

国家卓越工程师学院明月科创实验班

《微电路设计》课程结项报告



组 号： 第 3 组

姓 名： 郝博文，赵江涛，唐晨

学 号： 20234021，20232662，20235025

年 级： 23 级

班 级： 2 班

日 期： 2026/1/17

重庆大学国家卓越工程师学院 2025 年

目录

1 项目概述	4
1.1 项目背景	4
1.2 项目目标	4
1.3 项目分工与时间规划	5
1.4 扫地机器人方案设计	5
2 硬件选型与电路设计	7
2.1 核心器件选型	7
2.1.1 主控：STM32F407ZET6	7
2.1.2 电源：SCT2450CSTER + AMS1117-3.3 + SLM6900	7
2.1.3 IMU：JY901B 十轴陀螺仪	9
2.1.4 无线通信：DT-06 WiFi 模块	10
2.1.5 上位机：树莓派 5 与 N10 雷达	10
2.2 原理图设计	11
2.2.1 核心板	11
2.2.2 扩展板	12
3 PCB 设计与实现	15
3.1 核心板 PCB 设计	15
3.2 扩展板 PCB 设计	16
3.3 实物展示	19
3.3.1 核心板	19
3.3.2 扩展板	19
3.3.3 整体	20
3.3.4 纹波测试	20
4 嵌入式软件设计与实现	20
4.1 控制方案与代码架构	21
4.1.1 BSP 层（板级支持包）	21
4.1.2 MODULE 层（模块层）	22
4.1.3 APPLICATION 层（应用层）	22
4.1.4 系统通信与调试	23
4.2 运动模型与惯性导航	23
4.2.1 简单运动模型与底盘控制	23
4.2.2 惯性导航与传感器融合	25
4.2.3 运动模型与 PID 控制的协同工作	25
4.2.4 运动模型与惯性导航的总结	26
5 树莓派运动计算	26
5.1 建图和定点导航功能	26
5.1.1 建图功能	27
5.1.2 定点导航与路径规划	27
5.1.3 局部规划器概述	28
5.2 全局清扫和沿墙清扫	29
5.2.1 从定点导航到全局清扫功能的实现	30
5.2.2 从定点导航到沿墙清扫功能的实现	31
6 APP 端设计与通信协议	31

6.1 页面设计与交互功能	32
6.1.1 页面模块设计	33
6.1.2 核心设计特点	34
6.2 技术原理与协议设计	35
6.2.1 开发工具系统架构	35
6.2.2 通信协议设计	36
6.3 实现流程	36
7 整机集成与功能展示	39
7.1 整机效果	39
7.2 基础功能	40
7.2.1 按键/摇杆操纵移动	40
7.2.2 三级清洁力度	41
7.2.3 沿墙清扫	42
7.2.4 碰撞检测	43
7.2.5 防跌落检测	44
7.3 拓展功能	44
7.3.1 语音控制	44
7.3.2 快捷按键与自定义功能	45
7.3.3 自动全屋清扫	46
7.3.4 蜂鸣器奏乐	47
8 总结与改进方向	48
8.1 项目总结	48
8.2 后续改进建议	48
附:	50
实物多视图:	50
雷达支架建模:	50
b 站链接:	50

《微电路设计》结项报告

1 项目概述

1.1 项目背景

随着智能家居产业的高速发展与居民生活品质的提升，家庭清洁场景的智能化、高效化需求日益凸显。扫地机器人作为解放双手的核心智能家居产品，市场渗透率持续攀升，但用户对其清洁效率、路径规划精度、场景适应性的要求也愈发严苛。传统扫地机器人普遍依赖单一传感器或随机碰撞式导航方案，在复杂家庭环境中易出现漏扫、重复清扫、路径混乱等问题，难以满足精细化清洁需求。

在此背景下，本项目聚焦激光雷达建图与自主导航、精准手动遥控等核心功能，研发一款兼具智能化自主清洁与人性化手动操控的扫地机器人。通过激光雷达 SLAM（同步定位与地图构建）技术实现家庭环境的高精度地图绘制与最优路径规划，结合手动遥控模式适配复杂边角清洁、特定区域定点清扫等个性化需求，旨在解决传统扫地机器人的技术痛点，推动家庭清洁设备向“自主智能 + 人工干预”的双模式方向升级，为用户提供更高效、灵活的清洁体验。

1.2 项目目标



图 1.2.1 海尔 TL301 宣传图



图 1.2.2 海尔 TL301 清洁模式

项目基于海尔 TL30U1 扫地机器人进行重制与升级。海尔 TL30U1 是一款中低端家用扫拖洗一体机型，主打 7.6cm 超薄机身、多种清扫模式与自动回充功能，配备 2600-5500Pa 多档无刷电机吸力、旋转加压擦地系统等，可实现分区清扫、断点续扫等功能。然而，标准版红外避障对细小障碍物识别较弱，对于复杂环境的适应力略显不足，无法实现地图感知与路径规划等功能。

在此基础上，我们对扫地机器人进行了从内而外的整体改造，包括硬件 PCB 重制、传感器选型优化、嵌入式程序改进、添加上位机导航、小程序开发等，不仅能够更好地实现扫地机器人具备的原生功能，同时包含了地图感知与路径规划等进阶模式。

1.3 项目分工与时间规划

项目的有序展开离不开清晰明确的分工与整体规划，小组（第3组）的分工情况如下：

- ✧ 郝博文：硬件选型与方案设计、原理图绘制与 PCB 设计
- ✧ 赵江涛：嵌入式开发、扫图与导航开发
- ✧ 唐晨：小程序（APP）开发、焊接与测试

项目的整体时间规划与任务安排如下：



图 1.3.1 时间规划与任务安排

1.4 扫地机器人方案设计

本项目面向“自动导航扫地机器人”的完整系统实现，覆盖硬件方案规划、核心板/扩展板电路设计与 PCB 实现、嵌入式控制、上位机（树莓派）与导航建

图、以及 APP 端控制界面与通信协议等环节。

项目总体思路采用模块化拓扑：以核心板为主，扩展板为辅，将电源/驱动/传感器/通信等功能拆分模块，以提高容错性并便于维护与调试。

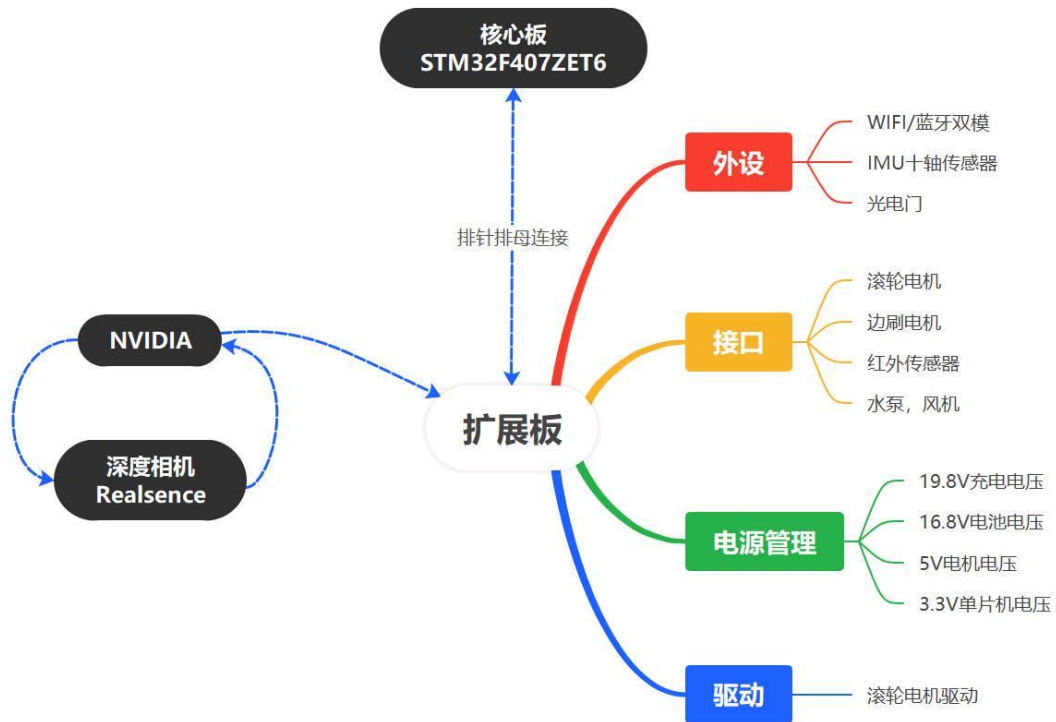


图 1.4.1 系统方案

系统以 STM32 控制板为下位机核心，外接关键模块：

- 运动控制与驱动（电机驱动等）
- 传感器接口（红外避障、光电门等）
- 电源管理模块（针对不同硬件的电源区分）
- 姿态与状态感知（IMU 传感器）
- 无线通信（WiFi 透传芯片）
- 上位机/导航建图（树莓派与 N10 雷达）

并通过 APP 实现手动遥控与参数调节等功能，最终实现“可遥控、可建图、可自动清扫”的整机效果。

2 硬件选型与电路设计

2.1 核心器件选型

本项目关键元器件的选择与选型理由如下：

2.1.1 主控：STM32F407ZET6

STM32F407ZET6 是意法半导体推出的一款高性能 32 位微控制器，基于 ARM Cortex-M4 内核且搭载硬件 FPU 浮点运算单元，主频最高 168MHz，拥有 512KB 闪存、192KB 运行内存，采用 144 引脚封装，外设资源丰富，集成多路串口、SPI、I2C、CAN 总线、以太网、ADC/DAC 及丰富定时器，算力强、性价比高，十分适用于扫地机器人项目的多外设接入与实时控制需求。

STM32F407ZET6 技术参数、封装参数	
类型	描述
安装方式	Surface Mount
频率	168 MHz
引脚数	144 Pin
电源电压	1.80V (min)
工作电压	1.8V ~ 3.6V
封装	LQFP-144
针脚数	144 Position
时钟频率	168 MHz
RAM大小	192 KB
位数	32 Bit
FLASH内存容量	512 KB
ADC数量	3 ADC
输入/输出数	140 Input
工作温度(Max)	85 °C
工作温度(Min)	-40 °C
耗散功率(Max)	500 mW
数模转换数	1 DAC
电源电压	1.8V ~ 3.6V
电源电压(Max)	3.6 V
电源电压(Min)	1.8 V

图 2.1.1.1 STM32F407ZET6 参数

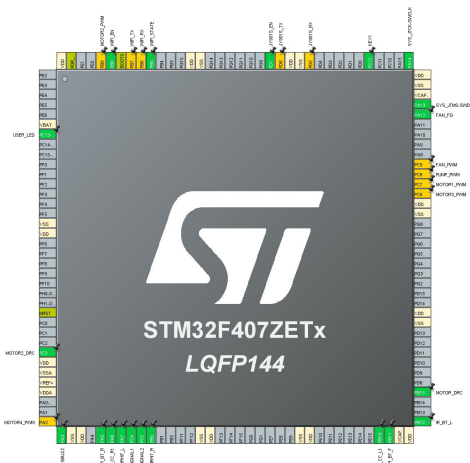


图 2.1.1.2 主控引脚配置

2.1.2 电源：SCT2450CSTER + AMS1117-3.3 + SLM6900

SCT2450CSTER: 16.8V→5V 降压芯片

SCT2450CSTER 是芯洲科技(Silicon Content Technology)推出的一款高性能 5A 同步降压 DC-DC 转换器，采用 ESOP-8L-EP 封装，专为宽电压输入场景设计，具有高效率、高集成度和完善的保护机制，是工业控制、汽车电子和嵌入式系统中电源管理的理想选择。

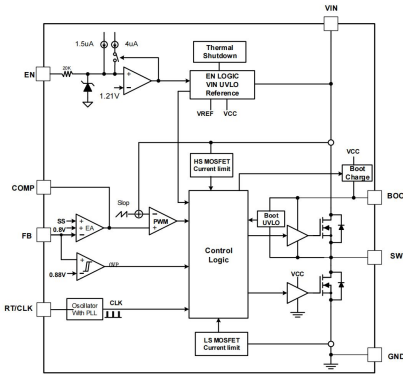


图 2.1.2.1 SCT2450CSTER 功能框图

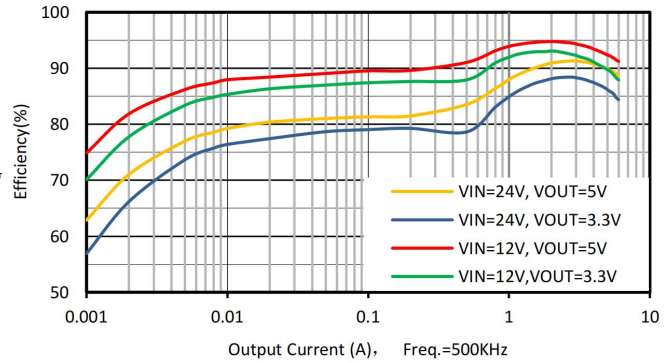


图 2.1.2.2 开关频率 500KHz 下输出效率

SCT2450CSTER 支持宽达 4.5V~50V 的输入范围，适配多种电源系统，采用全集成设计，无需外部 MOSFET，节省 PCB 空间，降低 BOM 成本。其内置的 4ms 软启动功能能够防止启动时的涌入电流，保护系统安全。具备外部环路补偿功能，提供优化环路稳定性或环路响应的灵活性。此外，芯片具有循环限流保护、热关机保护、输出过压/欠压保护，保障了芯片与电路的安全。

AMS1117-3.3: 5V→3.3V 降压芯片

AMS1117-3.3 是 Advanced Monolithic Systems (AMS) 推出的一款三端固定输出 3.3V 低压差线性稳压器(LDO)，以其低成本、外围电路简单、稳定性高等特点，成为嵌入式系统、智能设备与机器人开发中 3.3V 供电的主流选型，广泛用于 STM32 等微控制器及各类传感器、模块的供电。

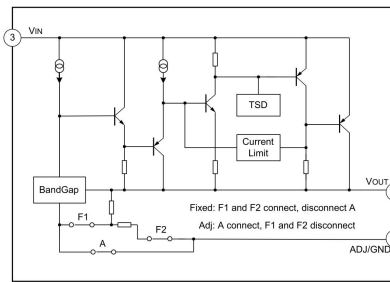


图 2.1.2.3 AMS1117-3.3 功能框图

Characteristics	Symbol	Value	Units
Input Supply Voltage	V_{IN}	18	V
Lead Temperature (Soldering, 5 seconds)	T_{Lead}	260	$^{\circ}C$
Junction Temperature Range	T_J	150	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 to 150	$^{\circ}C$
Power Dissipation	P_D	Internally Limited (Note1)	mW
Thermal Resistance from Junction to Ambient	SOT-223	120	$^{\circ}C/W$
	TO-252	100	$^{\circ}C/W$
	SOT-89	165	$^{\circ}C/W$
ESD Tolerance (Minimum)	ESD	2000	V

