

摘 要

本设计基于嵌入式系统开发了一款多功能机械臂控制系统，旨在实现机械臂的正逆解运动控制，示教及无线控制功能。系统以 STM32F103C8T6 微控制器为核心，结合步进电机驱动模块，微动开关和蓝牙模块，完成了硬件电路设计、Matlab 仿真、Protues 仿真及嵌入式软件编程。

在硬件设计上，采用 ZDT_Y42_V2.0 步进驱动板控制机械臂关节运动，反馈层采用工业级 AMR 磁编码器 MT6816，凭借其 $\pm 0.1^{\circ}$ 的角度分辨率与串口通信协议，构建起闭环控制回路，确保机械臂末端位置的动态校准，并利用蓝牙模块与 PC 端或手机端进行数据传输。软件层面基于 STM32 库函数开发控制程序，实现了机械臂逆解算法和多点示教功能，示教功能模块支持 5 组点位数据存储，利用梯形加减速算法对运动轨迹进行优化，将速度波动误差控制在 5% 以内，确保机械臂运动的平滑性。

测试结果表明，该系统能够稳定完成逆运动学解、多点示教功能以及旋转等基础动作，定位精度较高，相应延迟低。本项目验证了嵌入式系统在机电控制领域的应用可行性。为后续扩展视觉识别或互联网功能奠定了基础。

关键词：嵌入式系统；机械臂控制；STM32；多点示教

目 录

第一章 概述	1
1.1 设计的目的及意义	1
1.2 设计的主要内容	1
1.3 设计的实践安排	2
第二章 硬件电路设计	3
2.1 原理图设计	3
2.1.1 电源电路	3
2.1.2 低压检测与蜂鸣器电路	5
2.1.3 外设电路	6
2.2 PCB 设计	6
第三章 Matlab 与 Protues 仿真	9
3.1 Matlab 仿真	9
3.2 Protues 仿真	11
3.2.1 ADC 模块仿真	11
3.2.2 蜂鸣器模块仿真	12
3.2.3 按键与 OLED 模块仿真	12
第四章 实物搭建与运行	14
4.1 机械部分	14
4.2 控制部分	15
4.2.1 多点示教功能	16
4.2.2 蓝牙控制运动学逆解功能	18
第五章 结论与心得体会	20
5.1 结论	20
5.2 心得体会	20
附 录 A:基于 Matlab 的机械臂建模程序	21
附 录 B:基于 Matlab 的机械臂逆解程序	22

第一章 概述

1.1 设计的目的及意义

本设计以 STM32F103C8T6 高性能微控制器作为核心控制单元，致力于开发一套功能完备、性能可靠的机械臂示教与逆运动学演示系统。该系统的设计融合了先进的硬件电路架构与前沿的运动学算法，通过精心设计的主控板硬件电路，协同编码器、电机驱动模块，结合 Denavit - Hartenberg (DH) 参数法构建机械臂运动学模型，并运用代数逆解法实现复杂的运动学计算。其核心目标在于实现人工示教数据的高效存储与精准再现，同时对逆运动学解算的实时性进行严格验证，最终构建一个集成度高、可扩展性强的嵌入式控制平台，为机器人运动学相关教学与研究提供极具价值的实践案例。

在实际应用层面，该系统将抽象的理论算法成功转化为嵌入式实时控制，具有显著的双重价值。一方面，它能够帮助学生深入理解机械臂运动学与嵌入式开发之间的内在关联性，通过理论知识与实际操作的紧密结合，有效提升学生的实践能力与创新思维；另一方面，为工业机械臂的轻量化控制提供了重要的参考依据，助力工业领域在保证机械臂性能的前提下，实现结构优化与成本降低。此外，系统所具备的示教功能，能够让使用者通过直观的操作，深刻理解机械臂的运动过程；而逆解算法的验证过程，则有助于提升机械臂的控制精度，充分体现了“理论 - 算法 - 工程”全流程实践的重要价值，为相关领域的人才培养与技术发展奠定坚实基础。

1.2 设计的主要内容

本设计基于模块化设计理念，将系统划分为示教功能模块、逆运动学解算模块以及人机交互模块三大核心模块，各模块分工明确且相互协作，共同实现系统的整体功能。

1. 示教模块

示教模块作为系统实现人机协同操作的关键环节，采用直观便捷的操作方式。使用者可通过手动拖动机械臂，将其精准移动至各个关键点位置，随后按下特定按键，系统将迅速捕捉并存储该关键点的位置信息。这一过程借助高精度的传感器与高效的数据采集算法，确保所存储的示教数据具有极高的准确性与可靠性，为后续的运动再现提供精确的数据支持。

2. 逆运动学解算模块

逆运动学解算模块是系统的核心计算单元，其实现过程涉及多学科知识的综合运用。首先，利用 Matlab 强大的机器人工具箱，基于机械臂的结构参数与运动特性，建立精确的机械臂运动学模型。在此基础上，通过算法随机生成若干组

位于机械臂关节运动范围内的关节角度数据，运用 DH 参数法进行正运动学求解，从而获取机械臂末端的三维坐标。接着，依据所得到的大量机械臂末端三维坐标数据，通过空间分析算法推算出机械臂的运行空间，并进一步反推出满足该运行空间要求的连杆长度参数。最后，结合 DH 参数法与代数法，经过严密的数学推导与计算，得出适用于该机械臂的逆运动学解的方程，为机械臂的精确控制提供理论依据与算法支持。

3. 人机交互模块

人机交互模块作为用户与系统进行信息交互的桥梁，由独立按键、OLED 显示屏以及蜂鸣器等硬件电路组成，并通过精心设计的软件程序实现高效交互功能。其中，独立按键承担着模式切换与操作控制的重要任务，用户可通过按键轻松实现示教模式与逆运动学模式的切换，以及在示教模式下对操作的开始、停止与关键点记录等操作；OLED 显示屏则作为信息显示的窗口，实时、清晰地显示当前系统运行模式、示教模式的启动 / 停止状态、已存储的关键点个数，以及在逆运动学模式下系统的正常运行或错误状态等关键信息，使用户能够直观了解系统运行情况；蜂鸣器则在逆运动学模式下发挥重要的预警作用，当输入的坐标点超出机械臂的运行空间时，蜂鸣器将立即发出警报，提醒用户及时调整操作，确保系统运行的安全性与稳定性。

1.3 设计的实践安排

第一周周一：课程设计的选题与准备工作

第一周周二到周四：设计硬件电路并提交设计方案，绘制电路原理图与 PCB 图，电气特性检查，生成元器件清单列表。

第一周周五与第二周周一：进行仿真工作，可利用 Protues 或 Matlab 对自身选题的仿真。

第二周周二到周四：实物的搭建与运行，撰写课程设计报告。

第二周周五：答辩并回答问题。

第二章 硬件电路设计

2.1 原理图设计

系本设计使用步进电机控制机械臂关节，单个步进电机驱动板所需电流为 3-4A，一共四个关节，加上其余用电设备总电流为 18A 左右，所以选用容量为 3300mah 的 4s 航模锂电池，放电倍率为 25C，最大放电能力为 82.5A，额定放电能力为 30A 左右。

