

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

本文旨在探讨基于单片机的室内移动机器人的设计与应用，近年来随着机器人技术的发展，室内移动机器人作为机器人领域研究的一个重要方向，也已取得了突破性的发展，目前室内移动机器人已应用于家庭、医院、商场等多种复杂场景。本文着重于机器人的软件系统设计与实现，研究内容主要包括机器人主控系统设计与实现，传感器数据采集及驱动系统设计与实现与机器人无线控制系统设计与实现，具体研究内容如下。

首先，设计机器人主控系统，其核心处理器采用恩智浦（NXP）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。

其次，设计传感器数据采集及驱动系统，核心处理器采用了意法半导体（STMICROELECTRONICS）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处理解析后对电机进行相应控制。机器人无线控制系统同样以 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。

最后，在机器人综合系统的设计过程中，本文还进行了基础性功能测试与系统性功能测试以及机器人整体系统测试，以确保整个机器人系统的正常稳定运行。

关键词：单片机；室内移动机器人；嵌入式 Linux 系统；实时操作系统；传感器数据采集；CAN 通信

Abstract

This article aims to explore the design and application of indoor mobile robots based on microcontrollers. With the development of robotics technology in recent years, indoor mobile robots have become an important direction in the field of robotics research and have achieved breakthrough progress. Currently, indoor mobile robots have been applied in various complex scenarios such as homes, hospitals, and shopping malls. The focus of this article is on the design and implementation of the robot's software system, which mainly includes the design and implementation of the robot's main control system, sensor data acquisition and drive system, and the design and implementation of the robot's wireless control system. The specific research content is as follows.

Firstly, the main control system of the robot is designed with the core processor being the IMX6ULL microcontroller from NXP company, which features an ARM Cortex-A7 core and runs the embedded Linux operating system. This main control system achieves data and command interaction with the sensor data acquisition and drive system through the Linux CAN bus. It also drives the image sensor to capture environmental image data and exchanges wireless data information with the wireless control system through the Bluetooth serial port module.

Secondly, the sensor data acquisition and drive system is designed with the core processor being the STM32F103C8T6 microcontroller from STMicroelectronics company, featuring an ARM Cortex-M3 core and running the real-time operating system FreeRTOS. This system drives temperature and humidity sensors and robot attitude angle sensors through the STM32's IIC peripheral to acquire environmental temperature and humidity data as well as robot attitude angle data. It transmits data and robot motion control commands to the main control system via the STM32 CAN bus peripheral and processes and parses the received robot motion control commands through the CAN bus to control the motors accordingly. The robot's wireless control system, also centered around the STM32F103C8T6 microcontroller and running the FreeRTOS system, primarily drives the joystick through the STM32 ADC peripheral, processes the joystick movements algorithmically to generate robot motion control commands, and sends the data to the robot's main control system via the UART and Bluetooth serial port module. Additionally, the wireless control system displays real-time information of the sensor data received from the robot's main con

trol and the robot motion control commands issued by the main system on a TFT-LCD display driven by the STM32 SPI peripheral.

Finally, during the design process of the integrated robot system, this article also conducts basic functional testing, systemic functional testing, and overall system testing of the robot to ensure the normal and stable operation of the entire robot system.

Keywords: Single-chip microcontroller; Indoor mobile robot; Embedded Linux system; Real-time operating system; Sensor data acquisition; CAN communication

目录

摘要.....	I
Abstract	II
目录.....	IV
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 本文主要研究内容.....	2
第 2 章 机器人系统设计.....	4
2.1 引言.....	4
2.2 机器人系统方案设计.....	4
2.2.1 机器人驱动方案设计.....	5
2.2.2 机器人主控制系统方案设计.....	6
2.2.3 机器人无线控制系统方案设计.....	7
2.2.4 机器人传感器数据采集及驱动系统方案设计.....	8
2.2.5 机器人系统整体方案设计.....	10
2.3 机器人系统工作流程.....	10
2.3.1 机器人主控制系统工作流程.....	10
2.3.2 机器人传感器与驱动系统工作流程.....	11
2.3.3 机器人无线控制系统工作流程.....	12
2.3.4 机器人系统整体工作流程.....	13
2.3 本章小结.....	14
第 3 章 机器人硬件电路系统设计.....	15
3.1 引言.....	15
3.2 机器人主控制系统硬件电路.....	15
3.2.1 I.MX6ULL 核心工作电路.....	15
3.2.2 蓝牙串口模块电路.....	17
3.2.3 CAN 总线电路.....	18
3.3 机器人传感器数据采集及驱动系统硬件电路.....	19
3.3.1 STM32F103C8T6 核心工作电路.....	19
3.3.2 CAN 总线电路.....	21
3.3.3 温湿度传感器电路.....	22
3.3.4 姿态角度传感器电路.....	23
3.3.5 永磁有刷直流电机驱动电路.....	24
3.3.6 OLED 显示屏电路.....	24
3.4 机器人无线控制系统硬件电路.....	26
3.4.1 STM32F103C8T6 核心工作电路.....	26
3.4.2 TFT-LCD 显示屏电路.....	27
3.4.3 PS2 摇杆电路.....	28

3.5 本章小结	28
第 4 章 机器人软件系统设计	29
4.1 引言	29
4.2 机器人主控嵌入式 linux 系统.....	29
4.2.1 Linux CAN 驱动	29
4.2.2 Linux Uart 驱动	30
4.2.3 Linux V4L2 for Camera 驱动	32
4.2.4 基础性功能测试.....	33
4.2.5 系统性测试.....	35
4.3 机器人传感器数据采集及驱动系统.....	35
4.3.1 STM32F103 移植 FreeRTOS	35
4.3.2 STM32F103 UART 驱动.....	37
4.3.3 STM32F103 CAN 通信驱动	38
4.3.4 STM32F103 IIC 驱动传感器	39
4.3.5 STM32F103 PWM 驱动直流电机.....	40
4.3.6 基础性功能测试.....	41
4.3.7 系统性测试.....	43
4.4 机器人无线控制系统.....	43
4.4.1 STM32F103 UART 驱动.....	43
4.4.2 STM32F103 ADC 驱动 PS2 摇杆.....	44
4.4.3 STM32F103 SPI 驱动 TFT 显示屏.....	44
4.4.4 基础性功能测试.....	45
4.4.5 系统性测试.....	46
4.5 机器人系统整体测试.....	47
4.6 本章小结	47
第 5 章 总结.....	48
5.1 总结	48
致谢.....	49
参考文献.....	50

第1章 绪论

1.1 研究背景及意义

随着科技的进步，计算机和自动化技术逐渐融入人们的日常生活。其中，室内移动机器人作为智能家居和智能服务的重要部分，其设计与应用具有广泛的实际意义。基于单片机的室内移动机器人的设计与应用，不仅是对单片机技术、传感器技术、移动机器人技术的综合应用，更是对现代智能家居系统的完善和补充见上方^[1]。

首先，单片机作为一种微型计算机，在很多嵌入式系统和智能设备中扮演着核心角色。通过单片机，我们可以实现对室内移动机器人的精确控制，包括环境监测、环境图像数据监测、任务执行等。

其次，传感器技术的应用使得室内移动机器人能够更好地感知环境，与人交互。例如，通过红外传感器，机器人可以检测到障碍物；通过超声波传感器，可以测量距离；通过图像传感器技术，机器人可以看得见环境的图像数据。

最后，从应用角度看，室内移动机器人可以在家庭、办公室、医院等场合发挥重要作用。它们可以帮助人们完成一些重复、繁琐的任务，释放人力去做更有价值的工作。此外，它们还可以作为智能家居系统的一部分，与其他智能设备协同工作，提升家居生活的便利性和舒适性。

本文旨在研究并探索几个方面的问题，目的在于揭示其意义和重要性。

1. 机器人的嵌入式系统架构设计与室内移动机器人的运动方式。
2. 室内移动机器人的电子系统设计和多传感器的应用与数据处理。
3. 机器人系统的通信系统设计与机器人系统的性能优化。

综上所述，基于单片机的室内移动机器人的设计与应用这一选题，不仅有助于提高我们的技术应用能力，还能满足社会对智能化生活的需求，具有深远的实际意义。

1.2 国内外研究现状

室内移动机器人属于服务类型的机器人，是较为常见的类型，其具有较强的灵活性和机动性，可以用于各种应用领域，例如救援、探险、城市作战等。在机器人的研究中，机器人的嵌入式系统架构设计是一个非常重要的问题，其研究涉及到机器人的稳定性、安全性、性能等方面，对于室内机器人的实际应用具有重要意义^[2]。

室内移动机器人在无导航系统的情况下，主要依靠传感器和计算机技术来实现自主导航和定位。

在国内，无导航系统的室内移动机器人研究正在逐渐受到关注。一些研究团队致力于开发基于传感器融合的自主导航系统，通过感知环境中的特征、障碍物和已知的地图信息，实现机器人的自主探测和定位。此外，还有一些研究关注于基于视觉的导航方式，通过识别周围环境线索，使机器人能够进行有效的路径规划和定位。这些研究在智能家居、医疗护理等领域得到了初步的应用和验证^[3]。

在国外，无导航系统的室内移动机器人研究已经得到了广泛的关注和进展。一些国际知名企业如微软、英特尔等在无导航系统的室内移动机器人技术方面拥有多项专利和技术突破。这些技术基于传感器融合和计算机技术，通过感知环境中的特征和障碍物，实现机器人的自主导航和定位。此外，国外的研究还注重与虚拟现实、增强现实等技术的结合，使机器人能够更好地与用户进行交互和提供服务。这些研究成果已经在智能家居、医疗护理、商业服务等领域得到了广泛的应用和推广。

无导航系统的室内移动机器人研究是一个具有挑战性和前景的研究领域。国内外的研究者们正在通过不同的技术和方法进行探索和研究，并取得了一定的进展。未来，随着技术的不断进步和应用场景的不断拓展，无导航系统的室内移动机器人有望得到更广泛的应用和发展。同时，需要加强国际间的合作与交流，促进该领域的共同进步和创新。

