

沈阳工程学院

本科毕业设计论文

基于单片机的室内移动机器人的设计与应用

**Design and Application of Indoor Mobile Robot Based on
Microcontroller**



学 院：	自动化学院
专 业：	机器人工程
班 级：	机器人 202
学生姓名：	王令硕
学生学号：	2020311218
指导教师：	刘炜

沈阳工程学院学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目： 基于单片机的室内移动机器人的设计与应用

作者签名： _____ 日期： _____ 年 _____ 月 _____ 日

沈阳工程学院学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于沈阳工程学院，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目： 基于单片机的室内移动机器人的设计与应用

作者签名： _____ 日期： _____ 年 _____ 月 _____ 日

导师签名： _____ 日期： _____ 年 _____ 月 _____ 日

摘 要

移动机器人作为机器人领域研究的一个重要方向，目前也已取得了突破性的发展。移动机器人的种类繁多，其中不同的结构特性会影响机器人整体系统运行的稳定性，履带式移动机器人以高负载，高稳定性在众多移动机器人类型中脱颖而出，完美解决了移动机器人稳定性不足的缺陷。

本文以履带式移动机器人作为研究对象的基础平台，且重点关注机器人环境视觉感知问题，环境温湿度感知问题，和机器人自身姿态检测与非自动控制方式的控制策略问题。针对环境视觉感知问题的研究，本文以嵌入式 Linux 系统作为研究平台，对环境的图像数据进行采集与分析展示。除此之外嵌入式 Linux 系统不仅负责着机器人的环境视觉感知任务，还担任着机器人主控系统的角色，对机器人传感器采集及驱动系统发来的温湿度数据进行处理并通过无线数据传输的方式传输到机器人无线控制系统端。

对于环境温湿度感知问题和机器人自身姿态角度检测问题本文以基于 STM32 单片机的 FreeRTOS 实时操作系统作为机器人的传感器采集系统的研究平台。除此之外为了机器人系统运行的高效性，本设计直接将传感器感知系统与机器人驱动系统集成到了一起，为传感器数据采集及驱动系统。此系统负责机器人驱动与环境温湿度感知和机器人自身姿态角度检测的任务，传感及驱动系统通过 CAN 总线将采集到的机器人姿态角度数据与环境感知数据发送到机器人主控系统，机器人主控系统将数据进行处理后再通过蓝牙串口模块以无线的方式发送到机器人无线控制系统上进行显示。对于机器人的控制方式，本设计采用摇杆控制策略。机器人无线控制系统在 STM32 单片机运行的 FreeRTOS 操作系统上，通过 ADC 外设检测到摇杆 X/Y 的原始数据后，经过摇杆动作检测处理算法，处理生成机器人的运动控制命令，然后使用 UART 外设连接蓝牙串口模块，以无线的方式将机器人运动控制命令发送到机器人主控系统。主控系统通过 CAN 总线发送到传感器数据采集及驱动系统，传感及驱动系统根据收到的控制命令，进行相应的机器人驱动。

关键词 单片机，嵌入式 Linux 系统，实时操作系统，传感器，机器人

Abstract

Mobile robots, as an important research direction in the field of robotics, have also achieved breakthrough development. There are various types of mobile robots, among which different structural characteristics can affect the stability of the overall system operation of robots. Tracked mobile robots stand out among many types of mobile robots with high load and stability, perfectly solving the problem of insufficient stability of mobile robots.

This article focuses on the basic platform of tracked mobile robots as the research object, and focuses on the visual perception of the robot environment, as well as the perception of environmental temperature and humidity, as well as the control strategy of the robot's own posture detection and non automatic control methods. In response to the research on environmental visual perception, this article uses an embedded Linux system as the research platform to collect and analyze image data of the environment. In addition, the embedded Linux system is not only responsible for the robot's environmental visual perception task, but also plays the role of the robot's main control system, processing the temperature and humidity data collected by the robot's sensors and driving system, and forwarding it wirelessly to the robot's wireless control system.

For the issues of environmental temperature and humidity perception and robot pose angle detection, this thesis uses the FreeRTOS real-time operating system based on the STM32 microcontroller as the research platform for the sensor acquisition system of the robot. In addition, in order to improve the efficiency of the robot system operation, this design directly integrates the sensor perception system with the robot drive system, which is called the sensor data acquisition and drive system. This system is responsible for driving the robot, environmental temperature and humidity perception, and robot pose angle detection. The sensor and drive system sends the collected robot pose angle data and environmental perception data to the robot control system through the CAN bus. The robot control system processes the data and then wirelessly sends it to the robot wireless control system through the Bluetooth serial port module for display. The control method of this design adopts manual control strategy for research. The robot wireless control system runs on the FreeRTOS operating system of the STM32 microcontroller. After detecting the raw data of the joystick X/Y through the ADC peripheral, it is processed through the joystick action detection algorithm, Process and generate motion control commands for the robot, then connect the UART peripheral to the Bluetooth serial port module and wirelessly send the robot motion control commands to the robot control system. The control system sends the commands to the sensor data acquisition and driving system via the CAN bus, and the sensing and driving system drives the robot accordingly based on the received control commands.

Key words microcontroller, Embedded Linux system, Real time operating system, Sensors, Robots

目 录

摘 要	I
Abstract	II
目 录	III
第 1 章 绪论	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	1
1.2 国内外研究现状	1
1.3 本文主要研究内容	3
第 2 章 机器人系统设计	5
2.1 引言	5
2.2 机器人系统整体方案设计	5
2.3 机器人驱动方案选择	7
2.4 机器人主控制系统设计	8
2.5 机器人无线控制系统设计	10
2.6 机器人传感及驱动系统设计	12
2.6.1 机器人传感器数据采集方案设计	12
2.6.2 机器人力学建模与驱动电机选择	13
2.7 本章小结	17
第 3 章 机器人硬件电路系统设计	18
3.1 引言	18
3.2 机器人主控制系统硬件电路	18
3.2.1 I.MX6ULL 核心工作电路	18
3.2.2 蓝牙串口模块电路	19
3.2.3 CAN 总线电路	21
3.3 机器人传感器数据采集及驱动系统硬件电路	21
3.3.1 STM32F103C8T6 核心工作电路	21
3.3.2 CAN 总线电路	23
3.3.3 温湿度传感器电路	24
3.3.4 姿态角度传感器电路	25
3.3.5 永磁有刷直流电机驱动电路	27
3.3.6 OLED 显示屏电路	28
3.4 机器人无线控制系统硬件电路	30
3.4.1 STM32F103C8T6 核心工作电路	30
3.4.2 TFT-LCD 显示屏电路	31
3.4.3 摇杆电路	33
3.5 本章小结	33
第 4 章 机器人软件系统设计与测试	34
4.1 引言	34

4.2 机器人主控嵌入式 linux 系统.....	34
4.2.1 Linux CAN 驱动	34
4.2.2 Linux Uart 驱动	35
4.2.3 Linux V4L2 for Camera 驱动	36
4.2.4 基础性功能测试.....	37
4.2.5 系统性测试.....	39
4.3 机器人传感器数据采集及驱动系统.....	39
4.3.1 STM32F103 移植 FreeRTOS	39
4.3.2 STM32F103 UART 驱动.....	40
4.3.3 STM32F103 CAN 通信驱动	41
4.3.4 STM32F103 IIC 驱动传感器	41
4.3.5 STM32F103 PWM 驱动直流电机.....	42
4.3.6 基础性功能测试.....	43
4.3.7 系统性测试.....	46
4.4 机器人无线控制系统.....	46
4.4.1 STM32F103 UART 驱动.....	46
4.4.2 STM32F103 ADC 驱动摇杆	47
4.4.3 STM32F103 SPI 驱动 LCD 显示屏.....	49
4.4.4 基础性功能测试.....	50
4.4.5 系统性测试.....	51
4.5 机器人整体系统测试.....	51
4.6 本章小结	52
第 5 章 总结.....	53
致 谢.....	54
参考文献.....	55

第1章 绪论

1.1 研究背景及意义

1.1.1 研究背景

最近几年机器人技术已经成为现代科技领域的研究热点。特别是在室内环境中，移动机器人的应用场景日益增多^[1]。传统的室内环境，像工厂中的维护服务和管理多依赖于人力，人力不仅效率低下，而且在某些特殊或危险情况下，人工操作还存在着极大的风险。因此，研发出一款具有高效、智能、的室内移动机器人，已成为当前机器人领域的重点研究问题。

随着物联网、SLAM、人工智能等技术不断发展，室内移动机器人的研究也有了强大的技术支持。机器人可以通过智能感知、自主学习、智能决策等能力，更好地应对复杂的室内环境情况，解决各种复杂的任务^[2]。

尽管室内移动机器人的研究进程取得了一定的进展，但是仍面临诸多实际问题的技术挑战。如环境感知的精度和实时性、路径规划的智能性和有效性、人机交互的便捷性等问题，都需要进一步的研究和探索。

1.1.2 研究意义

室内移动机器人的应用，可以大幅提升室内环境中的工作效率，降低人力成本。例如：在商场、医院、工厂等场所，机器人可以代替人工进行巡检、清洁、运输等工作，减轻工作人员的负担，提高工作效率。在某些特殊或危险的环境下，如火灾现场、核电站等，人工操作存在着较大风险。而室内移动机器人可以在这些环境下代替人工进行作业，提高安全性和可靠性。同时，机器人还可以通过实时监控和数据分析检测，提前发现和预防潜在的安全隐患。室内移动机器人的研发和应用，还将带动相关产业的发展，如传感器技术、控制技术、通信技术、人工智能技术^[3]等。这些技术的发展，也将进一步推动机器人技术的发展，也为经济社会发展注入新的动力。

并且随着室内移动机器人技术的不断发展，未来机器人将能够更好地融入人们的生活日常，提供更加便捷、高效的服务。例如，通过智能家居系统，机器人可以帮助人们完成家务劳动、照顾老人和孩子、提供娱乐和休闲服务等，从而提升人们的生活质量和便利性。

1.2 国内外研究现状

室内移动机器人的研究最早可以追溯到二十世纪六七十年代，随着计算机、传感器和人工智能等技术的不断进步，这一领域取得了显著的进展。

上世纪 70 年代，美国斯坦福国际研究所（Stanford Research Institute, SRI）研制了 Shakey 机器人，这是 20 世纪最早的移动机器人之一。Shakey 机器人^[4]引入了人工智能的自动规划技术^[5]，具备一定的人工智能，能够自主进行感知、环境建模、行为规划并执行任务。

随后在八十年代，美国科学家、斯坦福大学的研究生 Moravec 制造了具有视觉能力可以自行在房间内导航并规避障碍物的“斯坦福车”（Stanford Cart）^[6]，这可以说是现代无人驾驶汽车的始祖。

随着技术的不断进步，室内移动机器人的发展也逐渐进入了智能阶段，这一阶段的机器人逐渐具备了情景感知、自主推理决策的能力，在 1985 年美国宾夕法尼亚州匹兹堡卡内基梅隆大学机器人研究所的 J.Crowley 提出了一种基于旋转式的超声波传感器的智能移动机器人导航系统^[7]，该系统基于一个动态维护的本地环境模型，称为复合本地模型。它整合了来自旋转超声波范围传感器、机器人触摸传感器和预先学习的全局模型的信息，以帮助机器人在其环境中移动。

1988 年美国密苏里州圣路易斯华盛顿大学科学与数学系的 E.Y.Rodin 和 S.M.Amin 提出了一种移动机器人的导航算法^[8]，该算法是用于解决在充满移动障碍物的环境中自主移动机器人的无碰撞路径规划和实时控制问题。所提出的方法基于几何表示/多目标 A*搜索和路径平滑/转向控制技术，并试图将几何结构引入到一个缺乏任何先前结构的范式中，以及多对象搜索技术。

移动机器人多用电机作为驱动机构，又因为是移动型的机器人，所以就不可能来连接着根长长的电线，为机器人提供电能，在 2000 年澳大利亚国立大的 S Oh, A Zelinsky, KTaylor, 提出了一种移动机器人的自动充电技术，研究使用的澳大利亚国立大学开发的 Nomad XR4000 移动机器人，利用其内置传感器控制与简单充电站对接^[9]。

在室内移动机器人的发展过程中，自主性、适用性和智能通信成为其重要的发展方向。随着无线网络和移动终端的普及，多机器人间的通信互联协作也变得更为方便，这为室内移动机器人的进一步发展提供了有力的支持。2010 年美国卡内基梅隆大学机器人研究所的 Joydeep Biswas 和卡内基梅隆大学计算机科学系 Manuela M.Veloso 提出了一种基于 WiFi 定位与室内导航的移动机器人^[10]，该研究团队通过构建 WiFi 信号特征地图，结合几何约束，设计了一种移动机器人自主导航的室内环境系统。此系统利用 WiFi 传感数据，通过连续的感知模型，将离散的基于图的 WiFi 信号强度采样转化为对环境的连续感知。这种方法提高了机器人在室内环境中的定位精度和导航能力。此外，他们还提出了一种 WiFi 定位算法，该算法能够连续利用感知模型来更新和优化机器人的位置信息。这一算法在实际应用中表现出了良好的性能，为自主室内移动机器人的定位与导航提供了新的解决方案。

2021 年，巴西布拉干萨理工大学的数字化和智能机器人研究团队介绍了一种改进室内定位系统的方法，可以提高移动机器人的定位精度^[11]。传统的 SLAM 和 AMCL 技术结合在动态环境和障碍物存在的情况下容易出现误差，为解决这一挑战，团队提出了一种新的定位系统，将 AMCL 算法与基于人工视觉检测散布在环境中的标志物的位置更新和校正相结合，以减少 AMCL 定位估计累积的误差，该方法基于 ROS，经过模拟环境测试和验证，结果表明，使用 SLAM 系统与传统 AMCL 结合，并根据人工视觉检测到的标志物进行校

正，可以改善机器人的轨迹。

相较于国外的室内移动机器人研究，国内的研究并不多，且起步比较晚，起步初期研究的技术也较为落后，1991年国内的中国科学院沈阳自动化研究所的李育发就对室内移动机器人基于多传感器融合的感知系统与室内建模进行了研究^[12]。

2001年中南大学信息工程学院的智控所的许永华等人，设计一款基于图像识别技术的具有视觉避障移动机器人，让室内移动机器人具有了视觉能力^[13]。

2003年浙江大学信电系信息与通信工程研究所的项志宇，刘济林设计了一款基于完整线段匹配算法的拥有自定位能力的室内移动机器人，让室内移动机器人拥有了自主定位的能力^[14]。

2005年大连理工大学的庄严设计了一款基于多传感器数据融合的定位与地图创建的移动机器人^[15]，此研究让室内移动机器人实现了实时定位与地图构建能力，让室内移动机器人的发展更进了一步。

2014年华中科技大学自动化学院的胡春旭等人设计了一款室内移动机器人，这款机器人基于ROS系统和SLAM技术且具有实时环境三维环境建模能力与导航定位能力^[16]，此项研究让机器人用了类人一样的环境感知能力。

2019年，中国矿业大学的机器人研究团队陈畅、朱华、王磊和刘宇在室内未知环境下移动机器人的视觉惯性同时定位与地图构建（SLAM）问题上提出了一种研究方法^[17]。在低纹理或重复纹理的室内环境中，移动机器人的图像特征容易丢失，使得在狭窄区域内来回移动的机器人姿态估计变得困难。为了解决这一跟踪问题，研究者提出了一种基于时空圆匹配序列（STCM）的单圈特征匹配方法，并基于该方法设计了STCM-SLAM技术。该技术紧密地结合了立体相机和惯性测量单元（IMU），以更好地估计室内移动机器人的姿态。此外，研究还采用了前后向光流来跟踪图像特征，与相关光流相比，STCM的绝对精度和相对精度分别提高了37.869%和129.167%。

2020年东南大学的袁杰设计了一款具有路径规划能力室内移动机器人^[18]，此研究在前人研究的实时环境三维建模与自主定位的基础上加入了路径规划和避障能力，让移动机器人更像人一样，不仅具有环境感知能力，还有路径规划和避障能力。

1.3 本文主要研究内容

本文以履带式移动机器人作为室内移动机器人的研究对象，且采用无导航人工控制的方式，为了实现室内移动机器人的课题研究，首先进行了机器人的详细系统方案设计，根据所设计的方案再进行各个子系统模块部分的硬件设计，之后在硬件平台的基础上进行驱动软件的设计。最后所有的子系统的软硬件都设计好之后，进行子系统的基础性功能测试和综合性系统测试。本文的具体章节安排如下。

第1章：对室内移动机器人的背景及其研究意义，与国内外研究现状和本文研究内容进行了概述。

第2章：对机器人系统进行了总体的方案设计和工作流程规划。例如：机器人驱动方式的选择，以及机器人主控系统设计，无线控制系统的设计与传感器数据采集及驱动系统

方案设计，系统设计内容主要包括单片机的选择，传感器型号的的选择，和通信方式的选择。

第 3 章：根据设计好的方案，进行子系统各个模块的硬件电路设计，例如：单片机核心工作电路设计与传感器显示屏电路设计还有 CAN 总线通信模块电路设计，与蓝牙串口模块通信电路设计，直流电机驱动电路设计等等。

第 4 章：在设计好的硬件电路基础上进行驱动软件编写，主要包括为 Linux 系统上的 CAN 总线驱动，UART 驱动与 V4L2 驱动和 STM32 上的 RTOS 移植，与 CAN 总线外设驱动，UART 驱动，ADC 驱动，与 IIC 驱动传感器和 SPI 驱动显示屏，PWM 驱动电机。写好驱动之后还需要进行基础性的功能测试，例如测试 CAN 是否可以正常收发数据，IIC 驱动的传感器是否可以正常采集数据等等，基础功能测试好之后再进行系统性的测试。

第 5 章：对本文工作内容进行全面性的总结。

第 2 章 机器人系统设计

2.1 引言

室内移动机器人是个综合系统如图 2.1，细分为机械系统，感知系统，硬件电路系统，控制系统，以及软件系统，所以机器人系统设计又分为机器人的机械结构设计，硬件电路设计，软件设计，运动控制算法设计，和感知系统设计。本课题的机器人系设计主要为硬件电路系统设计和感知系统设计，驱动系统设计以及软件系统设计，本设计为了提高机器人整个系统运行的高效性，将控制系统的驱动部分与感知系统集成成了一个整体，为传感器数据采集及驱动系统。