图 2.1.2.4 最大额定值表

AMS1117-3.3 外围设计极简，仅需两个滤波电容（输入 10 μ F、输出 22 μ F）即可稳定工作，大幅简化电路设计，降低 BOM 成本与 PCB 面积占用。其输出噪声低，纹波抑制比高，适合对电源质量敏感的数字电路与模拟电路供电（如扫地机器人中的 IMU 模块），良好的负载调整率小于 1.0%，电压线性度好。芯片内置过流保护与热关断功能，异常情况下自动保护芯片与负载。

SLM6900: 16.8V 锂电池充电芯片

SLM6900 是无锡松朗微电子(SLM Microelectronics)推出的一款高性能多节锂电池/磷酸铁锂电池开关充电管理 IC, 采用同步降压拓扑结构, 预置 3 节(12.6V) 和 4 节(16.8V)锂电池充电模式, 同时支持通过外围分压电阻灵活调节输出电压, 是扫地机器人、工业设备、无人机等多节电池供电系统的理想充电解决方案。

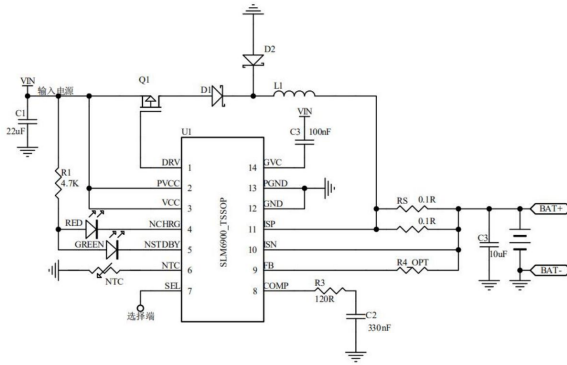


图 2.1.2.5 SLM6900 典型充电电路

符号	参数	条件	最小值	典型值	最大值	单位
V_{IN}	输入电源电压		6.8		28	V
I_{CC}	输入电源电流	待机模式 (充电终止)		0.75	1.5	mA
		停机模式 ($V_{IN} < V_{BAT}$ 或 $V_{IN} < V_{UV}$, 或 NTC 接地)			35	μ A
V_{FLOAT}	稳定输出 (浮充) 电压	SEL 接地	12.47	12.60	12.73	V
V_{FB}	SEL 接高, $V_{IN} = 20V$	SEL 接高, $V_{IN} = 20V$	16.63	16.80	16.97	V
		SEL 悬空	1.193	1.205	1.217	V
I_{BAT}	BAT 脚漏电流	$V_{BAT} = V_{FLOAT} + 0.2V$		10	15	μ A
I_{CHRG}	快充充电电流	待机模式 ($V_{IN} < V_{BAT}$ 或 $V_{IN} < V_{UV}$ 或 NTC 接地)			15	μ A
		$V_{BAT} > V_{TRKLL}$, $R_S = 0.05\Omega$, 电流模式	2.2	2.4	2.6	A
I_{TRKLL}	涓流充电电流	$V_{BAT} < V_{TRKLL}$, $R_S = 0.05\Omega$, 电流模式		550		mA
V_{TRKLL}	涓流充电门限电压	V_{BAT} 上升 (SEL 接地)		8.4		V
		V_{BAT} 上升, $V_{IN} = 20V$ (SEL 接高)		11.2		V
		V_{BAT} 上升 (SEL 悬空)		66		%VFB

图 2.1.2.6 芯片直流电特性

SLM6900 预置了 4 节(16.8V)锂电池充电模式, 直接适配扫地机器人的锂电池电源, 无需复杂配置即可直接使用。其采用同步降压转换器, 相比线性充电效率提升 30% 以上, 大电流充电时发热显著降低。充电软启动功能, 限制充电初期的浪涌电流, 具有循环限流保护和自动再充电功能, 电池电压下降时自动恢复充电, 保障了电源与电池安全。

2.1.3 IMU: JY901B 十轴陀螺仪



图 2.1.3.1 JY901B 产品图片

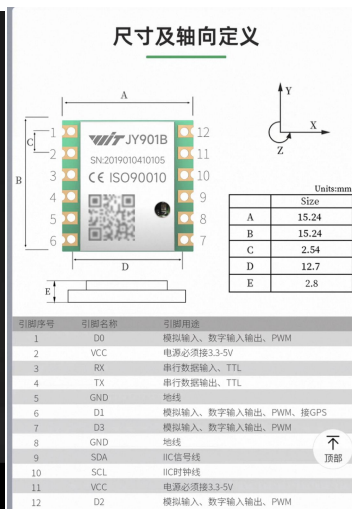


图 2.1.3.2 IMU 引脚配置

JY901B 是维特智能(WitMotion)推出的一款高性能十轴惯性测量单元(IMU),集成三轴陀螺仪 + 三轴加速度计 + 三轴磁力计 + 气压高度计,并内置高性能微处理器与动态卡尔曼滤波算法,可直接输出稳定的姿态数据(欧拉角/四元数),是机器人、无人机、工业控制等领域姿态感知与运动控制的理想选择。

2.1.4 无线通信: DT-06 WiFi 模块

DT-06 是深圳大工智能(SZdoit)基于 ESP-M2 (ESP8285)芯片开发的一款 TTL 串口转 WiFi 透传模块,采用与 HC-06 蓝牙模块完全兼容的物理接口与电气特性,可无缝替代蓝牙模块实现设备的 WiFi 联网升级,内置工业级透传固件与 WEB 配置界面,是物联网、智能设备远程监控的理想选择。



DT06无线WiFi透传模块
TTL转WiFi兼容蓝牙
AT固件/透传固件

支持AP、STA、AP+STA配置方式,支持的串口波特率从300bps~3686400bps,支持TCP Server/Client, UDP Server/Client, UDP局域网广播等多种功能。同时兼容AT指令配置,是目前很具有性价比的WiFi透传产品。

- 完全兼容原有的蓝牙透传模块电气接口和物理接口;
- 内置工业级透传固件V3.0,可靠稳定,WEB界面配置;
- 基于ESP-M2高性能WiFi模块,整体尺寸:34mm×17mm×4mm;
- 供电电压:4.5V~6.0V,TTL电压:3.3V(可兼容5.0V);
- 引出管脚:STATE, TXD, RXD, EN;
- 平均电流:80mA;WiFi数据发送时170mA;深度睡眠模式下20μA;
- 支持串口AT指令;
- 支持可再编程,OTA固件升级;
- 串口与WiFi数据实时无缝透传;
- 内置HTTP Web Server,支持网页配置各项参数;
- WiFi支持AP、STA、AP+STA三种模式;
- 支持WiFi STA模式下自动重连,TCP Client模式下自动重连;
- 串口支持设置波特率、数据位、奇偶校验、停止位、分包时间;
-等等

AT固件 透传固件

免费开普票

DT06无线模块 TTL转WiFi兼容蓝牙

图 2.1.4.1 DT-06 产品图片

图 2.1.4.2 DT-06 功能介绍

其兼容性强、配置简便、性能稳定,特别适合 APP 与扫地机器人下位机通信连路的搭建,实现设备的智能化与网络化,大幅降低开发难度与周期。

2.1.5 上位机: 树莓派 5 与 N10 雷达

树莓派 5:

树莓派 5 是树莓派基金会 2023 年 9 月发布、10 月底推出的新一代旗舰单板计算机,搭载博通 2.4GHz 四核 64 位 Arm Cortex-A76 处理器与 VideoCore VII GPU, CPU 和 GPU 性能较前代提升 2-3 倍,配备

2GB/4GB/8GB/16GB LPDDR4X 内存，首次集成自研 RP1 I/O 芯片与神经网络处理单元 (NPU) ，新增 PCIe 2.0 接口 (支持 NVMe SSD 扩展)、实时时钟和物理电源按钮，拥有双 4Kp60 HDMI 输出、双频 Wi-Fi、蓝牙 5.0 及兼容前代的 40 针 GPIO，兼具低功耗与强散热设计，非常适配扫地机器人的建图与导航功能开发，是性价比与扩展性兼具的嵌入式计算平台。



图 2.1.5.1 树莓派 5 产品图片

图 2.1.5.2 N10 雷达产品图片

N10 雷达:

镭神智能 N10 是一款主打高性价比的紧凑型单线 2D 激光雷达，采用 TOF 飞行时间测距技术与 905nm 人眼安全激光，可实现 0~12m 的精准测距，支持 360° 全方位扫描，具备 4500 点/秒的扫描速率与 6~12Hz 可调的扫描频率，角度分辨率在 0.48° ~0.96° 间自适应切换，其体积仅 $\phi 52 \times 36.1\text{mm}$ 、重量 60g，搭配串口及可选网口转接模块，拥有 IP65 防尘防水等级，内置优化算法可降低机器人 SLAM 计算负担，轻量易集成的特性使其广泛适用于移动机器人导航避障、室内外 SLAM 建图等场景，是扫地机器人环境感知开发的高性价比选择。

2.2 原理图设计

2.2.1 核心板

原理图层面延续模块化原则，核心板原理图的设计主要强调信号质量保障。使用四个 2*15 排针作为与扩展板的连接，可拆卸式设计便于单独焊接、测试、

烧录等。

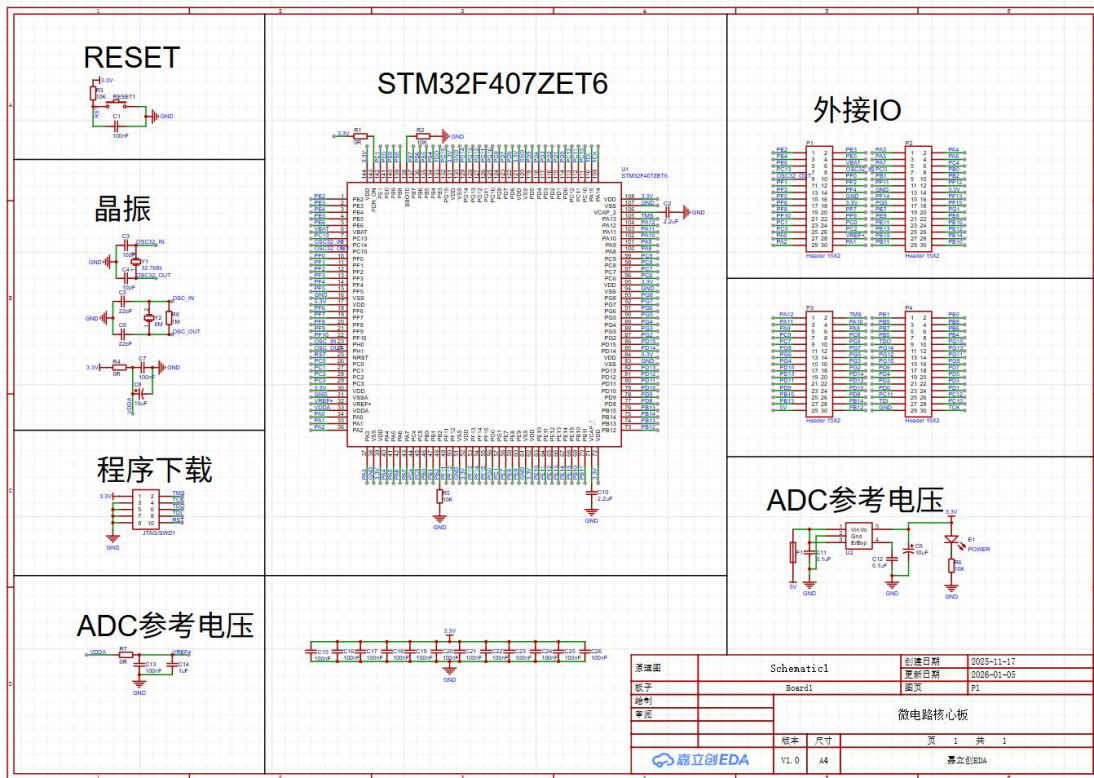


图 2.2.1.1 核心板原理图

核心板支持 STlink/Jlink 烧录，外接 32kHz 与 8MHz 高速晶振，可最大化 STM32F407ZET6 芯片性能。

核心板设计 ADC 参考电压，目的是保证 ADC 模数转换的精度、稳定性与一致性，从原理上保障测量的精度、稳定性与一致性。

主控芯片的 3V3 电源引脚旁设计大小电容组合，核心目的是全频段滤除电源噪声、稳定供电电压，为芯片内部的 CPU、外设、模拟电路提供纯净的电源，避免因电压波动或噪声干扰导致芯片工作异常、死机或性能下降。其中，大小电容的作用互补且不可替代，分别针对电源中的低频纹波和高频噪声，形成“高低频全覆盖”的滤波方案。

2.2.2 扩展板

扩展板原理图同样采用模块化设计（如图 2.2.2.1 左），通过四个 2*15 排母与核心板相连，可拆卸式设计便于单独焊接、测试等，并避免充电电路、驱动电路等噪声源对核心板造成影响。

扩展板按功能划分为：最小系统、电机驱动、电源管理、外设、接口及定位孔等模块，以降低耦合度并提升可维护性。

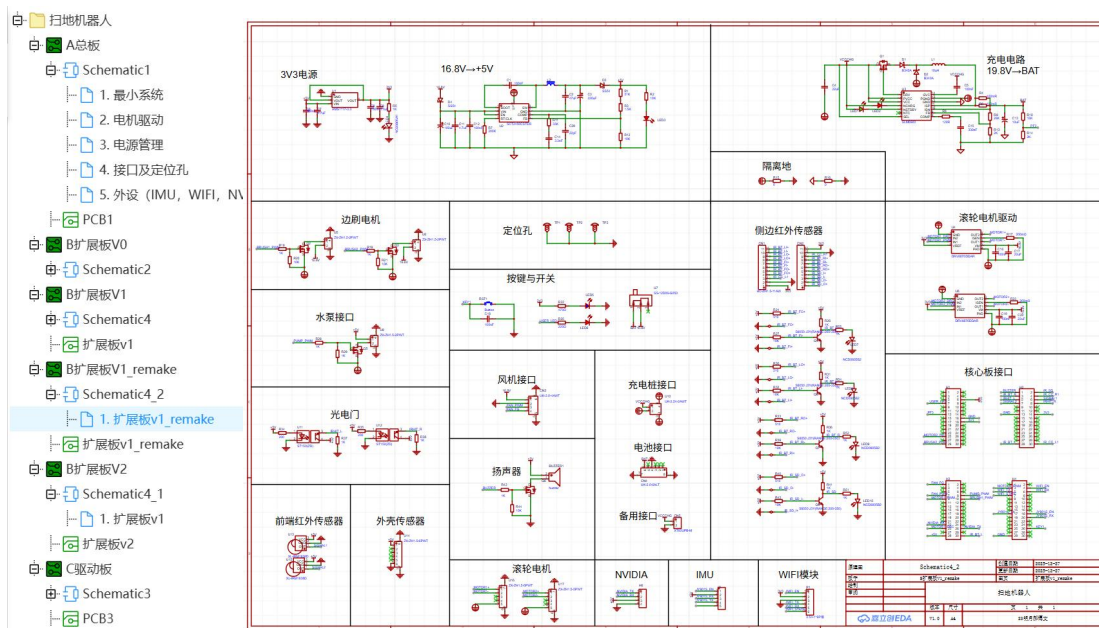


图 2.2.2.1 扩展板原理图

16.8V→5V 降压电路：

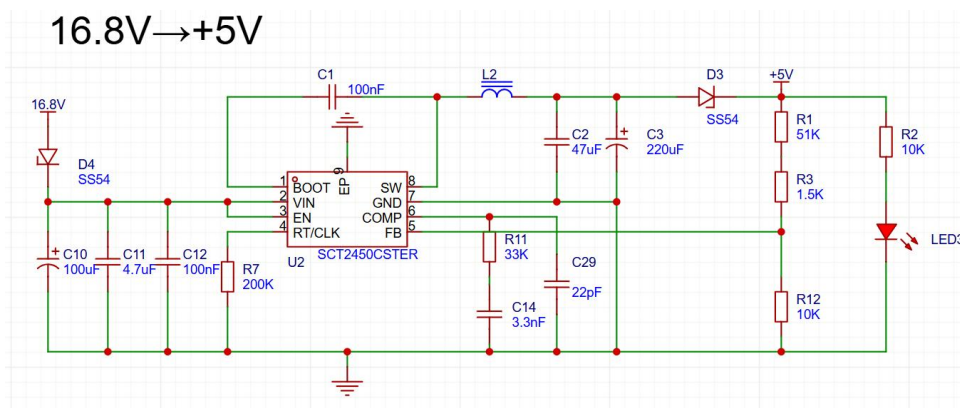


图 2.2.2.2 16.8V 降压电路拓扑

D4 (SS54) 串联在 16.8V 输入端，是典型的防反接/防倒灌做法，用于保护电路。C10、C11、C12 作为输入端的低/中/高频去耦组合，用于滤除电源端产生的高频噪声和低频纹波。BOOT - SW 引脚之间的 C1 作为自举电容，为芯片内部的高端功率 MOS 管提供足够的栅极驱动电压，保证 MOS 管可靠导通与关断。SW、L2、输出电容 C2/C3 共同构成标准 Buck 电路 LC 输出滤波。COMP 补偿网络取值参考芯片手册。FB 输出分压计算公式如下：