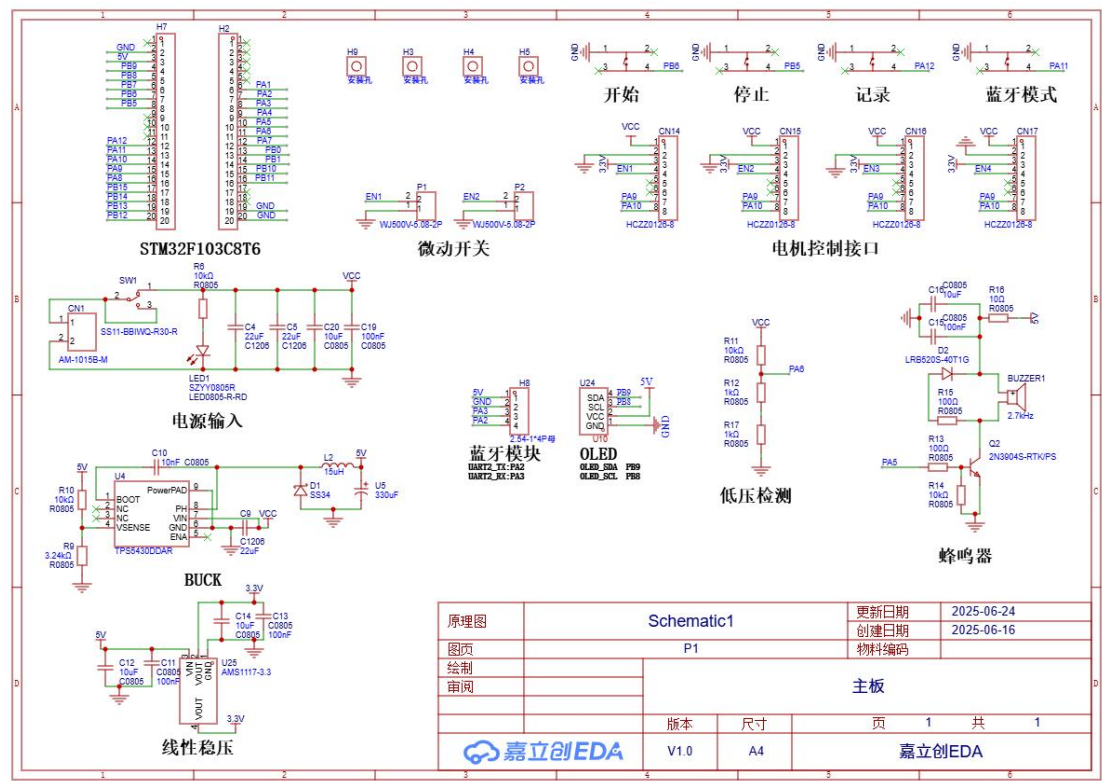


图 2-1 总原理图

总原理图主要由 STM32F103C8T6 最小系统板的 2x20p 排母接口,电源电路,4 路步进电机控制接口,蓝牙模块接口,OLED 显示屏接口,微动开关接口,低压检测电路,蜂鸣器电路,4 个安装孔和 4 个独立按键组成。

2.1.1 电源电路

电源电路主要由电源输入电路,BUCK 降压电路以及线性稳压电路三大部分组成。

1.首先是电源输入电路,如图 2-2 所示,电源接口端子选用的是 AM-1015 型号的公头 T 插,相对于 XT60,T 插更紧不容易松动。电源指示灯选用的是 0805 的 LED,限流电阻为 10K 欧姆,选用两个 22uf 的电容并联主要起储能作用,加上 10uf 与 100nf 的 0805 的电容用来滤除不同频段的噪声,以此来为后级电路提供更稳定的电压。

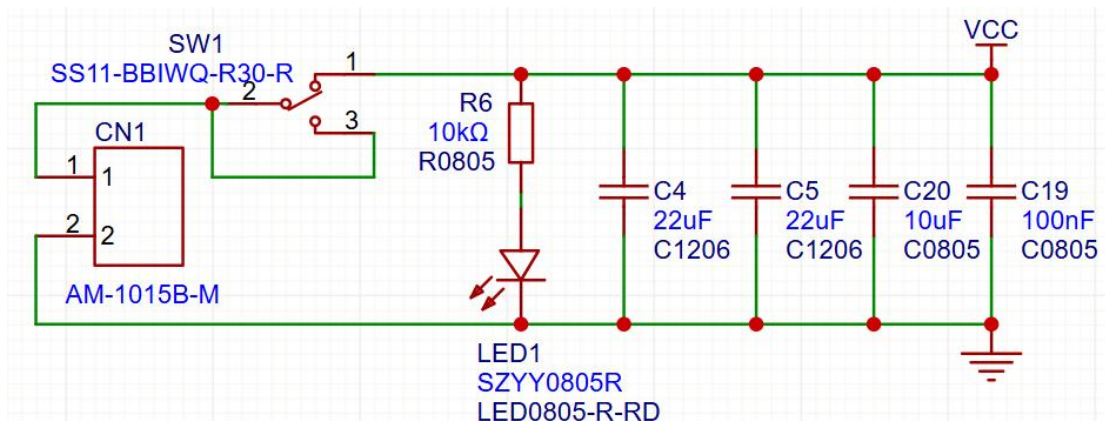


图 2-2 电源输入电路图

2.其次是 BUCK 降压电路,如图 2-3 所示,BUCK 芯片为 TI 公司的 TPS5430。

$$R9 = \frac{R10 \times 1.221}{V_{out} - 1.221} \quad (2-1)$$

TI 推荐 R10 取 10K 欧姆, V_{out} 为 5V, 通过式 (2-1) 可以算出 R9 约为 3.24K 欧姆。

$$L_{min} = \frac{V_{out(max)} \times (V_{in(max)} - V_{out})}{V_{in(max)} \times K_{ind} \times I_{out} \times f_{sw}} \quad (2-2)$$

对于 L2 的取值, , TPS5430 的 f_{sw} 为 500KHz, I_{out} 为 3A, K_{ind} 是表示电感纹波电流相对于最大输出电流的量的系数, TI 建议 K_{ind} 选 0.2-0.3 范围内, 这里选 0.25, 4s 的航模电池最大输出电压为 16.8V, 最后算出 L_{min} 约为 10uH, 这里选择 15uH。

对于输出电容 ($U5$) 的取值要考虑 ESR 对输出电压的影响 (输出纹波)。

$$C_{out} = \frac{1}{3357 \times L_{out} \times f_{co} \times V_{out}} \quad (2-3)$$

$$ESR_{max} = \frac{1}{2 \times \pi \times C_{out} \times f_{co}} \quad (2-4)$$

$$V_{pp(max)} = \frac{ESR_{max} \times V_{out} \times (V_{in(max)} - V_{out})}{N_c \times V_{in(max)} \times L_{out} \times f_{sw}} \quad (2-5)$$

$$f_{co} = \frac{f_{lc} \times f_{lc}}{85 \times V_{out}} \quad (2-6)$$

$$f_{lc} = \frac{1}{2 \times \pi \times \sqrt{L_c}} \quad (2-7)$$

$V_{pp(max)}$ 为期望的峰对峰输出波纹, 这里选择 0.1V, N_c 为并联输出电容的个数, 此处为 1, 通过式 (2-5) 算出 ESR_{max} 约为 0.213, 对于闭环交叉频率 f_{co} 由于内部补偿的设计, 希望将闭环交叉频率保持在 3khz 至 30khz 范围内, 因为该频率范围具有足够的相位升压以允许稳定运行, 这里选 12khz。再通过式 (2-4) 算出 C_{out} 约为 330uf。

对于二极管的选择, 因为 TPS5430 在额定输出电压为 5V 时可以提供高达 3A 的输出电流且开关频率高, 选择可通过 3A 电流的肖特基二极管 SS34。

最后, TI 规定用于高侧场效应管栅极驱动的 Boost 电容是将 0.01uf 低 ESR 电容从 BOOT 引脚连接到 PH 引脚。

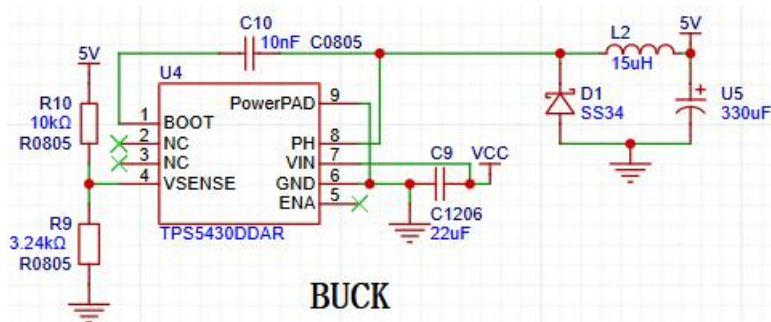


图 2-3 BUCK 降压电路图

3.最后是线性稳压电路，如图 2-4 所示，选用 AMS1117，将 5V 降为 3.3V，用于电机控制板的 Com 端，输入输出的 10uf 以及 100nf 的电容器用于滤波和储能。