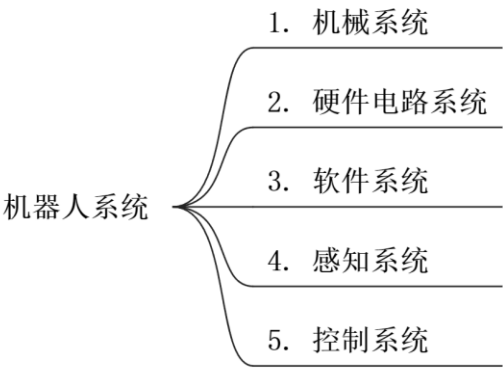


图 2.1 机器人系统组成结构图

2.2 机器人系统整体方案设计

本文主要研究机器人环境视觉感知问题，与环境温湿度感知问题，和机器人自身姿态检测与非自动控制方式的控制策略问题，根据需求将任务划分为以下三个子系统如图 2.2 所示，机器人的无线控制系统，与机器人主控系统，机器人传感器数据采集及驱动系统，下面对每个子系统的任务内容进行详细探讨。

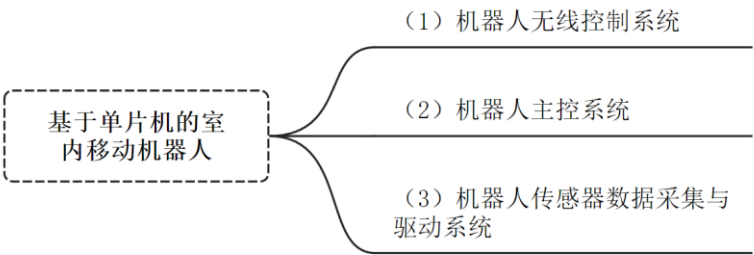


图 2.2 室内移动机器人系统组成图

（1）机器人无线控制系统

主要任务实现机器人的人工控制策略与接收机器人的环境感知数据，例如：环境温湿度数据，与机器人的自身姿态角度数据。

（2）机器人主控系统

主要任务实现机器人的环境视觉感知问题，与作为决策中心中转处理其他子系统的数据。

（3）机器人传感器数据采集及驱动系统

主要任务负责实现机器人的环境感知问题（例：环境中的温湿度数据），与机器人自身姿态角度数据感知，和机器人驱动问题。

根据系统方案设计，机器人系统整体包括机器人主控系统，与机器人无线控制系统，机器人传感器数据采集及驱动系统，其整体系统的硬件组成框架如下图 2.3 所示。

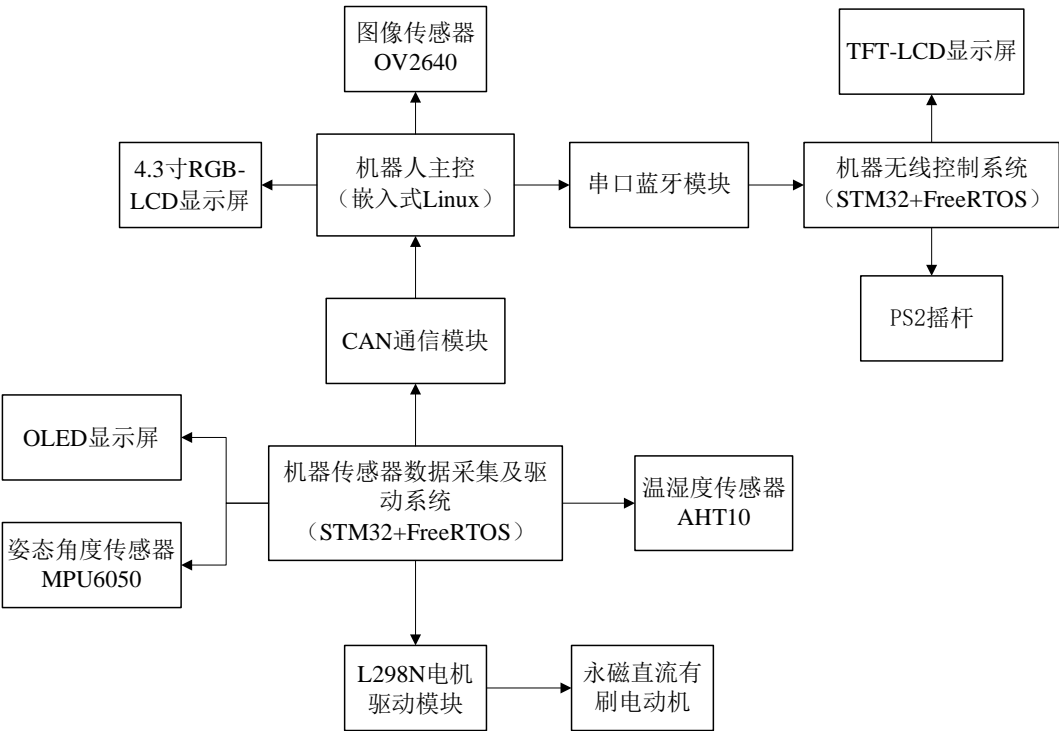


图 2.3 机器人整体系统硬件组成框架图

根据系统任务设计，主控系统与传传感器采集及驱动系统，无线控制系统为同时运行，且是并发关系。主控启动嵌入式 Linux 后开始执行环境图像数据采集任务，并将采集到环境图像数据实时显示在 RGB-LCD 显示屏上，因为主控是多线程运行同时 Linux CAN 总线也在从传感及驱动系统接收传感器数据，并将无线控制系统发来的机器人运动控制命令通过 CAN 总线发送到传感器与驱动系统，传感与驱动系统再分析机器人运动控制命令，并对直流电机做出相应的控制，系统整体的完整工作流程图如下图 2.4 所示。

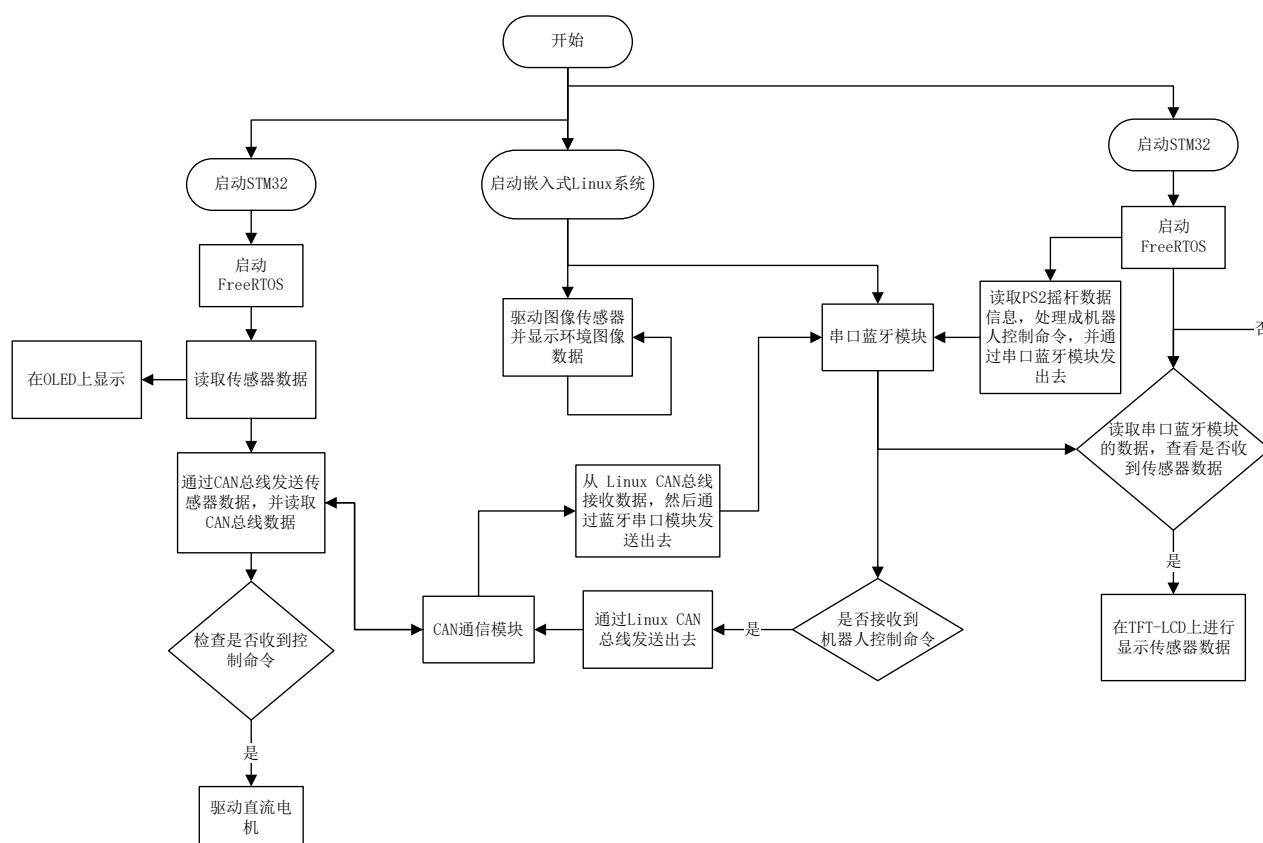


图 2.4 机器人系统整体工作流程图

2.3 机器人驱动方案选择

室内移动机器人多为轮式机器人，轮式机器人又分为独轮式如图 2.5 所示，双轮式如图 2.6 所示，和多轮式机器人如图 2.7 所示，轮式机器人的优点^[19]，移动速度快，易于控制，适用于平整的地面，但缺点是转弯半径大，遇到稍微凹凸不平的地面或者障碍物容易受阻，越障能力差，负载能力差，履带式机器人如图 2.8，则有优秀的地形适应能力，稳定性强，负载能力强，室内环境并不局限于商场家庭这种良好的地面环境，也可能是工厂中，废弃建筑物中的地面，这其中就会遇到众多的障碍物，并且负载根据实际场景不同还会加重。所以本文机器人驱动方式选择履带式驱动^[20]。

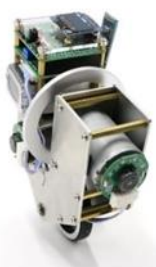


图 2.5 独轮式机器人



图 2.6 双轮式机器人



图 2.7 多轮式机器人



图 2.8 履带式机器人

2.4 机器人主控系统设计

（1）机器人主控软件操作系统选择

机器人主控系统主要负责机器人传感器数据的综合处理，例如接收温湿度传感器的数据与机器人姿态角度的数据，经过处理之后需要通过无线的方式发送给机器人无线控制系统，同时机器人主控系统还负责接收无线控制系统发来的机器人运动控制命令，然后机器人再间接控制电机来控制机器人的运动。除此之外机器人主控还负责图像数据的采集，在这样多任务以及大量数据处理的背景下，决定选择嵌入式 Linux 系统来作为机器人主控系统因为嵌入式 Linux 系统搭载的都是高性能多核处理器，且 RAM 空间也足够大，能够满足图像数据存储的需求，并且嵌入式 Linux 其实是经过裁剪优化后的 Linux，Linux 系统具有优秀且高效的多进程多线程处理能力以及高实时性。

（2）机器人主控与传感器采集及驱动系统通信方式选择

机器人主控与传感器采集及驱动系统之间的主要任务就是传输温湿度数据与机器人姿态角度数据，还有电机驱动命令，室内机器人的应用场景不仅限于家庭，还有其他的诸

多场景例如工厂中，商场，甚至战场中，这些环境下会有很多的物理干扰，最严重的就是电磁干扰，特别是在工厂这种环境中有很多的高压电线与高压用电设备，这些都会产生较强的电磁干扰，直接导致数据传输出现严重的失误，所以决定选择 CAN 总线通信作为主控与传感器与驱动平台的通信方式。因为 CAN 总线是双绞线形式且通信原理是通过两线的电压差来表示所传输的二进制数据 0 或者 1 的，双绞线形式让它在物理上更容易抗干扰，一根通信线受干扰另一根通信线也会跟着受干扰，所以所传输的数据内容并不会出现问题。

（3）机器人主控与无线控制系统通信方式的选择

主控与无线控制系统主要传输的是温湿度传感器数据与机器人自身姿态角度数据，还有机器人的运动控制命令，无线通信方式有很多种，例如 2.4GHZ 模块，蓝牙通信，与 WIFI 通信等等，2.4GHZ 模块通信距离长，速度快，但是缺点是技术难度大，功耗高，蓝牙通信则是低功耗，速度快，且技术难度低，WIFI 通信使用技术难度大，且局限性强，需要本地局域网空间，所以综合考虑功耗技术难度以及成本，最终选择蓝牙通信方式。

（4）机器人主控环境图像数据采集

室内移动机器人需要采集环境中的图像数据，就是需要驱动图像传感器去采集图像数据，本文选择使用嵌入式 Linux Video for linux2（简称 Linux V4L2）框架去驱动图像传感器来采集环境中的图像数据。Linux V4L2 是 Linux 内核中用于支持视频设备的框架。它提供了一组 API 和驱动程序，用于在 Linux 系统上捕获、处理和输出视频数据。V4L2 的主要目的是允许用户空间应用程序与视频设备进行通信，以控制视频流的捕获和处理。

综上所述，机器人主控系统设计包含软件系统的选择，以及通信方式与传感器驱动方式的选择，由此得出机器人主控系统组成框架如图 2.9。

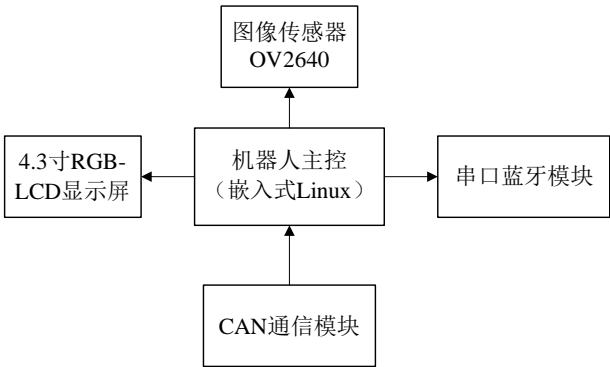


图 2.9 机器人主控系统组成框架图

根据主控系统设计，所有的任务都是基于嵌入式 Linux 系统运行的，所以主控系统工作的第一步就是上电启动嵌入式 Linux 系统，嵌入式 Linux 系统其实就是裁剪优化后的 Linux 系统，Linux 系统是个多进程多线程的操作系统，所以主控系统的开发，充分利用的 Linux 系统特性的优势，多线程开发，如下图 2.10 分为三个线程 CAN 总线数据处理任务线程，与图像传感器驱动任务线程，以及蓝牙串口模块数据处理任务线程，这三个线程是并行运行的，同时处理数据，例如同时进行 CAN 总线数据收发，图像传感器驱动数据采集与显示，及其蓝牙串口模块数据的收发。

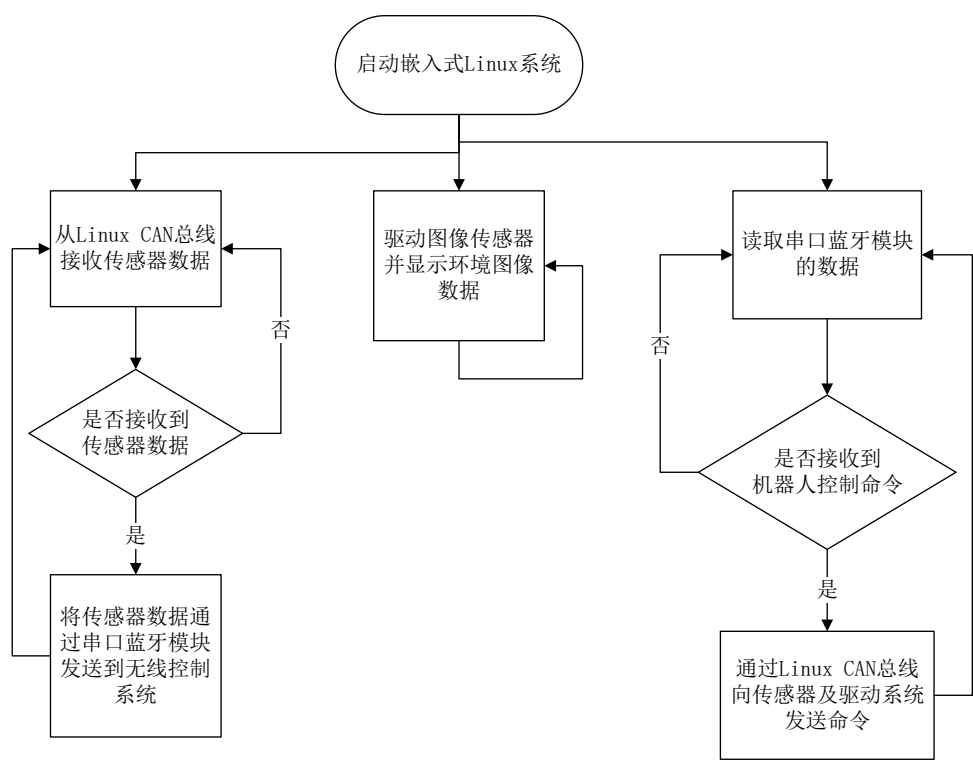


图 2.10 机器人主控制系统工作流程图

2.5 机器人无线控制系统设计

（1）软件系统的选择^[21]

无线控制系统主要负责获取摇杆的数据，经过处理算法处理后生成运动控制命令并向机器人主控发送，此外还进行接收机器人主控传输过来温湿度数据和机器人的姿态角度数据，并在显示屏上进行显示由于任务量较少，所以采用意法半导体（STMICROELECTRONICS）公司的 STM32F103C8T6 单片机作为系统的核心处理器，并运行 FreeRTOS 实时操作系统，因为实时操作系统能高效管理单片机执行的多任务，且有优秀的线程调度能力，与优秀的实时处理能力。

（2）摇杆驱动与无线通信方式^[22]

所选用的摇杆如图 2.11，可以通过 ADC 外设检测摇杆 X 轴 Y 轴方向上的电压值，通过处理算法处理就得出摇杆的信息了。无线通信方式与机器人主控制系统相同，选择低功耗且成本低的蓝牙串口模块，主控作为蓝牙从机，无线控制系统作为蓝牙主机

