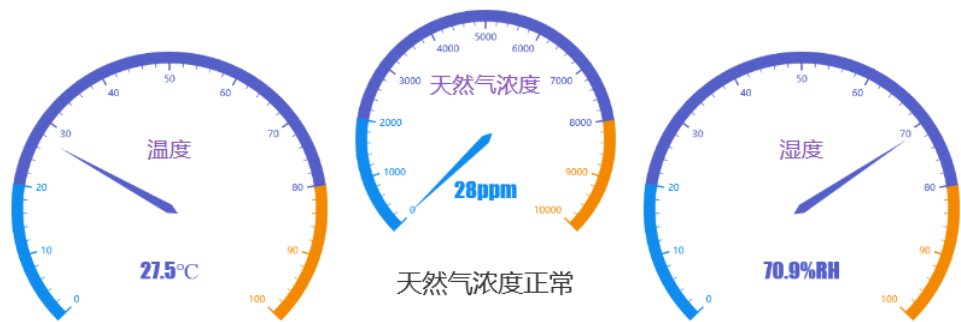


图5.3 云端显示数据

5.2

在数据传输中，波特率选择对传输速率会有一定影响，经过实验，完成了对波特率传输速度的测试，表 5.1 表示了屏幕支持的波特率下理论传输速率和实际传输速率。举例说明，如果使用最常用的 9600bps，传输大小为 16x16 的字符，每秒可以传输大概 29 个。

表5.1 不同波特率的数据传输速度

波特率	理论值	实际值
2.4KBps	0.24KB/s	0.23KB/s
4.8KBps	0.48KB/s	0.47KB/s
9.6KBps	0.96KB/s	0.94KB/s
115.2KBps	11.52KB/s	11.13KB/s
256KBps	25.6KB/s	23.27KB/s
2MBps	200KB/s	196.9KB/s

对串口驱动屏幕显示进行的测试结果如图 5.4 显示的。图中使用的屏幕为 4.3 寸的 TFT，测试时使用的波特率为 9600bps，对各种字符显示做了测试。图（a）为外接一个单片机对屏幕控制情况；图（b）为外接两个单片机对屏幕控制情况；图（c）为外接三个单片机对屏幕控制情况；图（d）为外接 4 个单片机对屏幕控制情况。外接单片机数量不同，屏幕可以做到自动分屏，避免设备控制混乱。从实验情况来看，无论是外接一个或者多个单片机对屏幕进行控制，屏幕显示效果较为流畅稳定。



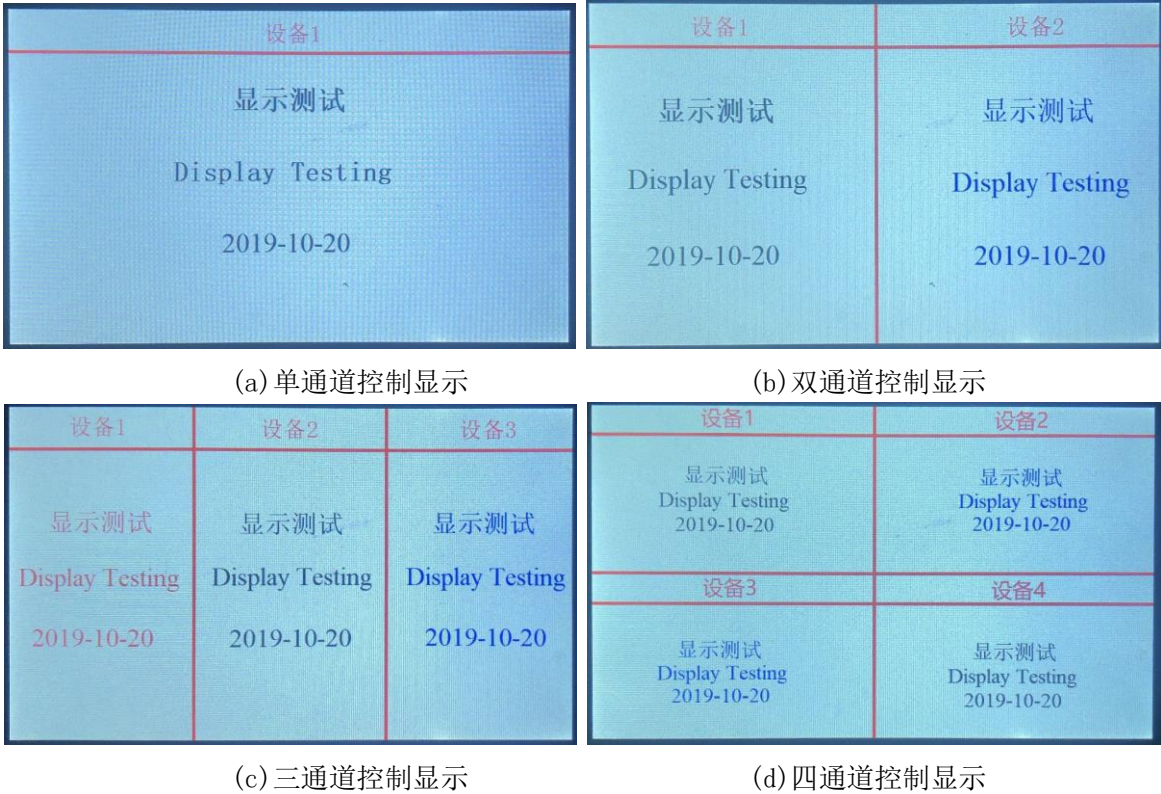


图5.4 液晶屏控制显示测试结果

在单片机和嵌入式开发中，TFT 液晶屏是经常用到的，如数据采集监控系统中，最初使用的是 LCD 屏幕，后为了增加显示区域和显示效果，改用 TFT 液晶屏，本次设计的液晶屏不仅在数据采集系统中使用，在嵌入式开发领域中也具有一定的作用。系统主要采用的方法是外部处理器通过串口与 STM32F429 进行数据交互，然后通过接收到的数据指令间接对屏幕进行控制。这样做的好处在于可以减轻处理器资源，简化操作步骤，即使没有串口的处理器也可以通过 IO 模拟串口进行处理，再者本设计具有很好的兼容性，经后续测试，本设计可以驱动 5 种尺寸和 15 种不同驱动芯片的屏幕，而且在 4 个通道同时使用时，屏幕显示效果较好，无卡顿现象。