1.3 本文主要研究内容

本文致力于室内移动机器人的系统设计与实现，深入探讨了从理论到实践的全流程。通过系统地研究国内外研究现状，明确了当前室内移动机器人技术的发展趋势和挑战，进而确定了本文的研究方向和目标。

首先，在机器人系统设计方面，本文提出了一个完整的室内移动机器人系统设计方案。该方案充分考虑了室内环境的复杂性和多样性，选择了适合室内导航和移动的驱动方式，包括轮式驱动和步进式驱动等^[4]。同时，针对机器人的主控系统，本文对比了多种方案，最终选择了基于高性能微处理器的设计方案，以确保机器人具有强大的计算能力和数据处理能力。此外，本文还设计了基于无线通信技术的机器人无线控制系统，实现了对机器人

的远程操控和监控。

其次，在硬件电路系统设计方面，本文详细阐述了各个模块的硬件电路设计。对于机器人主控系统，本文设计了包括电源电路、时钟电路、复位电路等在内的完整电路系统，以确保主控系统的稳定性和可靠性。对于传感器数据采集及驱动系统，本文设计了多种传感器的接口电路，包括温湿度传感器、姿态角度传感器等，以及永磁有刷直流电机的驱动电路。这些电路的设计使得机器人能够实时感知环境信息并准确控制运动。此外，本文还设计了 OLED 显示屏电路，用于显示机器人的状态信息和调试信息。对于无线控制系统，本文设计了基于 STM32 的无线控制模块，包括 PS2 摇杆电路和 TFT-LCD 显示屏电路，以实现机器人的远程控制和信息显示^[5]。

在软件系统设计方面，本文首先针对机器人主控系统进行了嵌入式 Linux 系统的设计和开发。通过编写 Linux CAN 驱动、Linux Uart 驱动以及 Linux V4L2 for Camera 驱动，实现了对机器人硬件的底层控制。同时，本文还对 Linux 系统进行了优化和定制，以适应室内移动机器人的应用需求。对于传感器数据采集及驱动系统，本文在 STM32F103 平台上移植了 FreeRTOS 实时操作系统，并编写了 UART 驱动、CAN 通信驱动、IIC 驱动传感器以及 PWM 驱动直流电机等关键代码。这些代码使得机器人能够实时采集和处理环境信息，并根据控制指令进行精确运动。对于无线控制系统，本文编写了 STM32 的 UART 驱动、ADC 驱动 PS2 摇杆以及 SPI 驱动 TFT-LCD 显示屏等代码，实现了对机器人的远程控制 and 状态显示以及机器人姿态角度数据和环境温湿度数据的显示。

最后，在整体测试方面，本文对室内移动机器人系统进行了全面的测试。首先，对各个模块进行了功能性测试，确保每个模块都能正常工作。然后，对整个系统进行了系统性测试，包括导航精度测试、避障能力测试以及人机交互性能测试等。测试结果表明，本文所设计的室内移动机器人系统具有较高的稳定性和可靠性，能够满足室内环境下的多种应用需求。

综上所述，本文主要研究了室内移动机器人的系统设计与实现，包括整体系统方案设计、硬件电路系统设计、软件系统设计以及整体测试等方面。通过本文的研究，不仅为室内移动机器人的进一步发展和应用提供了重要的理论支持和实践经验，也为相关领域的研究人员提供了有益的参考和借鉴。

第 2 章 机器人系统设计

2.1 引言

机器人是个综合系统如图 1.1，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，为了提高机器人整个系统运行的高效性，故将驱动系统与感知系统集成成一个整体为传感器数据采集及驱动系统，且机器人的控制方式不采用传统的自动控制或半自动控制方式，而是开创性的采用人工控制方式，因为当前的控制技术水平仍存在较强的局限性，在执行某些复杂任务的场景中人工控制方式反而更高效，更智能。

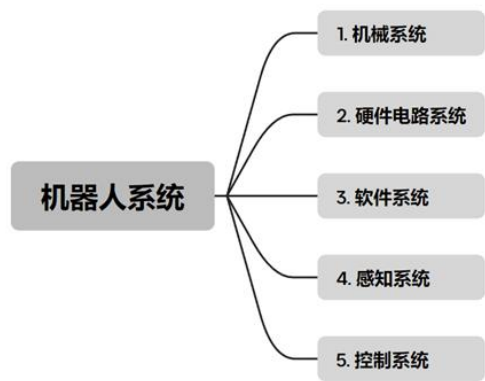


图 1.1 机器人系统组成结构

2.2 机器人系统方案设计

本文中所研究的室内移动机器人内容如图 1.2 所示，根据需求将任务划分为以下三个部分，机器人的无线控制系统，与机器人主控系统，及其机器人传感器数据采集及驱动系统，为了提高系统的集成度，与处理数据的高效性，所以将机器人感知系统与机器人驱动系统集成到了一个系统，机器人传感器数据采集及驱动系统，且控制系统不采用传统的自动控制的方式而采用人工控制的方式。

（1）机器人无线控制系统

主要负责接收机器人端的传感器数据，与机器人自身姿态数据，并向机器人发送运动控制命令，来控制机器人运动。

（2）机器人主控系统

负责接收传感器系统发过来的传感器数据，进行数据处理，并发送到机器人无线控制系统，除此之外机器人主控还负责进行环境图像数据的采集。

（3）机器人传感器数据采集及驱动系统

主要负责采集环境中的数据与机器人驱动。

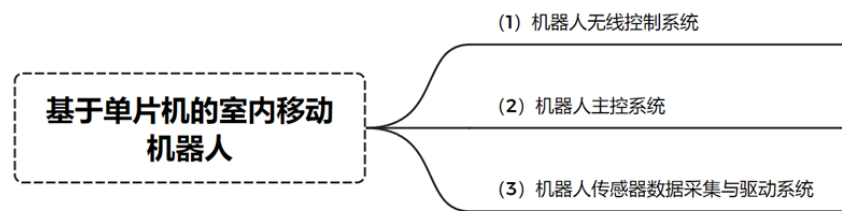


图 1.2 室内移动机器人研究内容

2.2.1 机器人驱动方案设计

轮式机器人又分为独轮式如图 1.3 所示，双轮式如图 1.4 所示，和多轮式机器人如图 1.5 所示，轮式机器人的优点，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 1.6，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 [6]，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。

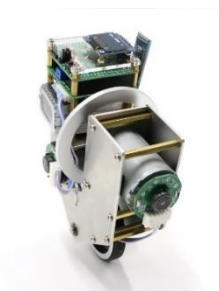


图 1.3 独轮式机器人



图 1.4 双轮式机器人



图 1.5 多轮式机器人



图 1.6 履带式机器人

2.2.2 机器人主控系统方案设计

主控负责数据处理和环境图像数据采集

（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^[7]。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^[8]，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 1.7。

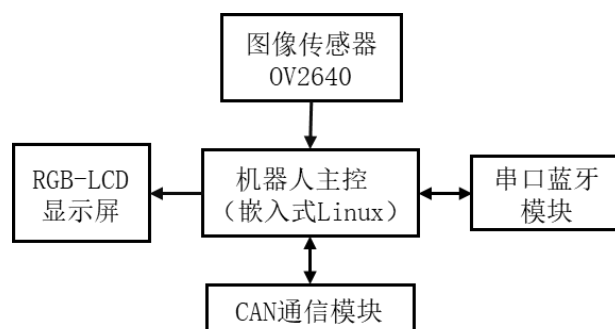


图 1.7 机器人主控系统组成框架图

2.2.3 机器人无线控制系统方案设计

使用人工控制取代自动控制，此系统类似于机器人运动控制系统，不过并不是自动控制或半自动控制，而是人工控制，由于自动控制技术水平的发展情况以及当前技术水平的局限性，本设计中的机器人系统选择人工控制，在执行某些复杂任务环境下，人工控制反而比自动控制或半自动控制更高效，更智能。

（1）软件系统的选择

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^[9]，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

所选用的摇杆为 PS2 摇杆如图 1.8，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 1.9。



图 1.8 PS2 摇杆



图 1.9 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，由此得出机器人主控系统组成框架如图 2.0。

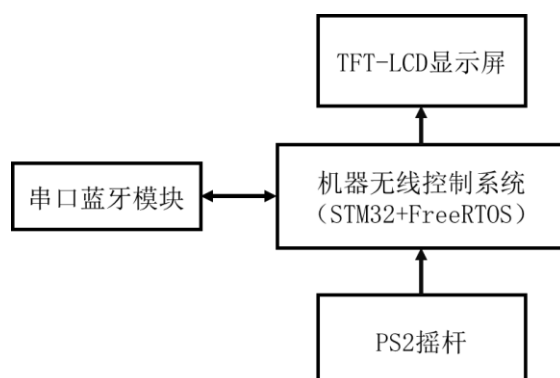


图 2.0 无线控制系统组成框架图

2.2.4 机器人传感器数据采集及驱动系统方案设计

此系统是将机器人感知系统与机器人驱动系统集成到了一起，主要作用就是采集机器人传感器的数据和驱动机器人。

（1）软件系统选择

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（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，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）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。温湿度采集选用国产奥松电子公司旗下的 AHT10 传感器，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^[10]。姿态角度传感器采用 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。

（3）机器人驱动方式选择

本文研究类型为移动机器人，并选择了履带式的驱动方式，履带式机器人的电机一般分为永磁无刷直流电动机（BLDC）如图 2.1 与永磁有刷直流电动机如图 2.2，永磁有刷直流电机具有驱动方式便捷简单且成本较低的优势。这种类型的电机在移动机器人中应用广泛，其结构简单、可靠性高，同时具有较高的功率密度和适应性，能够满足移动机器人对于驱动力和控制的需求^[10]。因此，选择永磁有刷直流电机作为履带式机器人的驱动方式是合理的选择，能够有效地支持机器人的移动和运动控制。