（3）显示屏的选择

根据任务需求，显示温湿度数据与机器人姿态角度数据，还有的运动控制命令，由于是多个不同类型数据的显示，且数据显示量较小，所以选择具有彩色显示能力的显示屏，1.44 英寸 TFT-LCD 显示屏如图 2.12。



图 2.11 摇杆



图 2.12 1.44 英寸 TFT-LCD

综上所述，机器人无线控制系统设计包含软件系统与处理器的选择，以及控制摇杆的选择与显示屏的选择，其系统组成框架如图 2.13。

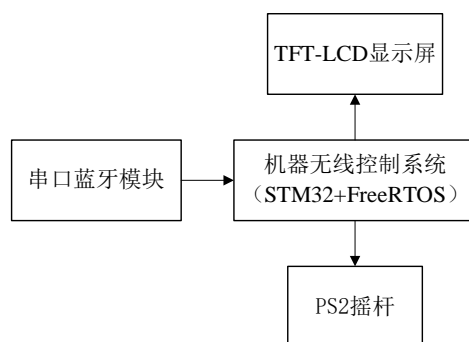


图 2.13 无线控制系统组成框架图

根据无线控制系统设计，系统运行首先启动 STM32，其次启动 FreeTOS，本系统与传感器数据采集及驱动系统相同所有的任务运行都是基于 FreeRTOS 的，所以要启动 FreeRTOS。

之后就要进行摇杆信息数据的采集与处理，处理成机器人运动控制命令通过蓝牙串口模块发送到机器人主控系统，同时也在 TFT-LCD 显示屏上显示发送出去的机器人运动控制命令，之后读取蓝牙串口模块接收机器人主控发来的传感器数据并在 TFT-LCD 显示屏上显示，系统整体工作流程如下图 2.14 所示。

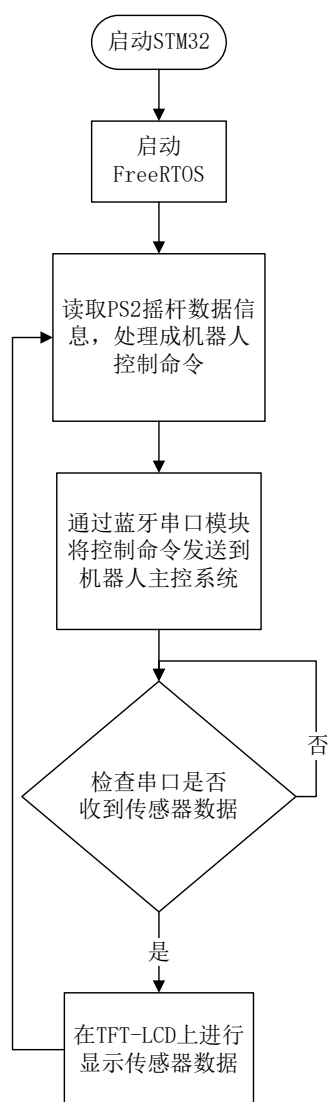


图 2.14 机器人无线控制系统工作流程图

2.6 机器人传感及驱动系统设计

此系统主要任务是进行传感器数据采集及电机驱动，与机器人主控进行数据传输，由于需要处理的数据量较小，所以采用意法半导体公司的 STM32F103C8T6 单片机作为系统的核心处理器，并运行 FreeRTOS 实时操作系统，实时操作系统能高效管理单片机执行的多任务，且有优秀的线程调度能力，与优秀的实时处理能力

2.6.1 机器人传感器数据采集方案设计

(1) 传感器型号选择

传感器采集与驱动系统主要负责温湿度采集与机器人姿态角度数据采集，传感器的选择直接影响到数据的准确性和系统的性能。温湿度采集选用用国产奥松电子公司旗下的 A

HT10 传感器^[23]，该传感器在智能产品设计领域有着广泛的应用，反映了其优秀性能。且 AHT10 传感器具有高精度、高灵敏度和低功耗的特点，能够准确地采集温度和湿度数据，为系统提供可靠的环境信息。姿态角度传感器^[24]采用 MPU6050，MPU6050 目前市场上最为常见且使用最广泛的姿态角度传感器，它优秀的精度得到了广大用户的认可，其灵敏度与性能也处于业界一流水平。

2.6.2 机器人力学建模与驱动电机选择

本文所研究的为移动型机器人，所采用驱动形式为履带式驱动，履带式驱动的本质为电机驱动转动轴，转动轴带动履带链条转动，实现机器人的移动，因此就需要考虑电机的选型问题，履带式机器人常用的电机一般分为两种，一种是永磁直流无刷电动机（BLDC）如图 2.15 和永磁直流有刷电动机如图 2.16，本设计采用永磁直流有刷电机^[25]，因为其具有驱动方式简单，结构简单，可靠性高的优势，而且这种类型的电机在移动机器人中应用也较为广泛，同时具有较高的功率密度和适应性，能基本满足移动机器人驱动和控制需求。



图 2.15 永磁无刷直流电动机（BLDC）



图 2.16 永磁有刷直流电动机

机器人驱动电机的选型中不仅要考虑电机的种类，还要考虑电机的实际功率负载参数，因为移动机器人通常负载较重，少则几千克多则几十千克，所以本设计采用大扭矩的减速版本的永磁直流有刷电机，减速版的直流有刷电机通过减速齿轮组降低了电机的转速，增大了电机的扭矩值，所以此电机能够输出大扭矩，满足移动机器人高负载的需求，本设计所选用的减速直流有刷电机具体参数如下表 2.1 所示。

表 2.1 永磁直流有刷减速电动机参数表

参数类型	参数值
额定电压值	8V
电压范围	6V~12V
额定电流	0.2A
额定扭矩	13Kg.cm
堵转电流	3A
堵转扭矩	18Kg.cm
额定转速	90±10RPM
减速比	1:31
减速前转速	10000RPM
输出轴	直径 6mmD 型偏心轴

虽然选用了能够输出大扭矩的减速直流有刷电机，但电机扭矩值各有不同，移动机器人负载也可能会发生变化，如果选择了一个小扭矩的驱动电机，将可能会发生移动机器人因为电机扭矩值过小而移动不了，导致电机堵转烧毁电机驱动系统与电机本身的情况。

为了研究此问题，将对机器人力学建模，进行受力分析，计算出电机扭矩值与移动机器人自身重量的线性关系，机器人理想模型受力分析图如下图 2.17 所示。

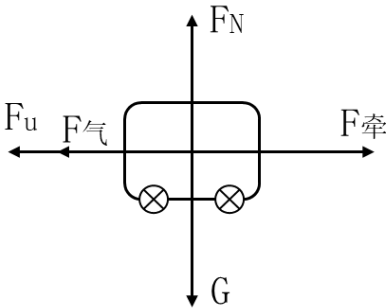


图 2.17 移动机器人模型受力分析图

由上图移动机器人模型受力分析可知，机器人一共受五个力，分别是重力 G ，牵引力 $F_{牵}$ ，支持力 F_N ，摩擦力 F_u ，空气阻力 $F_{气}$ 。

由受力分析可得机器人要想正常移动，牵引力 $F_{牵}$ 必须远远大于摩擦力 F_u 和空气阻力 $F_{气}$ 之和。也就是如下式 (2.1) 关系所示。

$$F_{牵} \gg F_{\mu} + F_{气} \tag{2.1}$$

重力是由地球引力所产生的力公式如下式 (2.2) 所示。

$$G = mg \ (g \approx 9.8m/s^2) \tag{2.2}$$

牵引力 $F_{牵}$ 为所选的永磁直流有刷电机转动的所产生的扭矩提供的力，电机扭矩为一种特殊的力矩，力矩公式如下式 (2.3) 所式。

$$M = F \times L \quad (2.3)$$

因为电机扭矩为一种特殊的力矩，所以由力矩公式（2.3）可推出电机扭矩公式，如式（2.4）所示， F 为电机转动产生的力， r 为转动半径。

$$M = F * r \quad (2.4)$$

由电机扭矩公式可得电机的牵引力 $F_{\text{牵}}$ 如下式（2.5）所示。

$$F = \frac{M}{r} \quad (2.5)$$

在水平状态下支持力 F_N 与重力为一对作用力和反作用力，其关系如下式（2.6）所示。

$$F_N = G \quad (2.6)$$

摩擦力是由于受压力所影响与路面产生的一种阻力，因为压力与支持力是一对作用力与反作用力，所以可以得出摩擦力的公式为（2.7）所示， μ 为摩擦系数， F_N 为支持力。

$$F_u = \mu * F_N \quad (2.7)$$

空气阻力 $F_{\text{气}}$ 为机器人移动是所产生的一种流体阻力，如下式（2.8）所示， C 为空气阻力系数， p 为空气密度， S 为迎风面积， V 为相对空气的移动速度。

$$F_{\text{气}} = \frac{1}{2} C p S V^2 \quad (2.8)$$

综上所述，由此可得出机器人正常移动所需的扭矩值如下式所示。

$$M \gg \left(\mu F_N + \frac{1}{2} C p S V^2 \right) * r \quad (2.9)$$

$$M \gg \left(\mu m g + \frac{1}{2} C p S V^2 \right) * r \quad (2.10)$$

式（2.10）所推算出的机器人驱动电机扭矩公式为纯理论公式，在实际工程中，由于室内机器人的移动速度比较慢，且遇不到高速流体的情况，所以空气阻力 $F_{\text{气}}$ 可忽略不计设置为 0。所以实际工程中的电机扭矩公式可以简化为式（2.11）所示， μ 为摩擦系数，不同的路面摩擦系数不相同， m 为移动机器人全身的质量， g 为重力加速度约为 $9.8m/s^2$ ， r 为旋转半径。

$$M \gg \mu * m * g * r \quad (2.11)$$

摩擦系数 μ 是个变化的值，在有水并且湿滑地板砖路面，和粗糙的地面值差值较大，所以为了机器人的大负载摩擦系数 μ 直接取值为 1.0，为了便于计算 g 取值为 $10m/s^2$ ，由此得出更精简实用的公式，为式（2.12），因此在实际电机选择中根据机器人质量就可计算出最小的理论电机扭矩值了，不过实际的移动机器人还有其他负载，所以选定的电机扭

矩值必须比这个最小理论值要大才可以。

$$M \gg 10 * m * r$$

(2.12)

综上所述，机器人传感器数据采集及驱动系统设计包含软件系统与处理器的选择，以及传感器的选择与机器人驱动方式和驱动电机扭矩值的选择，由此得出机器人传感器数据采集及驱动系统组成框架如图 2.18。

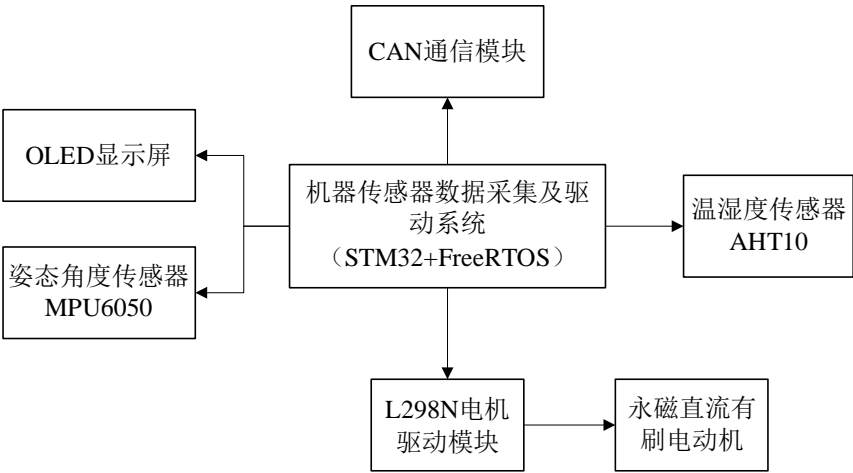


图 2.18 机器人传感器数据采集及驱动系统组成框架图

根据系统设计，首先给 STM32 上电启动，之后就是启动实时操作系统 FreeRTOS，这是个多线程的操作系统，此子系统上的所有任务都是基于 FreeRTOS 开发的，当 FreeRTOS 启动之后，进行温湿度数据读取，和姿态角度数据读取，读取完之后并在 OLED 显示屏上进行实时显示传感器数据，然后还要将采集到的传感器数据通过 CAN 总线发送到机器人主控制系统上，与此同时也从 CAN 总线读取主控制系统发来的机器人运动控制命令，经过解析后根据其中命令含义，然后对直流电机进行相应的控制，完整工作流程如图 2.19 所示。

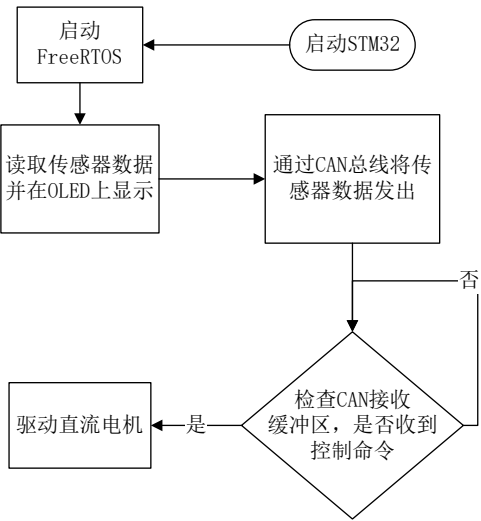


图 2.19 机器人传感及驱动系统工作流程图

2.7 本章小结

本章主要对机器人系统设计进行了全面而详细的介绍，在机器人系统方案设计部分，分别探讨了机器人驱动方案、主控系统方案、无线控制系统方案以及传感器数据采集及驱动系统方案。随后，对机器人系统的工作流程进行了详细的描述。阐述了主控系统的工作流程、传感器与驱动系统的工作流程以及无线控制系统的工作流程。

第3章 机器人硬件电路系统设计

3.1 引言

机器人系统的硬件电路设计是机器人软件系统与感知系统，驱动系统运行的基础。机器人系统的硬件电路系统设计涉及到多个方面，包括处理器核心工作电路的设计，传感器工作电路设计，以及通信电路设计与机器人驱动部分电路设计显示屏电路设计。

3.2 机器人主控系统硬件电路

3.2.1 I.MX6ULL 核心工作电路

机器人主控采用恩智浦（NXP）公司 ARM Coretex-A7 内核 I.MX6ULL 单片机作为核心处理器，I.MX6ULL 是一个低功耗，高性能类型的处理器，CPU 运行频率高达 900MHZ，L2 缓存达到了 128KByte 并且支持 16Bit 的 DDR3/LPDDR2。除此之外恩智浦公司还提供了 I.MX6ULL 低成本，低功耗，高度集成的优化功能应用开发方案，以满足物联网网关，终端节点和消费电子产品的开发要求。并且 I.MX6ULL 具有丰富的内核外设功能，例如多媒体功能，USB 外设，Ethernet 外设，CAN 外设以 UART，IIC，SPI 等多种外设，除此之外还支持视频处理和图像渲染的硬件加速，在操作系统上不仅支持嵌入式版本的 Linux，还是嵌入式实时操作系统，例如 FreeRTOS，这种操作系统支持的灵活性更适合开发人员的使用以应对特定需求的平台，目前 I.MX6ULL 已广泛应用于工业制造，消费电子产品，物联网（IoT），医疗设备，汽车电子等多个领域，接下来将介绍 IMX6ULL 的核心工作电路。

I.MX6ULL 单片机核心工作电路如下图 3.1，主要包括了电源电路，晶振电路，复位电路，与 BOOT 选择电路，MicroSD 存储卡电路。

（1）电源电路

主要作用就是为 I.MX6ULL 单片机芯片进行供电，让整个系统能正常运行。

（2）晶振电路

主要提供基准时钟信号，确保单片机能够按照预定的时间间隔执行指令和操作，还为内核外设提供时钟信号让其正常运行，例如 Timer，看门狗（WDT），PWM，USB，IIC，UART 等等。

（3）复位电路

单片机复位会在上电时进行复位，或者看门狗（WDT）计数溢出时复位，还有一种就是通过将 Reset IO 置高低电平进行主动复位，此电路设计就是将 Reset IO 通过按键连接电源地，当按键按下时让系统产生复位，重新开始运行。