5.3

语音呼叫功能特别适合运用在家庭有老年人的场景，对于活动不便的老年人，如果有紧急事情需要呼叫其他位置的子女，通过此设备可以方便快捷的进行呼叫，使呼叫更加方便，快捷。不仅如此，此系统也可以应用到病房呼叫系统，与医院网络进行连接，接入方式简单。经实验验证，整个系统运行稳定可靠，在室内网络信号良好，呼叫传输延迟在毫

秒级别，基本可以忽略。语音订制简单，可通过 USB 接口直接将音频文件复制到语音模块即可。整个系统上手简单，使用方便，可以在多个应用场景下满足需求。

5.4

定位算法的实现，主要依赖 STM32F030 对三角函数的处理。通过选取 10 个不同的角度值并对每个角度分别进行 50 次 sin、cos 的运算，得到 STM32F030 每处理一次三角函数时间如表 5.2 所示。

表5.2 不同角度正弦和余弦函数运算时间

角度	sin	cos
15.58°	0.378~0.380	0.415~0.417
45.79°	0.484~0.485	0.472~0.474
88.99°	0.452~0.454	0.476~0.478
152.48°	0.611~0.612	0.614~0.616
184.25°	0.607~0.609	0.583~0.585
221.17°	0.611~0.613	0.621~0.623
251.86°	0.611~0.613	0.605~0.608
270.42°	0.591~0.593	0.607~0.609
310.65°	0.625~0.627	0.606~0.609
352.39°	0.604~0.607	0.581~0.583

通过对三角函数运算测试，STM32F030能够较快的完成三角函数的运算。图5.5所示是在对184.25° 进行sin运算时，其中一次运算所需的时间为0.60725ms。

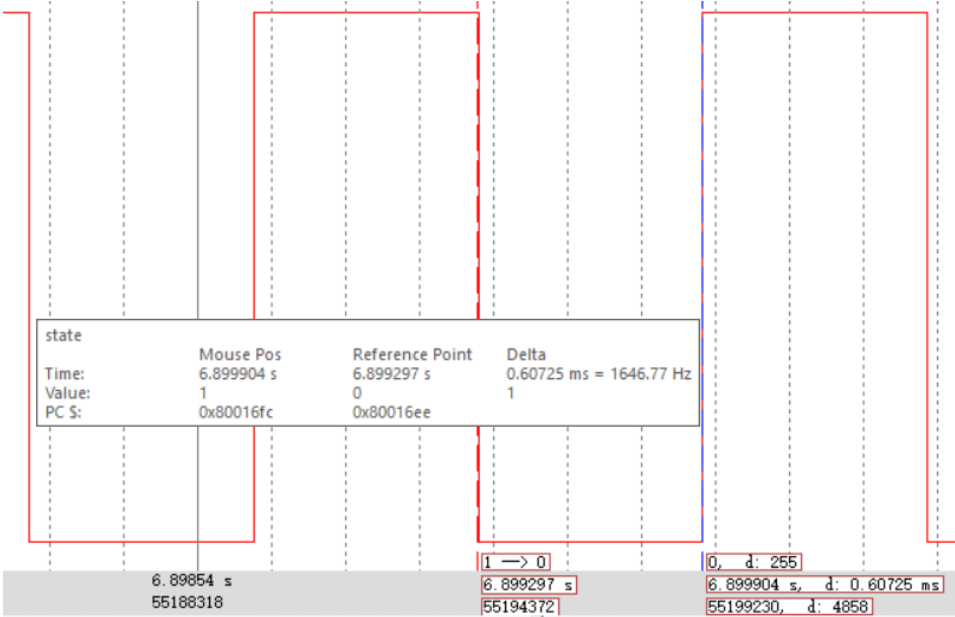


图5.5 184.25° 正弦函数运算时间

手机和PC连接WiFi后，登录网页地址，进行实时视频监控。图5.6和图5.7所示分

别是手机和PC对不同场景进行视频监控的结果。



图5.6 手机显示的监控视频



图5.7 PC显示的监控视频

将移动平台与无线视频监控相结合，通过NRF24L01和ESP8266实现了系统的近距离和远距离的遥控，增加了整个系统的适用范围，通过编码器和陀螺仪实现系统坐标定位，可根据需求更换安装在车体的传感器，以便获取所需数据。USB免驱的MJPEG格式输出的摄像头都可以直接使用。经过实验测试：遥控系统稳定，近、远距离遥控均无延迟现象；对车体定位较为准确，误差较小；手机和PC均可连接到AR9331发出的WiFi信号，并且通过手机和PC均可观看监控视频，也可以在外网的情况下通过服务器生成的外网访问地址观看监控视频，视频较为流畅，无卡顿现象。

5.5

本章主要对各个子系统进行实验验证分析。室内数据采集系统实验验证：室内数据采集系统可在 0.35 秒内可准确的对室内多点的数据进行监测和数据上传，证明了系统的实时性和准确性。为了增加适应性，可增加或减少节点数量，具有较好的可剪裁性。由于系统具有较好的实时性、准确性和可剪裁性，在智能监控领域有着较好的应用前景。液晶屏实验验证：MUC 接口的 TFT 液晶屏通过设计变为可以使用串口进行控制，最终实现了可以被外部带有串口的处理器所使用，降低了屏幕的驱动难度，节约了外部处理器资源。语音