图 2.1 永磁无刷直流电动机（BLDC）



图 2.2 永磁有刷直流电动机

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 2.3。

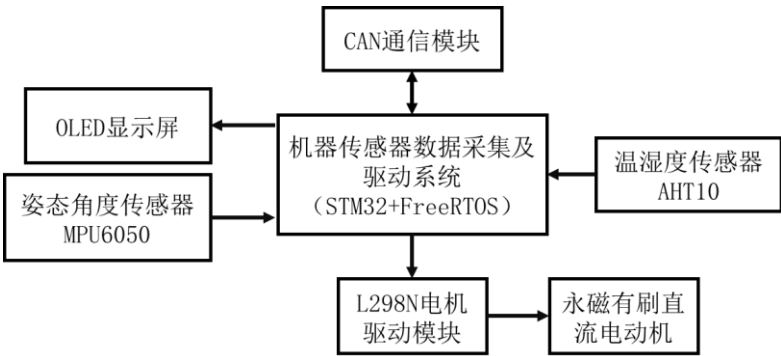


图 2.3 机器人传感器数据采集及驱动系统组成框架图

2.2.5 机器人系统整体方案设计

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，整体系统的组成框架如下图 2.4。

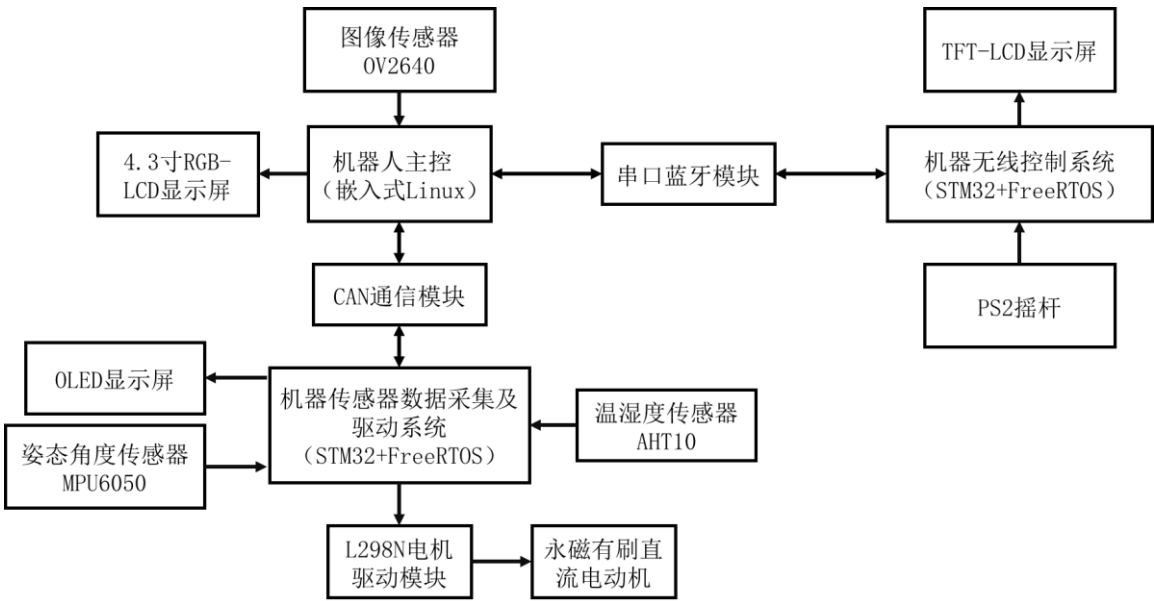


图 2.4 机器人综合系统组成框架图

2.3 机器人系统工作流程

2.3.1 机器人主控系统工作流程

主控系统中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 2.5 分为三个线程 CAN 总线数据处理任务线程 [12]，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。

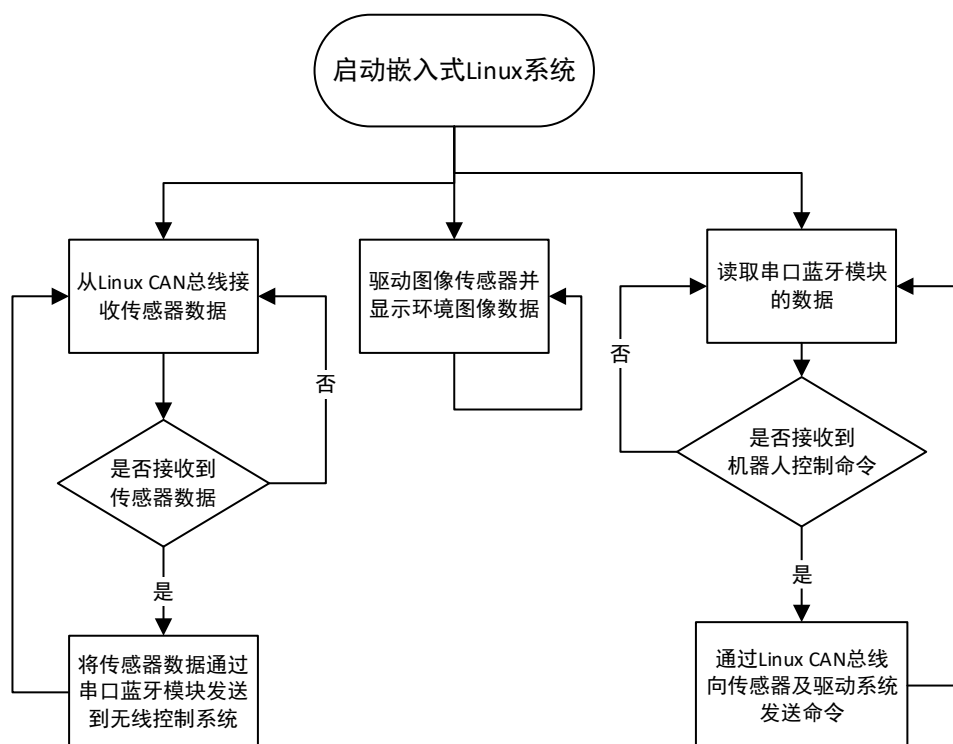


图 2.5 机器人主控制系统工作流程图

2.3.2 机器人传感器与驱动系统工作流程

首先给 STM32 上电启动它，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^[13]，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。

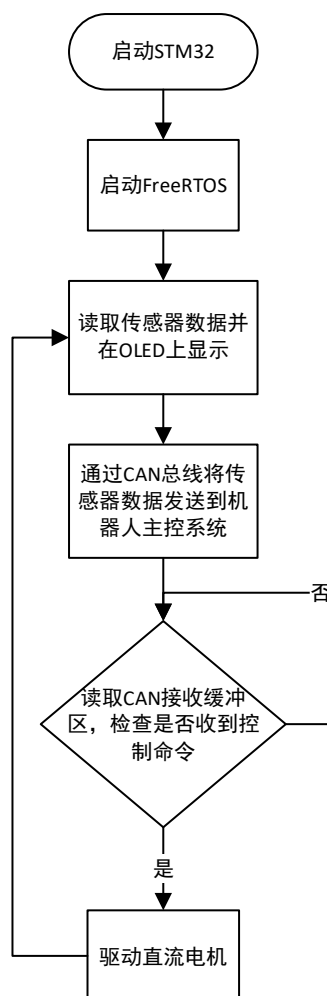


图 2.6 机器人传感及驱动系统工作流程图

2.3.3 机器人无线控制系统工作流程

此子系统的工作流程前几步与传感器数据采集及驱动系统相似，首先启动 STM32，其次启动 FreeRTOS，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 PS2 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制系统，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。

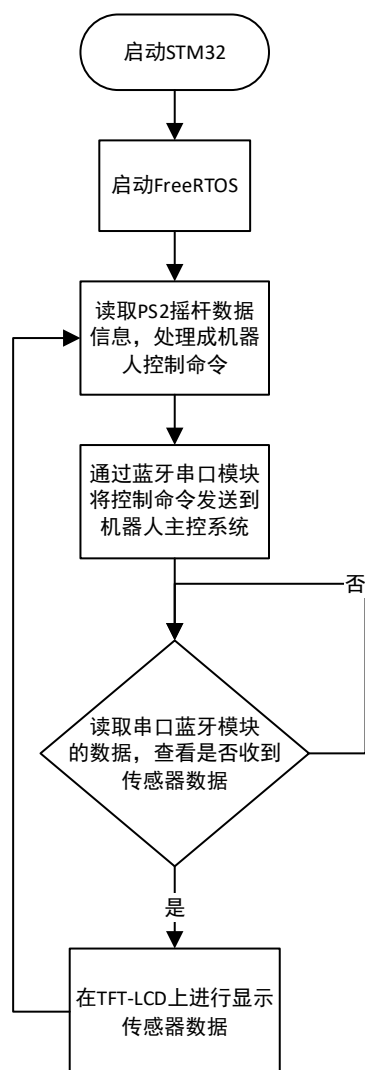


图 2.7 机器人无线控制系统工作流程图

2.3.4 机器人系统整体工作流程

主控系统与传感器采集及驱动系统，无线控制系统同时运行的，且是并行运行关系。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^[14]。