（4）BOOT 选择电路

单片机上电在开始的几个时钟周期内会根据不同的 BOOT IO 电平进行配置系统运行

模式，例如上电后从 FLASH 区域读取并运行程序，或者从 RAM 区域读取并运行程序。

（5）MicroSD 存储卡电路

MicroSD 存储卡是一种 FLASH 存储器具有掉电不丢失的非易失的特性，此电路主要是将 MicroSD 存储卡与 I.MX6ULL 单片机连接起来。

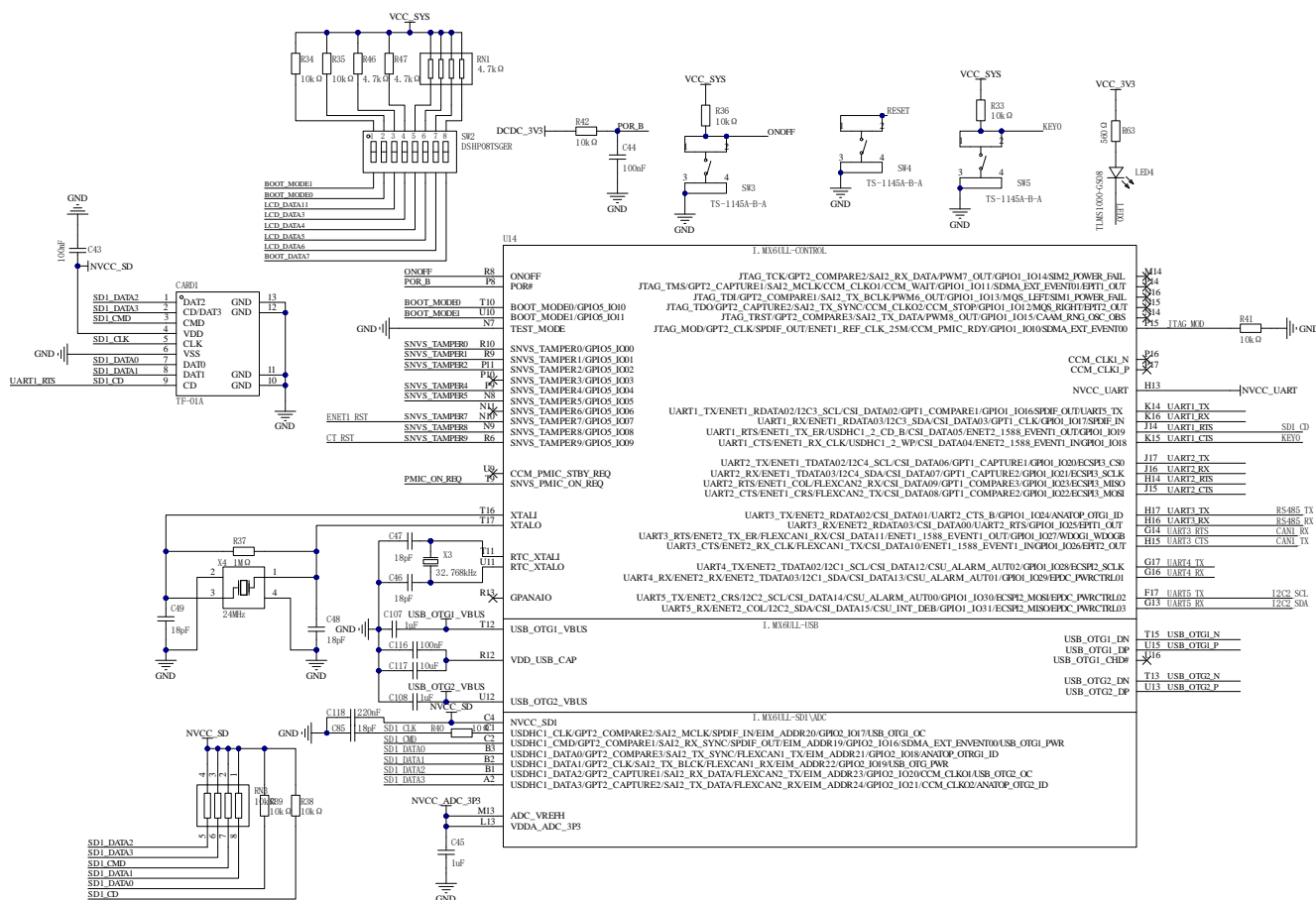




图 3.2 HC04 蓝牙串口模块实物图

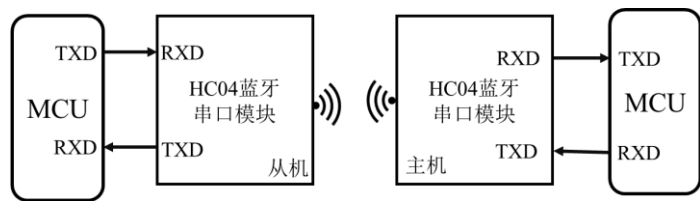


图 3.3 蓝牙串口模块主从机连接关系图

HC04 蓝牙串口模块电路设计如下图 3.4，主要包含复位电路，与 LED 信息提示电路和电源电路。

(1) 复位电路

主要作用，用来强制将蓝牙模块进行复位，当按键按下 Reset IO 置高电平，所有寄存器数据都将被重置，再重新开始工作。

(2) LED 信息提示电路

这个电路设计并不是简单的电源指示灯，而是可以提示蓝牙连接的状态，已连接还是未连接，已连接后 LED 等常亮，未连接状态 LED 灯闪烁。

(3) 电源电路

主要为蓝牙串口模块供电，具体设计为将输入的 5V 直流电经过稳压芯片降压输出+3.3V 的直流电为模块系统供应。

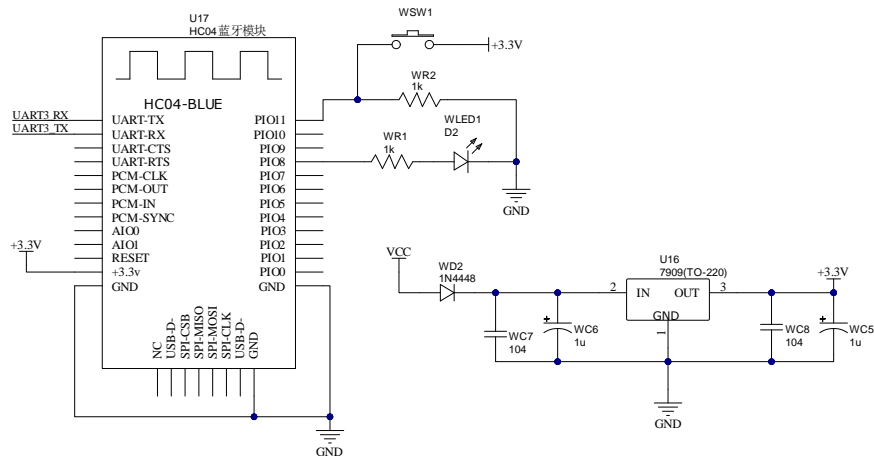


图 3.4 蓝牙串口模块电路图

3.2.3 CAN 总线电路

CAN 总线是目前工业制造领域以及汽车电子领域最常用的一种通信协议，CAN 总线能如此受欢迎且应用如此广泛的原因，主要因为其优秀的且独特的特性，CAN 总线属于是单双工，异步的通信协议，且是差模信号传输，通过 CAN_H 与 CAN_L 两线压差来表示二进制数据 0 或者 1，两根线并且是双绞线的形式，这种通信方式使得 CAN 总线具有强抗电磁干扰的能力，且通信速率高达 1Mbps/S，具体的通信过程是单片机的 CAN 接口接到对应的 CAN 电平转换芯片上，单片机向 CAN 外设缓冲区写入数据，然后经过 CAN 电平转换芯片发送到 CAN 总线上，CAN 总线上的设备也是通过 CAN 电平转换芯片将 CAN 电平转换为单片机的 TTL 电平。其 CAN 总线电路设计如下图 3.5，本课题所使用的 CAN 电平转换芯片为 TJA1050，详细参数见下表 3.1 所示，TJA1050 具有高精度且强抗干扰能力，具体电路设计部分有 TJA1050 芯片与滤波抗干扰电路，CAN_H 与 CAN_L 之间串联了个 120R 的电阻主要作用为吸收信号反射及回波，增强抗干扰模式。

表 3.1 TJA1050 芯片参数	
参数类型	参数值
最大传输速率	1Mbps
CANH,CANL 输入输出电压	-27V~40V
I/O 电压	支持 3.3V/5.5V
工作电压	4.75V ~ 5.25V
工作温度	-40℃ ~ +150℃

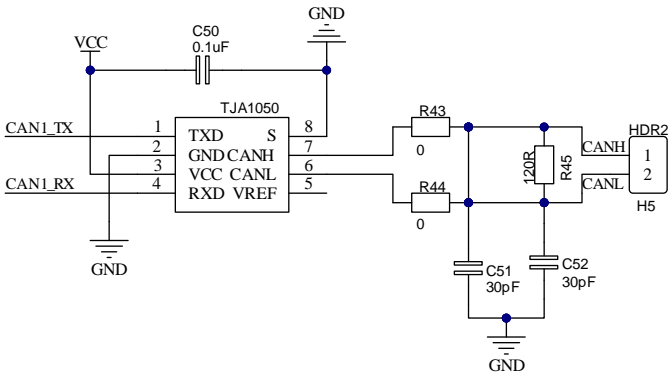


图 3.5 CAN 总线模块电路图

3.3 机器人传感器数据采集及驱动系统硬件电路

3.3.1 STM32F103C8T6 核心工作电路

机器人传感器数据采集及驱动系统所采用的核心处理器为 STM32F103C8T6 单片机，STM32F103C8T6 是意法半导体（STMicroelectronics）公司基于 ARM Cortex-M3 内核设计

的高性能，低功耗且具有强抗干扰性的 32 位单片机，详细参数见下表 3.2 所示。

表 3.2 STM32F103C8T6 单片机资源表

参数类型	参数值
内核	ARM Cortex-M3 32bit RISC 指令集
主频	72MHZ
SRAM	20KByte
Flash	64KByte
I/O 数量	48
工作电压	2.0~3.6V
外设资源	2*SPI, 3*USART, 2*I2C, 1*CAN, 37*OV 口, 2*ADC(12 位/16 通道), 3 个通用定时器(TIM2,TIM3,TIM4), 1 个高级定时器(TIM1), 支持 JTAG/SWD 接口

这款产品属于 STM32 系列中的 F1 系列，是该系列中一款非常流行的中低端型号，广泛应用于各种嵌入式系统和工业控制项目。STM32F103C8T6 核心工作电路如下图 3.6，主要包括，电源电路，晶振电路，复位电路，USB 通信电路，BOOT 电路，电源指示灯电路。

（1）电源电路

具体设计为输入 5V 的直流电压信号经过滤波电容滤波输入到稳压芯片，然后稳压芯片输出稳定的直流+3.3V 电压信号，对 STM32 单片机供电，用稳压芯片的主要目的就是対单片机系统输入稳定的电压让其工作更稳定，效率更高。

（2）晶振电路

主要为 STM32 单片机提供基准时钟信号，确保单片机能够按照预定的时间间隔执行指令和操作，还为内核外设提供时钟信号让其正常运行，例如 Timer，看门狗（WDT），PWM，USB，IIC，UART，ADC 等等。

（3）复位电路

单片机复位会在上电时进行复位，或者看门狗（WDT）计数溢出时复位，还有一种就是通过将 Reset IO 置高低电平进行主动复位，此电路设计就是将 Reset IO 通过按键连接电源地，当按键按下时的时候让系统产生复位，重新开始运行。

（4）BOOT 选择电路

单片机上电在开始的几个时钟周期内会根据不同的 BOOT IO 电平进行配置系统运行模式，例如上电后从 FLASH 区域读取并运行程序，或者从 RAM 区域读取并运行程序。

（5）USB 通信电路

STM32F103C8T6 具有 USB 外设，USB 是差分信号，通过两线压差来表示传输的二进制数据 0 或者 1，所以进行 USB 通信时需要把单片机输出的 TTL 电平转换成 USB 的电平信号，具体的电路设计为 USB 接口连接到 USB 电平转换芯片上，USB 电平转换芯片再与另一个 USB 设备连接，这就构成了 USB 通信网络。

（6）电源指示灯电路

主要目的用来指示单片机系统是否正常供电，有点指示的 LED 灯则亮，没电则不亮。

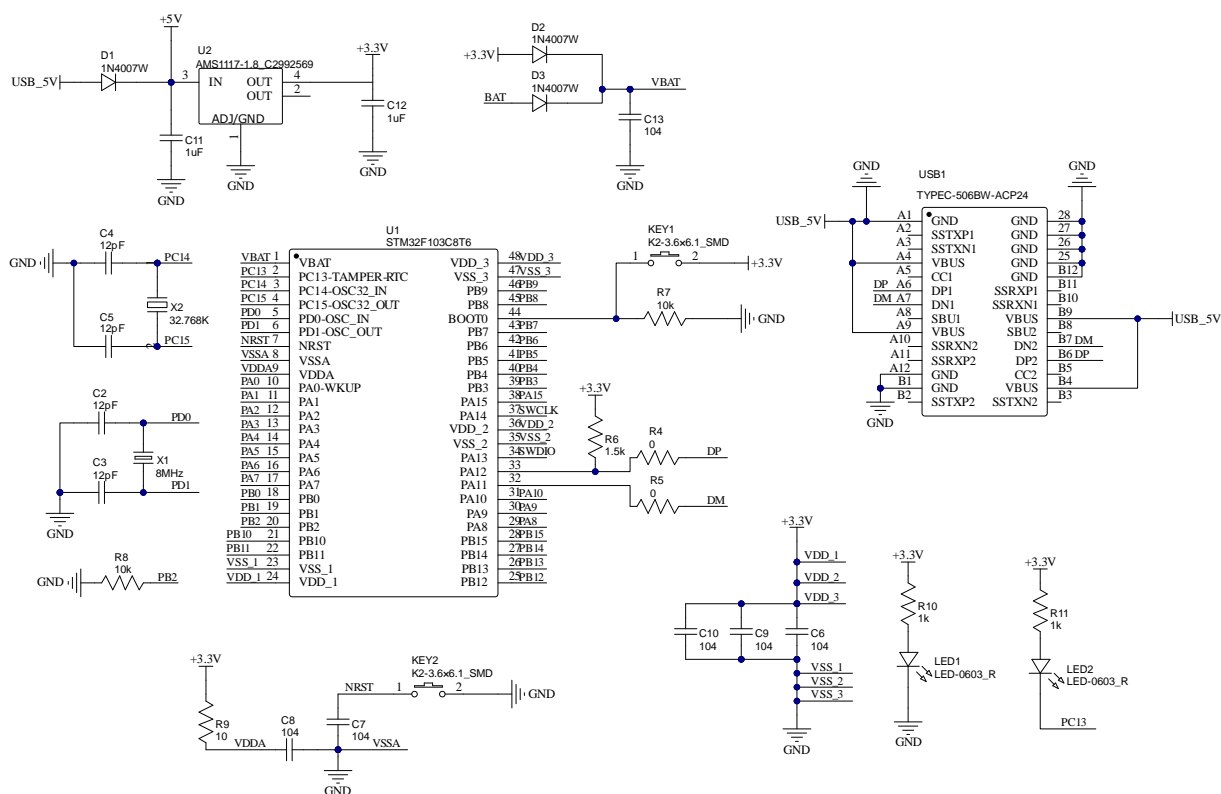


图 3.6 STM32F103C8T6 核心系统电路图

3.3.2 CAN 总线电路

CAN 总线是目前工业制造领域以及汽车电子领域最常用的一种通信协议，CAN 总线能如此受欢迎且应用如此广泛的原因，主要因为其优秀的且独特的特性，CAN 总线属于是单双工，异步的通信协议，且是差模信号传输，通过 CAN_H 与 CAN_L 两线压差来表示二进制数据 0 或者 1，两根线并且是双绞线的形式，这种通信原理使得 CAN 总线具有强抗电磁干扰的能力，且通信速率高达 1Mbps/S，具体的通信过程是单片机的 CAN 接口接到对应的 CAN 电平转换芯片上，单片机向 CAN 外设缓冲区写入数据，然后经过 CAN 电平转换芯片发送到 CAN 总线上，CAN 总线上的设备也是通过 CAN 电平转换芯片将 CAN 电平转换为单片机的 TTL 电平。其 CAN 总线电路设计如下图 3.7，本课题所使用的 CAN 电平转换芯片为 TJA1050，详细参数见下表 3.3 所示，TJA1050 具有高精度且强抗干扰能力，具体电路设计部分有 TJA1050 芯片与滤波抗干扰电路，CAN_H 与 CAN_L 之间串联了个 120R 的电阻主要作用为吸收信号反射及回波，增强抗干扰模式。

表 3.3 TJA1050 芯片参数

参数类型	参数值
最大传输速率	1Mbps
CANH,CANL 输入输出电压	-27V~40V
I/O 电压	支持 3.3V/5.5V
工作电压	4.75V ~ 5.25V
工作温度	-40℃ ~ +150℃

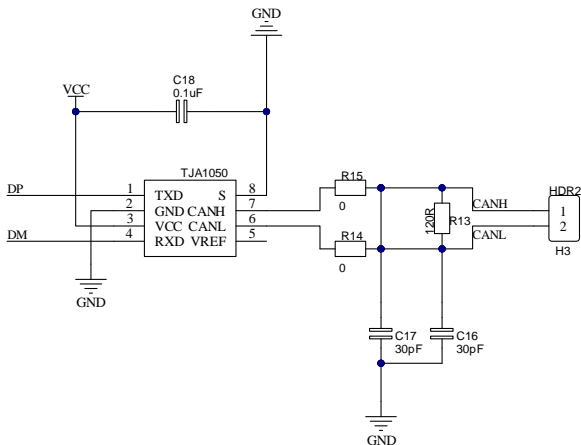


图 3.7 CAN 总线模块电路图

3.3.3 温湿度传感器电路

本设计采用数字式的温湿度传感器芯片来测量环境中的温湿度，温湿度传感器芯片采用的国产奥松电子的 AHT10 型号，此型号是市场主流的温湿度传感器，具有高精度，高灵敏度，以及低功耗，低价格等优势，目前已被广泛应用于消费电子与医疗器械领域，其详细参数见下表 3.4。

表 3.4 AHT10 芯片关键参数

参数类型	参数值
接口类型	IIC
工作电压	1.8V – 3.6 V
湿度精度	典型±2%
湿度分辨率	0.024%
温度精度	典型±0.3℃
温度分辨率	典型 0.01℃
工作温度	-40℃--85℃

下图 3.8 为 AHT10 传感器模块的电路图，主要包括电源电路，和传感器电路。

（1）电源电路

为温湿度传感器芯片 AHT10 供电，将输入的直流电压信号经过滤波电容滤波输入到稳压芯片 HX9193 中，然后稳压芯片输出稳定的直流+3.3V 电压信号，对芯片供电，用稳压芯片的目的其一为了让输入的电压范围更宽，让传感器模块适应性强，其二传感器芯片额定电压是 3.3V，稳定的电压能让传感器工作更稳定，效率更高。

（2）传感器电路

此部分主要包括了滤波抗干扰电路与 IIC 通信电路，SCL 与 SDA 引脚都接了上拉电阻因为温湿度传感器 AHT10 是 IIC 接口的，通过 IIC 接口与单片机进行通信，收发数据，IIC 协议规定的 IIC IO 属于是开漏输出模式，IIC 接口的 GPIO 只有低电平和高阻态两种模式，所以 IIC 接口 IO 没有输出高低电平的能力，当输出高阻态的时候 IIC 总线就输出高电平，代表着二进制数据 1。

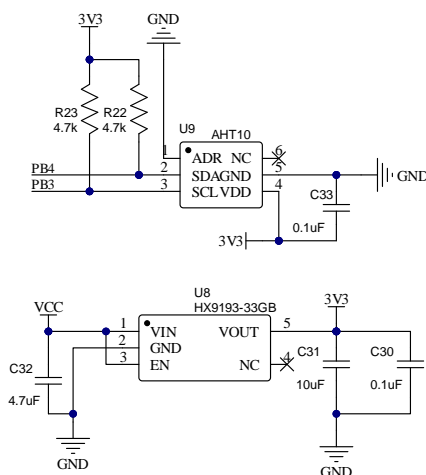


图 3.8 AHT10 温湿度传感器模块电路图

3.3.4 姿态角度传感器电路

姿态角度是对机器人姿态的一个量化，用具体的姿态角度数据来表示机器人的姿态，这在机器人自动控制中非常重要，例如在机器人平衡控制中需要不断的获取机器人的水平倾斜角度，来进行控制机器人直到让机器人达到平衡状态，也就是控制水平倾斜角度为零。