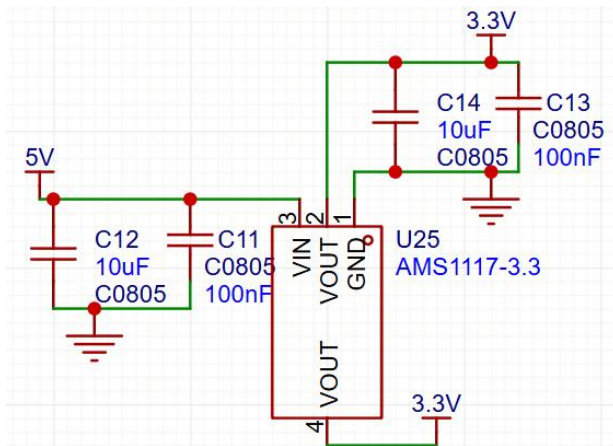


图 2-4 线性稳压电路

2.1.2 低压检测与蜂鸣器电路

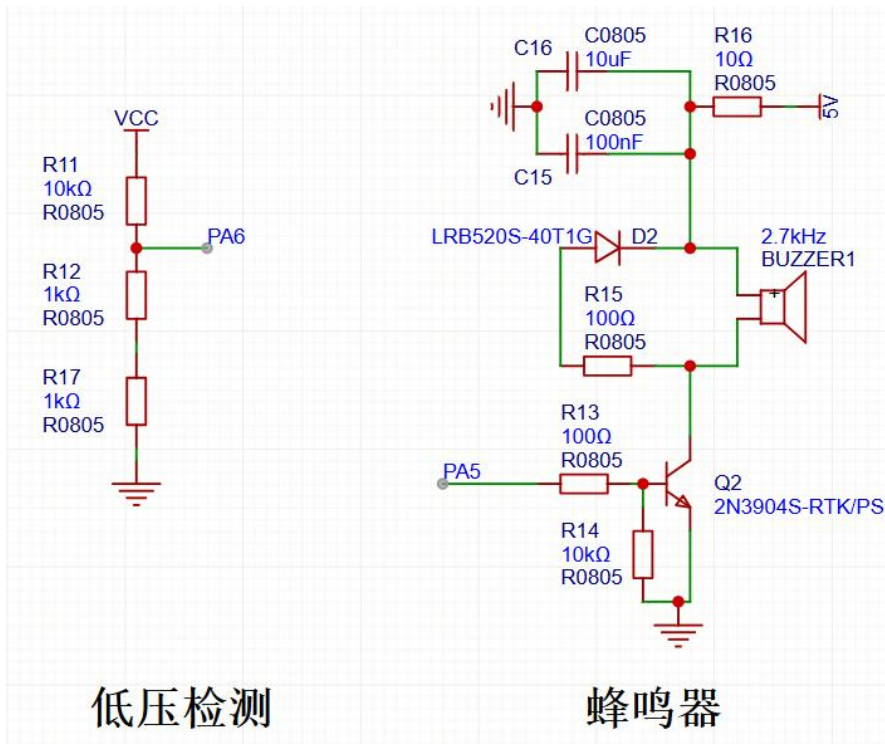


图 2-5 低压检测与蜂鸣器

航模电池的每节电芯满电电压为 4.2V, 3.6-3.7V 时电芯电量基本放空, 综上, 4S 航模电池的电压正常使用范围为 14.8-16.8V, 为防止航模电池过放, 设计一低压检测电路采集电源电压再配合蜂鸣器电路做出警报处理。

低压检测电路, 由于 stm32f103c8t6 内部集成了 ADC 电路的引脚电压耐受最高为 3.3V 左右, 即选用电阻分压的原理来降低电压采集, 从图 2-5 中低压检测电路可知采集电压范围为 2.47-2.8V, 通过欧姆定律可以算出 R11 上的最大功率为 19.6mW, R12 和 R17 都为 1.96mW, 一般 0805 封装的电阻最大功率为 125mW, 即合理设计。

蜂鸣器电路, 选用频率为 2.7kHz 的有源蜂鸣器, 通过 NPN 三极管控制该有源蜂鸣器的开关, R13 为限流电阻。R14 为下拉电阻, 该电阻在三极管关断时有利集电极电压的释放。D2 与 R15 共同作用于蜂鸣器关断时残余电量的释放, C15 与 C16 用于电源滤波。

2.1.3 外设电路

该电路提供 stm32 与蓝牙模块串口通信与电源的接口, stm32 与 OLED 显示屏 IIC 通信与电源的接口, stm32 与电机控制板串口通信、碰撞检测与电源的接口, 电机控制板与微动开关的接口以及 stm32 与 4 个独立按键的接口。

2.2 PCB 设计

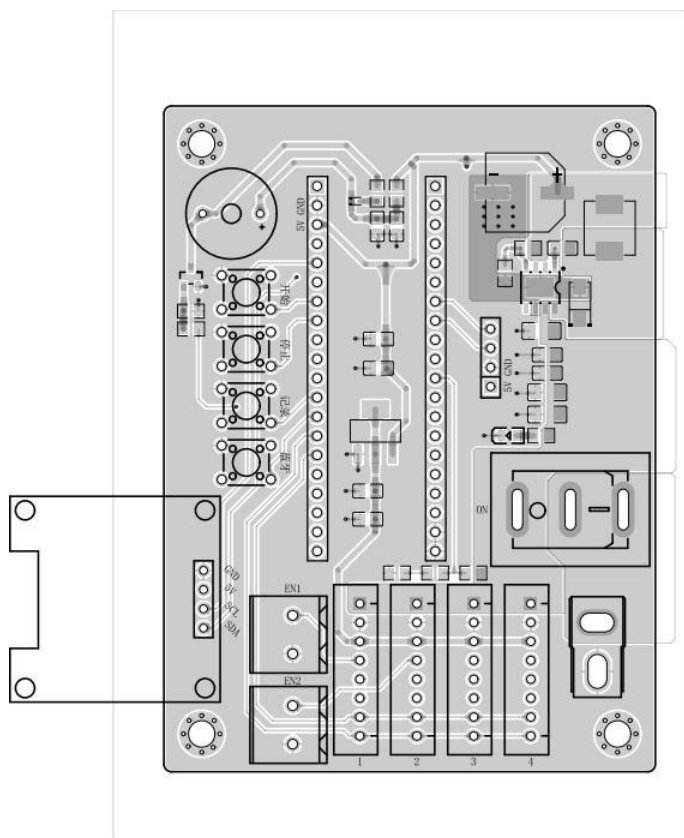


图 2-6 PCB 图

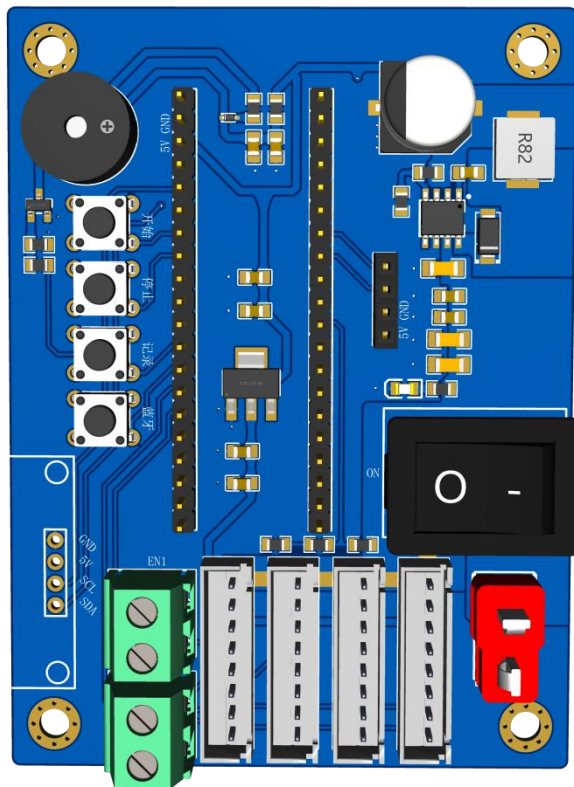


图 2-7 3D 图

硬件电路的除 stm32f103c8t6 其余元件都分布在 PCB 板的正面，右下角红色元件为公投 T 插，中下部白色元件 8pin 端子为电机控制接口，左下角绿色元件为微动开关（碰撞开关）接口，左部为独立按键，中部为线性稳压电路，右部为电源指示灯电路，右上部为 BUCK 降压电路，左上部为蜂鸣器电路。

对于 BUCK 电路，在 TPS5430 内 CMOS 开启时如图 2-7 所示，电流从输入电容-TPS5430 的 CMOS-电感-贴片电解电容-外围用电设备，图 2-7 的设计形成了一个环路大大减小了外部的干扰。