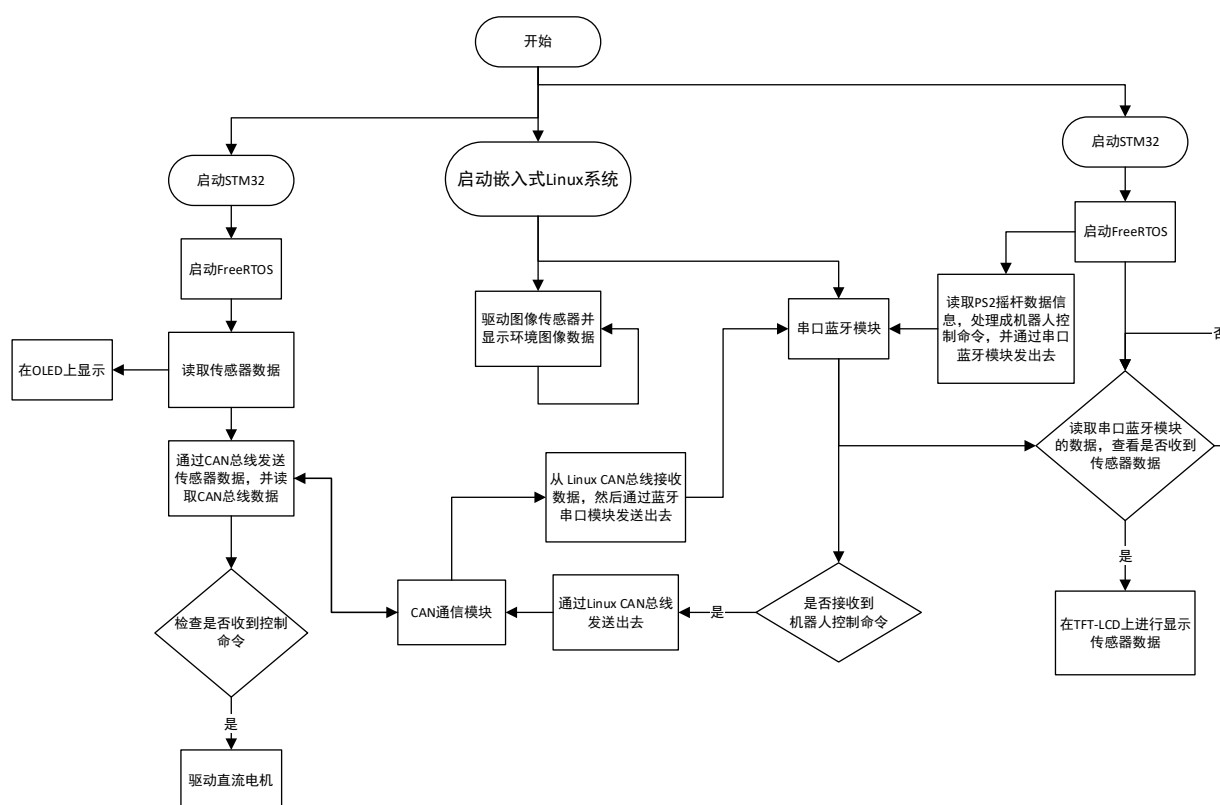


图 2.8 机器人系统整体工作流程图

2.3 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^[15]。

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 2.9，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) 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。

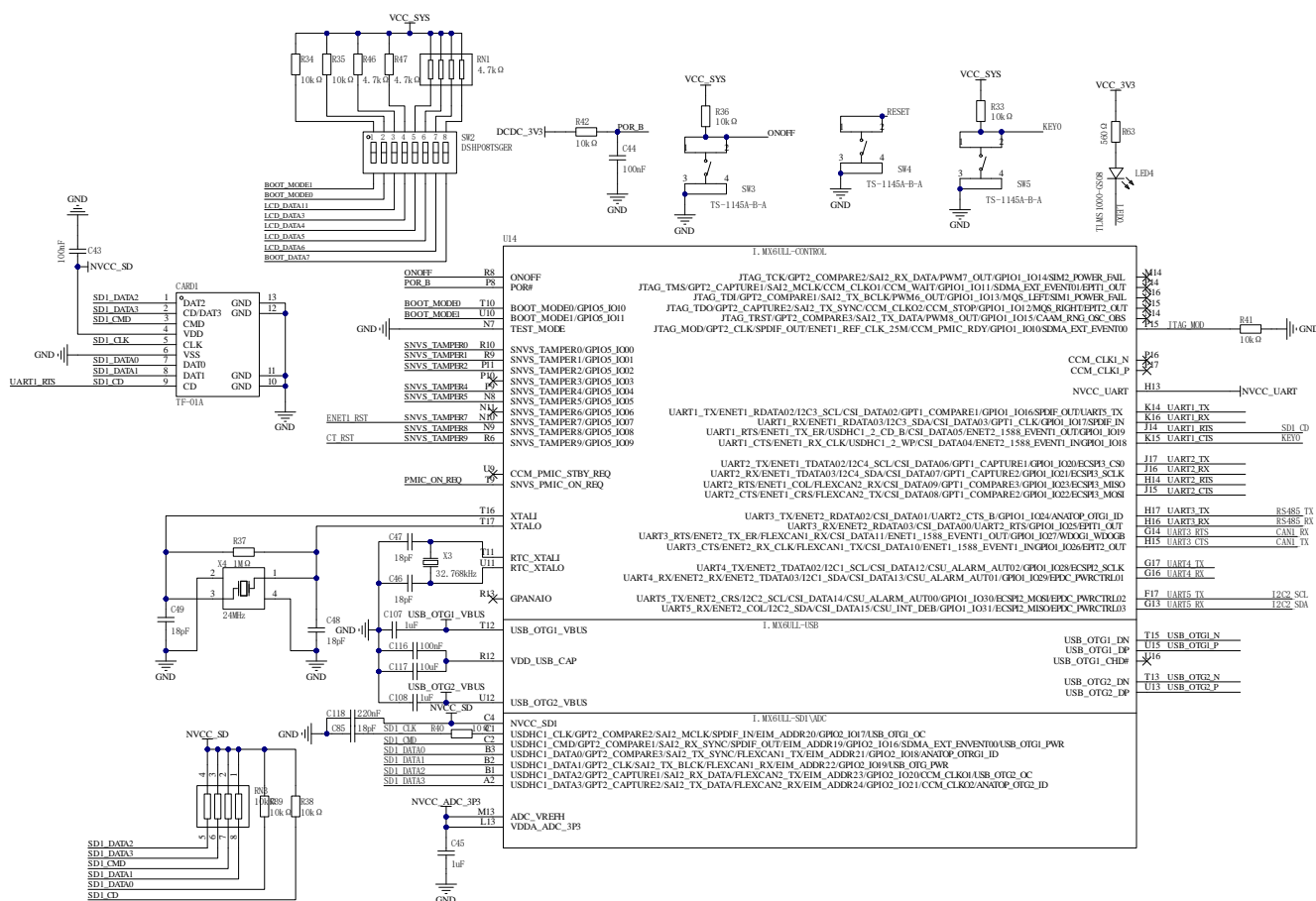


图 2.9 I.MX6ULL 核心工作电路图

3.2.2 蓝牙串口模块电路

无线通信方式采用的是蓝牙通信方式，本课题使用蓝牙串口模块来实现，具体型号为 HC04 如图 3.0，HC-04 蓝牙串口通信模块是新一代的基于 SPP&BLE 蓝牙协议的双模数传模块支持 BLE5.0。无线工作频段为 2.4GHz ISM，调制方式是 GFSK。模块最大发射功率为 6dBm，接收灵敏度为-92dBm。其工作原理为两个蓝牙串口模块^[16]，一个主机一个从机建立连接关系，然后蓝牙串口模块的 TXD，RXD 引脚接到单片机的 UART IO 上，蓝牙主机模块接的单片机向 UART 数据寄存器写入数据后蓝牙串口模块收到数据通过蓝牙方式传输到蓝牙串口从机模块，从机模块将数据传从到单片机 UART 数据接收寄存器中，这样蓝牙串口主机模块和从机模块就可以互传数据了，连接关系如图 3.1。



图 3.0 HC04 蓝牙串口模块实物图

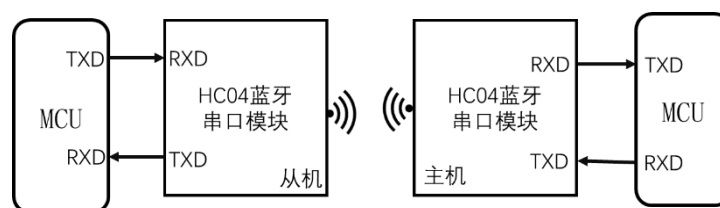


图 3.1 蓝牙串口模块主从机连接关系图

HC04 蓝牙串口模块电路设计如下图 3.2，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。

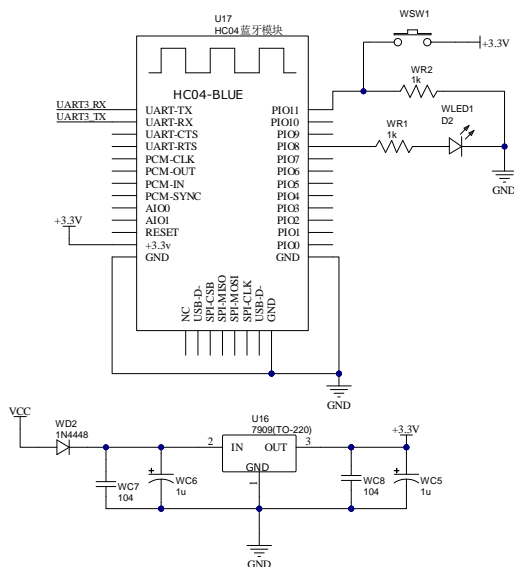


图 3.2 蓝牙串口模块电路图

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 3.3，本课题所使用的 CAN 电平转换芯片为 TJA1050，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。

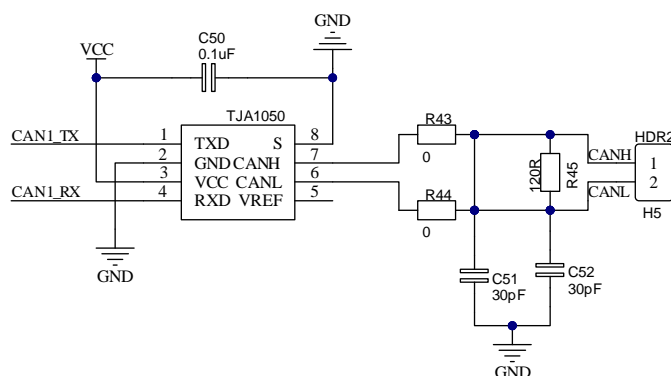


图 3.3 CAN 总线模块电路图

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。这款产品属于 STM32 系列中的 F1 分支，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。其主频为 72MHz，工作电压 2.0V~3.6V，工作温度 -40℃~+85℃，程序存储器（FLASH）容量 64KByte，数据存储器（SRAM）容量 20KByte，内核外设设有 USART，IIC，SPI，USB（从机模式），CAN，ADC，PWM，GPIO 以及通用 Timer 和高级 Timer，还有 ARM Cortex-M3 内核自带的 Timer SysTick，以及 JTAG/SWD 接口。STM32F103C8T6 核心工作电路如下图 3.4。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对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