$$V_{out} = 0.8V \times \left(1 + \frac{52.5k}{10k}\right) = 0.8V \times 6.25 = 5V$$

5V→3V3 降压电路:

3V3电源

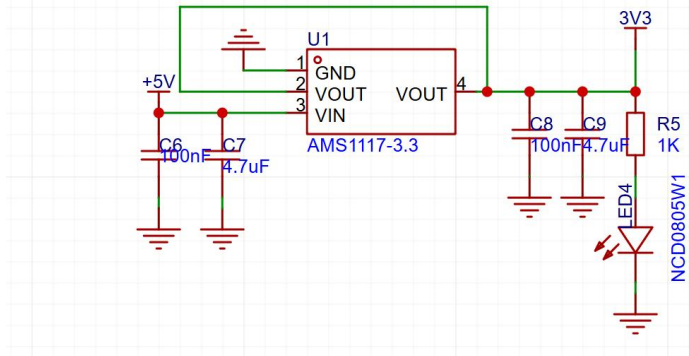


图 2.2.2.3 5V 降压电路拓扑

C6 和 C7 作为输入端和输出端的去耦电容，分别用于去除输入端的高频噪声和稳定输入电压。C8 和 C9 是输出端的去耦电容，用于确保稳压器输出的电压平稳且低噪声。此外，LED 指示灯用于指示电源的正常输出。

充电电路:

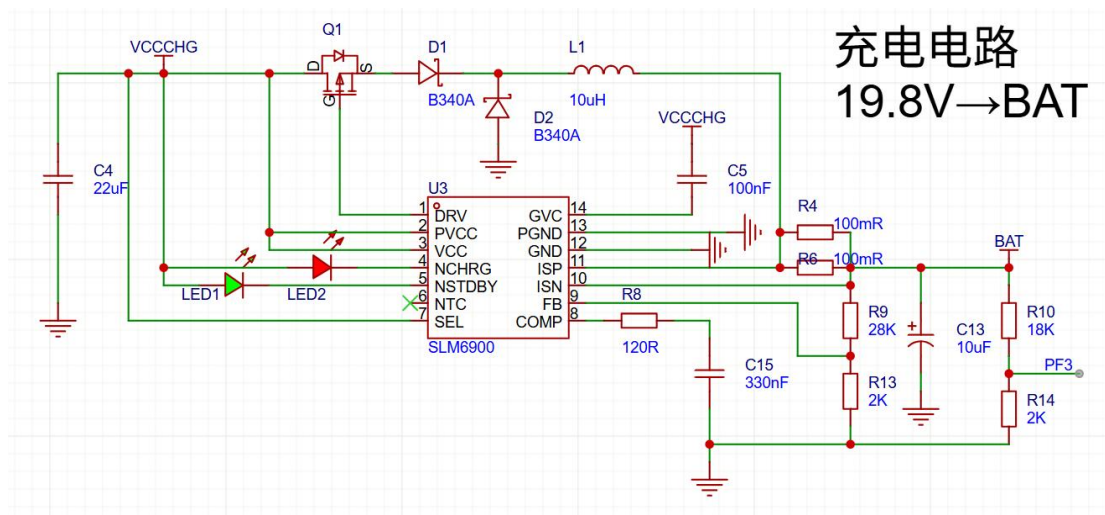


图 2.2.2.4 充电电路拓扑

锂电池充电电路使用 SLM6900 充电管理芯片。D1、D2 两个二极管用于防止反向电流流入电源部分，保护电路不受损坏。L1 电感用于储存能量，平滑充电

过程中的电流波动。电感器与芯片内部的开关配合工作，有助于实现高效的电压转换。C4、C5 作为去耦电容，用于滤波和稳定电压，减少噪声和电流波动。R4、R6 两组电阻用于监测充电电流，可以帮助进行电流检测，以便保护电池并进行适当的电流调节。

3 PCB 设计与实现

3.1 核心板 PCB 设计

核心板采用对称设计的双层板，尺寸为 2220mil x 2175mil，结构小巧紧凑。主控芯片放置在正面的最中央，便于向四周排针处走线。去耦电容放置在靠近电源引脚的位置，滤波效果更好。在布线密度上采用 6/8mil 密集布线策略，走线的宽度和距离设计符合“3W”原则；PCB 四周进行圆角处理，充分体现安全性和人性化角度的考量。整体走线呈现发散式，符合主控芯片走线原则。

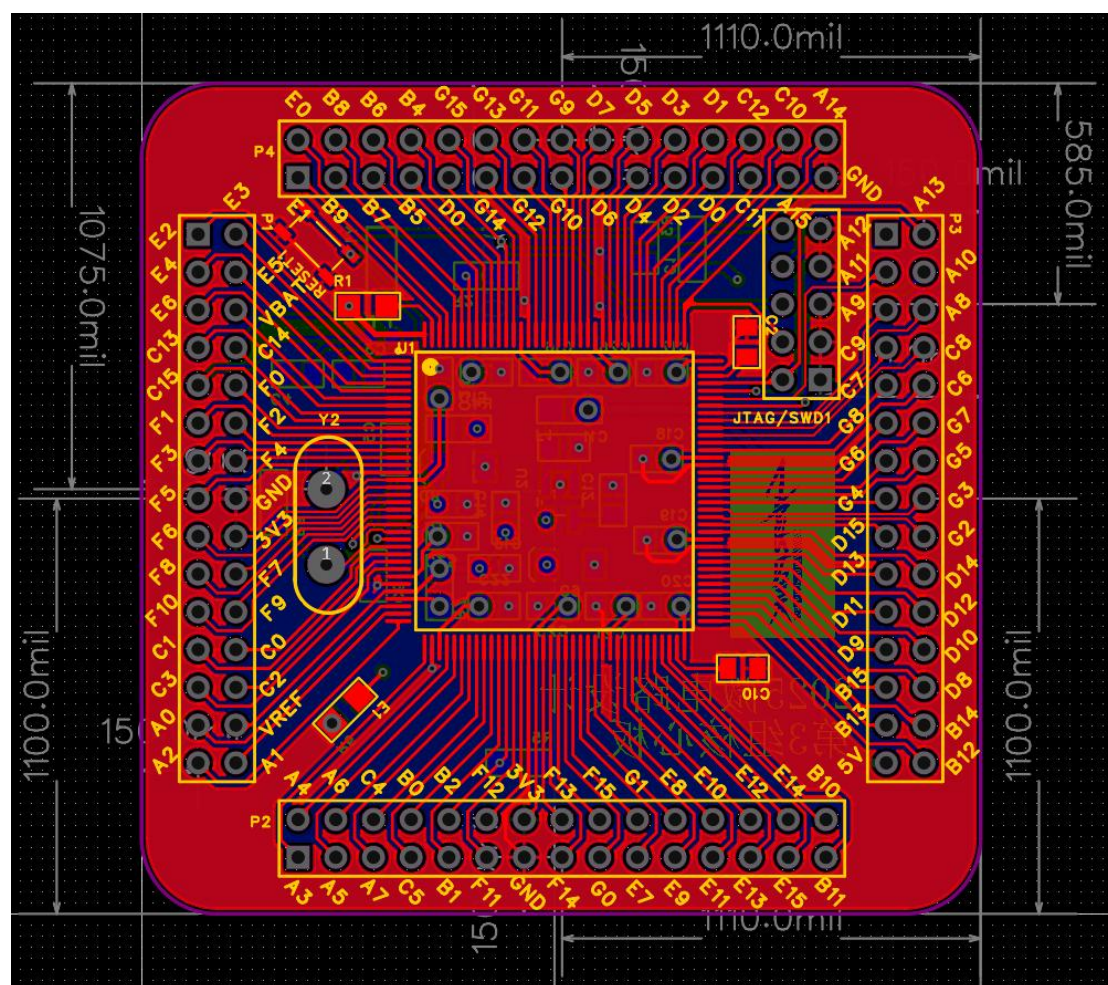


图 3.1.1 核心板 PCB 正面布局

PCB 背面全部放置贴片元件，提高集成度与装配一致性，便于焊接与测试。添加了课程、小组信息等文字标注与小组 logo，提高 PCB 辨识度与独特性。

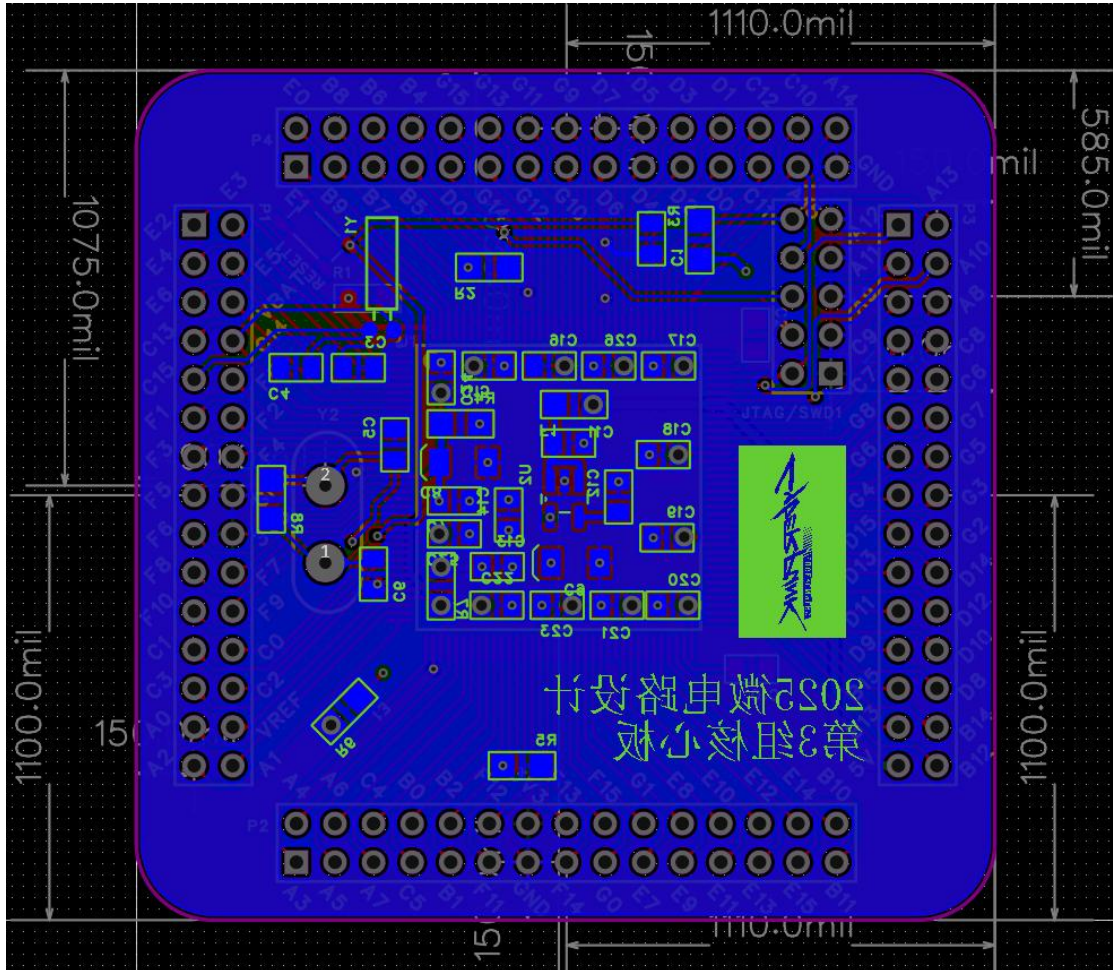


图 3.1.2 核心板 PCB 背面布局

3.2 扩展板 PCB 设计

扩展板采用对称四层板结构，从上到下为正面信号层→GND 层→电源层→背面信号层。螺丝孔、光电门、红外传感器等放置在测量出的精确位置，便于安装时和扫地机器人的结构衔接。外设接口放置在 PCB 边缘，方便接线与测试。

PCB 采用 45° 圆弧走线，整体呈现横平竖直样貌，美观工整，保持走线的阻抗连续性并降低尖角带来的潜在 EMI 问题；大电流走线处（如电机驱动回路、电池充放电线路、开关电源输出线）加粗走线宽度，提升电流承载能力、降低发热与电压压降、增强电路长期可靠性。

在大电流区域以及降压芯片、驱动芯片等位置进行“开窗”处理，便于散热与加工。在焊盘与走线之间添加泪滴作为平滑过渡结构，提升 PCB 的机械可靠性、制造良率和信号完整性。