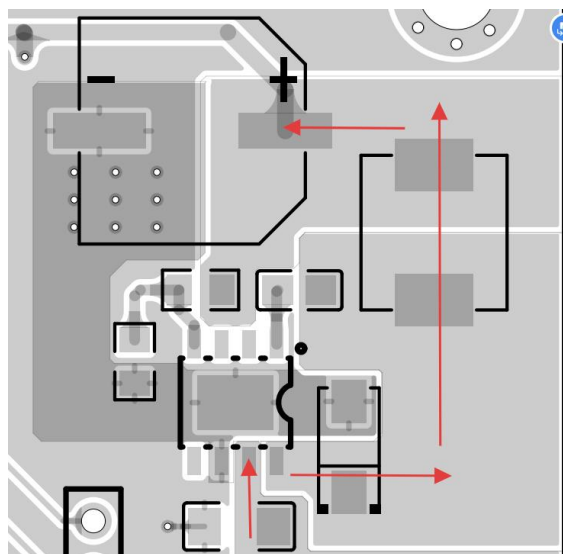


图 2-7 CMOS 开启电流走势图

在 TPS5430 内 CMOS 关闭时如图 2-8 所示，电流会从电感出发-贴片电解电容-GND-肖特基二极管 SS34-回到电感，图 2-8 的设计同样形成了一个环路也会减小了外部的干扰。

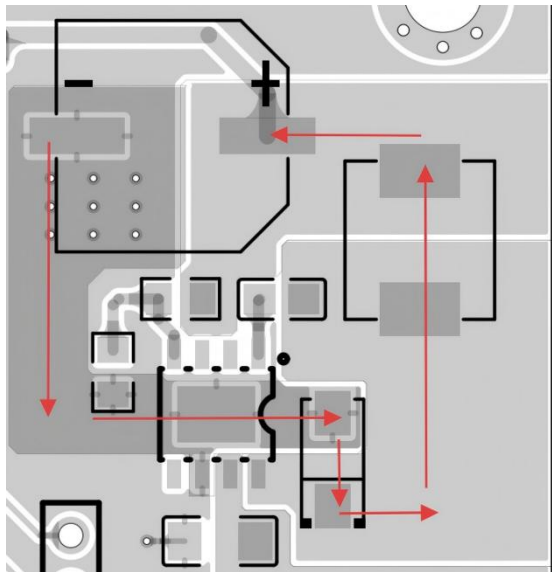


图 2-8 CMOS 关闭电流走势图

在给 4 个电机控制板供电的支路上，采用了开窗与铺铜这样可以更好的散热，在大电流通过时保证电路的稳定性。最后进行整体正反面铺地铜以及泪滴处理，可以提高线路以及 PCB 板的机械抗性。

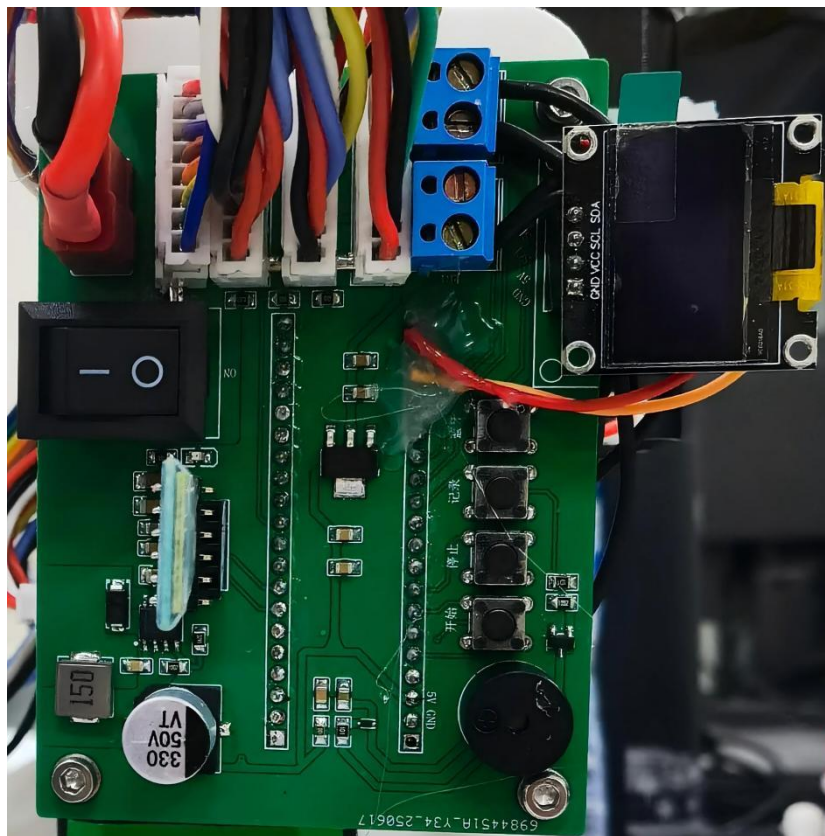


图 2-9 实物图

第三章 Matlab 与 Protues 仿真

在仿真部分，利用 Matlab 做机械臂仿真与 Protues 做传感器仿真，这样可以更直观且更严谨的体现本设计系统运作的原理。

3.1 Matlab 仿真

使用 Matlab 的第三方库（机器人工具箱）做机械臂的仿真。

1. 首先进行机械臂的建模，具体代码见附录 A，首先使用机器人工具箱的 Link 函数构建机械臂的 4 个关节并用 qlim 函数来定义该关节的可运行角度范围，将四个关节的 DH 参数写入 Link 函数，然后利用 SerialLink 函数将 4 个关节组合起来，SerialLink 函数会返回建好的机械臂模型 p560，可以使用 teach 函数来进行示教如图 3-1 所示。

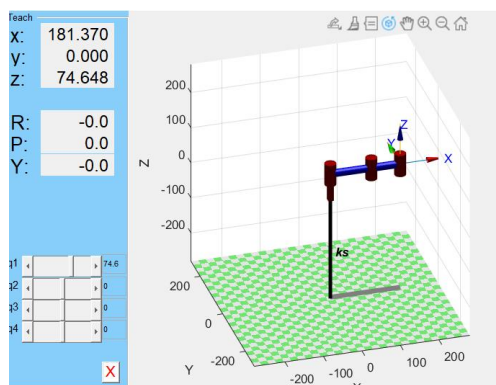


图 3-1 teach 图

2. 接着对机械臂的工作空间进行模拟，先使用 zeros 函数创建一个 N 行 3 列的矩阵，每一行的三个元素用来存储三维坐标(x,y,z)，通过一个 for 循环对四个机械臂关节角度随机取值操作循环 N 次，随机取值主要通过 rand(1, p560.n)实现，最后通过 scatter3 函数将所有坐标点显示出来，N 的取值越大点就会更密集，最后显示出来的工作空间的轮廓就越清晰。

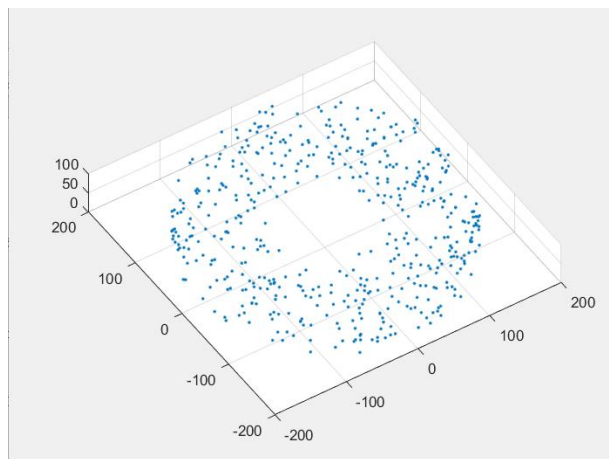


图 3-2 工作空间 500 点图

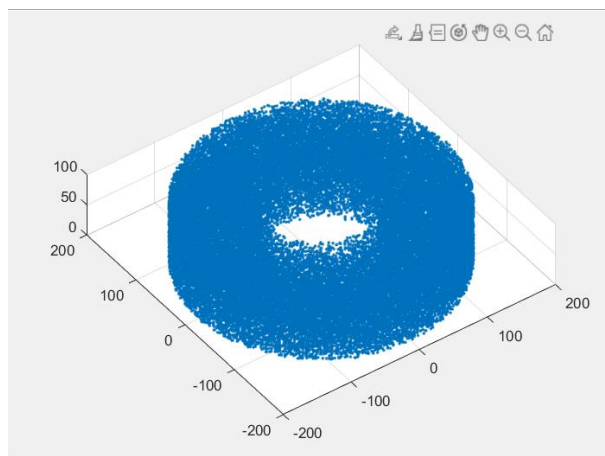


图 3-3 工作空间 50000 点图

比较图 3-2 与图 3-3 可以得出，N 的取值越大，最后机械臂的工作空间轮廓就更清晰。

3.最后对机械臂求逆解，具体代码见附录 B，见图 3-1 该机械臂的第一个关节为 z 轴方向上的平移关节，关节 2，关节 3，关节四都为 xy 平面上的摆动关节，所以 $d_0=z$ 。