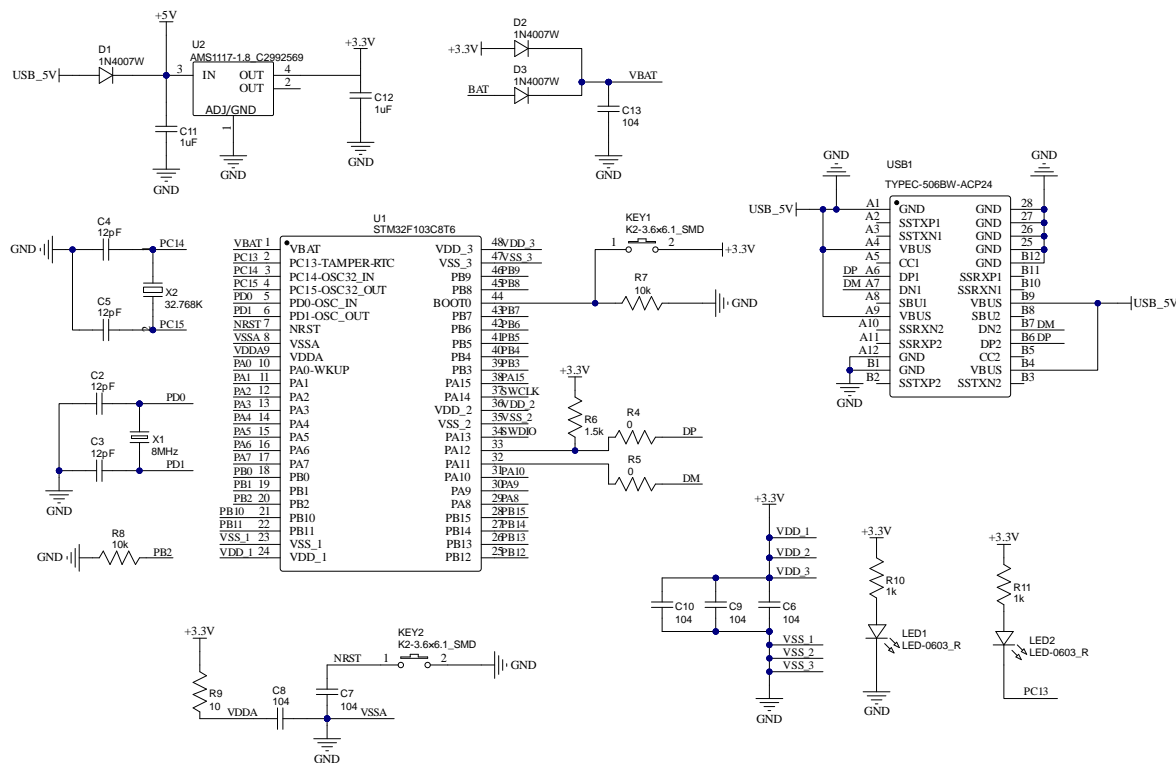
单片机上电在开始的几个时钟周期内会根据不同的 BOOT IO 电平进行配置系统运行模式，例如上电后从 FLASH 区域读取并运行程序，或者从 RAM 区域读取并运行程序。

（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。



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 3.3，本课题所使用的 CAN 电平转换芯片为 TJA1050，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。

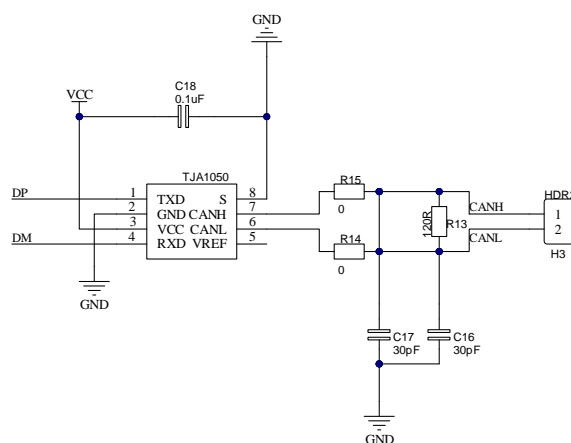


图 3.5 CAN 总线模块电路图

3.3.3 温湿度传感器电路

温湿度传感器芯片采用的是国产奥松电子的 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，其额定工作电压为 3.3V~3.6V，IIC 协议接口，湿度精度 $\pm 2\%$ ，湿度分辨率为 0.024%，湿度精度 $\pm 0.3^{\circ}\text{C}$ ，湿度分辨率 0.01°C ，测量温度范围 $-40^{\circ}\text{C}\sim 85^{\circ}\text{C}$ 。

下图为 AHT10 传感器模块的电路图如图 3.6，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。

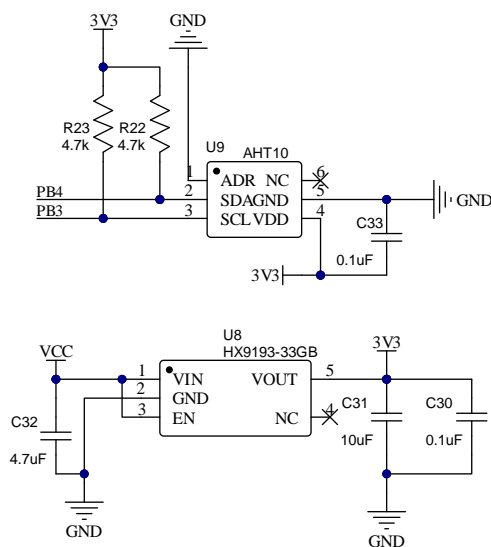


图 3.6 AHT10 温湿度传感器模块电路图

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 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，是一种常用的惯性测量装置，工作电压为 2.375V~3.46V，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。以下为 MPU6050 姿态传感器的工作电路如下图 3.7，主要包含电源电路，与 MPU6050 芯片 IIC 通信电路。

（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 MPU 6050 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。

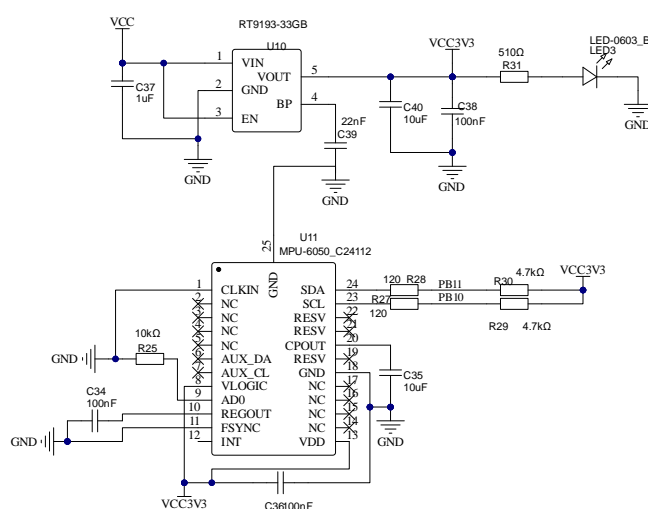


图 3.7 MPU6050 姿态角度传感器模块电路图

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，工作电压可达 46V，输出电流最高可至 4A，额定功率为 25W，且可靠性高，抗干扰能力强。下图 3.8 为 L298N 的驱动电路，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。

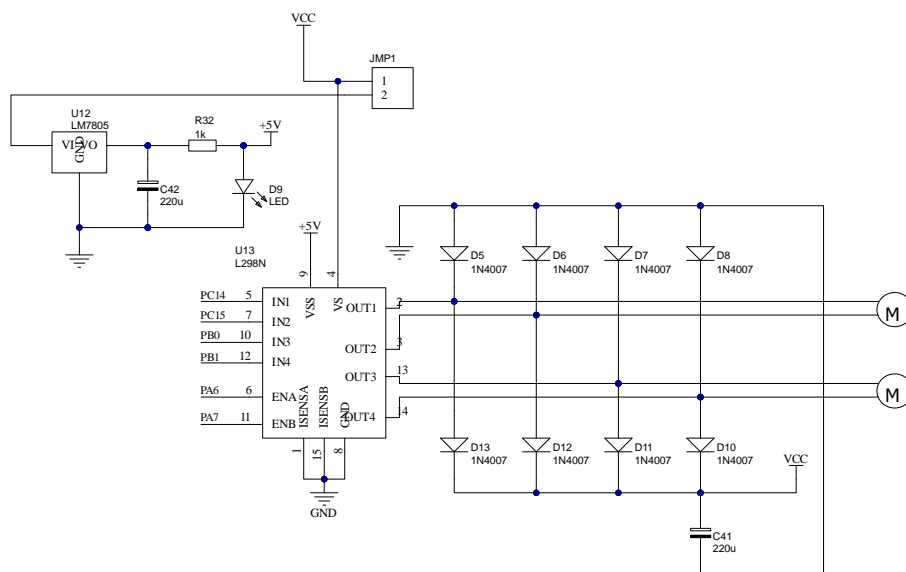


图 3.8 L298N 电机驱动电路图

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，所采用的传感器为 0.96 寸单色 OLED 显示屏，供电电压为 3.3V-5V，分辨率为 128*64，IIC 通信接口，下图 3.9 为 0.96OLED 显示屏电路图。电路设计部分主要包括了，电源电路与 IIC 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。