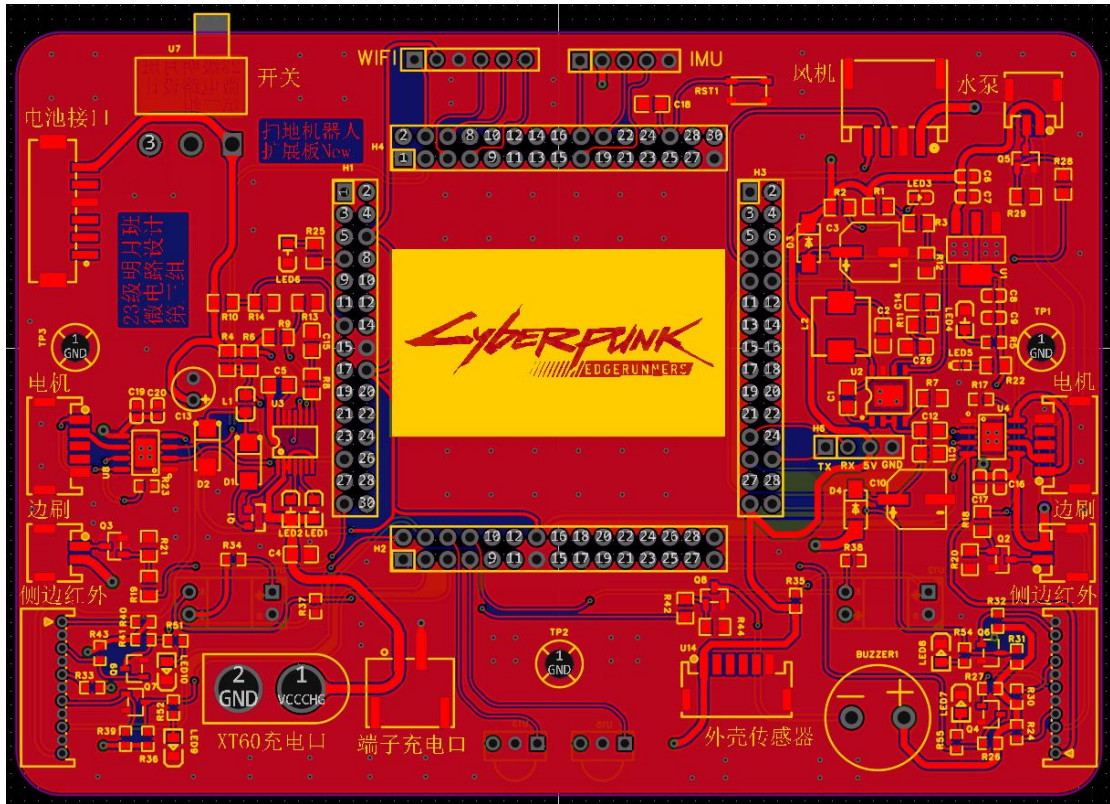


图 3.2.1 核心板 PCB 正面布局

区分 GND、AGND、PGND，从根源上隔离不同类型电路的干扰，避免大电流波动、高频噪声通过“地环路”耦合到敏感电路，保证系统的测量精度、运行稳定性和电磁兼容性（EMC），不同地之间通过 0Ω 电阻进行单点连接。



电源层专门设置不同电压的供电区域，提升电源稳定性、增强信号完整性、简化布线难度、优化散热性能。

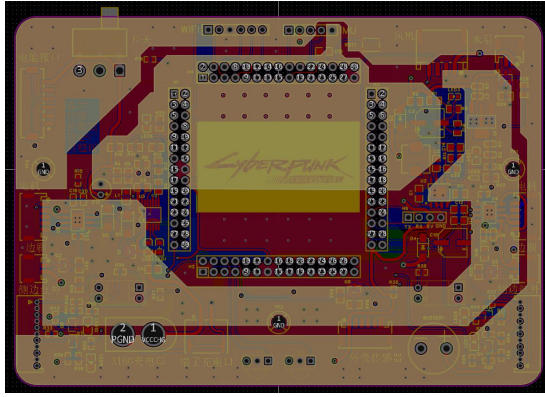


图 3.2.2 扩展板 GND 层

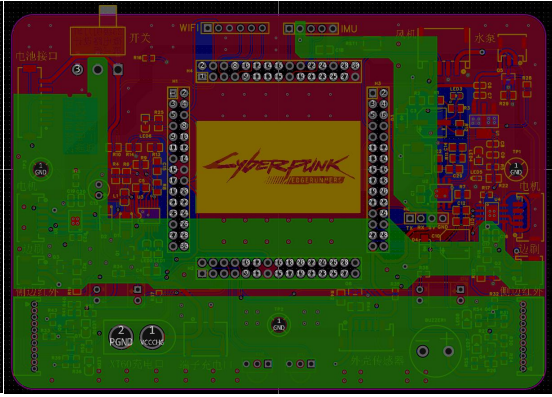


图 3.2.3 扩展板 PWR 层

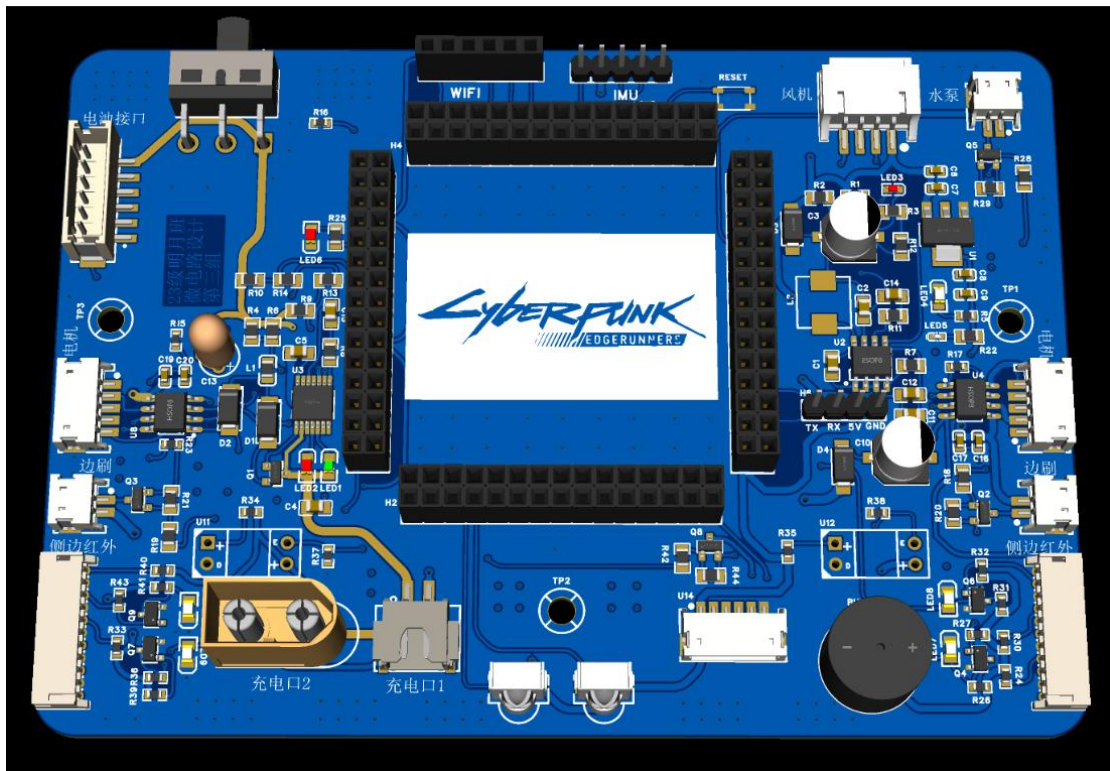
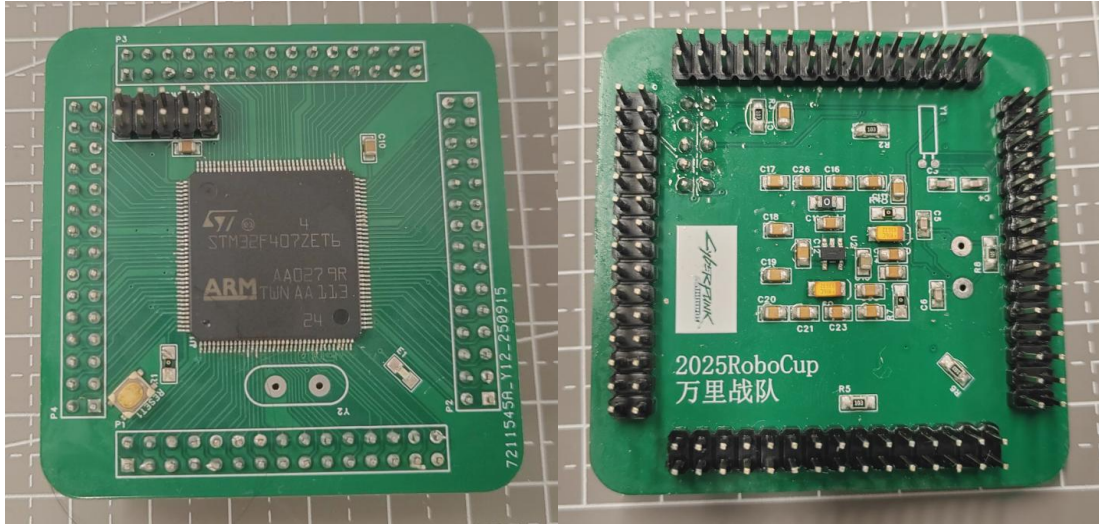


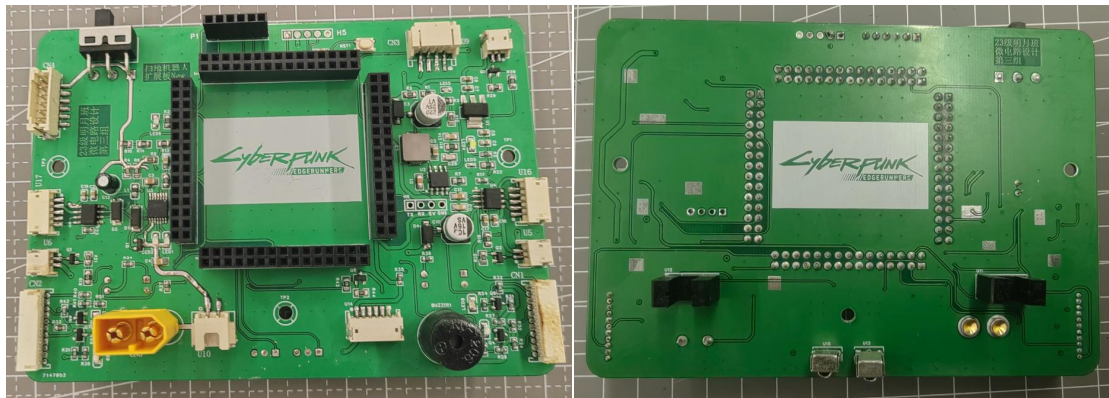
图 3.2.1 扩展板 PCB 3D 视图

3.3 实物展示

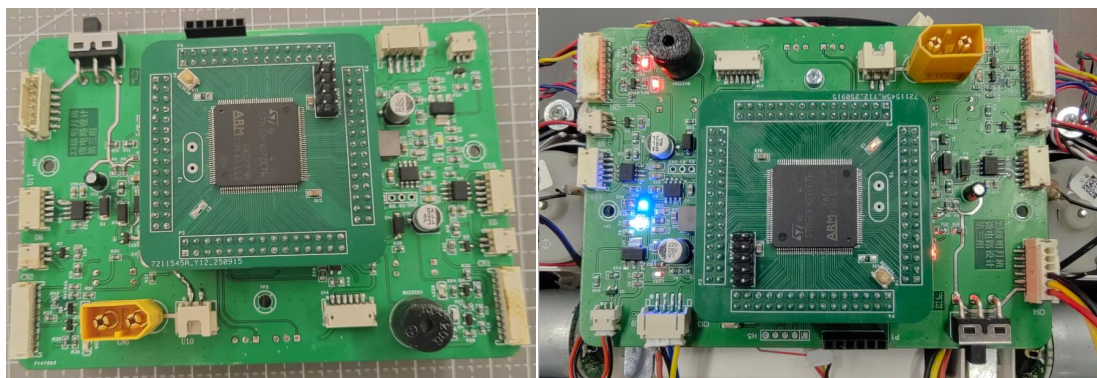
3.3.1 核心板



3.3.2 扩展板



3.3.3 整体



3.3.4 纹波测试

我们对电路板的 16.8V、5V、3.3V 电源输出纹波进行了测试 (如下图所示), 其中 16.8V 的平均峰峰值为 144.60mV, 5V 和 3.3V 的平均峰峰值为 28mV, 符合“百分之一法则”, 对与扫地机器人的功能实现完全足够。

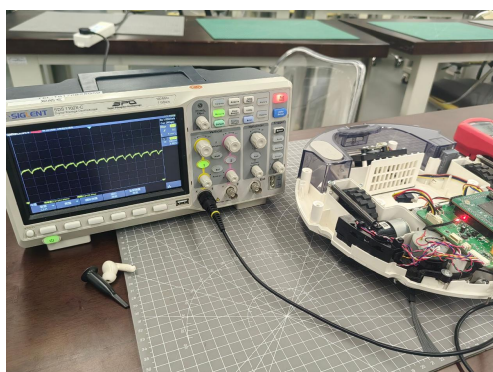


图 3.3.4.1 纹波测试

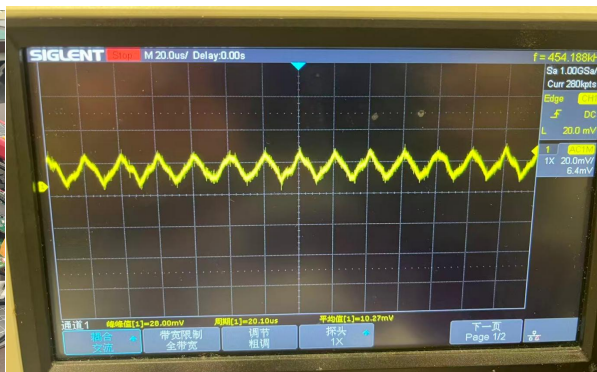


图 3.3.4.2 纹波结果

4 嵌入式软件设计与实现

嵌入式软件部分分为两部分, stm32f407 和树莓派, stm32 负责传感器数据采集、底层驱动; 树莓派负责运动控制。树莓派计算扫地机器人所需运动角度和速度并通过串口发送到 stm32 上, stm32 接受并通过运动解算将输入的角度和速度转为两个电机的转速, 从而实现运动。

4.1 控制方案与代码架构

在本项目中，嵌入式软件是实现扫地机器人核心功能的关键。整个系统采用了分层架构，代码组织清晰，便于后续的扩展和维护。我们将代码分为三个主要层次：BSP 层（板级支持包）、MODULE 层（模块层）和 APPLICATION 层（应用层）。这三层分别承担不同的功能，并且相互之间有清晰的职责划分。