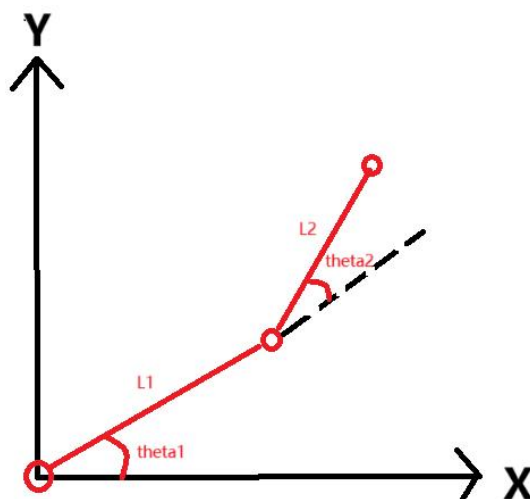


图 3-4 2 关节 3 关节 4 关节结构简图

如图 3-4， θ_1 为关节 2 的旋转角度， θ_2 为关节 3 的旋转角度，在 xy 坐标系下可得到如下公式：

$$x = L_1 \times \cos(\theta_1) + L_2 \times \cos(\theta_1 + \theta_2) \quad (3-1)$$

$$y = L_1 \times \sin(\theta_1) + L_2 \times \sin(\theta_1 + \theta_2) \quad (3-2)$$

对式(3-1)和式(3-2)平方后求和得：

$$x^2 + y^2 = L_1^2 + L_2^2 + 2 \times L_1 \times L_2 \times \cos(\theta_2) \quad (3-3)$$

因为 L_1 和 L_2 不为零，所以有：

$$\cos(\theta_2) = \frac{x^2 + y^2 - (L_1^2 + L_2^2)}{2 \times L_1 \times L_2} \quad (3-4)$$

上述方程有解的条件是，式(3-4)右边等式的取值范围为 $[-1,1]$ ，如果约束条件不满足，那么说明该点位于工作空间外。

如果在工作空间内：

$$\sin(\theta_2) = \pm \sqrt{1 - (\cos \theta_2)^2} \quad (3-5)$$

利用双参数反正切函数 atan 可以得到 θ_2 ：

$$\theta_2 = \text{atan2}(\sin \theta_2, \cos \theta_2) \quad (3-6)$$

这里使用 atan2 是因为其值域为 $[-\pi, \pi]$ ，且误差更小。

求取 θ_1 ，将式(3-1)与(3-2)拆分为如下四式：

$$x = k_1 \times \cos \theta_1 - k_2 \times \sin \theta_1 \quad (3-7)$$

$$y = k_1 \times \sin \theta_1 + k_2 \times \cos \theta_1 \quad (3-8)$$

$$k_1 = L_1 + L_2 \times \cos \theta_2 \quad (3-9)$$

$$k_2 = L_2 \times \sin \theta_2 \quad (3-10)$$

$$\text{设 } r = \sqrt{(k_1^2 + k_2^2)} \quad (3-11)$$

$$\gamma = \text{atan2}(k_2, k_1) \quad (3-12)$$

通过式(3-11)和式(3-12)带入式(3-7)和式(3-8)可得：

$$\frac{x}{r} = \cos \gamma \cos \theta_1 - \sin \gamma \sin \theta_1 \quad (3-13)$$

$$\frac{y}{r} = \cos \gamma \sin \theta_1 + \sin \gamma \cos \theta_1 \quad (3-14)$$

化简式(3-13)和式(3-14)可得：

$$\frac{x}{r} = \cos(\gamma + \theta_1) \quad (3-13)$$

$$\frac{y}{r} = \sin(\gamma + \theta_1) \quad (3-14)$$

通过式(3-14)/(3-13)可得：

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(k_2, k_1) \quad (3-15)$$

θ_3 的取值的多少决定了机械臂末端执行器的姿态，如果希望末端执行器姿态始终保持水平则：

$$\theta_3 = -(\theta_1 + \theta_2) \quad (3-16)$$

3.2 Protues 仿真

Protues 仿真主要针对于本设计系统的蜂鸣器，OLED 显示屏，按键以及 ADC 模块的仿真。

3.2.1 ADC 模块仿真

Stm32 的 ADC 采集功能在 protues 中使用 8 位逐次逼近 A/D 转换的 PCF8951 芯片采集滑动变阻器的滑动端电压来模拟。

由于 PCF8951 使用 IIC 通信，在仿真中使用 PA12 与 PA13 引脚进行软件模拟 IIC 通信，其三个地址引脚 AO, A1 和 A2 用于编程硬件地址，本仿真只使用了一个 PCF8951 所以该三引脚全部接地处理，未使用到其模拟输出功能即 AOUT

引脚悬空，AIN0 端接滑动变阻器的滑动端，其模拟电压范围为 AGND 到 VREF，由于本仿真电源电压范围为 0-5V，即 AGND 引脚接 GND，VREF 引脚接 VCC，详情见图 3-5。

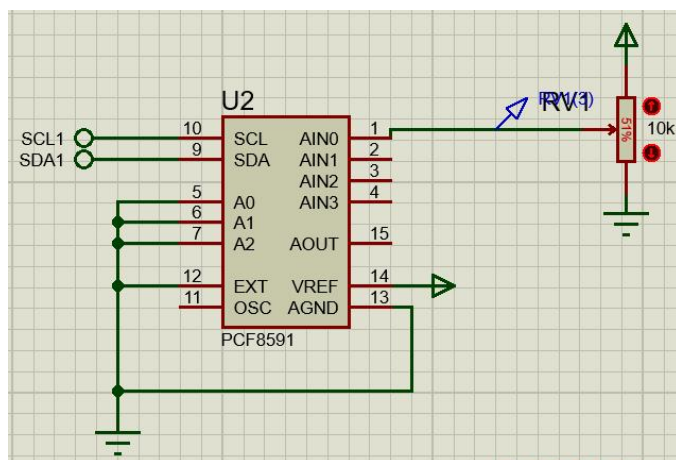


图 3-5 PCF8951 仿真图

3.2.2 蜂鸣器模块仿真

蜂鸣器使用的是 protues 中的有源蜂鸣器，其开关控制使用的是 NPN 型三极管，通过 stm32 的 PA15 引脚的推挽输出模式接到 NPN 三极管的基极来控制三极管的通断，R6(10K)为下拉电阻，R5(1K)为限流电阻，并在蜂鸣器两端并联一个 LED 用来指示蜂鸣器的通断情况，10R 电阻用来限流防止 LED 支路上的电流过大损坏 LED，详情见图 3-6。

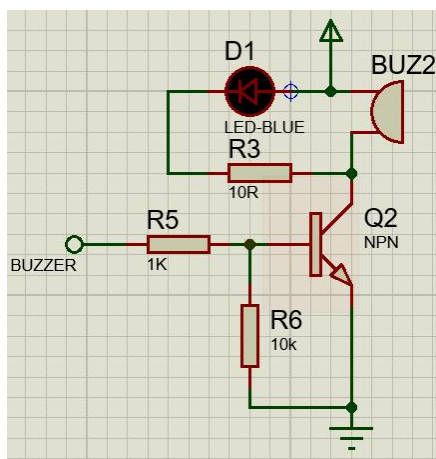
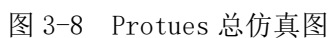
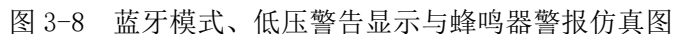
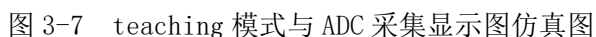


图 3-6 蜂鸣器仿真图

3.2.3 按键与 OLED 模块仿真

本设计运行模式显示与状态显示使用的是 0.96 寸有机电激光显示屏(OLED)，其分辨率为 128x64，通信方式为 IIC 通信，本设计使用 stm32 的 PB14 与 PB15 引脚模拟 IIC 通信与 OLED 通信。