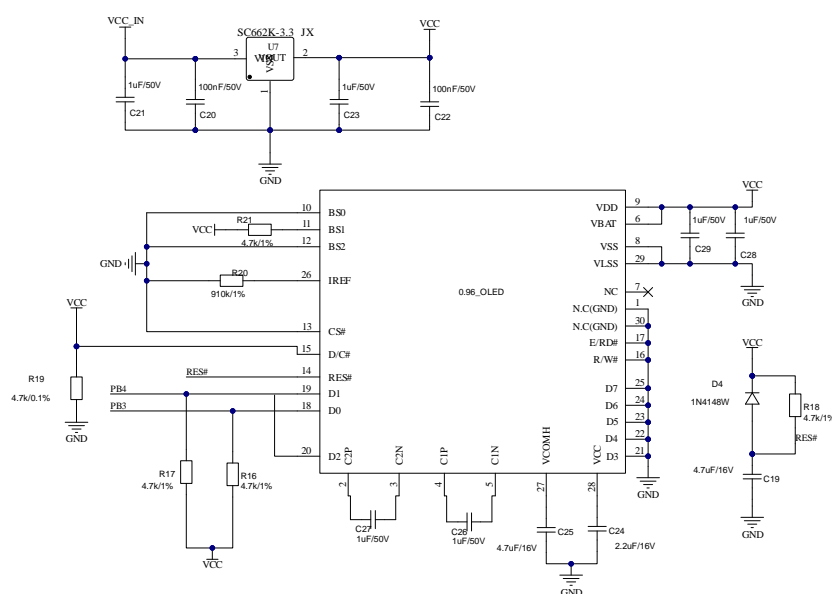


图 3.9 OLED 显示屏电路图

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。这款产品属于 STM32 系列中的 F1 分支，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。其主频为 72MHZ，工作电压 2.0V~3.6V，工作温度 -40℃~+85℃，程序存储器（FLASH）容量 64KByte，数据存储器（SRAM）容量 20KByte，内核外设有 USART，IIC，SPI，USB（从机模式），CAN，ADC，PWM，GPIO 以及通用 Timer 和高级 Timer，还有 ARM CortexM 内核自带的 Timer SysTick，以及 JTAG/SWD 接口。STM32F103C8T6 核心工作电路如下图 4.0。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为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。

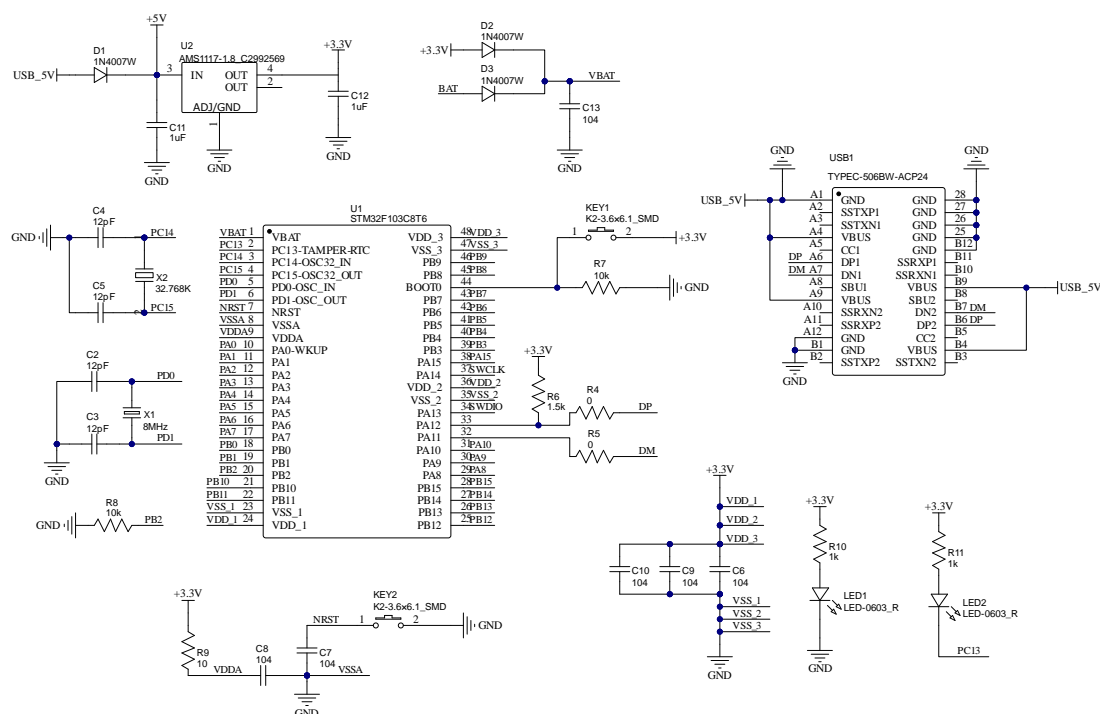


图 4.0 STM32F103C8T6 核心系统电路图

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-LCD1.44 英寸显示屏，额定电压为 3.3V，分辨率为 128*128，SPI 接口，能够显示彩色图像支持 RGB（红、绿、蓝）色彩模式且有非常优秀的低功耗能力。其单路设计图下图 4.1，此电路部分主要包含了电源电路，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。

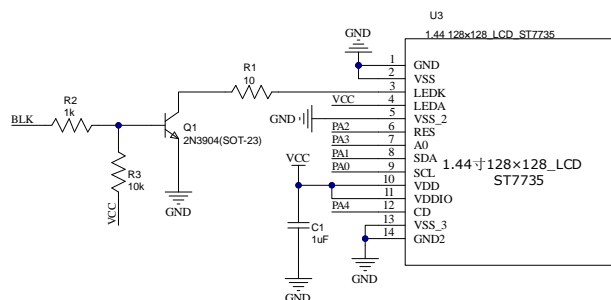


图 4.1 TFT-LCD 工作电路图

3.4.3 PS2 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 4.2，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。

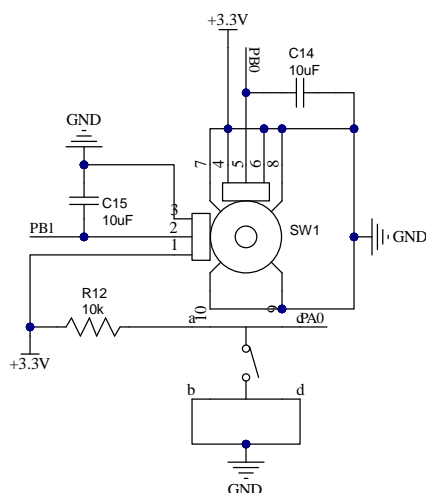


图 4.2 PS2 摇杆工作电路图

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

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，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 4.4，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（D

LC), 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。

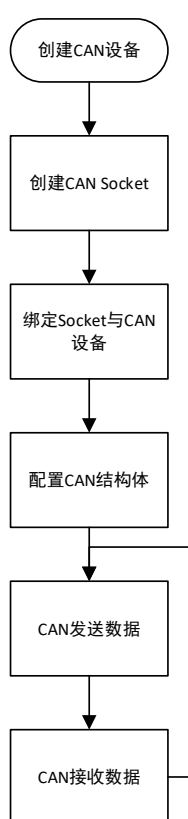


图 4.4 Linux CAN 工作流程图

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 4.5, 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。

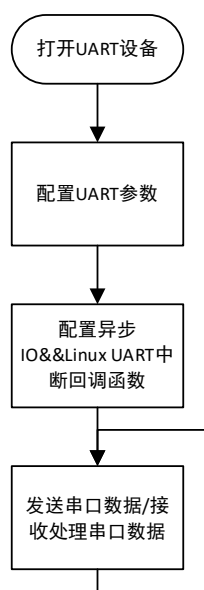


图 4.5 Linux UART 工作流程图

4.2.3 Linux V4L2 for Camera 驱动

Linux V4L2（Video for Linux 2）驱动是 Linux 内核中关于视频设备的子系统，它为 Linux 下的视频驱动提供了统一的接口，使得应用程序可以使用统一的 API 函数操作不同的视频设备，极大地简化了视频系统的开发和维护。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 4.6 所示。

（1）打开 Camera 设备

使用 `open()` 函数打开设备文件，获取文件描述符。

（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。

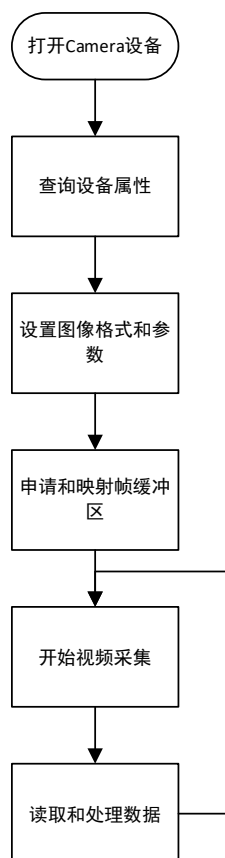


图 4.6 Linux V4L2 For Camera 工作流程图

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 4.7。



图 4.7 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 4.8。

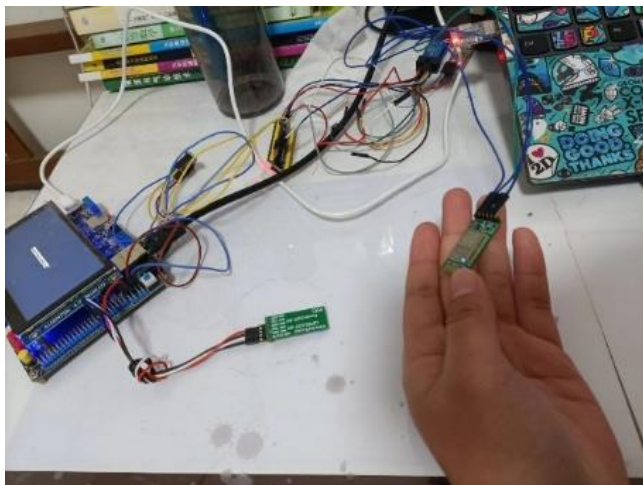


图 4.8 Linux UART 基础通信功能测试图

（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，实测如下图所示 4.9。



图 4.9 Linux V4L2 基础功能测试图

4.2.5 系统性测试

（1）机器人主控系统功能测试

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 5.0。



图 5.0 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 是一个迷你型的实时操作系统内核，具有免费、开源、小巧、可裁剪和可固化到 ROM 等特点。它主要适用于资源受限的嵌入式系统，可以根据应用程序的需求进行裁剪，只包含必要的功能模块，以减小内存占用。FreeRTOS 能够管理数量不受限制的任务，提供多任务管理、实时性、通信和同步、可移植性等重要特性。

在多任务管理方面，FreeRTOS 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堆栈和上下文，可以独立运行。实时性方面，FreeRTOS 提供了实时任务调度和中断处理机制，能够满足实时应用程序的需求，保证关键任务的及时执行。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和协作。

FreeRTOS 的内核代码是高度可移植的，可以在不同的处理器架构和开发平台上运行，例如移植在 STM32 单片机上，下边介绍移植的具体流程。

（1）获取 FreeRTOS 源码和 STM32 库

从 FreeRTOS 官网下载最新版本的 FreeRTOS 源码，获取 STM32F103C8T6 标准外设库。

（2）创建工程

使用 Keil MDK-ARM IDE 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 vTaskStartScheduler()来启动 FreeRTOS 调度器。