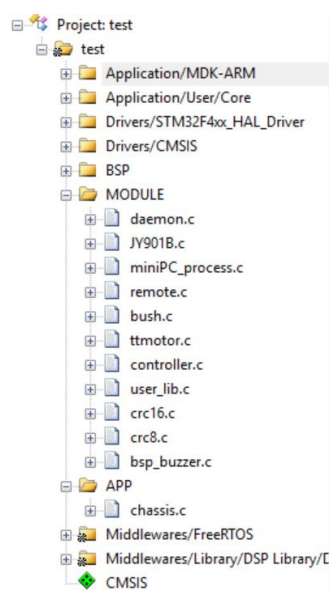


图 4.1.1 嵌入式框架

4.1.1 BSP 层（板级支持包）

BSP 层是整个系统与硬件之间的桥梁，主要负责对硬件外设的底层驱动与接口的支持。以下是该层的核心内容：

外设驱动： BSP 层提供了对各类硬件外设（如传感器、电机、通讯模块等）的驱动代码。具体来说，包括红外避障传感器、电机驱动器、激光雷达接口等的驱动。每个外设驱动模块提供基本的初始化、配置、读写操作等接口，使得上层可以通过这些接口与硬件进行交互。

GPIO 配置与控制： 为实现硬件控制，我们在 BSP 层配置了 STM32 微控制器的 GPIO（通用输入输出）引脚，并通过这些引脚控制电机、LED 指示灯、传感器等外设。GPIO 的配置基于 STM32 的硬件平台，确保信号的准确传输与接收。

定时器与中断管理： 为了实现实时任务调度，我们使用了 STM32 的定时器和中断机制。定时器用于周期性地触发任务，比如读取传感器数据、更新状态信息等。中断管理则用于处理来自外部设备（如按键、传感器）的实时信号。

4.1.2 MODULE 层 (模块层)

MODULE 层负责实现各个功能模块的具体逻辑，每个模块通常包括外设驱动的封装和业务逻辑的实现。该层的实现方式如下：

电机控制：在 MODULE 层中，我们编写了电机驱动控制模块，包括电机的启动、停止、速度调节等功能。通过 BSP 层提供的硬件接口，该模块能够控制电机的转动方向和速度，实现扫地机器人移动的基本动作。

传感器模块：包括红外避障传感器、激光雷达等的控制和数据读取。每个传感器都有相应的数据处理模块，处理从传感器接收到的原始数据，并将其转换为可用的控制信息。例如，激光雷达数据会通过 ROS 的传感器驱动模块传输到上位机系统进行建图处理。

导航与路径规划模块：在这个模块中，我们使用了如 AMCL（自适应蒙特卡洛定位）等算法来进行机器人定位。该模块与激光雷达、IMU（惯性测量单元）协同工作，通过不断调整机器人的位置和姿态，确保机器人在家庭环境中能够稳定行驶并进行清扫。

4.1.3 APPLICATION 层 (应用层)

APPLICATION 层是整个系统的核心，负责系统的高层功能和任务调度。具体功能如下：

任务调度：APPLICATION 层负责管理机器人的整体行为。例如，扫地机器人有多个功能模式，如自动清扫、手动遥控等，任务调度模块根据当前模式控制不同的操作流程。这个模块通过定时器来定期检查和更新机器人的状态，并根据传感器数据、用户输入等做出决策。

状态机与模式切换：为了更好地管理不同的工作模式，APPLICATION 层使用了状态机模式。机器人可以在“待机”、“清扫”、“充电”等多个状态之间切换。状态机帮助管理不同状态下的行为，例如在清扫状态下，机器人会进行路径规划和执行清扫任务，而在待机状态下，机器人会进入待机模式，等待下一次任务指令。

错误处理与恢复：该层还负责处理一些异常情况，例如电池电量不足、传感器故障、运动障碍等。当系统检测到异常时，会切换到错误状态，并采取相应的恢复措施，如自动返回充电站、停止清扫等。

4.1.4 系统通信与调试

在本项目中，机器人与上位机之间的通信通过 Wi-Fi 模块（如 DT-06 WiFi 模块）进行，采用了 TCP/IP 协议。为了保证通信的稳定性和可靠性，我们设计了一个专用的通信协议，确保指令传输的准确性和实时性。通信模块的主要功能包括：

指令传输：通过 DT-06 WiFi 模块，系统能够与上位机（如手机 APP）进行实时通信。用户可以通过 APP 发送指令，如开始清扫、停止清扫、调整清扫模式等，机器人接收到指令后执行相应操作。

传感器数据反馈：除了接收指令外，机器人还会将传感器数据（如电池电量、传感器状态等）实时反馈给上位机，用户可以在 APP 上查看机器人当前的工作状态。

调试功能：为了方便后期调试，我们在 APP 端和控制程序中加入了日志记录功能。通过日志，开发人员可以实时查看指令发送、接收情况，以及各个功能模块的运行状态。

4.2 运动模型与惯性导航

在本项目中，底盘运动控制与导航是基于嵌入式系统和惯性测量单元 (IMU) 实现的。运动控制系统主要负责底盘的行进、转向以及清扫功能的实现，而惯性导航则为机器人的精确定位和路径规划提供数据支持。以下内容将深入讲解底盘运动模型、PID 控制算法与惯性导航的实现。

4.2.1 简单运动模型与底盘控制

底盘的运动控制是本项目的核心功能之一。为了实现精准的底盘运动，系统采用了电机驱动与 PID 控制的结合方式来精确调节运动速度与方向。主要的运动模块包括轮式驱动和电机控制，在此基础上，通过 PID 控制器对速度和角度进行闭环调节，确保机器人能够按预定路径行进。

i 运动控制模型：

底盘的运动控制采用了前进、后退、转向等模式。通过底盘的四个电机（左、右行进轮及风机）来实现不同的运动状态。控制的主要逻辑如下：

线性速度 (v_y) 控制机器人沿直线行驶。

角速度 (vw) 控制机器人绕自身中心旋转。

角速度偏差矫正值 (vw_offset) 控制机器人

每个电机的速度参考值 (TTMotorSetRef) 根据线性速度、角速度的输入进行计算, 并被分别传递给左、右行进轮。具体来说, 左、右行进轮的参考速度通过以下公式计算:

$$\text{速度} = v_y \pm v_w \pm v_w_offset$$

```
void Chassis_bsp() {
    wangba.chassis_ctrl_cmd.current_yaw = O chassis.IMU_data->Yaw;
    MinimizeRotation(&chassis.IMU_data->Yaw, &wangba.chassis_ctrl_cmd.last_yaw, &wangba.chassis_ctrl_cmd.offset_w);
    wangba.chassis_ctrl_cmd.offset_w = (turn_flag==1)?PIDCalculate(&chassis_follow_pid, O chassis.IMU_data->Yaw, wangba.chassis_ctrl_cmd.last_yaw):PIDCalculate(&chassis_follow_pid, O chassis.IMU_data->Yaw, wangba.chassis_ctrl_cmd.last_yaw);
}
if(wangba.chassis_ctrl_cmd.vy == wangba.chassis_ctrl_cmd.vw == 0) {
    wangba.chassis_ctrl_cmd.offset_w = 0;
}
TTMotorSetRef(turn_l, wangba.chassis_ctrl_cmd.vy - wangba.chassis_ctrl_cmd.vw - wangba.chassis_ctrl_cmd.offset_w);
TTMotorSetRef(turn_r, wangba.chassis_ctrl_cmd.vy + wangba.chassis_ctrl_cmd.vw + wangba.chassis_ctrl_cmd.offset_w);
TTMotorControl();
}
if(wangba.chassis_ctrl_cmd.last_yaw>360) {
    wangba.chassis_ctrl_cmd.last_yaw=360;
}
if(wangba.chassis_ctrl_cmd.last_yaw<-360) {
    wangba.chassis_ctrl_cmd.last_yaw=-360;
}
}
```

图 4.2.1.1 底层控制运动模型

这确保了机器人的运动能够灵活调整, 不仅限于直线行驶, 还能在复杂环境中实现转向和调整。

ii. PID 控制:

PID (Proportional-Integral-Derivative) 控制算法被广泛应用于本项目中的底盘跟随控制和转向调整。PID 控制通过调节比例、积分和微分三个系数, 实现对机器人的精确控制。

比例 (P) : 根据当前误差来进行调整。

积分 (I) : 用于消除长期积累的误差, 防止稳态误差。

微分 (D) : 根据误差变化率进行调整, 增加系统的响应速度。

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

在底盘控制中, PID 控制器主要用于调整机器人当前方向与目标方向之间的偏差。chassis_follow_pid 为机器人定向控制的 PID 实例, 负责调整机器人的 yaw 角度 (方向)。PID 算法根据当前机器人 yaw 角与目标 yaw 角之间的误差, 计算出修正值, 并控制电机进行相应调整。

在 Chassis_bsp() 函数中, 通过 MinimizeRotation 函数最小化机器人的角度旋转, 将误差控制在合理范围内。接着, 使用 PIDCalculate 函数计算误差, 并调整机器人的偏航角, 使机器人沿着预定路径行驶。

4.2.2 惯性导航与传感器融合

惯性导航系统 (INS) 结合了 IMU (惯性测量单元) 的数据, 用于精确测量机器人的姿态 (如俯仰角、滚转角、偏航角等)。在本项目中, IMU 传感器提供了机器人当前的角度信息, 帮助导航系统进行精确定位。

i. IMU 数据采集:

IMU 传感器提供了机器人的角度信息, 特别是偏航角 (Yaw), 作为机器人的方向标志。每次机器人运动时, IMU 传感器通过 `Chassis_IMU_data->Yaw` 提供实时的角度数据。该数据用于更新机器人当前的方向, 作为控制算法的输入之一。

在 `Chassis_bsp()`函数中, 通过读取 IMU 传感器的 Yaw 值, 结合 PID 算法调整机器人的方向。系统不断比较当前偏航角和目标偏航角之间的差异, 通过 PID 控制器调整电机的速度, 以减少误差并将机器人导向目标位置。

ii. 传感器融合:

传感器融合技术在本项目中主要体现在 IMU 数据与电机数据的结合使用上。IMU 数据提供了机器人当前方向和姿态的实时更新, 而 PID 控制则根据目标方向和实际方向之间的误差, 动态调整电机控制信号。通过这种方式, 机器人能够根据实时的运动反馈, 进行调整和修正, 确保运动的精准性。

iii. 运动状态的判定与调整:

机器人在运行过程中, 除了惯性传感器的反馈, 还需依赖其他传感器 (如红外传感器、碰撞传感器等) 来提供环境信息, 以保证机器人不发生碰撞或跌落。

例如, 在 `avoid_fall()`函数中, 机器人通过检测前方障碍物来判断是否有掉落风险。若检测到障碍物, 机器人会自动调整方向或后退, 避免发生跌落等危险情况。此外, `Detect_Task()`函数负责读取红外传感器的数据, 判断是否存在障碍物, 并调整机器人的行驶方向。

4.2.3 运动模型与 PID 控制的协同工作

本项目的底盘运动模型不仅仅依赖于 IMU 数据的反馈, 还充分利用了 PID 控制算法来精确调节运动状态。每次运动状态变化时, 机器人会通过计算当前位置与目标位置之间的误差, 并使用 PID 控制进行调整。PID 控制的参数 (K_p , K_i , K_d) 根据机器人的响应特性进行调节, 以确保机器人的运动平稳且精确。

例如, `control_follow_Task()`函数控制机器人的沿墙壁清扫模式。在此模式下,

PID 控制器不断调整机器人与墙壁之间的距离，确保机器人能够稳定地沿墙壁行进。每次调整后，机器人会根据控制指令更新电机的输出，避免偏离预定路径。

4.2.4 运动模型与惯性导航的总结

通过结合 PID 控制与 IMU 传感器，本项目成功实现了扫地机器人的精确运动控制与惯性导航。PID 控制确保了机器人能够在不断变化的环境中稳定行驶，而 IMU 传感器提供的姿态数据为运动控制提供了实时反馈，使得机器人能够在执行任务时保持较高的精度。

在此基础上，机器人还结合了碰撞检测、避障机制以及传感器融合技术，进一步增强了运动控制的稳定性和可靠性。通过这些技术，机器人能够在复杂环境中灵活应对各种障碍物和变化的运动需求，最终实现高效、智能的清扫任务。

5 树莓派运动计算

5.1 建图和定点导航功能

本项目中的导航与建图模块采用了树莓派作为上位机平台，基于 ROS 生态系统实现。通过集成 Cartographer 进行实时地图构建，并使用 Navigation 进行导航定位与路径规划，最终实现机器人在家庭环境中的精确定位与路径规划。以下将详细介绍建图和定点导航的实现流程，重点讲解局部规划器的概述和功能实现。

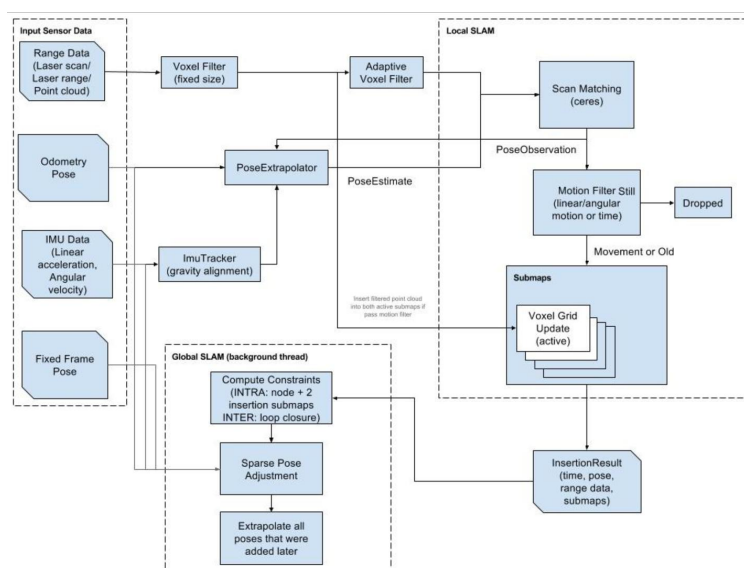


图 5.1.1 Cartographer 整体框架

5.1.1 建图功能

建图是机器人实现自主导航的基础。在本项目中，Cartographer 作为建图与定位的核心算法，负责在激光雷达的输入数据基础上构建环境地图。该过程依赖于雷达数据的实时采集与处理，并通过 ROS 中的 `**/scan` 话题发布数据，供 Cartographer 进行处理。建图的具体流程如下：

激光数据采集与发布：激光雷达通过 360° 全视野扫描采集环境中的障碍物信息，并将数据通过 ROS 发布为 `**/scan` 话题。这些数据包含了每个扫描点的距离信息和角度信息，是建图算法的基础输入。

Cartographer 处理与地图生成：Cartographer 订阅 `**/scan` 话题，并结合里程计 (odometry) 信息进行数据融合，实时生成机器人所在环境的地图。Cartographer 利用同步定位与地图构建 (SLAM) 技术，通过高效的算法将激光数据转换为地图。

保存地图文件：生成的地图文件以 `.pbstream` 或 `.yaml` 格式保存。`.pbstream` 文件包含了完整的地图数据及实时定位信息，而 `.yaml` 文件则用于存储地图的元数据，包括分辨率、地图尺寸等。