在待机状态下 OLED 会一直显示从 PCF8951 采集获取的 AD 值，当滑动其至低于设定的阈值时，OLED 显示屏会显示 LowVoltage 字样并且蜂鸣器会发出



第四章 实物搭建与运行

4.1 机械部分

机械臂一共有四个自由度。

第一个自由度为 Z 轴方向上的平移运动（上升与下降），由 42 步进电机轴上的齿轮带动丝杆上的齿轮使丝杆上的螺母做上下移动，如图 4-1 所示。

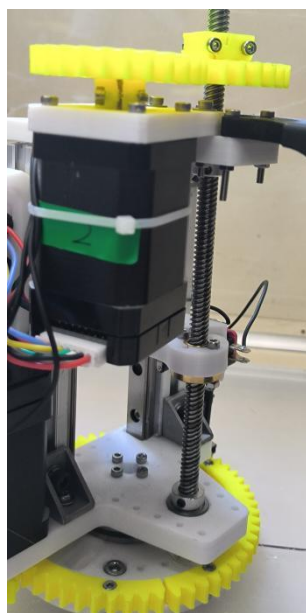


图 4-1 第一自由度实物结构图

第二个自由度为 Z 轴方向上的旋转运动，基座中心安装有轴承，轴承外圈安装有 3D 打印的齿轮都在固定基座上，轴承内圈上安装了旋转基座，由 42 步进电机轴上的小齿轮与固定基座外围的齿轮啮合，电机启动后，会使旋转基座绕 z 轴做旋转运动，如图 4-2 所示。

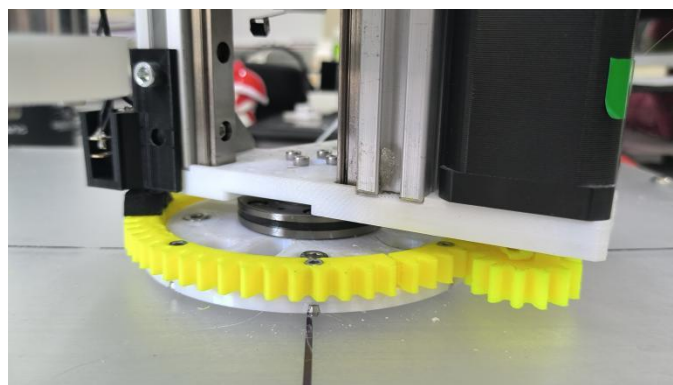


图 4-2 第二自由度实物结构图

第三个自由度仍为 Z 轴方向上的旋转运动，驱动该关节的电机固定在中臂上，中臂一端固定在丝杆螺母座上，一端固定在微小型滚珠直线导轨上，控制该自由度的电机轴上装有抱紧式联轴器，抱紧式联轴器上安装有小臂，如图 4-3 所示。

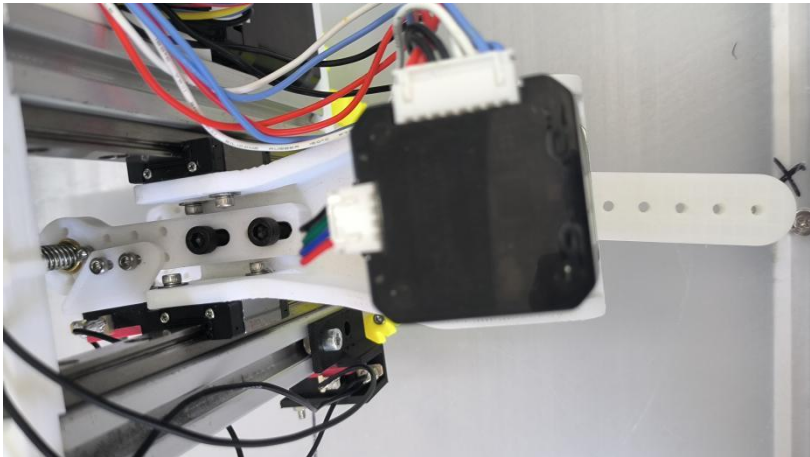


图 4-3 第三自由度实物结构图

第四个自由度同样是 Z 轴方向上的旋转运动，主要作用是控制末端执行器的姿态。

丝杆螺母座上安装有碰撞开关的固定件用于安装碰撞开关，安装有导轨的 2020 铝型材上也装有碰撞开关的固定件也用于安装碰撞开关。丝杆螺母座上的碰撞开关用于机械臂轴一的回零，铝型材上碰撞开关用于机械臂轴二的回零。