本次设计选用的姿态角传感器为美国 InvenSense 公司的 MPU6050 六轴姿态传感器芯片，MPU6050 是一款集成了三轴陀螺仪和三轴加速度计的微机电惯性测量单元 IMU，IMU 是一种常用的惯性测量装置，能够通过测量物体的加速度和角速度来获取其运动状态。且具有高精度和低功耗的特点，采用了数字输出，可以通过 IIC 接口与单片机进行通信，常用于惯性导航，运动控制，姿态测量等任务，通过其提供的数据，可以实现对物体的姿态，运动状态进行实时监测和控制。广泛用于无人机、机器人、虚拟现实设备、电子游戏手柄等领域，其详细参数见下表 3.5 所示。下图 3.9 为 MPU6050 姿态传感器的工作电路，主要包含电源电路，与 MPU6050 芯片 IIC 通信电路。

表 3.5 MPU6050 关键参数

参数类型	参数值
工作电压	2.375V~3.46V
功耗	500uA~3.9mA（工作电压 3.3V）
工作温度	-40℃~+85℃
通信接口	IIC(Max Speed: 400KHZ)
加速度测量范围	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$ 其中 g 为重力加速度常数, $g=9.8m/s^2$
加速度测量精度	0.1g
加速度最高分辨率	16384 LSB/g
陀螺仪测量范围	$\pm 250^{\circ}/s, \pm 500^{\circ}/s, \pm 1000^{\circ}/s, \pm 2000^{\circ}/s$
陀螺仪最高分辨率	131 LSB/ ($^{\circ}/s$)
陀螺仪测量精度	0.1 $^{\circ}/s$
DMP 姿态解算频率	最高 200HZ

（1）电源电路

电源电路的设计包括稳压部分和滤波抗干扰设计，稳压芯片输入 5V 的直流信号经过滤波抗干扰设计输出+3.3V 的直流信号在经过滤波输入到 MPU6050 的芯片电源引脚为 MPU6050 传感器供电。

（2）IIC 通信电路

MPU6050 是 IIC 接口的，IIC 协议规定 IIC IO 是开漏输出，无高电平输出能力所以 IIC SDA 和 SCL IO 需要接上拉电阻，为 IIC 提供输出高电平能力，因此 MPU6050 的 SDA 引脚与 SCL 引脚均接入了上拉电阻。

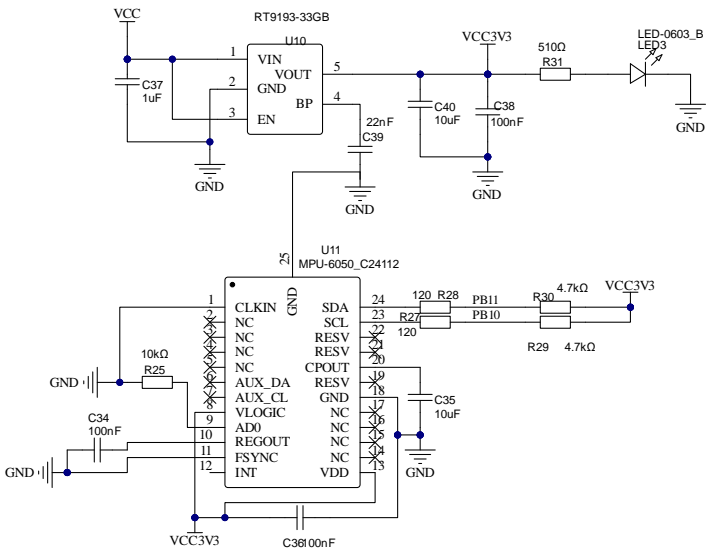


图 3.9 MPU6050 姿态角度传感器模块电路图

3.3.5 永磁有刷直流电机驱动电路

本设计中采用的是永磁有刷直流电动机，简称直流电机，电机驱动芯片采用的是意法半导体（STMicroelectronics）公司的 L298N 驱动芯片，L298N 是一款高电压、大电流双路全桥式电机驱动芯片，不仅可靠性高，而且抗干扰能力还强，其详细参数见下表 3.6 所示。

表 3.6 L298N 芯片参数表	
参数类型	参数值
逻辑电压	5V
驱动电压	5V~35V
逻辑电流	0mA~36mA
驱动电流	2A（MAX 单桥）
储存温度	-20℃ ~ +35℃
额定功率	25W
外围尺寸	43*43*27mm

下图 3.10 为 L298N 的驱动电路图，主要包括了电源电路和电机保护电路。

（1）电源电路

将输出的直流信号经过稳压芯片进行稳压输出+5V 直流信号然后再经过滤波电路进行滤波之后输入到芯片电源端为芯片进行供电。

（2）电机保护电路

电机在启动的瞬间会产生过电压，这个电压会超过电机驱动芯片 L298N 的额定值，如果不进行保护会直接损坏电机驱动芯片，甚至还可能损坏与其连接的单片机系统。

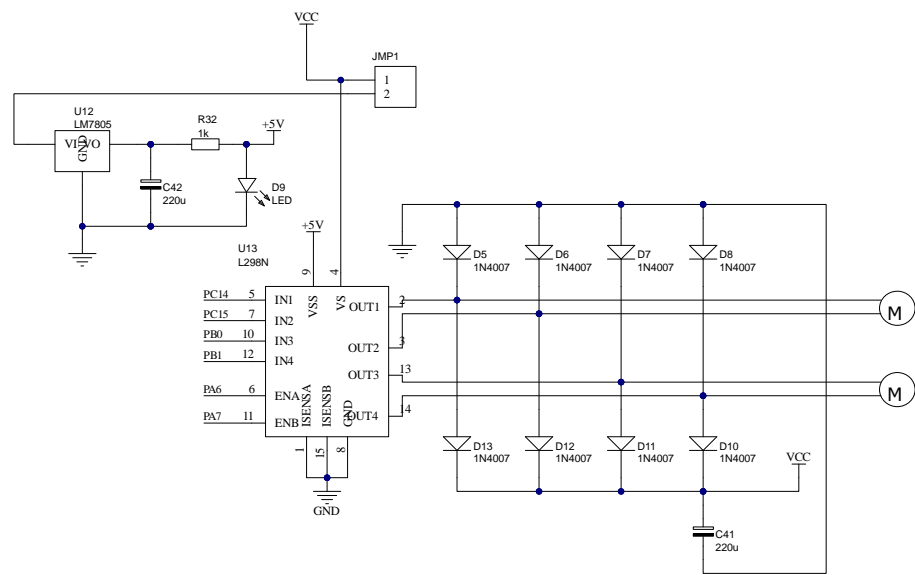


图 3.10 L298N 电机驱动电路图

本文所研究的机器人平台为履带式机器人，履带式机器人采用两个直流电机驱动，分为左电机和右电机，L298N 电机驱动芯片其内部为双 H 桥结构如下图 3.11 所示，使用 L298N 驱动两侧电机只需要给芯片的指定 IO 输入高低电平就可以了，如下表 3.7，3.8 为 L298N 驱动电机的 IO 真值表，ENA 为 PWM 接口，通过控制 PWM 的占空比就可以控制 L298N 芯片 OUT IO 输出的电压大小，从而控制电机转速。

表 3.7 L298N 驱动右侧电机真值表

电机状态	IN1	IN2	ENA (PWM)
正转	1	0	1
反转	0	1	1
停止	0	0	1

表 3.8 L298N 驱动左侧电机真值表

电机状态	IN3	IN4	ENB (PWM)
正转	1	0	1
反转	0	1	1
停止	0	0	1

注：1 表示单片机 IO 输出的高电平，0 表示单片机 IO 输出的低电平

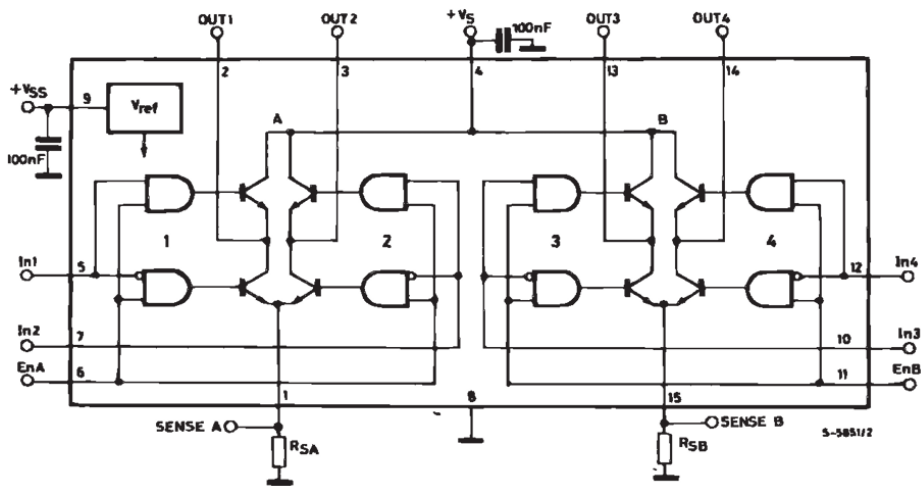


图 3.11 L298N 芯片内部电路原理图

3.3.6 OLED 显示屏电路

传感器数据采集及驱动系统使用了 OLED 显示屏进行辅助传感器数据显示，所采用的显示屏为 IIC 接口的单色 OLED 显示屏，其详细参数如下表 3.9 所示，

表 3.9 OLED 关键参数

参数类型	参数值
分辨率	128*64
功耗	0.08W
工作电压	3.3V~5V (DC)
工作温度	-40℃~+70℃
模块体积	27.6*27*3.7mm
通信协议	SPI 协议
驱动芯片	SSD1306

图 3.12 为 OLED 显示屏电路图，电路设计部分主要包括了，电源电路与 IIC 接口通信电路，和滤波抗干扰设计。

(1) 电源电路

电源电路的设计包括稳压部分和滤波抗干扰设计，稳压芯片输入 5V 的直流信号经过滤波抗干扰设计输出+3.3V 的直流信号在经过滤波输入到显示屏的芯片电源引脚为显示屏供电。

(2) IIC 通信电路

OLED 显示屏是通过 IIC 接口与单片机传输数据的，而 IIC 协议规定 IIC IO 是开漏输出，无高电平输出能力所以 IIC SDA 和 SCL IO 需要接上拉电阻，为 IIC 提供输出高电平能力，因此 OLED 的 SDA 引脚与 SCL 引脚均接入了上拉电阻。

(3) 滤波和抗干扰设计

在电路系统中，信号传输的过程中或者上电信号输入时会受到电磁干扰或者本身的信号质量问题，所以需要进行滤波抗干扰设计将干扰噪声信号滤除。

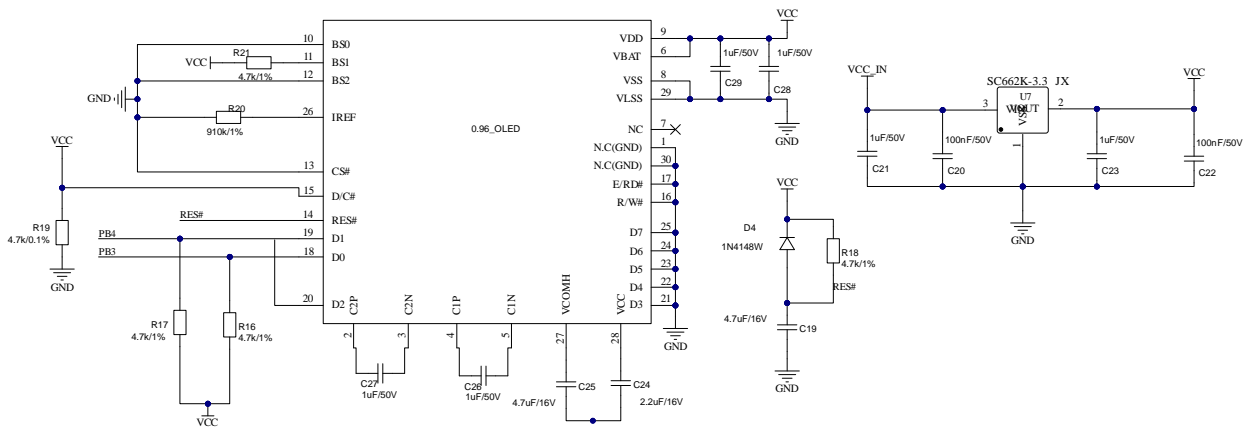


图 3.12 OLED 显示屏电路图

3.4 机器人无线控制系统硬件电路

3.4.1 STM32F103C8T6 核心工作电路

机器人传感器数据采集及驱动系统所采用的核心处理器为 STM32F103C8T6 单片机，STM32F103C8T6 是意法半导体（STMicroelectronics）公司基于 ARM Cortex-M3 内核设计的高性能，低功耗且具有强抗干扰性的 32 位单片机，详细参数见下表 3.10 所示。

表 3.10 STM32F103C8T6 单片机资源表

参数类型	参数值
内核	ARM Cortex-M3 32bit RISC 指令集
主频	72MHZ
SRAM	20KByte
Flash	64KByte
I/O 数量	48
工作电压	2.0~3.6V
外设资源	2*SPI, 3*USART, 2*I2C, 1*CAN, 37*OV 口, 2*ADC(12 位/16 通道), 3 个通用定时器(TIM2,TIM3,TIM4), 1 个高级定时器(TIM1), 支持 JTAG/SWD 接口

这款产品属于 STM32 系列中的 F1 系列，是该系列中一款非常流行的中低端型号，广泛应用于各种嵌入式系统和工业控制项目。STM32F103C8T6 核心工作电路如下图 3.13 主要包括，电源电路，晶振电路，复位电路，USB 通信电路，BOOT 电路，电源指示灯电路。

（1）电源电路

具体设计为输入 5V 的直流电压信号经过滤波电容滤波输入到稳压芯片，然后稳压芯片输出稳定的直流+3.3V 电压信号，对 STM32 单片机供电，用稳压芯片的主要目的就是対单片机系统输入稳定的电压让其工作更稳定，效率更高。

（2）晶振电路

主要为 STM32 单片机系统提供基准时钟信号，确保单片机能够按照预定的时间间隔执行指令和操作，还为内核外设提供时钟信号让其正常运行，例如 Timer，看门狗（WDT），PWM，USB，IIC，UART，ADC 等等。

（3）复位电路

单片机复位会在上电时进行复位，或者看门狗（WDT）计数溢出时复位，还有一种就是通过将 Reset IO 置高低电平进行主动复位，此电路设计就是将 Reset IO 通过按键连接电源地，当按键按下时的时候让系统产生复位，重新开始运行。

（4）BOOT 选择电路

单片机上电在开始的几个时钟周期内会根据不同的 BOOT IO 电平进行配置系统运行模式，例如上电后从 FLASH 区域读取并运行程序，或者从 RAM 区域读取并运行程序。

（5）USB 通信电路

STM32F103C8T6 具有 USB 外设，USB 是差分信号，通过两线压差来表示传输的二进

制数据 0 或者 1，所以进行 USB 通信时需要把单片机输出的 TTL 电平转换成 USB 的压差信号，具体的电路设计为 USB 接口连接到 USB 电平转换芯片上，USB 电平转换芯片再与另一个 USB 设备连接，这就构成了 USB 通信网络。

(6) 电源指示灯电路

主要目的用来指示单片机系统是否正常供电，有点指示的 LED 灯则亮，没电则不亮。

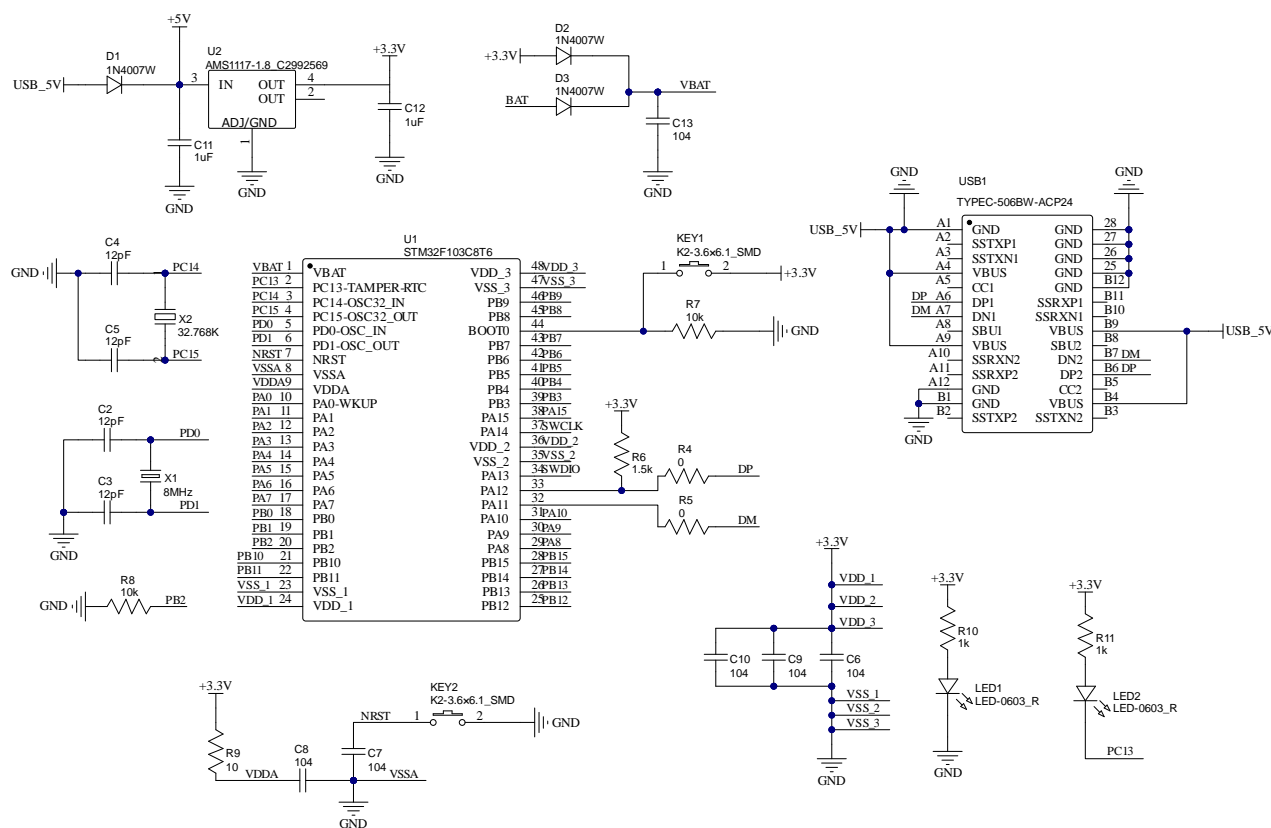


图 3.13 STM32F103C8T6 核心系统电路图

3.4.2 TFT-LCD 显示屏电路

为获知传感器数据信息，无线控制系统使用了显示屏作为数据显示的平台，本设计采用了 TFT-LCD 显示屏，它能够显示彩色图像，并支持 RGB（红、绿、蓝）色彩模式且有非常优秀的低功耗能力，其详细参数和接口参数见下表 3.11 与 3.12 所示。