呼叫系统实验验证：整个系统信息传输良好，无延迟现象，有很好的适应前景。移动视频监控平台实验验证：通过 NRF24L01 和 ESP8266 实现了系统的近距离和远距离的遥控，增加了整个系统的适用范围，通过编码器和陀螺仪实现系统坐标定位，可根据需求更换安装在车体的传感器，以便获取所需数据。系统支持所有 USB 免驱 MJPEG 输出格式的摄像头，可根据用户需求更换不同像素的摄像头。

## 6.1

物联网技术发展到今天，在技术方面已经非常成熟了，人们万物互联的理想也在逐渐实现。物联网目前已经应用在众多领域，如智能城市、智能农业、智能交通、智能家居等，并在这些领域展现出相当大的活力，并推进了这些行业的发展。未来物联网产业的发展将会更加广阔，应用的行业也将更加广阔，带来的变革也将是难以想象。

本文主要是结合物联网技术将智能家居应用到家庭，实现一套家庭物联网智能控制系统。通过研究，现将具体工作归纳如下：

(1)本文介绍了家庭物联网智能控制系统的研究背景、意义和物联网当前的现状，对整个物联网产业进行了阐述，并且详细分析了物联网产业对当今的影响、应用层面以及未来发展。

(2)在设计前首先对此次方案的可行性作出了具体分析，对每个模块的算法进行分析，通过实验验证和资料查询验证算法的可行性，通过验证，整个系统所用到的算法可行程度较高，但个别可能存在缺陷，后续如果发现需要进行改进。对于所涉及到的硬件部分，需要分享电路问题、元器件选型问题、硬件电路板设计问题、焊接问题等，经过分析也都可以通过。

(3)然后对整个系统的硬件进行分析设计。首先是对数据采集监控系统的硬件进行分析，给出硬件设计框图，分析硬件设计思路等；然后是对液晶屏改进进行硬件分析，首先分析当前屏幕的缺点，寻找替代方案，确定方案，在进行电路设计，系统设计等，最终完成硬件设计；然后是语音呼叫系统的设计，对于语音呼叫系统，硬件相对较为简单；然后是移动监控平台硬件设计，设计包括两部分，移动端和视频监控端，需要进行电机控制、舵机控制、近距离和远距离数据传输等。

(4)然后是对整个系统的软件部分进行分析。首先是对数据采集监控系统进行软件设计分析，概述了整个控制系统各个模块实现的功能，对于新设计的数据传输格式进行分析验证，以及网络连接、数据上传方法；接下来是对液晶屏改进方案进行软件设计分析，阐述了 uC/OS 系统移植方法，串口控制协议分析等，实现多串口同时操作屏幕的功能；然后是

对语音呼叫系统软件进行设计分析，软件主要是以路由器为中继，以 WIFI 模块为节点，将每个 WIFI 模块连接到路由器，通过节点不同的 IP 地址和端口号完成数据传输；最后是对移动视频监控平台进行软件设计分析，对整个车体的控制算法作出了设计分析，对 OpenWrt 系统的移植作出了介绍，对车体的定位算法作出了分析，对视频采集方法作出了分析。

(5)最后对整个系统作出验证，并记录验证结果。

## 6.2

物联网正在朝着万物互联的方向发展，未来将具有非常广阔的前景，目前在对物联网的技术的研究具有非常大的挖掘空间，然而每一个新技术的突破，都将会对物联网的发展作出巨大的提升。

借助物联网技术可以实现很多人们需要的功能，也可以改变人们日常生活的很多方面，使人类向着更好的方向发展。对于我来说，作为一个电子专业的学生，即将毕业，未来工作选择上，物联网相关工作将会还是一个不错的选择。

对于本次的设计还存在许多不足之处。比如，在视频监控方面，并不能通过外网进行访问，需要做内网穿透才可以；语音呼叫并不是传递真正的语音信号，在这一方面还需要改进，做到可以真正传输人此刻说话的声音。因此，为了所做的工作还有很多。不仅在此方面，在其他方面、比如智慧交通、智慧农业等方面也还有很多未知等着去探索。

- 
- [1] 沈林涛,王凯.基于物联网技术的实验设备监控系统设计与实现[J].软件导刊,2021,20(02):119-123.
- [2] 王睿.大数据时代物联网技术的应用与发展[J].网络安全技术与应用,2021(04):67-68..
- [3] 《物联网技术》简介[J].物联网技术,2021,11(02):122.
- [4] 邱宏斌.一种基于ESP8266模块的物联网设计思路[J].电子世界,2017(07):157.
- [5] 黄健,钟益民,方挺,李文钦.面向工业电机温度监测的LoRa无线传感网络设计[J].兰州工业学院学报,2021,28(01):76-80.
- [6] 张琴,戴阳,杨胜龙,张衡,崔雪森.基于LoRa的低功耗水产养殖水质监测系统设计与实现[J].传感器与微系统,2019,38(11):96-99.
- [7] 魏源通,戴亚文.基于LoRa的工业监测自组网系统设计[J].计算机测量与控制,2019,27(02):225-228.
- [8] 肖华,王亚刚.基于物联网与云架构的实验室监督管理系统[J].软件导刊,2018,17(12):135-138.
- [9] 蒲逸然.基于物联网技术的新型设备管理框架探讨[J].现代商贸工业,2020,41(13):192-193.
- [10] Othmane Friha,Mohamed Amine Ferrag,Lei Shu,Leandros Maglaras,Xiaochan Wang.Internet of Things for the Future of Smart Agriculture:A Comprehensive Survey of Emerging Technologies[J].IEEE/CAA Journal of Automatica Sinica,2021,8(04):718-752.
- [11] ALiero Muhammad Saidu,Qureshi Kashif Naseer,Pasha Muhammad Fermi,Jeon Gwanggil. Smart Home Energy Management Systems in Internet of Things networks for green cities demands and services[J]. Environmental Technology & Innovation,2021,22(prepublish):835-841.
- [12] Zikria Yousaf Bin,Ali Rashid,Afzal Muhammad Khalil,Kim Sung Won. Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions.[J]. Sensors (Basel, Switzerland),2021,21(4):225-230.
- [13] Chandra Umesh,Shukla Gaurav,Maheshwari Himani,Kaur Rasmeet. Internet of Things (IoT) in Agriculture[J]. Informatics Studies,2021,7(2):768-773.
- [14] Othmane Friha,Mohamed Amine Ferrag,Lei Shu,Leandros Maglaras,Xiaochan Wang.Internet of Things for the Future of Smart Agriculture:A Comprehensive Survey of Emerging Technologies[J].IEEE/CAA Journal of Automatica Sinica,2021,8(04):718-752.
- [15] Mentsiev A U,Gatina F F. Internet of things and data analysis in agriculture[J]. IOP Conference Series: Earth and Environmental Science,2021,677(3):703-710.
- [16] Monteleone Sergio,Moraes Edmilson Alves de,Tondato de Faria Brenno,Aquino Junior Plinio Thomaz,Maia Rodrigo Filev,Neto André Torre,Toscano Attilio. Exploring the Adoption of Precision Agriculture for Irrigation in the Context of Agriculture 4.0: The Key Role of Internet of Things.[J]. Sensors (Basel, Switzerland),2020,20(24):595-604.
- [17] Benssalah Mustapha,Sarah Izza,Drouiche Karim. An Efficient RFID Authentication Scheme Based on Elliptic Curve Cryptography for Internet of Things[J]. Wireless Personal