下图为机械臂的装配图。

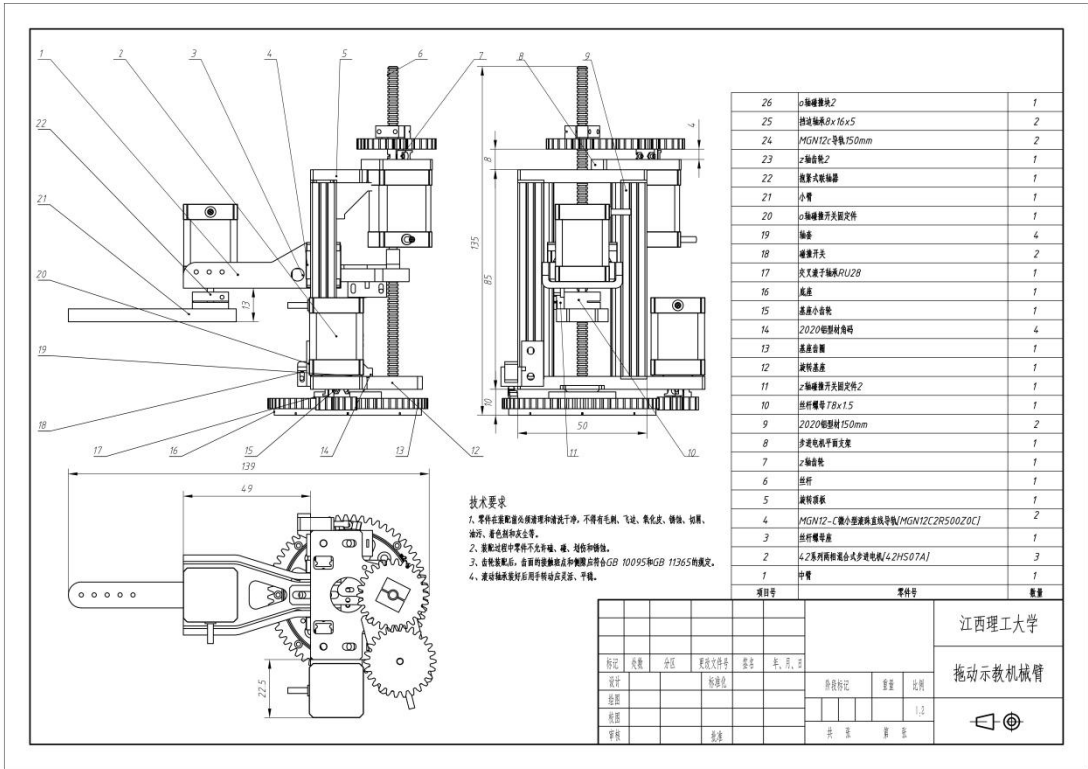


图 4-4 机械臂装配图

4.2 控制部分

本设计的控制系统主要分为多点示教功能与蓝牙控制运动学逆解功能两大部分。

4.2.1 多点示教功能

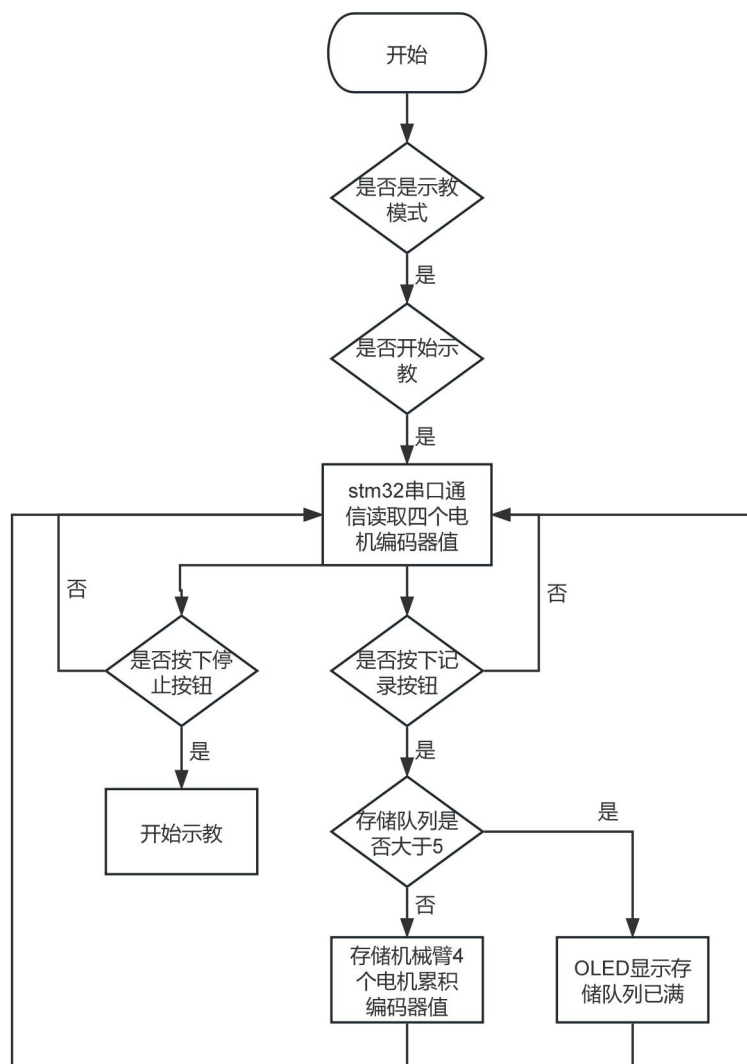


图 4-5 多点示教流程图

在系统上电后，机械臂自动回到零点并且系统处于待机状态，所有步进电机处于自锁状态，等待示教模式与逆运动学模式的选择，详见图 4-6。

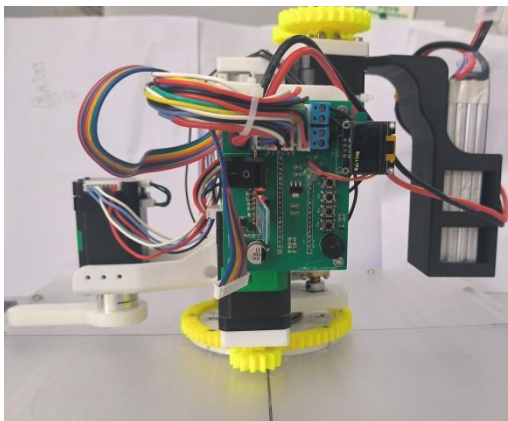


图 4-6 机械臂上电后初始状态图

按下开始按键（具体见图 2-1），进入示教模式，系统此时处于准备状态，OLED 显示 `teachingMode`，此时电机仍处于自锁状态，详见图 4-7。

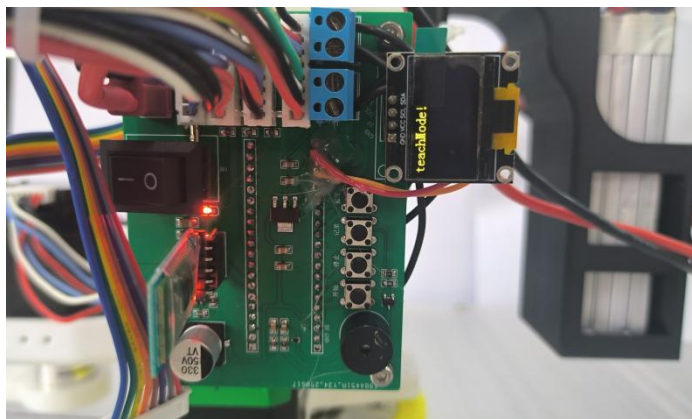


图 4-7 机械臂开始示教状态图

再次按下开始按键，OLED 显示 `start` 提示此时所有电机解除自锁状态可以手动拖动电机，并且此时 `stm32` 通过串口通信读取四个电机编码器值。

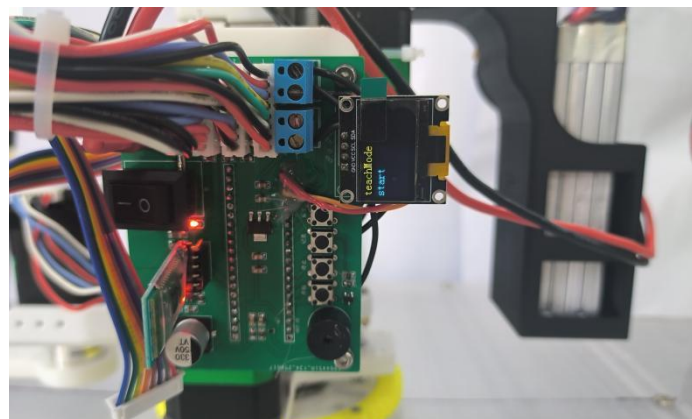


图 4-8 机械臂可拖动状态图

按下记录按键 `stm32` 会自动存储此时每个电机的累积编码器值，最多存储 5 次，如果按下停止按键，机械臂会自动回零开始示教。

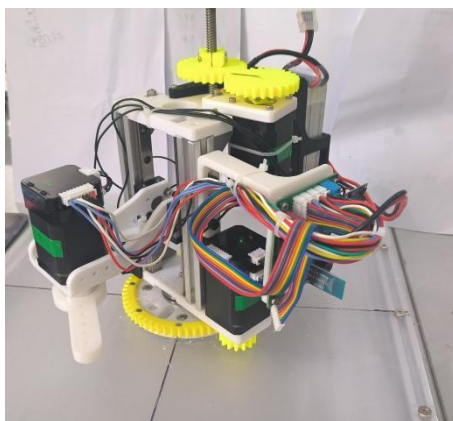


图 4-9 机械臂示教结束状态图

在按下记录按钮时，OLED 显示屏会显示已存储累积编码器值的次数。在开始示教时，`stm32` 会从存储队列中依次取出累积编码器值，得到起始点

和终止点，中间通过阶梯型加减速平滑过渡。

4.2.2 蓝牙控制运动学逆解功能

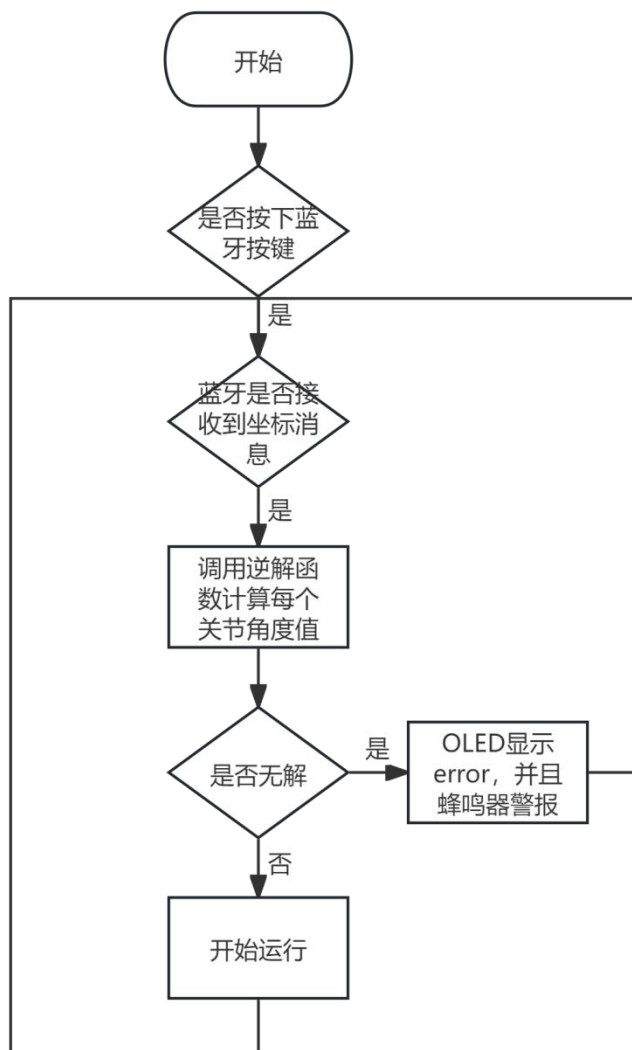


图 4-10 蓝牙控制运动学逆解流程图

在系统上电后，机械臂自动回到零点并且系统处于待机状态，所有步进电机处于自锁状态，OLED 显示屏显示 Waiting 等待示教模式与逆运动学模式的选择。详见图 4-6。