表 3.11 TFT-LCD 显示屏参数

参数	参数值
分辨率	128*128 (RGB)
驱动芯片	ST7735

续表 3.11 TFT-LCD 显示屏参数

参数	参数值
接口类型	SPI
背光类型	LED*1
工作电压	2.8V~3.3V
工作电流	15mA
功耗	0.06W
工作温度	-20℃ ~ +70℃

表 3.12 TFT-LCD 显示屏接口参数

序号	符号	说明
1	GND	电源地
2	VCC	电源正,3.3V
3	SCL	SPI 时钟线
4	SDA	SPI 数据线
5	RES	LCD 复位
6	DC	LCD 数据/命令选择脚
7	CS	LCD 显示屏片选信号, 低电平有效
8	BLK	背光控制开关, 默认打开背光, 低电平关闭

图 3.14 为 TFT-LCD 显示屏电路原理图，此电路设计要包含了电源电路，和滤波抗干扰电路，电源电路为 LCD 显示屏进行供电保证 LCD 显示屏能正常工作，滤波电路抗干扰电路主要是除去噪声信号，保证 LCD 显示屏能稳定工作。

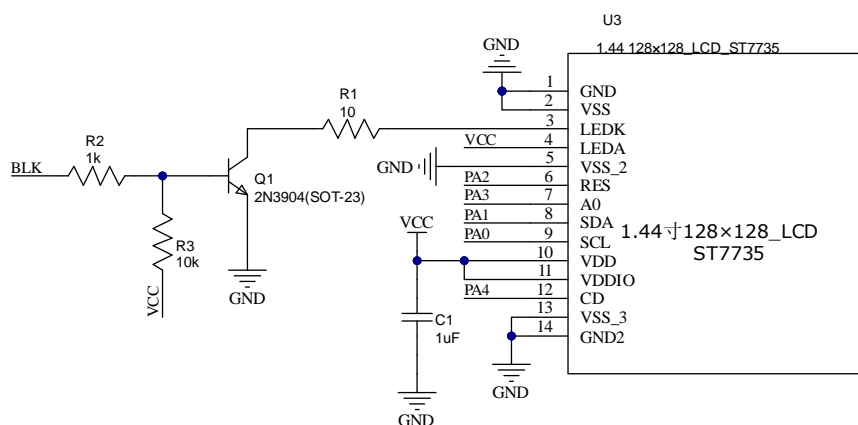


图 3.14 TFT-LCD 工作电路图

3.4.3 摇杆电路

本系统设计中摇杆主要作用就是为采集机器人运动控制的命令信息，摇杆类似于一个传感器，通过上下左右拨动摇杆其 X 轴电压和 Y 轴电压值会有不同，然后通过单片机 ADC 外设就可以采集摇杆的 X 轴 Y 轴电压值了，经过处理算法的处理就得到了摇杆的推动信息。其电路设计如下图 3.15，主要包含了电源电路和滤波设计，电源电路为摇杆进行供电，滤波电路抗干扰电路主要是除去噪声信号，保证输入到摇杆的电压信号是正确的。

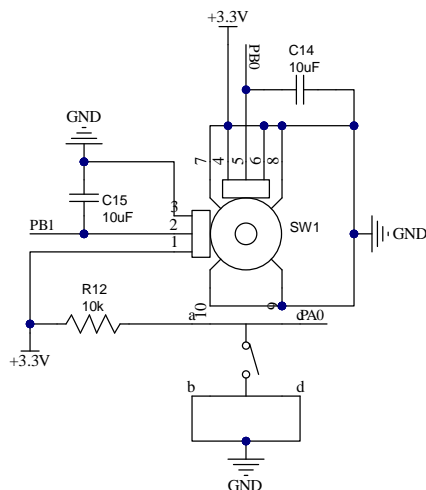


图 3.15 摇杆工作电路图

3.5 本章小结

本章主要介绍了机器人硬件电路系统设计的各个部分，主要包括机器人主控制系统硬件电路、机器人传感器数据采集及驱动系统硬件电路、以及机器人无线控制系统硬件电路。在每个部分中，详细讨论了单片机核心工作电路、各种传感器电路以及显示屏，摇杆，蓝牙等模块的电路设计。

第4章 机器人软件系统设计与测试

4.1 引言

机器人软件系统是机器人智能化的前提, 软件系统并不是单独而存在的, 而是与感知系统, 控制系统交叉融合共存, 并且软件系统也作为一个平台为感知系统, 运动控制系统提供发展的基石。本章主要介绍机器人主控在嵌入式 Linux 系统平台上的任务开发, 与单片机 (MCU) 在嵌入式实时操作系统 (RTOS) 平台上的任务开发。

4.2 机器人主控嵌入式 linux 系统

4.2.1 Linux CAN 驱动

Linux CAN 驱动是用于 Linux 操作系统中控制器局域网络 (CAN) 通信的驱动程序。CAN 总线是一种在汽车和工业自动化领域广泛应用的串行通信协议^[26], 它允许多个设备在不需主机控制的情况下进行通信。Linux CAN 驱动的主要作用是提供对 CAN 总线设备的访问和控制, 使得应用层程序能够与 CAN 总线上的其他设备进行数据交换。具体的 Linux CAN 工作流程如下图 4.1, 首先进行 CAN 设备创建, 其次就是创建 CAN Socket, 之后就是绑定 Socket 与 CAN 设备, 配置 CAN 数据结构体, 所有的配置工作都完成之后就要进行 CAN 收发工作了。

(1) 创建 CAN 设备

Linux 系统中所有的设备都是以文件的形式存在, 所以使用设备前需要先创建设备, 具体步骤为使用 System(“ip link set can0 up type can bitrate 100000 triple-sampling on”)API 来创建 CAN 设备和设置相应的波特率。

(2) 创建 CAN Socket

使用 socket 函数创建一个 CAN 套接字, 并指定通信域为 PF_CAN, 套接字类型为 SOCK_RAW, 以及 CAN_RAW 协议。

(3) 将 CAN 与 Socket 绑定

Socket 与 CAN 设备进行绑定, 以便通过套接字进行 CAN 数据的发送和接收。

(4) 配置 CAN 设备参数

Linux 系统对 CAN 寄存器进行了高度的封装, 不必再像单片机寄存器编程那种繁杂的方式去编程了, 而是直接配置 Linux 系统的 CAN 结构体参数就可以配置其 UART 的相应寄存器的数值了。例如: 通过 CAN_TxFrame.can_dlc 结构体参数配置 CAN 的数据长度 (DLC), 通过 CAN_TxFrame.can_id 结构体参数 CAN 设备 ID 以便于与其他 CAN 设备通信。

(5) 发送 CAN 数据

将 UART 接收数据缓冲区的数据复制到 CAN 发送数据缓冲区, 实现了将无线控制系

统发来的机器人运动控制命令数据通过蓝牙串口模块接收到 UART 接收数据缓冲区中, 然后再把此数据通过 CAN 发送到机器人传感及驱动系统中, 传感及驱动系统控制电机驱动。

(6) 接收 CAN 总线数据

将 CAN 接收数据缓冲区中的数据读取出来, 复制到 UART 发送数据缓冲区中, 实现将传感及驱动系统发来数据经过机器人主控发送到无线控制系统中。

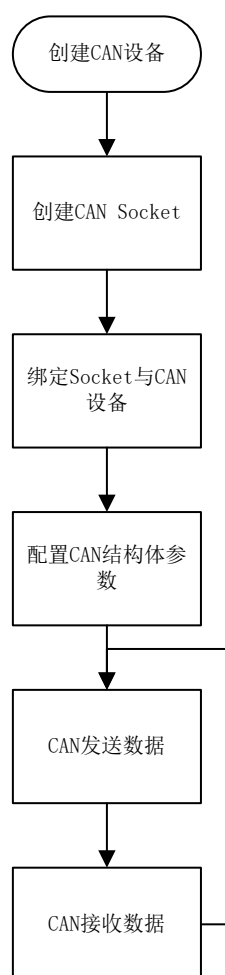


图 4.1 Linux CAN 工作流程图

4.2.2 Linux Uart 驱动

Linux UART 驱动是用于 Linux 操作系统中串行通信接口 (UART) 的驱动程序。UART 是一种通用异步收发传输器, 常用于微控制器和其他设备之间的通信。Linux UART 驱动负责在操作系统与串行设备之间建立通信, 并管理数据的发送和接收。其大致工作流程如下图 4.2, 主要为打开 UART 设备, 配置 UART 参数, 与配置 Linux UART 中断回调函数, 最后就是收发串口数据。

(1) 打开 UART 设备

在 Linux 系统中, 所有的设备均以文件的形式存在, 所以使用设备前需要先打开相应

的设备文件。

(2) 配置 UART 参数

Linux 系统对 UART 寄存器进行了高度的封装, 不必再像单片机寄存器编程那种繁杂的方式去编程了, 而是直接配置 Linux 系统的 UART 结构体参数就可以配置其 UART 的相应寄存器的数值了。例如: 通过 `cfg.baudrate` 结构体参数配置 UART 的通信速率, 通过通过 `cfg.parity` 结构体参数配置 UART 的奇偶校验模式。

(3) 配置 Linux UART 中断回调函数

在 Linux 系统中, 当又 UART 接收数据时, 并不会直接进中断服务函数 (ISR) 进行数据处理, 因为这样做会打断操作系统正在运行的任务, 导致效率低下, Linux 操作系统则是在 UART 中断服务函数中设置了个回调函数, 当产生 UART 接收中断时候直接调用回调函数, 用户在回调函数中读取 UART 接收的数据。

(4) 读取和发送 UART 数据

在 Linux 系统中 UART 的数据是存储在 UART 文件中, 要发送数据的时候将 UART 发送缓冲区的数据, 使用 `write()` 文件操作函数写入数据到文件中, 读取数据的时候, 在 Linux UART 中断回调函数中使用 `Read()` 文件操作函数将接收到的数据从文件中读取到 UART 接收数据缓冲区中。

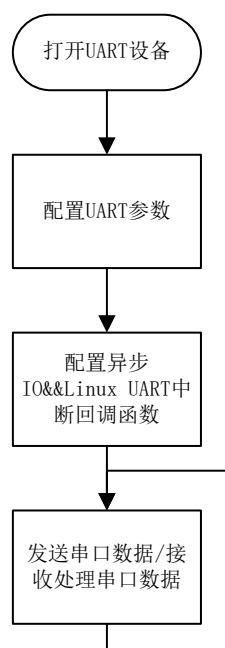


图 4.2 Linux UART 工作流程图

4.2.3 Linux V4L2 for Camera 驱动

Linux V4L2 (Video for Linux 2) 驱动框架是 Linux 内核中关于视频设备驱动而开发的一个子系统, 它为 Linux 下的视频驱动提供了统一的接口^[27], 使得 Video 应用程序可以使用统一的 API 函数操作不同的视频设备, 极大地简化了 Video 设备的开发流程。V4L2 特

别适用于图像传感器的驱动,通过它,可以高效地控制图像传感器的工作,实现图像的采集和处理。Linux V4L2 驱动图像传感器的具体工作流程如下图 4.3 所示。

(1) 打开 Camera 设备

首先使用 `open()` 函数打开相应的 Video 设备文件,获取 video 设备文件的描述符。

(2) 查询设备属性

使用 `ioctl()` 函数结合 V4L2 提供的控制命令(如 `VIDIOC_QUERYCAP`)来查询设备的属性和能力。

(3) 设置图像格式和参数

根据应用需求,设置图像的格式(如分辨率、颜色空间等),使用 `ioctl()` 函数结合相关命令(如 `VIDIOC_S_FMT`)来设置采集格式和参数。

(4) 申请和映射帧缓冲区

申请视频采集所需的帧缓冲区,使用 `mmap()` 函数将帧缓冲区从内核空间映射到用户空间,便于应用程序读取和处理数据。

(5) 开始视频采集

将申请到的帧缓冲区排队到视频采集的输入队列中,调用 `ioctl()` 函数结合适当的命令(如 `VIDIOC_STREAMON`)来启动视频采集。

(6) 读取和处理数据

从视频采集的输出队列中取出送到帧缓冲区,读取其中的图像数据,对图像数据进行处理,如显示图像数据,处理后的帧缓冲区重新放入输入队列,循环往复以持续采集视频数据。

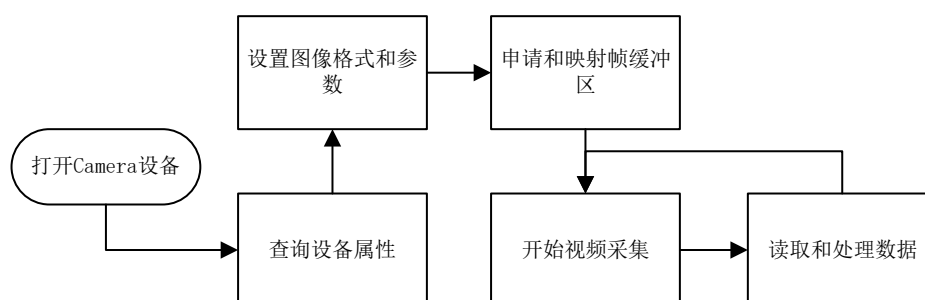


图 4.3 Linux V4L2 For Camera 工作流程图

4.2.4 基础性功能测试

(1) Linux CAN 基础通信功能测试

本测试是将主控与传感器及驱动系统连接在了一起,进行 CAN 通信测试,主控向传感器及驱动系统发送了 8 个字节的数据,传感及驱动系统向主控也发送了 8 个字节数据,实测如下图 4.4。



图 4.4 Linux CAN 基础通信功能测试图

(2) Linux UART 基础通信功能测试

UART 测试主要测试 UART 是否能正常收发数据，本测试将主控 UART 连接到蓝牙串口模块上，另一端的蓝牙串口模块与 USB 转 TTL 模块连接，这两个组成了主机和从机，Linux 端向串口发送数据，如果 UART 功能正常的话 PC 端的串口助手上就可以看到发来的数据，实测如下图 4.5。

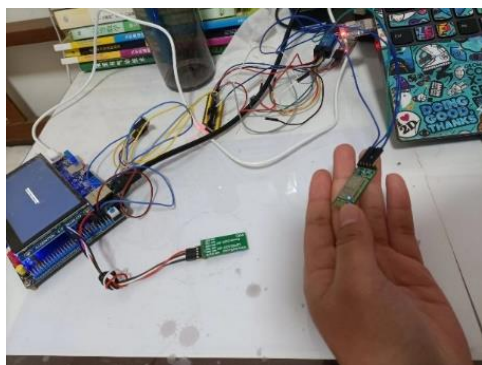


图 4.5 Linux UART 基础通信功能测试图

(3) Linux V4L2 for Camera 基础功能测试

V4L2 驱动图像传感器通过读取缓冲区中的图像数据并显示在 LCD 显示屏上，实测如下图 4.6。



图 4.6 Linux V4L2 基础功能测试图

4.2.5 系统性测试

主控系统测试属于是综合性测试,将图像传感器驱动与 CAN 通信和 UART 通信三个功能结合到一起,具体实测如下图 4.7,实现了主控端的环境视觉感知能力与数据处理中枢的任务。



图 4.7 机器人主控系统测试图

4.3 机器人传感器数据采集及驱动系统

4.3.1 STM32F103 移植 FreeRTOS

FreeRTOS 是一个实时操作系统的内核,具有高度可移植性,可裁剪性,可以裁剪移植到不同的处理器架构平台上运行,并且 FreeRTOS 还有具有免费、开源、小巧等关键特性。

FreeRTOS 在多任务管理方面可以创建和管理多个任务,并使用优先级和时间片轮转等调度算法进行任务调度。每个任务都有自己的线程栈空间,可以独立运行。在实时性方面,FreeRTOS 提供了实时任务调度和中断处理机制,能够满足高实时应用程序的需求,保证关键任务能及时执行。此外,它还提供了多种通信和同步机制,如信号量、消息队列、事件标志组等,用于任务之间的通信和同步,确保任务之间的数据共享的安全性。

下边介绍在 STM32 单片机上移植 FreeRTOS 具体流程。

(1) 获取 FreeRTOS 源码

从 FreeRTOS 官网下载最新版本的 FreeRTOS 源码。

(2) 创建工程

使用 Keil MDK-ARM 集成开发环境创建一个 STM32F103C8T6 新的工程。

(3) 配置工程

根据 STM32 的标准外设库初始化时钟,配置好 GPIO, UART, 等基础外设功能。

(4) 加入 FreeRTOS 源码

将 FreeRTOS 源码加入到工程目录中。

(5) 修改 FreeRTOS 配置

在 FreeRTOS 源码中找到配置文件 FreeRTOSConfig.h, 修改配置选项。例如: 定义任务数量, 堆栈大小, 调度器等等。

(6) 实现 FreeRTOS 钩子函数

实现 FreeRTOS 提供的钩子函数(hook functions), 例如任务创建、删除、切换时的钩子函数。

(7) 实现硬件定时器

FreeRTOS 需要一个硬件定时器来生成滴答中断, 以支持时间管理和任务调度。需要根据配置一个 STM32F103C8T6 定时器作为 FreeRTOS 的滴答定时器, 并实现相应的中断服务程序。