- Communications,2020,117(3):473-480.
- [18] Li Weiguang,Yuan X.,Elhoseny M.. Design of smart campus management system based on internet of things technology[J]. Journal of Intelligent & Fuzzy Systems,2021,40(2).
- [19] 王栋,袁伟,吴迪.基于WiFi物联网的图书馆环境监测系统[J].计算机科学,2018,45(S2),532-534+557.
- [20] 罗俊,孙国耀.基于WiFi无线传感器网络的工业环境监测平台[J].仪表技术与传感器,2018,(08),65-68.
- [21] 汪文娟,杨余旺,黄宇民,等.无线传感器网络节点通信抗干扰实验研究[J].计算机工程与应用,2013,49(9):102-105.
- [22] 何诚刚.基于LoRa的无线监测系统设计[J].山东农业大学学报(自然科学版),2018,49(3):528-530.
- [23] 王军,张福弟,王磊,等.基于STM32的R61581驱动液晶屏显示研究方法[J].液晶与显示,2016,31(1):93-96.
- [24] 马永杰,宋晓凤,李雪燕,等.基于嵌入式车流量实时监测算法研究与实现[J].液晶与显示,2018,33(9):787-792.
- [25] 徐白冰,王建立,曹景太.基于物联网的温湿度与风速信息采集系统设计[实现[J].液晶与显示,2016,31(10):936-942.
- [26] 毛博,徐恪,金跃辉,等. DeepHome: 一种基于深度学习的智能家居管控模型[J].计算机学报,2018,41(12): 2689-2701.
- [27] 杨丽,冯娟,卢秀丽,等.基于物联网智能家居安全监控系统设计[J].现代电子技术,2019,42(8): 55-58.
- [28] 李慧敏,樊记明,杨笑.基于STM32和OV7670的图像采集与显示系统设计[J].传感器与微系统,2016,35(9):114-117.
- [29] 原群盛,钱松荣.基于STM32动态监测系统设计[J].液晶与显示,2016,31(7):680-685..
- [30] 刘振海,骆冬根,王改.基于FSMC接口的CF卡数据存储系统设计[J].现代电子技术,2019,42(10):15-18.
- [31] 孟蓉歌,张春化,王佳,等.符合 AUTOSAR标准的 $\mu$ C/OS-III移植[J].计算机应用,2017,37(S2):248-250,258.
- [32] 谢小云,许艳,邓玉林.基于超高频RFID技术的矿山斜井安全监控系统的设计[J].江西理工大学学报,2017,38(5):70-74.
- [33] 唐娜娜,郑喜凤,武金忠,等.基于4G-LTE的脱机LED控制系统的设计[J].液晶与显示,2018,33(1):55-60.
- [34] 徐哲鑫,张艺林,林萧,等.基于OpenWrt与ZigBee的智能家居路由器设计[J].计算机工程,2017,43(3):94-98,104.
- [35] 肖雅丹,王占山.基于物联网的煤矿综合自动化控制系统研究[J].控制工程,2018,25(5):816-822.
- [36] 李时杰,何怡刚,罗旗舞,等.基于LoRa的电气设备温度湿度监测终端设计[J].传感器与微系统,2018,37(4):89-91.
- [37] 资文彬,李民政,彤新伟.基于LoRa的无线抄表系统设计与实现[J].桂林电子科技大学学报,2018,38(2):106-111.
- [38] 刘琛,邵震,夏莹莹.低功耗广域LoRa技术分析与应用建议[J].电信技术,2016,8(5):43-50.