（9）创建任务

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

（10）编译和调试

编译工程并下载到 STM32F103C8T6 中，使用调试器结合 Keil IDE 进行调试，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。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 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

CAN 全称为 Controller Area Network，即控制器局域网，是一种应用在现场、在微机化测量设备之间实现双向串行多节点数字通讯系统。它是由德国 Bosch 公司为汽车监测和控制而设计的，并逐步发展成用于其他工业控制领域的现场总线。CAN 已成为国际标准化组织的 ISO 11898 标准。

CAN 协议建立在 ISO/OSI 模型之上，其模型结构有三层。它采用点对点、一点对多点及全局广播 3 种方式发送和接收数据。所有的消息都以固定的格式发送，当两个以上的单元同时开始发送消息时，会根据 ID（标识符）决定优先级，对各消息 ID 的每个位进行逐个仲裁比较。因此，在总线上增加单元时，连接在总线上的其它单元的都不需要改变。

CAN 的特点主要包括：

多主控制：在总线空闲时，所有的单元都可开始发送消息，最先访问总线的单元可获得发送权。多个单元同时开始发送时，发送高优先级 ID 消息的单元可获得发送权。

消息发送：CAN 协议中，所有的消息都以固定的格式发送，消息的 ID 不是地址，而是一种表征优先级的标识符。

远程数据请求：可通过发送“遥控帧”请求其他单元发送数据。当总线上发生持续数据错误时，可将引起此故障的单元从总线上隔离出去。

高速数据传输：CAN 可提供高达 1Mbit/s 的数据传输速率，使得实时控制变得非常容易。

强大的抗电磁干扰能力：硬件的错误检定特性增强了 CAN 的抗电磁干扰能力。

CAN 在车载各电子控制装置 ECU 之间交换信息，形成汽车电子控制网络，如发动机管理系统、变速箱控制器、仪表装备、电子主干系统中都嵌入了 CAN 控制装置。一个由 CAN 总线构成的单一网络中，理论上可以挂接无数个节点，但在实际应用中，节点数目受网络硬件的电气特性所限制。

总之，CAN 是一种具有多主控制、优先级判定、远程数据请求等特性的通信协议，广泛应用于汽车和其他工业控制领域，实现可靠、高效的实时数据传输。

STM32 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

线的的数据。

4.3.4 STM32F103 IIC 驱动传感器

IIC（Inter-Integrated Circuit）实际上是 IICBus 的简称，中文名为集成电路总线。它是一种串行通信协议，也被称为 I2C（Inter-IC）总线。IIC 总线由飞利浦公司在 1980 年代为连接主板、嵌入式系统或手机与低速周边设备而发展。

IIC 总线的工作原理基于两根线：串行数据线(SDA)和串行时钟线(SCL)。SDA 线用于传输数据，而 SCL 线用于传输时钟信号。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。通信过程由主设备控制，主设备可以是微控制器、DSP、FPGA 等。

IIC 总线在多个领域有广泛的应用，如工业自动化、智能家居、医疗设备等。例如，在工业自动化中，IIC 用于连接各种传感器和执行器，实现生产过程的自动化控制。在智能家居系统中，IIC 用于连接智能设备，如智能灯泡和智能插座，实现家庭设备的远程控制和自动化管理。在医疗设备中，IIC 用于连接传感器和执行器，实现远程监控和治疗。

此外，IIC 总线在现代电子领域，特别是在物联网(IoT)和嵌入式系统中，仍然得到广泛应用。其简单性、可靠性以及适用性使得它在连接各种不同类型的设备时具有优势。它使用两根信号线来连接多个设备，大大节省了硬件引脚，并支持多主设备和多从设备的连接。

STM32 中使用标准库开发 IIC 外设的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

PWM 是脉宽调制（Pulse Width Modulation）的缩写，是一种模拟控制方式。它通过调制晶体管基极或 MOS 管栅极的偏置，根据相应载荷的变化来改变晶体管或 MOS 管的导通时间，从而实现对开关稳压电源输出的改变。这种方式能使电源的输出电压在工作条件变化时保持恒定，是利用微处理器的数字信号对模拟电路进行控制的一种非常有效的技术。

PWM 技术被广泛应用于从测量、通信到功率控制与变换的许多领域中。在电机控制中，PWM 被用于驱动电机，通过调整 PWM 信号的占空比来控制电机的转速和扭矩。本系统中就通过 STM32 的 PWM 外设来驱动直流电机。

STM32 的 PWM（脉冲宽度调制）是一种用于控制电子设备的技术，通过调整信号的脉冲宽度和周期，可以模拟出不同的电压或功率级别。

PWM 信号主要由两个关键参数决定：占空比和频率。占空比是指在一个脉冲循环内，通电时间（可能是高电平，也可能是低电平）相对于总时间所占的比例。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电压或功率。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或功率也将为输入电压或功率的一半。

在 STM32 中，PWM 功能是通过定时器驱动的。定时器的周期决定了 PWM 的周期，而为了控制高低电平的比例，会在定时器的基础上加上一个比较寄存器。这个比较寄存器与 IO 口结合，用于输出 PWM 波。STM32 中的 PWM 属于定时器的功能，因此通过直接配置定时器就可以使用 PWM。此外，还需要加入一个比较计数值来确定一个周期内翻转电平的时机，并且需要 GPIO 输出方波，这通常利用 GPIO 的复用功能实现。

STM32 微控制器提供了多个 PWM 通道，每个通道可以配置为不同的输出引脚，并具有灵活的配置选项，如频率、占空比、极性等。这使得开发者能够使用 STM32 的 PWM 功能实现精确的电子设备控制。

总的来说，STM32 的 PWM 功能强大且灵活，能够满足各种电子设备控制的需求。通过合理配置 PWM 参数，可以实现对电机、LED 等设备的精确控制，从而在各种应用中实现高效、稳定的性能。

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 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 5.1 FreeRTOS 运行图

（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 5.2.

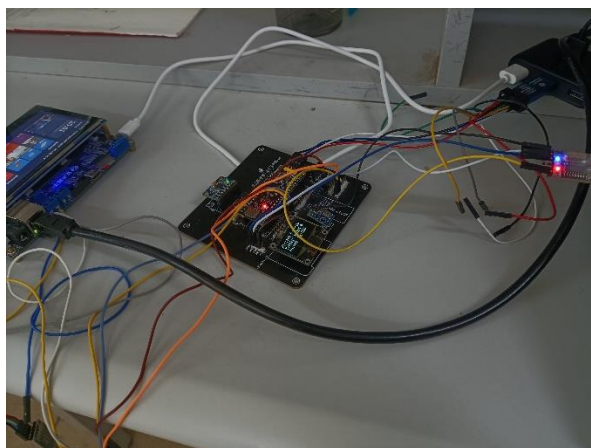


图 5.2 STM32 UART 测试图

（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 5.3，与主控双方各发送 8 个字节的数据。

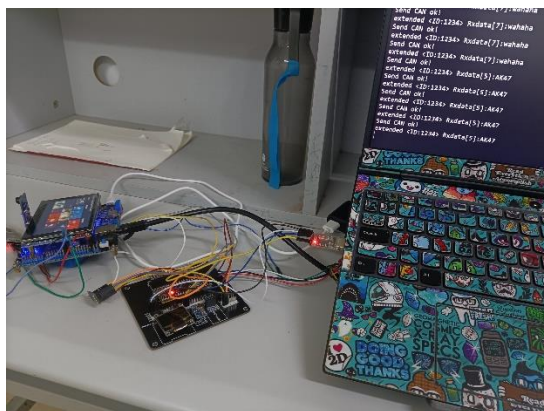


图 5.3 STM32 CAN 测试图

（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 5.4。

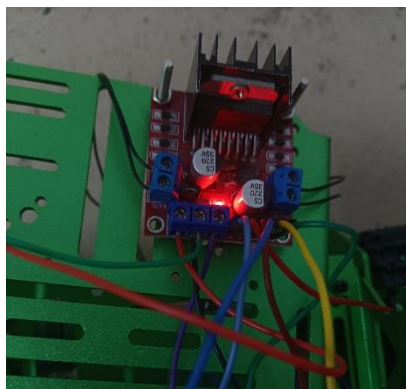


图 5.4 STM32 PWM 驱动电机测试图

（5）STM32 IIC 驱动 OLED 测试

下图 5.5 为 OLED 显示测试图，将读取到的温湿度数据通过 OLED 显示出来。

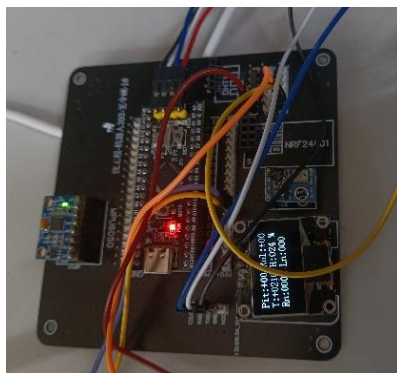


图 5.5 STM32 OLED 测试图

4.3.7 系统性测试

（1）机器人传感数据采集及驱动系统功能测试

如下图 5.6，将此系统中的多个功能结合到一起，共同运行，进行实测。

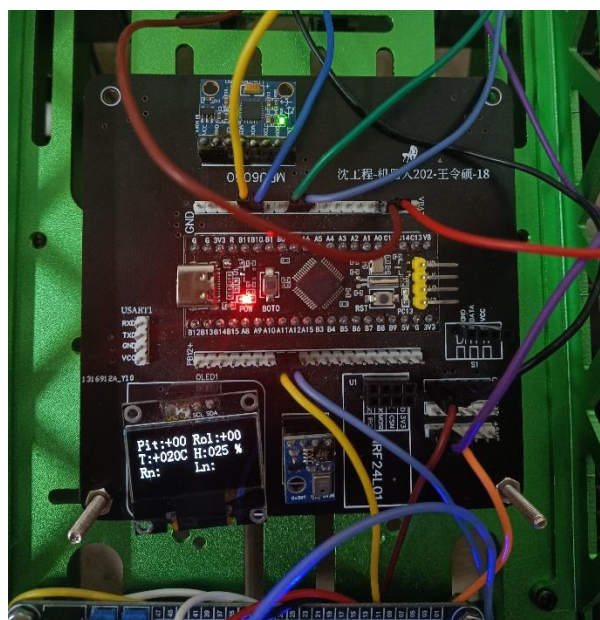


图 5.6 机器人传感及驱动系统测试图

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 PS2 摇杆

STM32 使用标准库开发 ADC（模数转换器）驱动 PS2 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）处理 ADC 通道的数据