(8) 初始化 FreeRTOS

在 main 函数中调用 API 接口 vTaskStartScheduler();来启动 FreeRTOS 调度器。

(9) 创建任务

使用 FreeRTOS 提供的 API 接口 xTaskCreate()创建任务, 并自定义任务函数。

(10) 编译和调试

编译工程并下载到 STM32F103C8T6 中, 使用 UART 进行打印反馈调试, 确保 FreeRTOS 能够正常启动, 任务能够按照预期运行。

4.3.2 STM32F103 UART 驱动

UART 全称通用异步收发传输器(Universal Asynchronous Receiver/Transmitter), 是一种异步收发传输器, 也是设备间进行异步通信的关键模块, 是极其常用的内核通信外设。它负责处理单片机与单片机之间的串行数据收发, 并规定了帧格式。只要通信双方采用相同的帧格式和波特率, 就可以在无时钟信号的情况下, 仅用两根信号线(Rx 和 Tx)完成数据通信, 因此也被称为异步串行通信。本课题使用 STM32 标准库开发应用 UART 其步骤通常包括以下几个主要环节。

(1) 初始化 GPIO

UART 通信涉及 TX(发送)和 RX(接收)两个引脚。首先配置 STM32 的 GPIO(通用输入输出)引脚以支持 UART 通信, 并设置成推挽输出。

(2) 初始化 UART 外设

配置 UART 参数, 波特率(数据传输速率)、数据位、停止位和校验位。这些参数必须与目标设备相匹配, 以确保与其他设备正确通信。配置好之后使用 STM32 标准库提供的函数来初始化 UART, 设置其工作模式(如异步模式)和相关的参数。

(3) 发送 UART 数据

将待发送的数据写入 UART 的发送缓冲区, 在 STM32 中使用标准库的开发方式, 并不需要像传统的编程方式一样直接向 UART 寄存器写入数据, 而是使用 STM32 官方提供

的标准库函数,通过向 UART 发送数据函数传参数来发送 UART 数据。

(4) 接收 UART 数据

STM32 中的 UART 数据的接收,使用中断处理的方式来接收,在标准库提供的中断服务函数里边读取 UART 接收函数的数据,放入 UART 接收数据缓冲区中实现 UART 数据接收。

4.3.3 STM32F103 CAN 通信驱动

STM32 中的 CAN 驱动步骤主要包括以下几个环节:

(1) 初始化 CAN 外设

初始化 STM32 CAN 相关的 GPIO,设置 CAN 控制器的时钟,以及配置 CAN 工作模式等等。

(2) CAN 参数配置

配置 CAN 外设的波特率为 500Kbps/S,工作模式为正常模式,过滤器等等。

(3) 收发 CAN 数据

配置好 CAN 数据结构体参数后,使用标准库的 CAN 收发函数进行接收发送 CAN 总线的数据。

4.3.4 STM32F103 IIC 驱动传感器

IIC (Inter-Integrated Circuit) 是 IIC Bus 的简称,中文名为集成电路总线。它是一种串行同步半双工的通信协议,在国内也被称为 I2C (Inter-IC) 总线。IIC 总线是飞利浦公司在 1980 年代为解决 MCU/SOC 与其他外围设备通信问题,而提出的一种简单的通信协议总线。

IIC 总线总共有两根线,一根串行数据线(SDA)和一根串行时钟线(SCL)。SDA 线用于传输数据,而 SCL 线用于提供时钟信号。每个设备在 IIC 总线上都有一个唯一的地址,用于在总线上识别设备。

本文研究中使用 STM32 的标准库开发 IIC 外设,其步骤主要包括以下几个环节:

(1) 初始化 IIC 外设

初始化 STM32 IIC 模块的时钟,GPIO 配置等。

(2) IIC 参数配置

配置 IIC 通信的通信速率(波特率),工作模式(主模式),IIC 地址。

(3) 读写控制 IIC 设备

在此系统中,IIC 主要驱动了温湿度传感器,姿态角度传感器与 OLED,进行读取温湿度传感器数据,姿态角度传感器数据和数据显示,通过 IIC 接口向传感器发送读写命令,进行读写温湿度传感器数据和再 OLED 上显示数据。

4.3.5 STM32F103 PWM 驱动直流电机

本课题机器人驱动所采用的为永磁直流有刷电动机, 永磁直流有刷电动机的驱动原理是靠控制输入电压值大小进行控制其转速的, 下图 4.8 为直流电动机工作原理模型图, 电刷通电连接换向器, 电枢转子通电产生励磁磁场, 励磁磁场与永磁磁场相互作用, 电枢转子就进行转动了, 但是单片机通用 IO 口输出的是数字信号二进制数据 0 或 1, 其本质为 TTL 电平 3.3V/0V, 3.3V 表示数字信号二进制数据 1, 0V 表示数字信号二进制数据 0。这样的电平只能控制电机的转动与停止, 不能控制电机的转速高低, 例如: 电机加速减速, 因此可以使用 PWM 技术来进行直流电机转速控制。

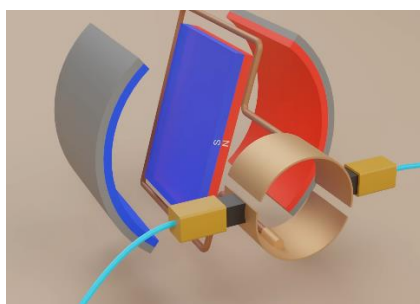


图 4.8 永磁直流有刷电机工作原理模型图

PWM 是脉宽调制 (Pulse Width Modulation) 的缩写, 是一种模拟的控制方式如下图 4.9。是通过对一系列脉冲的宽度进行调制, 来获得等效作用的数字信号。

PWM 信号有两个关键参数: 占空比和频率。占空比是指在一个脉冲周期内, 高电平相对于整个周期 T 所占的比例。频率则是指 PWM 信号完整周期的重复次数, 通常以赫兹 (Hz) 为单位。通过调整占空比, 可以控制输出信号的平均幅值。例如, 如果 PWM 信号的占空比为 50%, 即高电平时间等于总周期的一半, 那么输出信号的平均电压就为 $1/2$ 的 TTL 高电平。

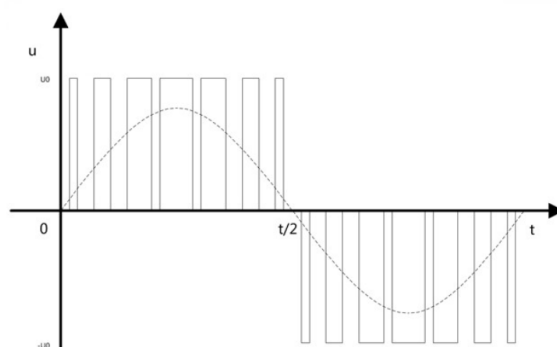


图 4.9 PWM 等效信号原理图

在 STM32 单片机中 PWM 功能是通过 Timer 驱动的, Timer 配置成向上计数模式后根据比较器的设置, Timer 计数值低于比较值时 IO 输出高电平, 高于比较器值时输出高电平, 然后下一个 Timer 周期再以此往复, 这个输出 IO 高低电平的波形过程就是 PWM (脉宽调

制)原理如下图 4.10。

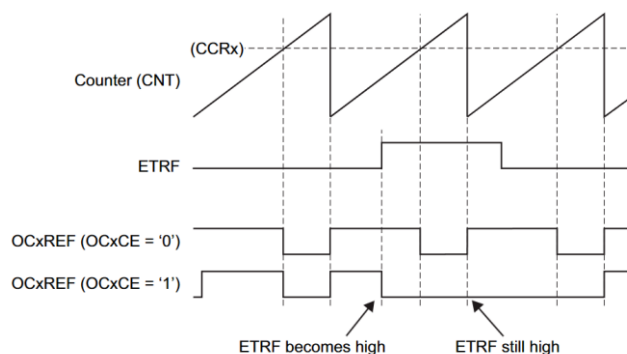


图 4.10 STM32 PWM 实现原理图

此外 STM32 单片机提供了多个 PWM 通道, 每个通道都可以配置为不同的输出引脚, 并具有灵活的配置选项, 如配置频率、占空比、极性等。

STM32 使用标准库开发 PWM 的步骤主要包括以下几个关键步骤:

(1) 配置时钟

配置 AHB、APB 等时钟分频系数, 配置 STM32 Timer 时钟初始化。

(2) 初始化 PWM 外设

配置 PWM 的基本参数, 如预分频值、自动重载值等。启用定时器的 PWM 输出模式, 并设置选择相应的 PWM 通道。

(3) 配置 PWM 输出

设置 PWM 输出的比较匹配寄存器值, 这决定了 PWM 的占空比, 配置 PWM 模式为中心对齐模式, 输出极性为高电平有效。

(4) 使能 PWM 输出

启动 PWM, 开始生成 PWM 波形, 之后根据需要, 动态调整 PWM 的占空比或频率。

4.3.6 基础性功能测试

(1) STM32 FreeRTOS 基础功能测试

在 STM32F103C8T6 上移植好 FreeRTOS 之后创建第一个线程 task1, 线程内运行打印 task1 字符的任务, 如下图 4.11 通过串口助手观察到成功打印出 task1 字符说明, FreeRTOS 移植成功了。



图 4.11 FreeRTOS 运行测试图

(2) STM32 UART 基础通信功能测试

本测试是 STM32 通过串口模块直接与 PC 端相连接进行串口收发测试, 实测如下图 4.12 所示。

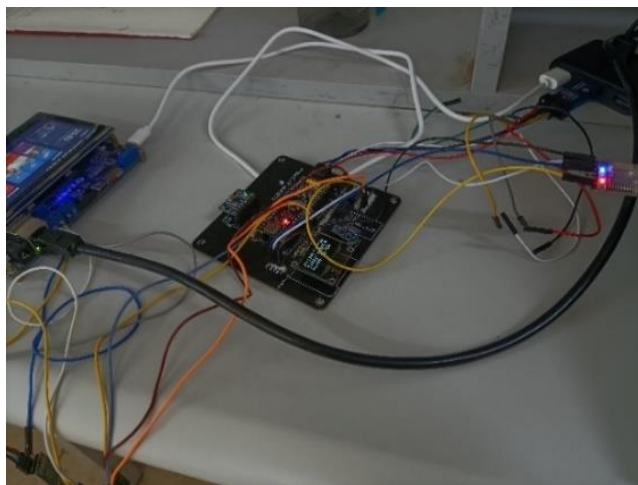


图 4.12 STM32 UART 测试图

(3) STM32 CAN 基础通信功能测试

STM32 的 CAN 通信测试是与机器人主控端连接进行测试的, 实测如下图 4.13 所示与主控双方各发送 8 个字节的数据。

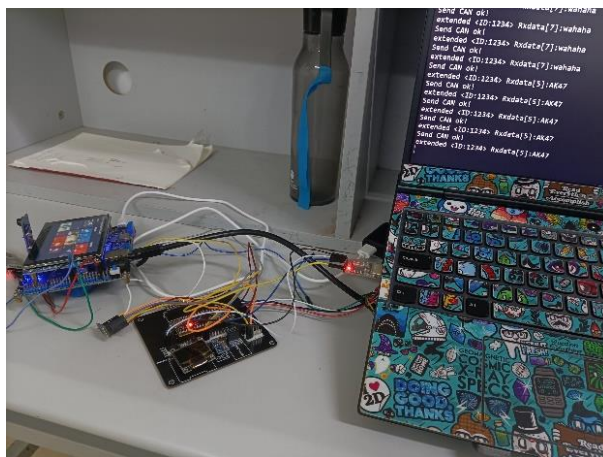


图 4.13 STM32 CAN 测试图

(4) STM32 PWM 电机驱动测试

PWM 驱动电机通过像 L298N 模块输入 PWM，间接驱动电机，实测如下图 4.14。

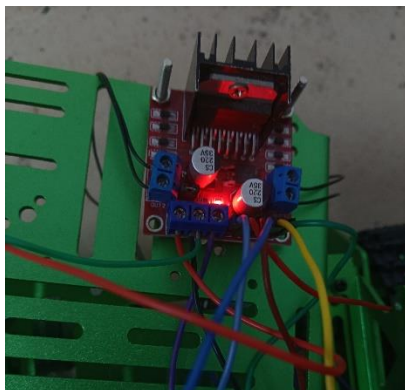


图 4.14 STM32 PWM 驱动电机测试图

(5) STM32 IIC 驱动 OLED 测试

下图 4.15 为 OLED 显示测试图，将读取到的温湿度数据通过 OLED 显示出来。

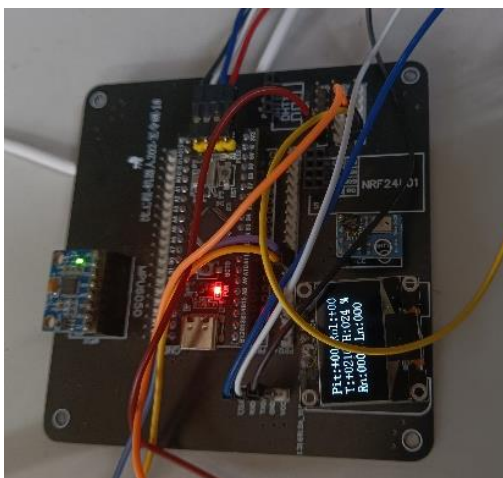


图 4.15 STM32 OLED 测试

4.3.7 系统性测试

如下图 4.16, 将此系统中的多个功能结合到一起, 共同运行, 进行测试, 实现了传感机及驱动系统端的机器人环境感任务, 与机器人驱动, 和检测机器人姿态角度数据的任务。

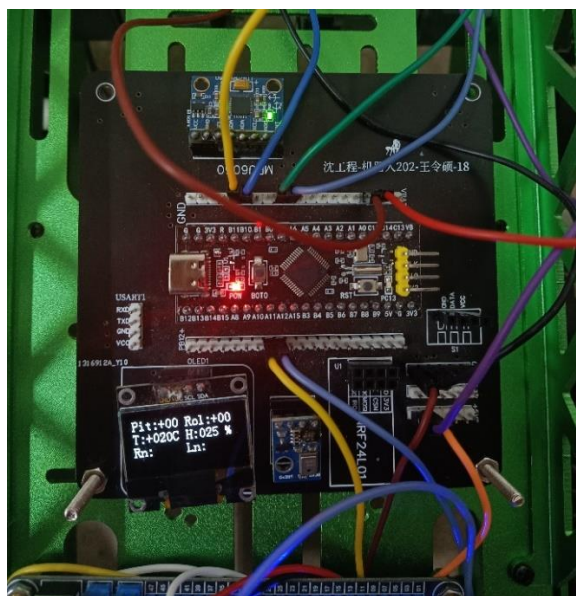


图 4.16 机器人传感及驱动系统测试图

4.4 机器人无线控制系统

4.4.1 STM32F103 UART 驱动

UART 全称通用异步收发传输器 (Universal Asynchronous Receiver/Transmitter)，是一种异步收发传输器，也是设备间进行异步通信的关键模块。它负责处理数据总线和串行口之间的串/并、并/串转换，并规定了帧格式。只要通信双方采用相同的帧格式和波特率，就可以在未共享时钟信号的情况下，仅用两根信号线 (Rx 和 Tx) 完成通信过程，因此也被称为异步串行通信。本课题使用 STM32 标准库开发应用 UART 其步骤通常包括以下几个主要环节。

(1) 初始化 GPIO

UART 通信涉及 TX (发送) 和 RX (接收) 两个引脚。首先配置 STM32 的 GPIO (通用输入输出) 引脚以支持 UART 通信，并设置成推挽输出。

(2) 初始化 UART

配置 UART 参数，波特率 (数据传输速率)、数据位、停止位和校验位。这些参数必须与目标设备相匹配，以确保与其他设备正确通信。配置好之后使用 STM32 标准库提供的函数来初始化 UART，设置其工作模式 (如异步模式) 和相关的参数。

(3) 发送 UART 数据

将待发送的数据写入 UART 的发送缓冲区, 在 STM32 中使用标准库的开发方式, 并不需要像传统的编程方式一样直接向 UART 寄存器写入数据, 而是使用 STM32 官方提供的标准库函数, 通过向 UART 发送数据函数传参数来发送 UART 数据。

(4) 接收 UART 数据

STM32 中的 UART 数据的接收, 使用中断处理的方式来接收, 在标准库提供的中断服务函数里边读取 UART 接收函数的数据, 放入 UART 接收数据缓冲区中实现 UART 数据接收。

4.4.2 STM32F103 ADC 驱动摇杆

ADC 是模拟数字转换器 (Analog-to-digital converter)。它能将输入的模拟信号转换为数字信号, 方便单片机进行数据处理, 因为单片机只能处理数字信号 0/1, 转换过程如下图 4.17, STM32 的 ADC 是一种 12 位逐次逼近型的模拟数字转换器, 具有多达 18 个通道, 可以测量 16 个外部和 2 个内部信号源。其工作原理大致分为采样、量化、编码和输出四个步骤。STM32 的 ADC 具有多种工作模式, 如多通道扫描单转换模式、单通道连续转换模式以及注入转换模式等, 这些模式可以根据不同的应用需求进行灵活配置。

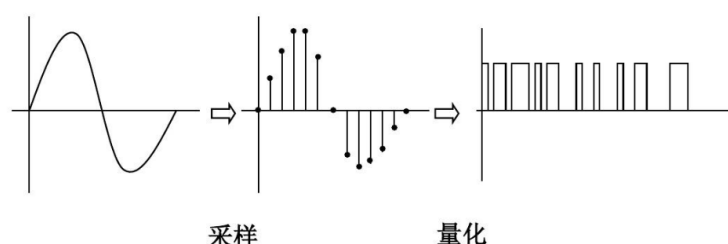


图 4.17 模拟信号转数字信号过程图

本课题研究使用 STM32 的标准库开发 ADC (模数转换器) 驱动摇杆的具体步骤包括以下几个环节:

(1) 配置 GPIO 和时钟

配置与 ADC 相关的 GPIO 引脚为模拟输入模式, 并开启相应的时钟。确保 ADC 能够正确读取摇杆的模拟信号。

(2) 配置 ADC

使用 STM32 的标准库配置 ADC 的参数, 例如: 分辨率、采样时间, ADC 工作模式为连续转换模式, 输入通道等。

(3) ADC 通道数据处理

STM32 的使用了两个 ADC 通道分别连接着摇杆的 X 轴信号 IO 和 Y 轴信号 IO, 当拨动摇杆在 X 轴方向运动或者 Y 轴方向运动时 X 轴的 ADC 通道或者 Y 轴的 ADC 通道就可以检测到波动的电压数据, 由于电磁干扰的问题, 单片机检测到的原始 ADC 数据非常的不稳定, 波动较大, 所以需要进行滤波处理, 得到一个稳定的 X/Y 数据, 在单片机 ADC