- [39] 邱宏斌.一种基于ESP8266模块的物联网设计思路[J].电子世界, 2017 (7) :157.
- [40] 张宏伟,党瑞荣. 基于Internet对潜油电泵温度压力的远程监控[J]. 计算机科学,2016,43(S1),551-554.
- [41] 杨立身,张安伟,王磊,等.基于STM32的  $\mu$  C/OS-II与  $\mu$  C/GUI整合移植与显示优化[J].液晶与显示,2014,29(6):950-956.
- [42] 王泽,肖继学,张汉中,等.  $\mu$  C/OS-III的实时性测试[J].测控技术,2016,35(5):105-110.
- [43] 罗尔聪,郭宇.  $\mu$  C/OS-III任务调度器在Coq中的验证[J].计算机工程,2015,41(3):53-58.
- [44] 王军,张福弟,王磊,等. 基于STM32的R61581驱动液晶屏显示方法研究[J]. 液晶与显示,2016,31(1):93-96.
- [45] 汤莉莉,黄伟.基于STM32的FSMC接口驱动TFT彩屏设计[J].现代电子技术,2013,36(20):139-141,144.
- [46] 赵富强,王长坤,李露,等.基于STM32、以太网和ZigBee技术的智能家居系统的设计[J].测控技术,2016,35(4):94-97.
- [47] 罗志远,张涛,许骏.基于STM32的多通道照度技术的设计[J].传感技术学报,2019,32(4):618-624.
- [48] Holovatyy Andriy. Development of IoT Weather Monitoring System Based on Arduino and ESP8266 Wi-Fi Module[J]. IOP Conference Series: Materials Science and Engineering,2021,1016(1):198-203.
- [49] Barukčić I.,Liang Guangyu,Liu Yanyang,Feng Kangwei,Pan Yuefeng,Liu Yaxiong,Yuan Man. Design of home intelligent robot of internet of things[J]. MATEC Web of Conferences,2021,336:996-1002.
- [50] 卢忠亮,沈慧芳,陈杰.基于射频识别的智能物流小车[J].江西理工大学学报, 2011, 32(3):53-56.
- [51] 罗刘敏,王明霞,郭艳花,等.基于单片机的智能小车控制系统设计[J].仪表技术与传感器, 2018,(1):123-126.
- [52] 张阳伟,李成凤,佟良,等.嵌入式远程视频监控系统[J].测控技术,2018,37(12):137-140.
- [53] 陈为民,胡向臻.基于智能视频监控的安防系统设计[J].江西理工大学学报,2013,34(1):74-78.
- [54] 王键,王林惠,岳学军,等.基于WiFi的无人机视频传输系统设计与试验[J].农业工程学报,2015,31(2):47-51.
- [55] 张传胜.基于RFID技术的智能监控平台的设计[J].液晶与显示,2015,30(5):825-831.
- [56] 杨辉,严永锋,陆荣秀.基于模糊PID控制算法的管廊通风系统设计[J].控制工程,2019,26(12):2181-2187.
- [57] 朱志强,江紫亚,何玉庆,等.PID控制器的频域特性与无模型参数调节[J].控制与决策,2014,29(10):1833-1838.
- [58] 罗红波,施显阳,魏鹏,等.凿岩石车多关节机械臂的 PID 位置控制研究[J].工程科学与技术,2019,51(4):157-162.

## A

## 1、室内数据采集监控系统部分程序

数据校验和串口输出程序：

```

1 unsigned short CRC_CHECK(unsigned char *Buf, unsigned char CRC_CNT)
2 {
3     unsigned short CRC_Temp;
4     unsigned char i,j;
5     CRC_Temp = 0xffff;
6
7     for (i=0;i<CRC_CNT; i++){
8         CRC_Temp ^= Buf[i];
9         for (j=0;j<8;j++){
10             if (CRC_Temp & 0x01)
11                 CRC_Temp = (CRC_Temp >>1 ) ^ 0xa001;
12             else
13                 CRC_Temp = CRC_Temp >> 1;
14         }
15     }
16     return(CRC_Temp);
17 }
18 void OutPut_Data(USART_TypeDef* USARTx)
19 {
20     int32_t temp[4] = {0};
21     uint32_t temp1[4] = {0};
22     uint8_t databuf[10] = {0};
23     uint8_t i;
24     unsigned short CRC16 = 0;
25     for(i=0;i<4;i++)
26     {
27
28         temp[i] = (int)OutData[i];
29         temp1[i] = (unsigned int)temp[i];
30
31     }
32
33     for(i=0;i<4;i++)
34     {
35         databuf[i*2] = (unsigned char)(temp1[i]%256);
36         databuf[i*2+1] = (unsigned char)(temp1[i]/256);
37     }
38
39     CRC16 = CRC_CHECK(databuf,8);
40     databuf[8] = CRC16%256;
41     databuf[9] = CRC16/256;
42
43     for(i=0;i<10;i++)
44     {
45         USART_SendData(USARTx, databuf[i]);
46         while( USART_GetFlagStatus(USARTx, USART_FLAG_TXE) == RESET );
47     }
48     while(USART_GetFlagStatus(USARTx,USART_FLAG_TC)==RESET);
49 }