通过这种方式，Cartographer 能够为机器人创建高精度的二维地图，为后续的定位和路径规划提供可靠的空间数据支持。

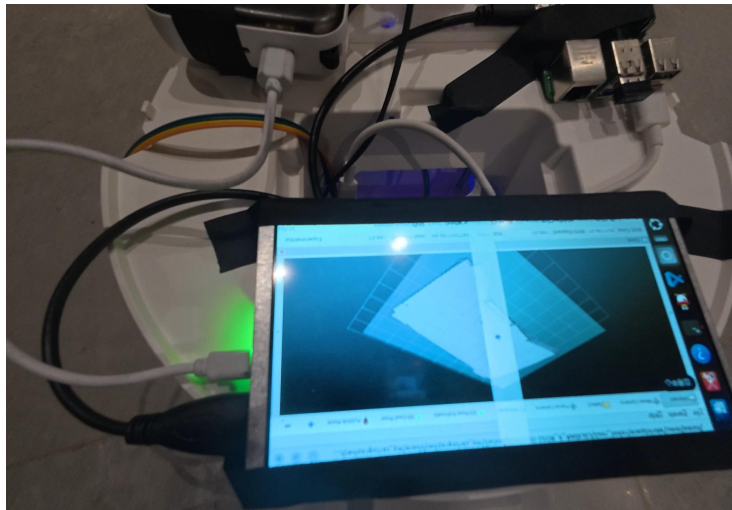


图 5.1.1.1 建图过程

5.1.2 定点导航与路径规划

在完成地图构建后，机器人需要利用这些地图数据进行定位与路径规划，确保能够有效地完成清扫任务。ROS 中的 Navigation 包承担了导航与路径规划的核心

心任务，包括地图加载、定位、路径规划和运动控制等功能。

地图加载与 AMCL 定位：通过 Navigation 包，机器人首先加载保存的地图文件（如**.pbstream 或.yaml 文件），并利用 AMCL（自适应蒙特卡洛定位）**算法进行定位。AMCL 通过粒子滤波方法，结合 IMU 数据和激光雷达数据，估算机器人在地图中的精确位置。该算法能够在环境中存在一定不确定性的情况下，提供可靠的定位结果。

路径规划：在定位完成后，机器人根据目标位置生成路径。路径规划采用全局规划与局部规划相结合的方式。全局规划通过计算从当前位置到目标位置的最短路径，而局部规划则处理在执行过程中遇到的障碍物，并生成局部的避障路径。

发布运动控制命令：在路径规划过程中，Navigation 包通过**/cmd_vel**话题发布速度命令，控制底盘的运动。控制命令根据当前的速度和角度调整底盘的行进方向，以确保机器人能够沿着规划路径顺利移动。

5.1.3 局部规划器概述

局部规划器是 move_base 导航系统中的最核心组件之一，直接决定了机器人是否能够按照全局路径成功行驶，并在行驶过程中处理动态避障问题。局部规划器不仅对全局路径起到补充作用，还能够在机器人遇到障碍物时，实时计算避障路径，避免碰撞并保持运动的流畅性。以下将详细介绍局部规划器的工作原理及其实现技术。

i. 局部规划器的作用：

局部规划器的主要功能是根据全局规划路径和实时环境数据计算机器人当前的运动轨迹，并对动态障碍物进行避让。它需要解决以下几个问题：

全局路径的跟踪：局部规划器会根据全局路径规划算法生成的路径（通常是一个一系列的目标点）来调整机器人实际行驶的轨迹。全局路径是一个离散的目标点集合，局部规划器需要将这些目标点转化为连续的轨迹，并根据环境的变化实时调整。

动态避障：机器人在运动过程中可能会遇到新的障碍物或移动的物体，局部规划器需要通过实时的激光雷达数据、IMU 传感器数据等输入，判断前方是否有障碍物，并在此基础上重新计算避障路径。

轨迹平滑与优化：局部规划器不仅需要实时计算避障路径，还需要考虑路径的平滑性与可行性。它通过优化算法平滑转弯和路径中的急转弯部分，确保机器人行驶的平稳性与稳定性。

ii. 局部规划器的工作流程：

局部规划器的工作流程主要分为以下几个步骤:

输入数据: 局部规划器接收来自全局路径规划器的目标路径, 并订阅来自激光雷达和其他传感器的实时数据 (如障碍物位置、机器人姿态、速度等)。

路径跟踪: 局部规划器通过轨迹跟踪算法计算机器人的运动轨迹, 并与全局路径进行比对。如果机器人偏离了全局路径, 局部规划器会进行调整, 确保机器人能够尽可能沿着全局路径行驶。

避障与重新规划: 当机器人检测到新的障碍物时, 局部规划器会实时计算新的避障路径。这些路径会绕过障碍物, 并确保机器人能够继续按照目标方向前进。局部规划器通常采用一些避障算法, 如动态窗口法 (DWA)、开环避障等。

发布控制命令: 经过计算后, 局部规划器会发布速度命令 (通过**/cmd_vel**话题) 控制机器人的移动, 调整机器人的速度和方向, 确保其能够避开障碍物并顺利前进。

iii. 局部规划器与全局规划器的协同工作:

全局规划器和局部规划器的协同工作是实现高效导航的关键。全局规划器负责生成一个全局路径, 而局部规划器则负责将这一路径转化为实际的行驶轨迹, 并在过程中应对动态环境变化。

全局规划器负责计算从起始位置到目标位置的最短路径, 通常使用 A*算法、Dijkstra 算法等进行全局路径搜索。

局部规划器负责根据实时环境数据对全局路径进行修正和优化。它会考虑到机器人当前位置与目标路径的误差, 以及障碍物的影响。

在这种协同工作下, 机器人能够在不断变化的环境中, 保持路径的稳定性和避障能力。局部规划器通过动态调整机器人的运动, 使其能够在全局路径上顺利行驶, 同时避免碰撞。

5.2 全局清扫和沿墙清扫

在本项目中, 机器人首先实现了基于定点导航的功能, 能够根据用户指定的目标位置进行精确定位与路径规划。在此基础上, 我们进一步扩展了全局清扫与沿墙清扫的功能, 以适应复杂家庭环境中的不同清扫需求。全局清扫和沿墙清扫功能不仅要求机器人能够自主规划路径, 还需要具备避障能力、路径优化能力以及对环境的适应能力。

5.2.1 从定点导航到全局清扫功能的实现

全局清扫功能要求机器人能够覆盖整个清扫区域，而不仅仅是根据单一目标点进行导航。在实现全局清扫功能时，我们需要在定点导航的基础上，添加对环境的全面感知与路径规划优化。具体实现过程如下：

i. 全局路径规划的调整与扩展：

在定点导航模式下，机器人通常基于用户指定的目标位置进行导航，路径规划是一次性的，完成之后机器人便会停留在目标位置。然而，全局清扫需要机器人在整个清扫区域内不断移动，并确保每个区域都能够被覆盖到。

为此，我们引入了全局路径规划的循环执行。在全局清扫模式下，机器人首先通过 Cartographer 生成一张环境地图，并根据地图信息计算出清扫区域的边界和各个区域的优先级。全局规划器计算出一个整体路径，机器人按路径规划依次清扫整个区域。路径规划采用的是分区策略，每次清扫时，机器人根据当前的位置信息和环境的实时变化，动态调整行进路径。

ii. 动态避障与路径调整：

全局清扫过程中，机器人可能会遇到新出现的障碍物，因此需要实时避障。为了处理这种情况，我们使用了局部规划器（如 DWA 动态窗口法）来实现避障功能。局部规划器在接收到全局路径的目标点后，会对路径进行局部调整，绕过障碍物并重新规划运动轨迹。机器人在执行全局清扫任务时，始终保持对环境的感知，并能够在动态环境中根据障碍物位置和实时反馈调整运动路径。

iii. 路径优化与覆盖率：

全局清扫功能的另一个关键问题是路径的优化。为了提高清扫效率并减少重复行驶，我们采用了最短路径规划和路径优化策略。通过算法优化，机器人在每次运动时，都会选择最短、最直接的路径来覆盖整个清扫区域，避免重复行驶，减少无效的能源消耗。此外，机器人还会定期检查清扫区域的覆盖率，确保没有漏扫的地方。

IV. 实现过程中的关键技术：

环境建图与定位：通过 Cartographer 算法，机器人能够实时生成环境地图，识别不同的清扫区域，并根据地图信息进行全局路径规划。

全局路径规划：全局路径规划算法（如 A*算法、Dijkstra 算法）负责计算从当前位置到目标清扫区域的最短路径。

动态避障与局部规划：在清扫过程中，局部规划器负责处理动态障碍物，并为机器人重新计算避障路径。

通过这些技术的结合，机器人能够实现全局清扫功能，自动覆盖整个清扫区域，并有效避开障碍物，保证清扫任务的顺利完成。

5.2.2 从定点导航到沿墙清扫功能的实现

沿墙清扫功能是为了应对家庭环境中墙角和狭小空间的清扫需求。与全局清扫不同，沿墙清扫强调的是沿着房间的边缘行驶，以确保墙角和边缘的垃圾能够被彻底清扫。实现这一功能的关键在于机器人能够持续监测墙壁与自身的相对位置，并根据这一位置实时调整行进路径。具体实现过程如下：

i. 墙壁感知与跟踪：

为了实现沿墙清扫，机器人需要通过红外传感器或激光雷达来检测与墙壁的距离。通过传感器数据，机器人能够实时感知到与墙壁的相对位置，并在此基础上调整行驶轨迹。具体来说，机器人会根据与墙壁的距离，计算出最优的运动轨迹，并保持一定的距离进行行驶。常见的操作是通过 PID 控制算法来调整机器人的运动速度和转向角度，保持与墙壁的稳定距离。

ii. 路径规划与调整：

在沿墙清扫模式下，机器人并不需要计算到达某个目标点的路径，而是按照墙壁的轮廓进行清扫。因此，机器人在沿墙行驶时，其路径规划是基于墙壁的形状和机器人当前的位置。局部规划器根据传感器数据实时调整机器人运动的方向，确保机器人始终能够沿着墙壁行驶。

当机器人遇到直角墙角或障碍物时，局部规划器会及时调整机器人的转向角度，避免撞到墙壁并确保继续沿着墙壁清扫。具体的转向策略是根据墙壁的形状和角度来调整机器人的航向，确保机器人能够平稳地绕过墙角，并继续沿着新的墙段清扫。

iii. 避障与重新规划：

在沿墙清扫过程中，机器人会根据实时传感器数据判断前方是否存在障碍物。若检测到障碍物，局部规划器将重新计算避障路径，并调整机器人的行进方向，以绕过障碍物并继续沿墙行驶。通常，沿墙清扫的避障路径规划是短距离的局部调整，而不是全局路径的重新规划。

6 APP 端设计与通信协议

APP 设计选用微信小程序，基于微信开发者工具平台进行开发。

选用该技术路线的原因在于：在家庭使用环境中，微信小程序控制比网页端控制更贴合现实场景需求。家庭用户的操控习惯多为碎片化、便捷化导向，网页端控制需额外打开浏览器、输入指定网址，操作链路冗长，存在一定使用门槛；而微信作为家庭用户高频使用的社交工具，小程序可直接嵌入其中，无需额外下载安装，实现“即开即用”，点击即可启动控制界面，完美适配家庭场景下随时启停、临时调整清扫指令的需求，大幅降低用户的操作成本。



图 6.1 小程序封面

总的来说，我们设计的小程序通过与 DT-06 WiFi 串口透传模块通信，实现对扫地机器人的控制及辅助开发。

小程序页面分为网络连接页、功能实现页、辅助开发页三类，具备页面分类清晰、自动状态检测、动态监测及支持自定义开发的特点；技术层面，开发工具涵盖微信小程序控制面板等软件与 DT-06 串口透传模块等硬件，系统总体架构遵循“用户界面 - 网络层 - 硬件层”的交互流程，同时设计通信协议，明确各功能按键对应的 Function 位及二进制指令示例；技术实现阶段按照“创建后台程序→编辑基础功能→与 WiFi 模块通信→搭建通信协议→与下位机联调、优化 UI”的流程推进，并完成二进制协议 Function 位的完整分配，为小程序功能落地提供支撑。

6.1 页面设计与交互功能

在页面设计上，为实现对扫地机器人的控制及辅助开发，页面按“连接-控制-扩展”逻辑拆解为三类，兼顾实用性与可扩展性，核心设计特点体现在自动状态检测、动态监测及自定义开发支持，配合可视化交互组件提升操作体验。各页面针对实际操作痛点针对性设计，解决调试及使用中的核心问题，各个页面效果如图 6.1.1 所示。



图 6.1.1 各页面效果展示

6.1.1 页面模块设计

我们将小程序页面拆解为网络连接页、功能实现页、辅助开发页三类，每一类都针对实际操作中的痛点做了针对性设计：

i. 网络连接页：

除了基础的 TCP 客户端配置 (IP、端口、二进制协议选择)，额外开发了“连接预检测”功能——在点击“连接”按钮前，小程序会自动检测当前设备的网络类型 (WiFi/移动网络)，若为移动网络则弹出“非局域网可能连接失败”的提示；通信日志模块不仅记录指令收发信息，还会自动解析日志中的错误码 (如“Error-read CONNSET”对应“连接未建立”)，并标注错误原因，解决了前期“日志一堆代码看不懂问题”的调试难点。



图 6.1.1.1 网络连接提示 (未连接状态)

ii. 功能实现页：

设计了“按键 + 摇杆”双控制模式：按键区采用“功能分组”布局，将“启停控制”、“清扫模式”、“辅助功能”分为三列，避免误触；摇杆区则

绑定了“动态速度映射”逻辑——摇杆滑动的距离对应机器人的移动速度（滑动越远速度越快），同时加入“转向角滑块”，可精准调节机器人的转向弧度（范围为 $-90^{\circ}\sim+90^{\circ}$ ）。在测试时，我们发现机器人在转向角超过 60° 时易侧翻，于是在代码中增加了“角度超限自动截断”的限制，确保操作安全。

iii. 辅助开发页：

为了适配后续机器人功能的扩展，我们设计了“自定义功能”的可视化配置界面——支持输入指令的十六进制码、选择功能图标、设置触发方式（单击 / 长按），并将配置数据存储在小程序的本地缓存中；同时增加了“快捷按键导出 / 导入”功能，方便将自定义按键分享给其他开发者。

6.1.2 核心设计特点

在页面与功能设计中，有以下几个特点：

① 功能页面划分：

将“连接配置 - 实时控制 - 扩展开发”三类低关联度的功能拆分到独立页面，比如网络连接页仅保留通信参数与日志，避免了单页功能拥挤导致的误操作；同时在页面顶部增加“功能面包屑”，方便快速切换页面。

② 自动状态检测：

通过 `setInterval` 定时轮询 DT-06 模块的状态反馈（每 500ms 请求一次），在小程序顶部导航栏实时显示“已连接（绿色） / 断开（红色） / 异常（黄色）”的状态标识；若检测到连接断开，会自动暂停指令发送，并弹出重试提示，解决了“连接断了还发指令导致指令堆积”的问题。



图 6.1.2.1 通信日志

③ 动态监测：

通信日志模块支持“按类型筛选”（仅看错误日志 / 仅看指令日志），且每条日志会标注“[发送] / [接收] / [错误]”的类型标签；我还在日志中加入了指令的“执行状态”（如“[发送] 启动指令 → [接收] 执行成功”），让整个指