数据滤波时常用的滤波算法有均值滤波算法如式(4.1)将输入的数据进行累加然后求平均值, 中值滤波算法如式(4.2)找出输入的大量数据的中间值, 一阶滞后滤波如式(4.3)将本次输入的数据与上次输入 a 倍数据相乘之后输出。

$$y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n-i] \quad (4.1)$$

$$y[n] = \text{Median}(x[n-N+1], x[n-N+2], \dots, x[n]) \quad (4.2)$$

$$y[n] = \alpha \cdot x[n] + (1-\alpha) \cdot y[n-1] \quad (4.3)$$

上述几种滤波算法各有优劣, 均值滤波简单高效速度快, 但是滤波效果不是太好, 中值滤波效果较好但是延迟高且需求的存储空间大, 一阶滞后的滤波效果又有限, 并且需要人工调试, 应用过程繁杂, 所以综合考虑本设计采用均值滤波算法对 X 轴 Y 轴数据进行滤波整定, 对 X/Y 轴的原始数据多次采样后, 进行累加求和, 之后进行求均值处理, 就得到一个稳定的 X/Y 轴的数据, 接下来就可以根据 X/Y 轴的数据和摇杆动作相结合处理生成机器人的运动控制命令了。

(4) 摇杆动作处理

经过滤波算法处理后的电压数据为浮点数, 在单片机编程中处理起来比较麻烦, 所以再次进行翻倍处理, 直接对数据进行乘 10, 得到的一个三位数的电压值, 因为输入的电压值为 TTL 电平 0V~3.3V, 所以经过乘 10 处理后, 数据的范围为 0~330 之间, 例如: 中间值 170 左右为摇杆静止状态, 也就是让处于机器人静止状态, 当检测 X 轴大于 170 时表示摇杆向左拨动, 产生一个机器人左行的运动控制命令, 当 X 轴小于 170 且小于 155 时产生一个机器人右行的运动控制命令, X/Y 轴数据与机器人运动控制命令关系如下表 4.1, 4.2。

表 4.1 X 轴数据命令表

X 轴	机器人运动命令
>170	左行
>155&&<170	静止
<155	右行

表 4.2 Y 轴数据命令表

Y 轴	机器人运动命令
>170	后退
>155&&<170	静止
<155	前进

ADC 驱动摇杆整体处理算法工作流程如下图 4.18, 图 4.19 所示。

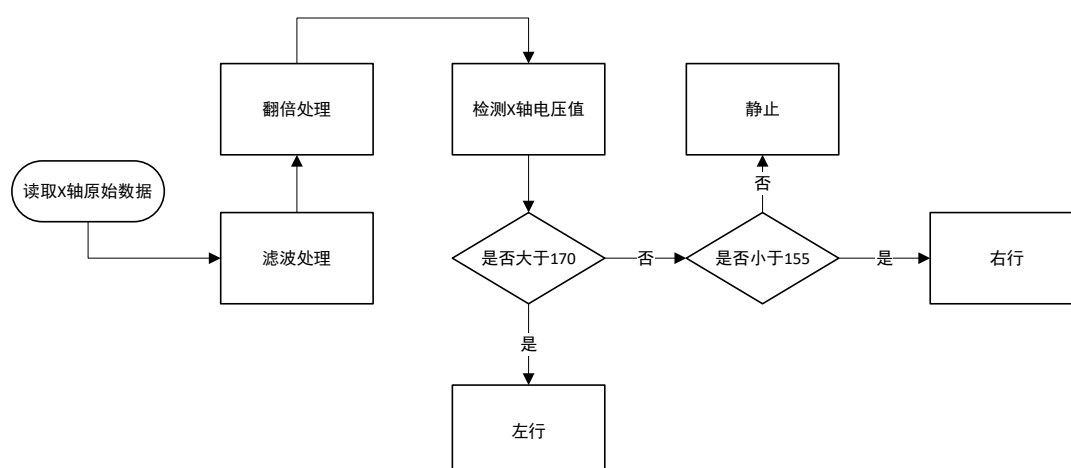


图 4.18 摇杆 X 轴数据处理流程图

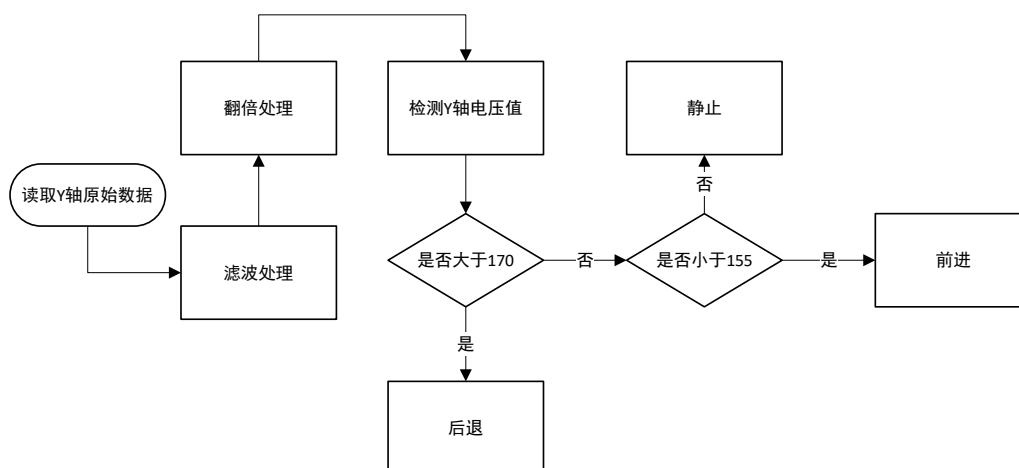


图 4.19 摇杆 Y 轴数据处理流程图

4.4.3 STM32F103 SPI 驱动 LCD 显示屏

使用 STM32 开发 SPI 驱动的具体步骤包括以下几个环节：

(1) 配置时钟

启用 SPI 相关的时钟和 GPIO 时钟。

(2) 配置 GPIO

设置相应 SPI1 的 GPIO 为推挽输出功能。

(3) 初始化 SPI

使用 STM32 标准库中的 SPI 函数初始化 SPI 接口。配置 SPI 的工作模式为主机模式、数据位数 8Bit、时钟极性和相位等参数

(4) 显示屏初始化

通过 SPI 接口向显示屏发送初始化命令序列。用于设置显示屏的参数、工作模式和显示区域等。配置像素格式为 RGB565。

(5) 显示屏应用代码编写

编写用于控制显示屏的函数,如清屏、绘制像素、绘制线条和文本等。这些函数内部将使用 SPI 通信函数与显示屏进行通信。

4.4.4 基础性功能测试

(1) STM32 ADC 驱动摇杆功能测试

实测如下图 4.20,将摇杆连接好之后,波动摇杆,经过相应的处理算法,显示屏上会显示相关的数据信息。

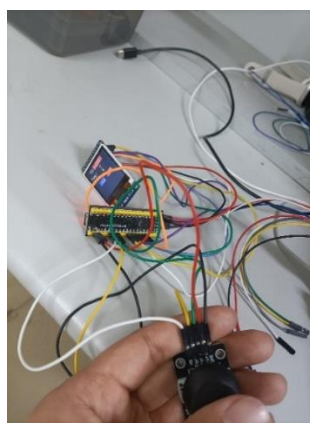


图 4.20 摇杆测试图

(2) STM32 SPI 驱动 TFT 显示屏功能测试

将显示屏接好之后,测试基础的显示字符与色彩的功能实测如下图 4.21。

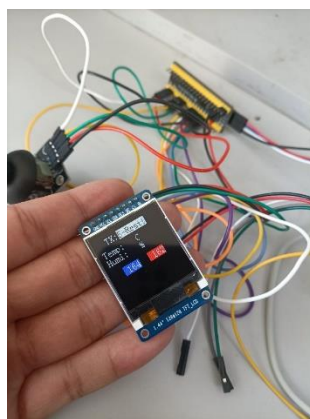


图 4.21 TFT-LCD 显示屏测试图

(3) STM32 UART 功能测试

将 STM32 上的 UART 通过串口模块与 PC 端相连接,然后进行 UART 收发数据测试。实测如下图 4.22。

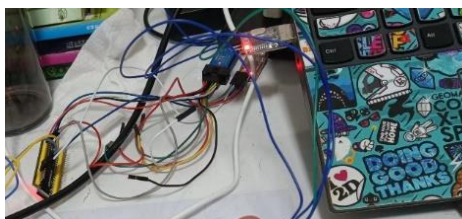


图 4.22 无线控制系统 STM32 UART 测试图

4.4.5 系统性测试

将本系统摇杆功能模块的驱动与显示屏以及蓝牙串口模块的驱动结合到一起,进行整体性测试,实测如下图 4.23,实现了机器人非自动控制策略的控制方式,与接收显示环境数据。

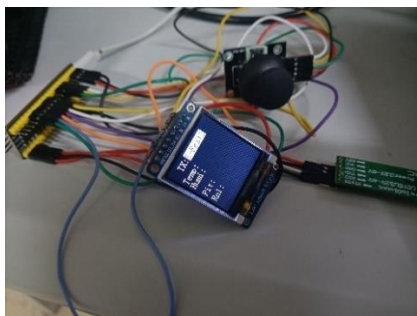


图 4.23 无线控制系统测试图

4.5 机器人整体系统测试

整体测试是将三个子系统分别为机器人主控系统与机器人传感器数据采集及驱动系统,和机器人无线控制系统,整合到一起,进行系统性的联调测试,首先无线控制系统端拨动摇杆通过 ADC 检测数据并处理生成运动控制命令发送到机器人主控系统上,基于嵌入式 Linux 的主控系统运行着环境视觉感知任务的同时,并对接收到的运动控制命令进行处理后,通过 CAN 总线发到机器人传感及驱动系统上,传感及驱动系统根据收到的相应的机器人运动控制命令来驱动机器人进行运动,与此同时此系统还执行着机器人环境温湿度数据感知任务与机器人姿态角度数据检测任务,并将采集到的数据通过 CAN 总线发送到机器人主控系统上,主控系统再经过处理后通过 Linux UART 外设连接蓝牙串口模块以无线的方式发送到机器人的无线控制系统进行数据显示,本测试如下图 4.24,顺利完成了机器人的环境视觉感知问题的研究与机器人环境温湿度数据感知问题的研究,机器人姿态检测问题,以及机器人驱动方式和非自动化控制策略问题的研究。



图 4.24 机器人系统整体测试图

4.6 本章小结

本章对机器人软件系统的设计进行了全面阐述，包括基于 Linux 的主控系统、传感器数据采集及驱动系统、无线控制系统等关键部分的设计和实现。同时，进行了基础性功能测试和系统性功能测试以及综合性系统测试以确保稳定性和可靠性。

第 5 章 总结

本文围绕室内移动机器人的环境感知能力与驱动问题和非自动控制策略展开了深入的研究，本文研究的主要工作内容包含以下几个部分。

（1）机器人主控系统设计，其核心处理器采用恩智浦公司的 ARM Cortex-A7 内核的 I.MX6ULL 单片机，并搭载运行嵌入式 Linux 系统。该主控系统实现了通过 Linux CAN 总线与传感器数据采集及驱动系统进行数据命令交互，以及驱动图像传感器采集环境图像数据，和通过蓝牙串口模块与无线控制系统进行无线数据信息的交互。

（2）传感器数据采集及驱动系统设计，核心处理器采用了意法半导体公司的 ARM Cortex-M3 内核的 STM32F103C8T6 单片机，并搭载运行实时操作系统 FreeRTOS。该系统通过 STM32 的 IIC 外设驱动温湿度传感器与机器人姿态角度传感器，实现对环境温湿度数据与机器人姿态角度数据的获取，并通过 STM32 CAN 总线外设与主控系统进行传输数据和机器人运动控制命令，根据 CAN 总线接收到的机器人运动控制命令进行解析后对电机进行相应控制。

（3）机器人无线控制系统设计同样以 STM32F103C8T6 单片机为核心，搭载运行实时操作系统 FreeRTOS 系统，主要任务包括通过 STM32 ADC 外设驱动摇杆，对摇杆动作进行算法处理并生成机器人运动控制命令，通过 UART 将数据发送到蓝牙串口模块然后再发送到机器人主控系统上。其次无线控制系统还通过 STM32 SPI 外设驱动的 TFT-LCD 显示屏实时显示 UART 外设收到的机器人主控发来的传感器数据信息与主控系统发出的运动机器人运动控制命令。

致 谢

首先，我要郑重感谢我的导师刘炜老师，在我的毕业设计论文写作过程中，给予我专业且细致的指导与帮助，他的专业知识和独到的见解在我的毕业设计过程中起到了至关重要的作用。刘老师不仅在学术上给予我细致入微的指导，更在思考认知和学术研究的问题上对我产生了深远的影响。

其次我要感谢我的得力伙伴赵福帧同学，在我做毕设过程中与我共同商讨所遇到的技术难题，并提供关键的解决思路，使得我的毕设工作更进一步。

此外，还要感谢在我做毕设过程中遇到难题，迷茫时，对我提供帮助指导的网友们，和在本科四年学习过程中给予我启发和支持的各位老师，他们不仅传授给我专业知识，更是教会了我如何独立思考和解决问题，为我未来的发展奠定了坚实的基础，老师们的教诲和悉心指导让我受益终生，在我人生道路上指引着我前行。

最后我要感谢所有在我毕业设计过程中给予我帮助和支持的老师和同学们。正是因为你们的支持与鼓励，我才能够顺利完成毕业设计，迈向人生新的起点。衷心感谢你们！

参考文献

- [1] 宋显铭.基于改进A*与改进DWA算法的室内移动机器人路径规划研究[D].大连交通大学,2023.
- [2] HUO Fengcai, CHI Jin, HUANG Zijian, REN Lu, SUN Qinjiang, CHEN Jianling. Review of Path Planning for Mobile Robots[J].Journal of Jilin University (Information Science Edition), 2018, 36(6): 639-647.
- [3] 徐淑萍,杨定哲,熊小墩.多传感器融合的室内机器人SLAM[J].西安工业大学学报,2024,44(01):93-103.
- [4] Albus, J.S. and Evans, J.M., 1976. Robot systems. Scientific American, 234(2), pp.76-87.
- [5] Giralt, G. (1984). Mobile Robots. In: Brady, M., Gerhardt, L.A., Davidson, H.F. (eds)Robotics and Artificial Intelligence. NATO ASI Series, vol 11. Springer, Berlin, Heidelberg.
- [6] Moravec H P. The Stanford cart and the CMU rover[J]. Proceedings of the IEEE, 1983, 71(7): 872-884.
- [7] Crowley J. Navigation for an intelligent mobile robot[J]. IEEE Journal on Robotics and Automation, 1985, 1(1): 31-41.
- [8] Rodin E Y, Amin S M. Intelligent navigation for an autonomous mobile robot[C]//Proceedings 1988 IEEE International Symposium on Intelligent Control. IEEE Computer Society, 1988: 366,367,368,369-366,367,368,369.
- [9] Oh S, Zelinsky A, Taylor K. Autonomous battery recharging for indoor mobile robots[C]//Proceedings of the Australian conference on robotics and automation. 2000.
- [10] Biswas J, Veloso M. Wifi localization and navigation for autonomous indoor mobile robots[C]//2010 IEEE international conference on robotics and automation. IEEE, 2010: 4379-4384.
- [11] de Oliveira Júnior A, Piardi L, Bertogna E G, et al. Improving the mobile robots indoor localization system by combining slam with fiducial markers[C]//2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE). IEEE, 2021: 234-239.
- [12] 李育发. 移动式机器人环境感知系统--基于多传感器信息融合的室内环境建模[D]. 辽宁:中国科学院沈阳自动化研究所,1991.
- [13] 许永华,蔡自兴,刘娟,等. 基于图像分析的室内自主式移动机器人避障系统[J]. 计算技术与自动化,2001,20(z1):198-201.
- [14] 项志宇,刘济林. 基于环境特征跟踪的移动机器人定位[J]. 仪器仪表学报,2003,24(4):391-394.
- [15] 庄严. 移动机器人基于多传感器数据融合的定位及地图创建研究[D].大连理工大学,2005.

- [16] 胡春旭,熊泉,任慰,等.基于嵌入式系统的室内移动机器人定位与导航[J].华中科技大学学报(自然科学版),2013,41(S1):254-257+266.
- [17] Chen C, Zhu H, Wang L, et al. A stereo visual-inertial SLAM approach for indoor mobile robots in unknown environments without occlusions[J]. IEEE Access, 2019,7: 185408-185421.
- [18] 袁杰. 室内复杂环境下移动机器人路径规划技术研究[D].东南大学,2022.
- [19] 陈睿韵,田文斌,鲍海波,等.农业轮式机器人三维环境感知技术研究进展[J].智慧农业(中英文),2023,5(04):16-32.
- [20] 杜传胜.大型仓库消防机器人控制系统设计与路径规划方法研究[D].山东建筑大学,2023.
- [21] 谢鹏程.基于STM32 和FreeRTOS的独立式运动控制器设计与研究[D].华南理工大学,2012.
- [22] 赵晓丽.微型四旋翼飞行器的嵌入式视频监控终端的研发[D].吉林大学,2016.
- [23] 吴树添,全钊锋,张春美,等.基于互联网+智能化背景下的微型多功能气象仪设计[J].电子制作,2024,32(05):11-14.
- [24] 张承岫,李铁鹰,王耀力.基于MPU6050 和互补滤波的四旋翼飞控系统设计与研究[J].传感技术学报,2016,29(07):1011-1015.
- [25] 林文建,钟杭,黎福海,等.两轮自平衡机器人控制系统设计与实现[J].电子测量与仪器学报,2013,27(08):750-759.
- [26] 唐恒飞.基于CCP协议的新能源汽车控制器标定系统研究[D].上海工程技术大学,2021.
- [27] 刘常杰,李斌,郭寅,等.焊接机器人实时引导中光条快速提取算法研究[J].激光技术,2015,39(05):642-647.