```

## 2、多通道串口驱TFT液晶屏显示系统部分程序

FSMC初始化函数:

```

41 void TFTLCD_FSMC_INIT(void)
42 {
43     FSMC_NORSRAMInitTypeDef  FSMC_NORSRAMInitStructure;
44     FSMC_NORSRAMTimingInitTypeDef  FSMC_ReadTimingInitStructure;
45     FSMC_NORSRAMTimingInitTypeDef  FSMC_WriteTimingInitStructure;
46
47     RCC_AHBPeriphClockCmd(RCC_AHBPeriph_FSMC,ENABLE); // 使能FSMC时钟
48     FSMC_ReadTimingInitStructure.FSMC_AddressSetupTime = 0x01; // 地址建立时间 (ADDSET) 为2个HCLK 1/36M=27ns
49     FSMC_ReadTimingInitStructure.FSMC_AddressHoldTime = 0x00; // 地址保持时间 (ADDHLD) 模式A未用到
50     FSMC_ReadTimingInitStructure.FSMC_DataSetupTime = 0x0f; // 数据保存时间为16个HCLK
51     FSMC_ReadTimingInitStructure.FSMC_BusTurnAroundDuration = 0x00;
52     FSMC_ReadTimingInitStructure.FSMC_CLKDivision = 0x00;
53     FSMC_ReadTimingInitStructure.FSMC_DataLatency = 0x00;
54     FSMC_ReadTimingInitStructure.FSMC_AccessMode = FSMC_AccessMode_A; // 模式A
55
56     FSMC_NORSRAMInitStructure.FSMC_Bank = FSMC_Bank1_NORSRAM4; // 这里我们使用NE4
57     FSMC_NORSRAMInitStructure.FSMC_DataAddressMux = FSMC_DataAddressMux_Disable; // 不复用数据地址
58     FSMC_NORSRAMInitStructure.FSMC_MemoryType = FSMC_MemoryType_SRAM; // FSMC_MemoryType_SRAM
59     FSMC_NORSRAMInitStructure.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_16b; // 存储器数据宽度为16bit
60     FSMC_NORSRAMInitStructure.FSMC_BurstAccessMode = FSMC_BurstAccessMode_Disable; // FSMC_BurstAccessMode_Disable;
61     FSMC_NORSRAMInitStructure.FSMC_WaitSignalPolarity = FSMC_WaitSignalPolarity_Low;
62     FSMC_NORSRAMInitStructure.FSMC_AsynchronousWait = FSMC_AsynchronousWait_Disable;
63     FSMC_NORSRAMInitStructure.FSMC_WrapMode = FSMC_WrapMode_Disable;
64     FSMC_NORSRAMInitStructure.FSMC_WaitSignalActive = FSMC_WaitSignalActive_BeforeWaitState;
65     FSMC_NORSRAMInitStructure.FSMC_WriteOperation = FSMC_WriteOperation_Enable; // 存储器写使能
66     FSMC_NORSRAMInitStructure.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
67     FSMC_NORSRAMInitStructure.FSMC_ExtendedMode = FSMC_ExtendedMode_Enable; // 读写使用不同的时序
68     FSMC_NORSRAMInitStructure.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
69     FSMC_NORSRAMInitStructure.FSMC_ReadWriteTimingStruct = &FSMC_ReadTimingInitStructure; // 读写时序
70     FSMC_NORSRAMInitStructure.FSMC_WriteTimingStruct = &FSMC_WriteTimingInitStructure; // 写时序
71
72     FSMC_NORSRAMInit(&FSMC_NORSRAMInitStructure); // 初始化FSMC配置
73     FSMC_NORSRAMCmd(FSMC_Bank1_NORSRAM4, ENABLE); // 使能BANK1
74 }

```

## 3、语音呼叫系统部分程序

数据传输、按键判断程序：

```

216 void KEY_Player(void)
217 {
218     if(analogValue >= ADC_ValueRange[0][0] && analogValue <= ADC_ValueRange[0][1] /*&& digitalRead(D1) == LOW*/)
219     {
220         delay(50);
221         if(analogValue >= ADC_ValueRange[0][0] && analogValue <= ADC_ValueRange[0][1])
222         {
223             Udp.beginPacket("192.168.4.113", 8888);
224             Udp.write("T1");
225             Udp.endPacket();
226
227             digitalWrite(D7,HIGH);
228             delay(100);
229             digitalWrite(D7,LOW);
230         }
231     }
232     if(analogValue >= ADC_ValueRange[1][0] && analogValue <= ADC_ValueRange[1][1] /*&& digitalRead(D1) == LOW*/)
233     {
234         delay(50);
235         if(analogValue >= ADC_ValueRange[1][0] && analogValue <= ADC_ValueRange[1][1])
236         {
237             JQ8400_MusicChoice(2);
238         }
239     }
240     if(analogValue >= ADC_ValueRange[2][0] && analogValue <= ADC_ValueRange[2][1] /*&& digitalRead(D1) == LOW*/)
241     {
242         delay(50);
243         if(analogValue >= ADC_ValueRange[2][0] && analogValue <= ADC_ValueRange[2][1] )
244         {
245             JQ8400_MusicChoice(3);
246         }
247     }
248     if(analogValue >= ADC_ValueRange[3][0] && analogValue <= ADC_ValueRange[3][1] /*&& digitalRead(D1) == LOW*/)
249     {
250         delay(50);
251         if(analogValue >= ADC_ValueRange[3][0] && analogValue <= ADC_ValueRange[3][1])
252         {
253             JQ8400_MusicChoice(4);
254         }
255     }
256     if(analogValue >= ADC_ValueRange[4][0] && analogValue <= ADC_ValueRange[4][1] /*&& digitalRead(D1) == LOW*/)
257     {
258         delay(50);
259         if(analogValue >= ADC_ValueRange[4][0] && analogValue <= ADC_ValueRange[4][1])
260         {
261             Udp.beginPacket("192.168.4.112", 8887);
262             Udp.write("T2");
263             Udp.endPacket();
264
265             digitalWrite(D7,HIGH);
266             delay(100);
267             digitalWrite(D7,LOW);
268         }
269     }
270     if(analogValue >= ADC_ValueRange[5][0] && analogValue <= ADC_ValueRange[5][1] /*&& digitalRead(D1) == LOW*/)
271     {
272         delay(50);
273         if(analogValue >= ADC_ValueRange[5][0] && analogValue <= ADC_ValueRange[5][1])
274         {
275             Udp.beginPacket("192.168.4.112", 8887);
276             Udp.write("TT");
277             Udp.endPacket();
278
279             Udp.beginPacket("192.168.4.113", 8888);
280             Udp.write("TT");
281             Udp.endPacket();
282
283             digitalWrite(D7,HIGH);
284             delay(100);
285             digitalWrite(D7,LOW);
286         }
287     }
288 }