令流转过程可视化，大幅提升了调试效率。

④ 支持自定义开发：

辅助开发页的“自定义功能”模块，不仅支持指令配置，还能绑定“参数变量”——比如设置“清扫速度”时，可将指令中的速度字段设为变量，通过滑块实时调整数值后再发送，这解决了“固定指令无法调整参数”的局限，也是我针对机器人调试需求自主设计的功能。

6.2 技术原理与协议设计

小程序采用“软件+硬件”协同架构，以微信开发者工具与 DT-06 WiFi 串口透传模块为核心，构建“用户界面-网络层-硬件层”交互体系，通过自定义通信协议实现与机器人下位机的稳定数据传输。

6.2.1 开发工具系统架构

软件端使用微信开发者工具完成前端界面（WXML+WXS）与逻辑层（JS）开发；硬件端选用 DT-06 WiFi 串口透传模块作为小程序与机器人下位机的通信中介，负责指令透传与状态反馈。系统总体架构如图 6.2.1 所示，各层级功能分工明确：

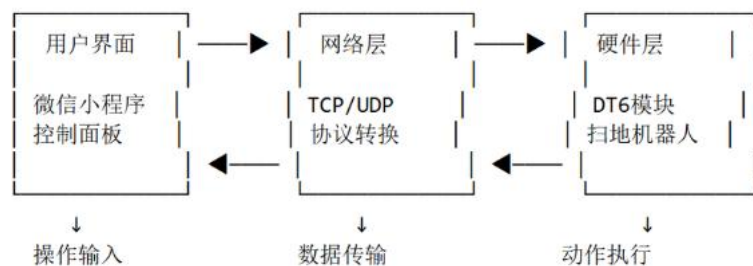


图 6.2.1.1 系统整体架构

用户界面层：提供可视化操作界面与交互组件，接收用户操作指令，展示设备状态、通信日志等信息。

网络层：核心实现 TCP 连接管理与指令缓存队列功能，网络延迟时暂存未发送成功的指令，待连接恢复后按顺序发送，避免指令丢失。

硬件层：通过 DT-06 模块完成 WiFi 信号与串口信号的转换，建立小程序与下位机的数据链路。

6.2.2 通信协议设计

为保障指令传输的唯一性与稳定性，自定义二进制通信协议，格式为“帧头+功能位+参数位+校验位+帧尾”，具体设计如下：帧头固定为“89”，用于标识指令起始；功能位扩展为 2 字节（原 1 字节），解决功能编码重复导致的指令误触发问题；参数位根据功能需求配置对应参数（如转向角、移动速度）；校验位采用异或运算，确保指令传输完整性；帧尾固定为“FF”，标识指令结束。

协议明确各功能按键对应的 Function 位分配，例如“停止”对应 0x0000、“全屋清扫”对应 0x0002，并提供二进制指令示例，同时完成 Function 位的完整分配，为功能落地提供支撑。自定义通信协议展示如图 6.2.2.1 所示。



Function位	十六进制	功能名称
0x00	0x00	手动控制
0x01	0x01	沿墙清扫
0x02	0x02	全屋清扫
0x03	0x03	停止/暂停
0x04	0x04	返回充电
0x05	0x05	导航模式
0x06	0x06	设置速度
0x07	0x07	设置角度
0x08	0x08	设置清洁等级
0x09-0xFF	-	保留

图 6.2.2.1 自定义二进制通信协议

6.3 实现流程

整个微信小程序开发的流程为：创建后台程序 → 编辑基础功能 → 与 WiFi 模块通信 → 搭建通信协议 → 与下位机联调 → UI 优化，体现了从“链路打通”到“体验完善”的工程推进逻辑，具体流程如下：

微信小程序开发

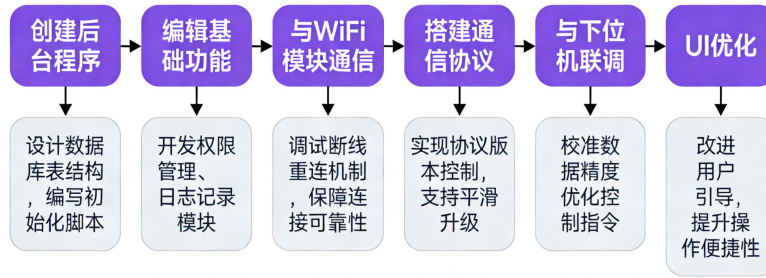


图 6.2.2.1 流程示意

1. 创建后台程序:

在微信开发者工具中完成项目初始化，明确项目名称、AppID 及目录结构，搭建基础开发框架。配置 app.json 文件，定义页面路由，确保页面跳转路径清晰可追溯；使用状态提示栏、日志面板等进行辅助开发。在 app.js 中定义 globalData 全局变量池，存储连接状态、指令缓存队列、波特率配置等核心数据；额外添加全局异常捕获函数，对程序运行中的报错（如变量未定义、接口调用失败）进行捕获并记录，辅助调试。

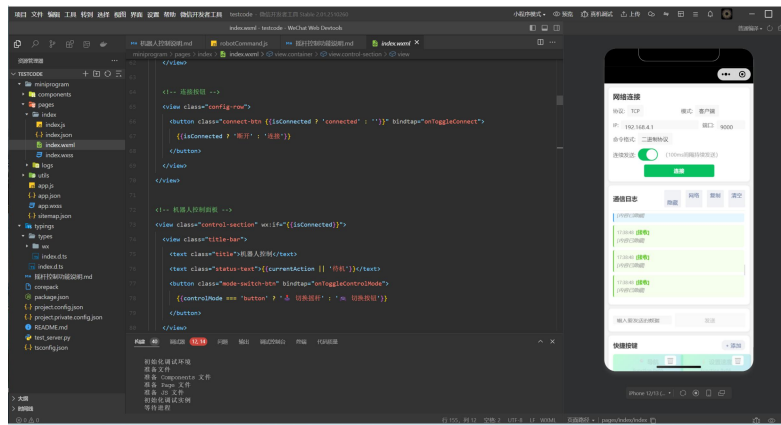


图 6.3.3.1 微信开发者工具开发展示

2. 编辑基础功能:

开发按键、摇杆等前端交互组件，绑定点击、滑动事件，实现“操作→指令编码”基础逻辑。摇杆组件选用自定义 Canvas 绘制，支持 360° 全方位操控，绑定 touchstart、touchmove、touchend 事件，实现摇杆跟随手指滑动的实时响应；针对初期摇杆滑动卡顿、速度映射不精准的问题，优化 touchmove 事件触发机制，通过定时器控制触发频率从每帧触发改为每 50ms 触发，减少事件触发次数，同时优化速度映射算法，将摇杆滑动距离（像素值）按比例转换为机器人移动速度（0-100 级），并使用算法消除手指自然颤抖。

3. 与 WiFi 模块通信:

编写 TCP 连接核心 JS 代码, 构建 “连接 - 断开 - 重连 - 状态监听” 的完整通信链路。连接逻辑中, 通过 wx.createSocket () 创建 TCP 客户端连接, 传入配置好的 IP 地址、端口号及二进制协议参数, 同时添加连接成功、连接失败、连接关闭的回调函数, 实时更新全局连接状态并同步至界面; 断开功能支持手动点击断开按钮及异常自动断开, 手动断开时主动清空指令队列, 异常断开时触发重连机制。针对 DT-06 WiFi 模块的丢包现象, 设计定时重复发送指令的逻辑, 每 50ms 秒发送一次, 维持模块与小程序的信息稳定传输, 避免模块因数据丢包导致传输错误; 额外添加网络状态实时监测, 若连接过程中网络从 WiFi 切换为移动网络, 立即暂停指令发送并弹出提示。



图 6.3.3.2 串口通信调试过程

4. 搭建通信协议:

封装指令编码与解码两大核心函数, 完成自定义二进制协议的落地实现。编码函数按 “帧头 + 功能位 + 参数位 + 校验位 + 帧尾” 格式拼接完整协议帧, 首先固定帧头 (0x89) 与帧尾 (0xFF), 再根据前端操作传入的功能标识赋值功能位 (如全屋清扫对应 0x0002), 参数位则根据功能需求填充对应数据 (如转向角滑块值转换为 8 位二进制数、速度值对应参数字段); 校验位通过异或运算实现, 遍历帧头至参数位的所有字节并计算异或结果, 确保指令传输过程中无数据丢失或篡改。解码函数负责解析机器人下位机返回的状态数据帧, 先校验帧头帧尾合法性, 合法帧则拆分功能位、参数位, 将参数位数据转换为直观的界面展示信息 (如电量参数对应百分比显示、故障码对应文字提示), 非法帧则标记为错误日志并丢弃; 同时对接通信日志模块, 将编码后的发送指令、解码后的接收数据实时写入日志, 标注指令类型与执行状态。

5. 与下位机联调、优化 UI:

联合下位机开发人员测试各指令执行效果，验证“沿墙清扫”“返回充电”等功能的准确性。针对操作反馈优化 UI，将功能按钮尺寸从 40rpx 调整为 60rpx，添加点击反馈动画，提升操作体验。增大点击区域，同时添加点击反馈动画（轻微缩放 + 颜色加深），提升操作触感；优化页面布局层级，将常用功能（摇杆控制、启停按键）置于页面上半部分，不常用功能（参数配置、日志筛选）折叠至下拉面板，减少页面滚动操作。



图 6.3.1 结合实机进行调试

7 整机集成与功能展示

7.1 整机效果



图 7.1.1 扫地机器人实物图

这是我们的实物效果图。整机外观规整，各模块布局合理，硬件电路与传感器接线牢固且做好绝缘防护，供电模块稳定输出适配电压；下位机与上位机通过 DT-06 WiFi 串口透传模块实现通信链路打通，小程序可精准接收机器人状态反馈并下发控制指令，无信号丢失或延迟异常问题；雷达建图功能正常，能自动感知周边环境，实现路径规划。

7.2 基础功能

7.2.1 按键/摇杆操纵移动

按键与摇杆双模式操纵移动是上位机小程序手动遥控的核心交互方式，指令经二进制协议编码后，通过 DT-06 WiFi 串口透传模块实时下发至下位机，响应延迟控制在 30ms 以内，实现机器人的实时动作跟随。

按键操纵采用“方向 + 功能”组合布局，方向按键包含前、后、左、右四个基础方向，支持点动单次移动与长按连续移动两种触发模式，搭配清扫力度切换按键，可在移动过程中同步调整清洁强度；摇杆操纵则采用 360° 全方位操控设计，绑定动态速度映射逻辑，同时集成转向角调节功能，精准控制机器人运动。两种操纵模式可在小程序界面一键切换，满足不同用户的操控习惯与复杂户型的移动需求。

速度与转角的灵活可调，将上位机小程序摇杆的滑动幅度与机器人移动速度精准绑定，滑动距离越大，速度等级越高，速度调节范围覆盖 0-100 级，可适配不同户型场景需求——小幅度滑动对应低速精细调整，适用于桌椅缝隙等狭窄区域清扫；大幅度滑动对应高速移动，适用于开阔空间的快速转移。同时小程序界面配备速度数值显示栏，实时反馈当前速度档位，方便用户精准把控。转角支持 $-90^{\circ} \sim 90^{\circ}$ 全范围角度控制，通过摇杆偏转角度或界面滑块两种方式实现：摇杆偏转方向对应机器人转向方向，偏转角度对应转向弧度；界面滑块则提供更精细化的角度输入，满足精准转向需求。调节后的转角参数会同步上传至小程序，结合雷达建图数据，可直观显示机器人转向轨迹，辅助用户预判移动路径。



图 7.2.1.1 移动功能

图 7.2.1.2 直线误差

特点：我们自主更换双路编码器，实现更精确控制，在直线行走测试中 2m 误差不大于 3cm（如图 7.2.1.2）。



图 7.2.1.3 更换电机

7.2.2 三级清洁力度

三级清洁力度是手动遥控模式下适配不同清洁场景的核心功能，通过小程序界面的独立按键实现一键切换。主控根据接收的功能位指令，精准调节清扫执行机构的滚刷转速与吸尘功率，实现清洁强度的分级控制。三个挡位的功能定位与适用场景明确区分：

一级清洁：滚刷转速与吸尘功率调至基础档位，噪音小，适用于地面不太脏的情况。

二级清洁：滚刷与吸尘功率提升至均衡档位，是清洁的默认模式，可高效清理正常的垃圾（如测试中的纸屑），兼顾清洁效率与设备续航。

三级清洁：滚刷转速与吸尘功率拉满，针对需要大幅清理的场景设计，能更

有力打扫，缺点是噪音较大。



图 7.2.2.1 清洁档位调节



图 7.2.2.2 清洁效果

清洁力度切换响应延迟小于 50ms，支持在机器人移动过程中无缝切换，小程序界面会同步显示当前挡位标识（绿色一级、蓝色二级、红色三级），并实时反馈滚刷运动状态，方便用户根据清洁需求精准调控。

7.2.3 沿墙清扫

沿墙清扫功能是自动清洁模式下针对墙角、墙边等卫生死角的专项优化功能，可通过微信小程序一键启动。

该功能依托机器人侧边搭载的红外测距传感器与 N10 激光雷达融合感知，实时检测机身与墙体的间距，通过闭环控制算法将贴合距离稳定控制在 3-5cm，既保证滚刷能深入清理墙角缝隙的积尘，又避免机身与墙体发生硬性碰撞。

采用贴墙匀速行进+小角度转向的路径策略，遇到 90° 直角墙角时，机器人会先以 5cm 半径小幅转向，再重新校准贴墙距离继续清扫。小程序会同步显示沿墙清扫状态、实时贴墙距离及清洁进度，支持用户手动干预暂停或退出，进一步提升功能的实用性与可控性。



图 7.2.3.1 沿墙清扫展示

7.2.4 碰撞检测

该结构的核心工作原理是：机器人前端周向布置两组杠杆式防撞触臂，触臂与机身通过弹性铰链连接，常态下触臂保持伸展状态，与机身内部的光电门发射端、接收端保持对齐，光电门处于通路状态；当机器人行进过程中触臂接触到障碍物时，触臂受压向内收缩，触发杠杆结构偏转，直接遮挡光电门的光路，使光电门由通路转为断路，随即产生电平跳变信号。