读取到的 ADC 通道的数据是不够准确的，还需要进行滤波处理，例如均值滤波，以提高 ADC 数据的准确性和稳定性。

（4）摇杆控制逻辑代码

根据 ADC 通道的数据值与摇杆的动作，来解析摇杆的信息数据。

4.4.3 STM32F103 SPI 驱动 TFT 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 PS2 摇杆功能测试

实测如下图 5.7，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。

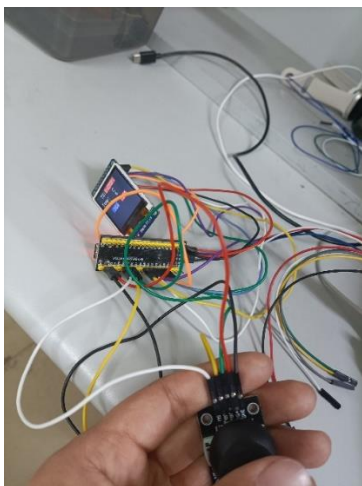


图 5.7 PS2 摇杆测试图

（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 5.8。

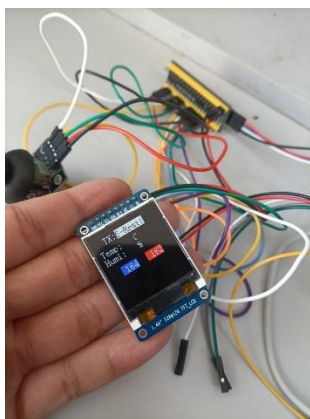


图 5.8 TFT-LCD 显示屏测试图

（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 5.9。

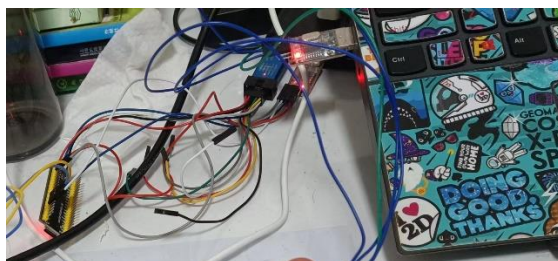


图 5.9 无线控制系统 STM32 UART 测试图

4.4.5 系统性测试

（1）机器人无线控制系统功能测试

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 6.0。

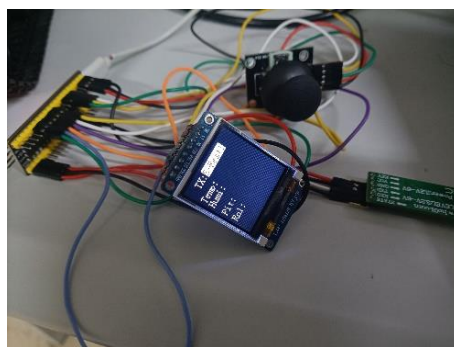


图 6.0 无线控制系统测试图

4.5 机器人系统整体测试

将三个子系统主控系统，传感器数据采集及驱动系统，和无线控制系统，整合到一起，进行系统性的联调测试，实测如下图 6.1。

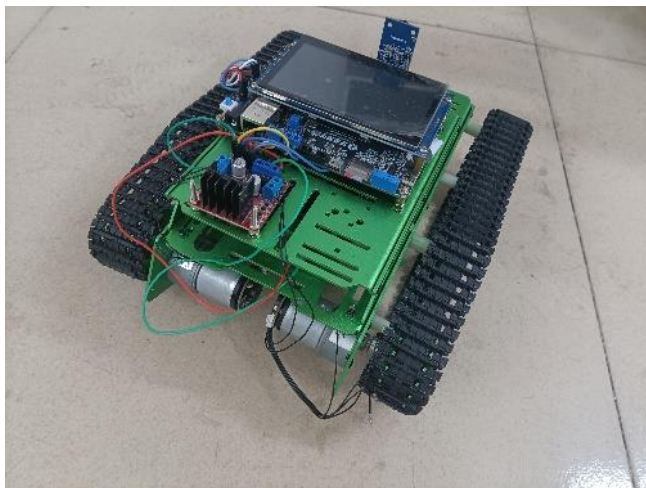


图 6.1 机器人系统整体测试图

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

5.1 总结

本文围绕基于单片机的室内移动机器人的设计与应用展开深入研究，从主控系统、传感器数据采集及驱动系统、无线控制系统等多个方面进行了详细阐述。主要工作内容包含以下几个部分。

（1）主控系统的设计采用了基于 ARM Cortex-A7 内核的 i.MX6ULL 单片机，并成功集成了嵌入式 Linux 操作系统。这一系统不仅实现了与传感器数据采集系统的无缝数据交互，还通过图像传感器增强了机器人的环境感知能力。此外，通过蓝牙模块实现了与无线控制系统的数据通信，提升了机器人的操作便利性和互动性。

（2）传感器数据采集及驱动系统的开发基于 ARM Cortex-M3 内核的 STM32F103C8T6 单片机，并运行实时操作系统 FreeRTOS，确保了对环境温湿度和机器人姿态的精确监测。通过 CAN 接口与主控系统的高效通信，提高了数据采集的准确性和系统的响应速度，从而优化了机器人的整体性能。

（3）无线控制系统同样基于 STM32F103C8T6 单片机和 FreeRTOS 操作系统，通过 ADC 接口接收 PS2 摇杆的输入信号，并将其转换为机器人的运动控制指令。同时，利用 SPI 接口驱动的 TFT 显示屏实时反馈机器人的状态信息，并通过 UART 接口与蓝牙模块实现与主控系统的无线通信，极大地提高了用户对机器人控制的直观性和易用性。

致谢

首先，我要表达我最深切的感激之情给我的毕设导师刘炜老师，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，更在思考方式和解决问题的能力上对我产生了深远的影响。

我还要感谢我的同学们和朋友们，他们在我遇到困难和挑战时给予了我无私的帮助和鼓励。我们共同度过的讨论和学习时光，不仅丰富了我的知识体系，也让我收获了宝贵的友谊和合作经验。

我对于学校提供的资源和设施也怀有深深的感激。特别是图书馆的工作人员，他们的专业和热情为我的研究提供了极大的便利。在文献查找和资料整理方面，他们的帮助让我能够更高效地进行学术研究。

此外，我要向我的家人表达最诚挚的感谢。他们的理解、支持和鼓励是我在学术道路上不断前进的动力源泉。在我忙于研究和写作的日子里，家人的关爱和鼓励是我最大的精神支柱。

最后，我要感谢所有在我学习旅程中给予我帮助的人。每一次的交流和讨论都让我受益匪浅，也让我的研究更加深入和全面。感谢你们的宝贵意见和真诚支持，没有你们，我的毕业设计不会如此顺利完成。这份致谢，是我对你们所有人的感激之情的真挚表达。

参考文献

- [1] 张明路, 丁承君, 段萍. 移动机器人的研究现状与趋势. 河北工业大学学报, 2004.3 2(2):110~115 孟祥莲, 孙平, 高洪志. 单片机原理及应用[M]. 人民邮电出版社:, 2015 9.290.
- [2] 侯志强, 韩崇昭. 视觉跟踪技术综述. 自动化学报, 2006, 32(4):603-617 韩京清. 自抗扰控制技术[J]. 前沿科学, 2007(01):24-31.
- [3] 吴转峰, 刘卫国, 付丽, 朗宝华. 微型红外追踪及无线遥控车设计. 微特电机, 2006. 5 田彦涛, 孙中波, 李宏扬等. 动态双足机器人的控制与优化研究进展[J]. 自动化学报, 2016, 42(08):1142-1157.
- [4] Song S, Kidziński Ł, Peng X B, et al. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation[J]. Journal of neuroengineering and rehabilitation, 2021, 18(1): 1-17
- [5] 王志文, 郭戈. 移动机器人导航技术现状与展望. 机器人, 2003.25(5):470~474
- [6] Borenstein. J , Everett H R , Feng L. Navigating mobile robots: A Survey. systems and techniques. Wellesley , Mass :AK Peters , 1996. 130~217
- [7] Borenstein. J. Control and kinematics design of multi-degree-of freedom mobile robots with compliant linkage: A Survey. IEEE Trans. Robotics and Automation, 1995, 11(1):21~35.
- [8] 郑敏, 赵玉芹, 田红伟. 基于多传感器数据融合的机动目标跟踪算法研究. 探测与控制学报, 2006.28(5):43~45
- [9] 蔡自兴. 智能控制及移动机器人研究进展. 中南大学学报(自然科学版), 2005.36(5): 721~726
- [10] Khabib. O. Real-time obstacle avoidance for manipulators and mobile robots: The International Journal of Robotics Research, 1986, 5(1):1258~1269
- [11] Liu Q, Ren T, Chen Y. Characteristic analysis of a novel in-pipe driving robot[J]. Mechatronics, 2013, 23(4):419-428.
- [12] 常玉连, 邵守君, 高胜. 石油工业中管道机器人技术的发展与应用前景[J]. 石油机械, 2006, 34(9):122-126.
- [13] 刘清友, 李维国. Sondex 水平井井下爬行工具介绍[J]. 国外测井技术, 2008(5):57-59.
- [14] 程义. 城市地下管道检测机器人设计[D]. 湖北工业大学, 2018.

- [15] 夏文涵, 王凯, 李彦, 等. 基于 TRIZ 的管道机器人自适应检测模块创新设计[J]. 机械工程学报, 2016, 52(5): 58-67.
- [16] 陈松, 李天剑, 王会香, 等. 排水管道机器人综述[J]. 机器人技术与应用, 2014(1): 23-27.