```

## 4、移动视频监控平台部分程序

接收遥控信号接收、舵机控制程序：

```

22 int main(void)
23 {
24     uint8_t tmp_buf[33];
25     uint8_t DJ1_ANGLE = 5; // 舵机1角度
26     uint8_t DJ2_ANGLE = 15; // 舵机2角度
27
28     delay_init(48);
29     STM32F030_GPIO_Configure(); // GPIO引脚初始化
30     USART1_Init(115200);
31
32     NRF24L01_Init(); // NRF初始化
33     printf("\r\n 000\r\n");
34     while(NRF24L01_Check()) // 等待NRF24L01准备就绪
35     {
36     }
37     NRF24L01_RX_Mode();
38     STM32F030_ADC_Config(); // ADC初始化
39     OLED_Init(); // OLED 初始化
40     OLED_Clear(); // OLED 清屏
41     STM32F0_NVIC_Config();
42     STM32F0_Timer_Config(999,47);
43     STM32F0_Timer14_Config(999,47);
44     TIM_SetCompare1(TIM3, 0);
45     TIM_SetCompare2(TIM3, 0);
46     TIM_SetCompare3(TIM3, 0);
47     TIM_SetCompare4(TIM3, 0);
48     STM32F0_Timer_SG90(200,4800);
49     LED1 = 0; LED2 = 1; LED3 = 1;
50     USART2_Init(115200);
51     tmp_buf[0] = 10;
52     tmp_buf[4] = 10;
53     TIM_SetCompare1(TIM1, DJ1_ANGLE); // 通道1为舵机1
54     TIM_SetCompare4(TIM1, DJ2_ANGLE); // 通道2为舵机2
55     delay_ms(500);
56     while(1)
57     {
58         ADC_ConvertedValueLocal[0] =(float) ADC_ConvertedValue[0]/4096*3.3;
59         if(NRF24L01_RxPacket(tmp_buf)==0)
60         {
61             if((tmp_buf[0]*255+tmp_buf[1]) > 3000 && ((tmp_buf[4]*255+tmp_buf[5]) > 1000 && (tmp_buf[4]*255+tmp_buf[5]) < 3000))
62             {
63                 TIM_SetCompare1(TIM3, 0);
64                 TIM_SetCompare2(TIM3, 1000);
65                 TIM_SetCompare3(TIM3, 1000);
66                 TIM_SetCompare4(TIM3, 0);
67             }
68             else if((tmp_buf[0]*255+tmp_buf[1]) < 1000 && ((tmp_buf[4]*255+tmp_buf[5]) > 1000 && (tmp_buf[4]*255+tmp_buf[5]) < 3000))
69             {
70                 TIM_SetCompare1(TIM3, 1000);
71                 TIM_SetCompare2(TIM3, 0);
72                 TIM_SetCompare3(TIM3, 0);
73                 TIM_SetCompare4(TIM3, 1000);
74             }
75             else if((tmp_buf[4]*255+tmp_buf[5]) > 3000 && ((tmp_buf[0]*255+tmp_buf[1]) < 3000 || (tmp_buf[0]*255+tmp_buf[1]) > 1000))
76             {
77                 TIM_SetCompare1(TIM3, 500);
78                 TIM_SetCompare2(TIM3, 0);
79                 TIM_SetCompare3(TIM3, 500);
80                 TIM_SetCompare4(TIM3, 0);
81             }
82             else if((tmp_buf[4]*255+tmp_buf[5]) < 1000 && ((tmp_buf[0]*255+tmp_buf[1]) < 3000 || (tmp_buf[0]*255+tmp_buf[1]) > 1000))
83             {
84                 TIM_SetCompare1(TIM3, 0);
85                 TIM_SetCompare2(TIM3, 500);
86                 TIM_SetCompare3(TIM3, 0);
87                 TIM_SetCompare4(TIM3, 500);
88             }
89
90             else if((tmp_buf[4]*255+tmp_buf[5]) > 3000 && ((tmp_buf[0]*255+tmp_buf[1]) > 3000 || (tmp_buf[0]*255+tmp_buf[1]) < 1000))
91             {
92                 TIM_SetCompare1(TIM3, 500);
93                 TIM_SetCompare2(TIM3, 0);
94                 TIM_SetCompare3(TIM3, 500);
95                 TIM_SetCompare4(TIM3, 0);
96             }
97             else if((tmp_buf[4]*255+tmp_buf[5]) < 1000 && ((tmp_buf[0]*255+tmp_buf[1]) > 3000 || (tmp_buf[0]*255+tmp_buf[1]) < 1000))
98             {
99                 TIM_SetCompare1(TIM3, 0);
100                 TIM_SetCompare2(TIM3, 500);
101                 TIM_SetCompare3(TIM3, 0);
102                 TIM_SetCompare4(TIM3, 500);
103             }
104             else
105             {
106                 TIM_SetCompare1(TIM3, 0);
107                 TIM_SetCompare2(TIM3, 0);
108                 TIM_SetCompare3(TIM3, 0);