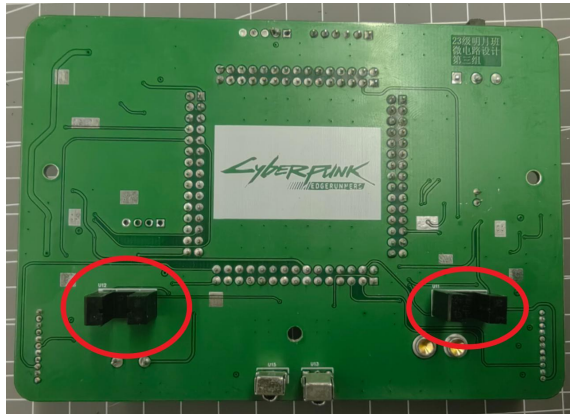


图 7.2.4.1 光电门设置

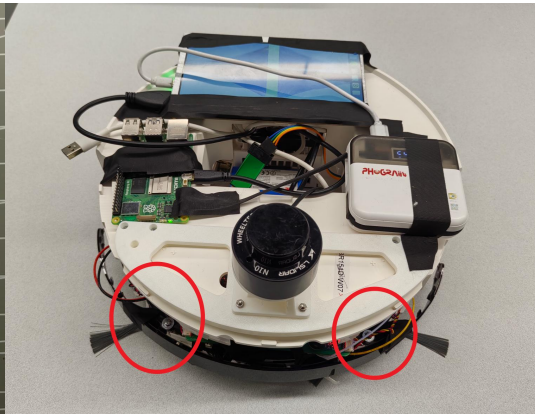


图 7.2.4.2 防撞触臂

该信号会实时传输至主控，主控接收到碰撞信号后，会立即执行三级防护动作：第一级是紧急制动，控制驱动轮停止转动，避免持续碰撞加剧损伤；第二级是反向退避，驱动机器人向后小幅移动（约 8cm），脱离与障碍物的接触，此时弹性铰链带动触臂复位，光电门光路恢复通路；第三级是转向绕行，根据碰撞位置（左前/右前/正前）转向角度（左转/右转 15° - 30° ），规避障碍后重新回归清扫路径。

7.2.5 防跌落检测

防跌落检测功能基于机器人底部均匀布置的多组红外测距传感器实现, 该红外传感器采用“发射 - 接收”工作模式, 可实时检测机身与下方地面的高度差, 默认设定 2cm 为跌落风险阈值。

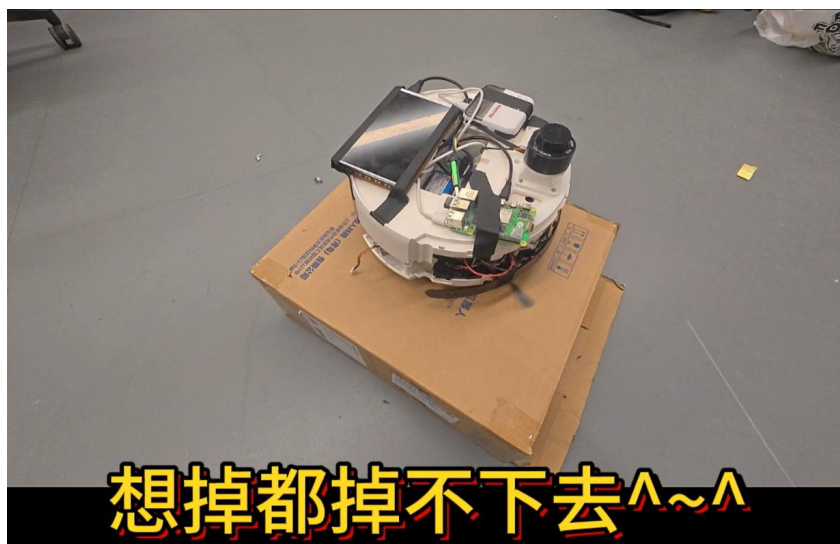


图 7.2.5.1 防跌落效果展示

当机器人行进至楼梯口、阳台边缘等悬空区域时 (测试时为纸箱), 底部红外传感器会因地面反射信号骤降, 判定存在跌落风险, 随即向主控传输高电平触发信号。主控接收到风险信号后, 会立即对驱动轮紧急制动, 避免机身继续向悬空区域移动; 之后控制机器人反向退避约 10cm, 脱离风险区域。

7.3 拓展功能

7.3.1 语音控制

语音控制功能基于 ASRPRO 语音识别模块开发, 是扩展开发类中提升人机交互便捷性的核心功能, 支持通过自然语音指令实现机器人的免手动操控; 与主控通过串口通信, 预先烧录了“打开风机”“沿墙清扫”等常用指令集; 其工作流程简洁高效, 用户发出语音指令后, ASRPro 模块快速完成语音采集、特征提取与指令匹配, 将识别结果转化为标准化控制指令, 发至主控, 主控接收指令后驱

动机器人执行对应动作。无需手动操作小程序，大幅提升了使用的便捷性与智能化程度。

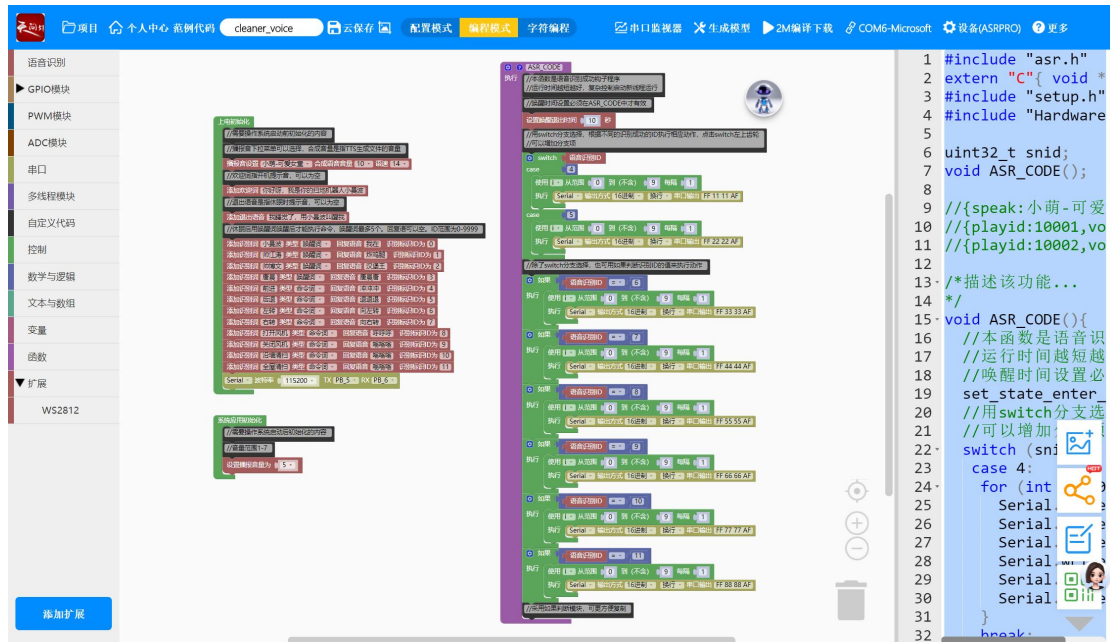


图 7.3.1.1 语音控制逻辑

7.3.2 快捷按键与自定义功能

快捷按键与自定义功能是我们面向普通用户便捷操作与开发者二次开发需求的核心扩展功能，依托微信小程序上位机的辅助开发页实现，指令配置完成后经 DT-06 WiFi 透传模块下发至主控，实现功能的灵活定制与快速调用。

快捷按键：针对家庭日常使用场景设计，支持用户将高频操作指令（如一键启动沿墙清扫、切换三挡清洁力度）绑定至小程序界面的自定义快捷按钮，大幅缩短操作链路；同时支持按钮图标、名称的个性化修改，适配不同用户的使用习惯。



图 7.3.2.1 快捷按键展示

自定义功能：为扫地机器人项目调试提供了灵活的扩展接口，用户可在小程序辅助开发页输入自定义指令的二进制/十六进制编码，选择功能图标，并设置触发方式（单击/长按），配置完成后的数据会存储于小程序页面，避免重复配置；该功能支持指令集的导出与导入，极大提升了系统的扩展性与适配性。



图 7.3.2.2 自定义功能区展示

7.3.3 自动全屋清扫

自动全屋清扫是系统的核心自动化功能，依托树莓派 5+N10 激光雷达构建的 SLAM 感知系统实现，全程无需人工干预，可高效完成家庭全区域清洁作业。在启动自动全屋清扫前，需先通过雷达完成 360°全视野环境建图，树莓派 5 会对采集的环境数据进行降噪、拼接与边界识别，生成包含房间轮廓、障碍物位置的高精度二维地图，并同步上传至微信小程序上位机。

用户在小程序界面确认地图后，点击“自动全屋清扫”按钮，指令经二进制协议编码下发至下位机，树莓派 5 主控会基于地图数据规划弓字形清扫路径，该路径规划逻辑可最大化覆盖清洁区域，避免重复清扫与漏扫，清扫覆盖率可达 95% 以上。

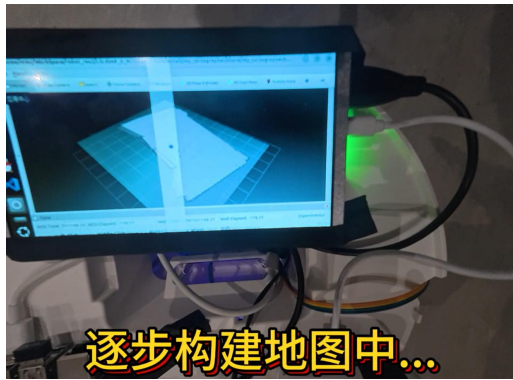


图 7.3.3.1 建图展示



图 7.3.3.2 全屋清扫

清扫过程中，系统会根据实时感知数据动态调整作业策略，通过碰撞传感器与雷达融合感知，实现障碍物的提前规避，遇到家具、墙角等障碍时自动绕行，绕行后重新规划路径回归原清扫轨迹；底部红外防跌落传感器实时监测高度差，杜绝跌落风险。

7.3.4 蜂鸣器奏乐

在本项目中，功能切换时的无源蜂鸣器奏乐功能是为提升用户交互体验而设计的。该功能能够在用户进行不同操作时，通过蜂鸣器音效提供实时反馈，增强操作的可感知性和操作确认感。

无源蜂鸣器是一种通过发出声音来进行提示或反馈的设备。与有源蜂鸣器不同，无源蜂鸣器本身并不自带产生声音的电路，而是依赖于外部驱动信号来产生声音。它通过驱动信号的频率和占空比来控制蜂鸣器的声音特征。通过调节 PWM 信号的频率和占空比，可以实现音高（频率）和音量（音强）的调节，从而播放不同的音效。

通过预先将音符的频率与乐理信息写入 STM32，并根据这些频率计算 PWM 的占空比和重载值，我们可以实现精确的音符控制，从而驱动无源蜂鸣器播放不同的音效。 驱动无源蜂鸣器，从而实现奏乐。

耀泽电子
YAOZEEDIANZI



图 7.3.4.1 无源蜂鸣器

```
60 // freq_hz = 0 表示静音
61 void buzzer_set_freq(uint32_t freq_hz)
62 {
63     if (freq_hz == 0)
64     {
65         buzzer_off();
66         return;
67     }
68     uint32_t timclk = buzzer_tim_clk_hz();
69     uint32_t arrpl = (BUZZER_ARR + 10);
70     // 计算 psc (做边界保护)
71     uint32_t psc_calc = timclk / (freq_hz * arrpl);
72     if (psc_calc == 0) psc_calc = 10;
73     psc_calc -= 10;
74     if (psc_calc > 0xFFFFF) psc_calc = 0xFFFFF;
75     uint16_t psc = (uint16_t)psc_calc;
76     uint16_t ccr = (uint16_t)(arrpl / 20); // 50% duty
77     buzzer_on(psc, ccr);
78 }
79
80 static void buzzer_play_tone(uint16_t freq, uint16_t ms)
81 {
82     buzzer_set_freq(freq);
83     osDelay(ms);
84     buzzer_off();
85 }
86
87 void buzzer_play_song(int song_list)
88 {
89     if (song_list == 1) {
90         // ... (code for playing a specific song)
91     }
92 }
```

图 7.3.4.2 蜂鸣器奏乐部分代码展示

8 总结与改进方向

8.1 项目总结

本项目完成了从器件选型、原理图与 PCB 设计、硬件焊接装配，到嵌入式分层软件、上位机 ROS 建图导航、APP 端 UI 与通信协议的全流程实现，并最终在整机上实现了“手动控制 + 建图导航 + 清扫相关策略”的综合功能闭环。

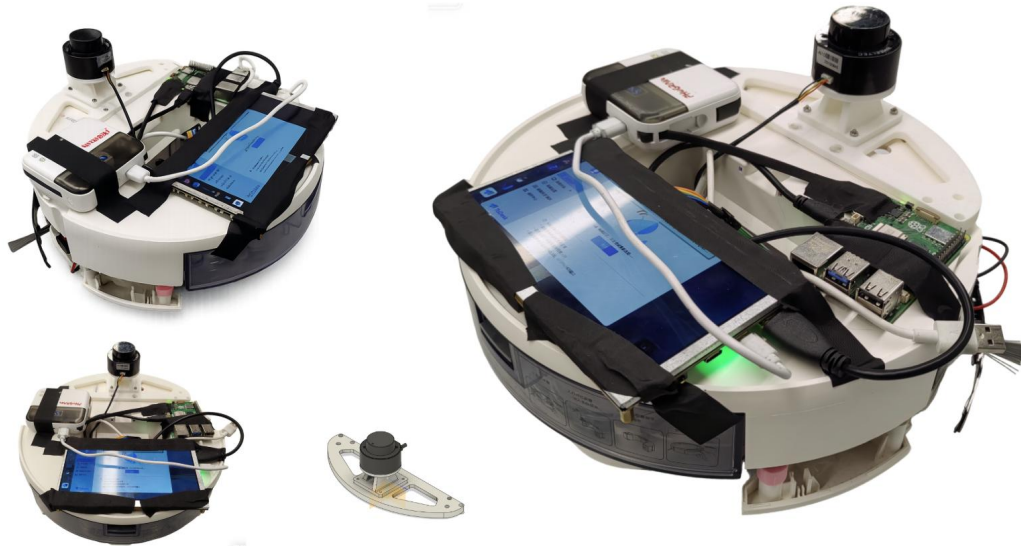
8.2 后续改进建议

1. **电源与 EMI 进一步优化:** 对大电流回路、降压芯片热设计与地回流进行更细致的仿真/测试闭环，提升可靠性。
2. **运动控制与定位精度提升:** 在编码器与 IMU 融合方面可进一步优化滤波与参数标定，提升 AMCL 定位稳定性与导航成功率。
3. **APP 控制抗干扰性:** 补充丢包/重传、版本号与校验机制，提升无线环境下的可用性。
4. **清扫策略完善:** 在沿墙、覆盖规划、异常处理等策略上可进一步扩展，使功能更接近真实产品级逻辑。

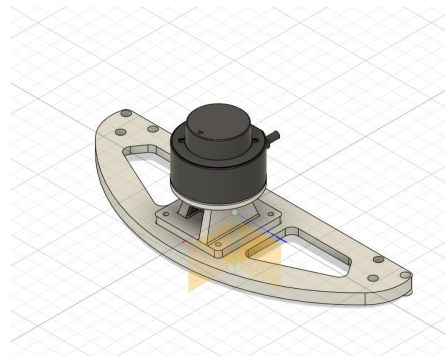
5. **数据化测试:** 引入更多的性能参数对测试结果进行量化分析、精准验证与客观评估。

附：

实物多视图：



雷达支架建模：



b 站链接：

[扫地机器人 Remake! 重庆大学 23 级明月班微电路设计课程项目~_哔哩哔哩_bilibili](#)