```

```
109     TIM_SetCompare4(TIM3, 0);
110 }
111
112 if(tmp_buf[8] == 1)
113 {
114     TIM_Cmd(TIM14, ENABLE);
115     if(DJ_Count >= 450)
116     {
117         DJ1_ANGLE ++;
118         TIM_SetCompare1(TIM1, DJ1_ANGLE);
119         if(DJ1_ANGLE>25){ DJ1_ANGLE = 25;}
120         DJ_Count = 0;
121     }
122 }
123 else if(tmp_buf[8] == 3)
124 {
125     TIM_Cmd(TIM14, ENABLE);
126     if(DJ_Count >= 450)
127     {
128         DJ1_ANGLE --;
129         TIM_SetCompare1(TIM1, DJ1_ANGLE);
130         if(DJ1_ANGLE<=5){ DJ1_ANGLE = 5;}
131         DJ_Count = 0;
132     }
133 }
134 else if(tmp_buf[8] == 6)
135 {
136     TIM_Cmd(TIM14, ENABLE);
137     if(DJ_Count >= 450)
138     {
139         DJ2_ANGLE --;
140         TIM_SetCompare4(TIM1, DJ2_ANGLE);
141         if(DJ2_ANGLE<=5){ DJ2_ANGLE = 5;}
142         DJ_Count = 0;
143     }
144 }
145 else if(tmp_buf[8] == 5)
146 {
147     TIM_Cmd(TIM14, ENABLE);
148     if(DJ_Count >= 450)
149     {
150         DJ2_ANGLE ++;
151         TIM_SetCompare4(TIM1, DJ2_ANGLE);
152         if(DJ2_ANGLE<=5){ DJ2_ANGLE = 5;}
153         DJ_Count = 0;
154     }
155 }
156 else if(tmp_buf[8] == 2)
157 {
158     TIM_Cmd(TIM14, ENABLE);
159     if(DJ_Count >= 450)
160     {
161         DJ1_ANGLE = 5; DJ2_ANGLE = 15;
162         TIM_SetCompare1(TIM1, DJ1_ANGLE);
163         TIM_SetCompare4(TIM1, DJ2_ANGLE);
164         DJ_Count = 0;
165     }
166 }
167 else
168 {
169     TIM_Cmd(TIM14, DISABLE);
170 }
171
172 if(ESP_Dat == 'Q')
173 {
174     TIM_SetCompare1(TIM3, 0);
175     TIM_SetCompare2(TIM3, 1000);
176     TIM_SetCompare3(TIM3, 1000);
177     TIM_SetCompare4(TIM3, 0);
178 }
179 else if(ESP_Dat == 'H')
180 {
181     TIM_SetCompare1(TIM3, 1000);
182     TIM_SetCompare2(TIM3, 0);
183     TIM_SetCompare3(TIM3, 0);
184     TIM_SetCompare4(TIM3, 1000);
185 }
186 else if(ESP_Dat == 'Z')
187 {
188     TIM_SetCompare1(TIM3, 500);
189     TIM_SetCompare2(TIM3, 0);
190     TIM_SetCompare3(TIM3, 500);
191     TIM_SetCompare4(TIM3, 0);
192 }
193 else if(ESP_Dat == 'Y')
```

```
194  
195 {  
196     TIM_SetCompare1(TIM3, 0);  
197     TIM_SetCompare2(TIM3, 500);  
198     TIM_SetCompare3(TIM3, 0);  
199     TIM_SetCompare4(TIM3, 500);  
200 }  
201 else if(ESP_Dat == 'S' && (tmp_buf[4]*255+tmp_buf[5])>1000 && (tmp_buf[4]*255+tmp_buf[5])<3000  
202        && (tmp_buf[0]*255+tmp_buf[1])>1000 && (tmp_buf[0]*255+tmp_buf[1])<3000)  
203 {  
204     TIM_SetCompare1(TIM3, 0);  
205     TIM_SetCompare2(TIM3, 0);  
206     TIM_SetCompare3(TIM3, 0);  
207     TIM_SetCompare4(TIM3, 0);  
208 }  
209 delay_ms(10);  
210 }
```

自 2018 年 9 月踏入江西理工大学研究生院至今已经接近 3 年的时间了，这三年时光真的是转瞬即逝，这意味着我整个研究生学习阶段即将落幕。这是个伤心的日子，真的觉得这么久都没有好好把握却已经过完；还没有和老师、朋友、同学相处够却已经过完；还没有将自己的研究水平上升更高的高度却已经过完。这也是一个高兴的日子，虽然研究生阶段已经过完，但人生还很长，还需要去继续探索下一个人生阶段。三年以来，在诸多良师益友的帮助下，自己成长很多，无论在知识、见识、学识上都有很大的飞跃。在此，我要向在这三年以来对我伸出援助之手的老师、朋友、同学表达我的感激之情。

首先，我要表达对我的导师任老师真诚的感激与感谢。恩师于我，如春晖之于寸草，得求学于此，生而有幸！正是老师的敦敦教诲，让我在研究生阶段受益匪浅，无论是在学习、论文、生活当中，任老师都给了我无数的帮助，让我这三年学到了丰富的知识。在此，我要再次表达我对任老师最真诚的感谢，我也要送上我的祝福，祝福老师及其家人平安顺遂，美满幸福。

三年的学习当中，有一个地方陪伴了我大部分的学习时光，那就是江西理工大学信息工程学院 N307 实验室，在这个实验室里有很多像我一样的小伙伴，有师兄师姐、师弟师妹，在这里更像一个温暖的家，大家一起学习，一起讨论，一起进步。在论文撰写期间，也得到了实验室许多小伙伴的支持，在此感谢你们的帮助，我们的友谊我将铭记于心。

然后我要感谢我的女朋友，我和我女朋友一起从江西理工大学本科毕业，之后我读研究生，她出去工作，等我。感谢她这三年来的不离不弃与陪伴，感谢一路上对我的照顾与爱，希望我们以后会越来越好，一起组建温暖的大家庭。

最后，我要谢谢我的家人，是你们的鼓励与支持让我进入了研究生阶段的学习，给了我无穷的动力，在我迷茫期间，有家和家人给了我温暖，给了我人生的指引。感谢家人，未来我将会更加努力，书写人生新篇章。

最后，我要感谢家人的关爱与支持，正是家人的存在让我感受到了温暖，也是家人给了我学习的动力，让我有更强的意志力一路学习下去。感谢你们，未来的路将更加美好。

研究生阶段即将谢幕，时光荏苒，总有不舍，不过还是要以积极的态度面对以后的人生。在未来的工作和学习中，我也将会拿出勇于拼搏的精神去迎接。再次对有恩于我、帮助过我的你们致以最诚挚的感谢。感谢成长路上遇见你们，我将铭记于心。



**已发表论文:**

1. 任克强, 王传强. 基于物联网的室内数据采集监控系统[J]. 液晶与显示, 2020, 35(2): 136-142.
2. 任克强, 王传强. 基于 STM32F4 的多通道串口驱动 TFT 液晶屏显示系统设计[J]. 液晶与显示, 2020, 35(5): 449-455.
3. 任克强, 王传强. 嵌入式可移动无线视频监控平台的设计[J]. 传感器与微系统, 2020(已录用).
4. 任克强, 王传强. 基于 STC8F1K08S2 的串口驱动 OLED 显示系统设计[J]. 电子器件, 2021 (已录用).

**获得奖励:**

1. 在 2019-2020 学年中获得“三好研究生标兵”.
2. 获得江西省研究生数学建模比赛三等奖.
3. 参加中国研究生电子设计竞赛江西理工大学选拔赛两次分别获得二等和三等奖.