按下蓝牙按键（具体见图 2-1），进入蓝牙模式，OLED 显示 BTMode。

此时通过手机蓝牙发送三维坐标信息，蓝牙模块接受到后通过串口通信发送给 stm32，stm32 接受到三维位置信息后，调用逆解函数（具体内容见 3.1matlab 仿真的逆解部分），如果存在多解，首先判断得出的每个关节的角度值是否在该关节的角度活动范围内，再筛选出与当前关节角度相差值最小的解，最后将该解对应的每个关节角度值转换为步进电机的脉冲值，再通过梯型加减速来过渡。

发送的消息格式为：首先会发送三个字节的固定包头 0x41 0x42 0x43 用于每个消息帧的包头校验，第四个字节为 x 轴坐标的绝对值的十六进制，第五个字节为 x 轴坐标的符号（为零时 x 为正，为一是 x 为负），第六个字节为 y 轴坐标的绝对值的十六进制，第七个字节为 y 轴坐标的符号（为零时 y 为正，为一是 y 为负），第七个字节为 z 轴坐标的十六进制，无 z 轴坐标的符号因为 z 为 0 时就是地面/基座，最后一个是固定的包尾信息 0x5A，用于每个消息帧的包尾校验。

如图 4-11 所示发送的三维坐标为(64,-76,81)，实物结果图见图 4-12。



图 4-11 蓝牙控制手机 APP 图

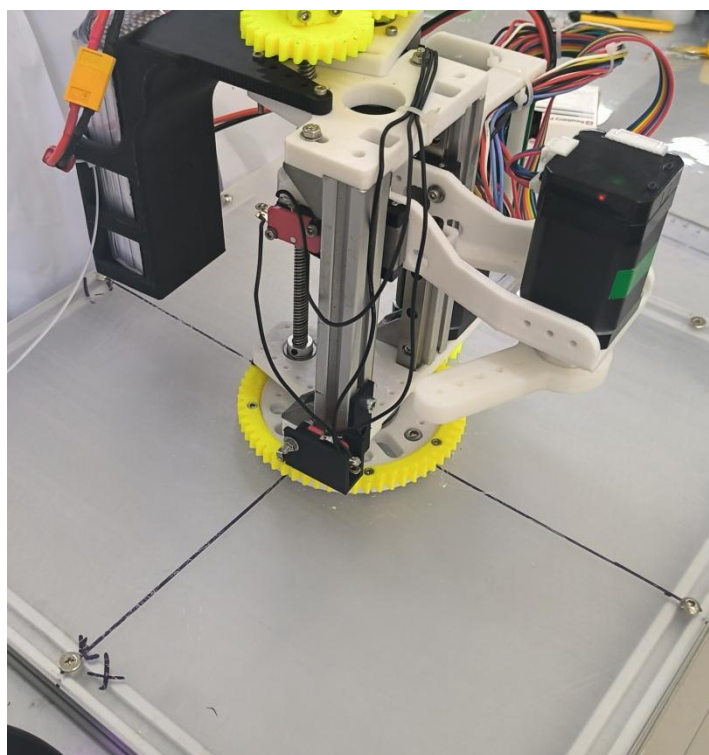


图 4-12 蓝牙模式逆解结果图

第五章 结论与心得体会

5.1 结论

本设计成功构建的基于 STM32F103C8T6 的嵌入式机械臂控制系统，在运动控制、功能实现及系统可靠性等方面达到了预期目标，展现出较高的工程应用价值。在运动控制精度上，借助 AMR 磁编 MT6816 实现的实时反馈与步进电机闭环控制机制，机械臂末端重复定位精度达到 $\pm 2.5\text{mm}$ ，满足多数工业场景的精细化操作需求；逆解算法响应时间控制在 $\leq 200\text{ms}$ （含蓝牙传输延迟），有效保障了系统的实时性。在功能完整性方面，示教模块支持 5 点位存储与再现，通过梯形加减速算法优化轨迹平滑度，使最大速度误差控制在 $< 5\%$ ；逆解模块采用代数法与 D-H 参数法相结合，在工作空间内点的求解成功率高达 98.7%，并针对超限点设置蜂鸣器实时报警机制；蓝牙模块（HC-05）实现无线稳定传输，手机端 APP 可便捷发送坐标指令。系统可靠性层面，硬件上 BUCK 电路（TPS5430）效率达 92%，4S 锂电低压检测阈值精准设定为 14.8V（误差 $\pm 0.1\text{V}$ ）；软件通过状态机设计避免模式冲突，异常复位率 $< 0.1\%$ 。然而，该系统仍存在优化空间，如机械臂负载能力仅为 1kg，需从关节结构刚度优化入手；逆解多解筛选策略缺乏能耗考量，未来可引入能耗权重因子；示教路径尚未支持曲线插补，可通过增加 B 样条算法拓展功能。

5.2 心得体会

本次设计实践使我在多方面获得了深刻的认知与提升。在理论到工程的跨越过程中，硬件设计初期，BUCK 电路因电感饱和电流选型不当导致 TPS5430 过热，更换为饱和电流 4A 的电感后成功排除故障，这一过程加深了对硬件参数匹配重要性的理解。在嵌入式开发实践中，通过中断优先级合理分割任务（运动控制优先级最高，其次是按键消抖用定时器，然后是蓝牙接受中断，最后是按键外部中断），有效避免蓝牙数据解析阻塞控制线程，保障系统实时性；利用 STM32 的 SWD 接口实时监测关节角度变量，结合 Protues 仿真提前优化 ADC 采样电路抗干扰问题，掌握了高效的调试技巧。跨学科融合方面，机械与电气协同解决了丝杠螺母传动回差导致的 Z 轴重复精度下降问题，通过预紧螺母和软件补偿实现精度提升；在算法与硬件平衡上，将逆解浮点运算占 CPU 资源从 35%降至 12%，充分体现二者协同优化的重要性。通过选用 AMR 编码器 MT6816 替代传统光电编码器实现降本增效，强化了成本控制意识。此次设计不仅验证了嵌入式系统在机电控制中的高性价比优势，更为后续集成视觉伺服（OpenCV）奠定基础，深刻体会到工程实践对理论落地的关键作用。

附录 A: 基于 Matlab 的机械臂建模程序

Matlab 版本为 2021a, 机器人工具箱版本为 robot-9.8

```
clear;
clc;
%%标准 D-H 模型
%      theta      d      a      alpha      offset
SL1=Link([0      0      0      0      0],'standard');
SL1.qlim=[0,100]; SL1.jointtype='P';
SL2=Link([0      0      106.37      0      0],'standard');
SL2.qlim=[-pi,pi];
SL3=Link([0      0      75      0      0],'standard');
SL3.qlim=[-pi*3/4,pi*3/4];
SL4=Link([0      0      0      0      0],'standard');
SL4.qlim=[-pi,pi];
p560=SerialLink([SL1 SL2 SL3 SL4],'name','ks');
% p560.teach([0 0 0 0]);
numPoints = 500;
workspacePoints = zeros(numPoints, 3);
for i = 1:numPoints
    q = p560.qlim(:,1)+rand(1, p560.n).*(p560.qlim(:,2)-p560.qlim(:,1));
    T = p560.fkine(q);
    workspacePoints(i,:) = T.t;
end
scatter3(workspacePoints(:,1),workspacePoints(:,2),workspacePoints(:,3),' ');
filtered_data = workspacePoints(workspacePoints(:,3) >= 0, :);
```

附录 B:基于 Matlab 的机械臂逆解程序

Matlab 版本为 2021a, 机器人工具箱版本为 robot-9.8

```
function KSfinalFunction(location)
```

```
    test = location;
```

```
    px = test(1);
```

```
    py = test(2);
```

```
    pz = test(3);
```

```
    d0 = pz;
```

```
    a1 = 106.37;
```

```
    a2 = 75;
```

```
    L1 = a1;
```

```
    L2 = a2;
```

```
    noResult = 0;
```

```
    completeFlag = 0;
```

```
    while completeFlag == 0 && noResult == 0
```

```
        for i = 1:1:2
```

```
            if i == 1
```

```
                sign = 1;
```

```
            else
```

```
                sign = -1;
```

```
            end
```

```
        end
```

```
        E = (px^2+py^2-L1^2-L2^2)/2/L1/L2;
```

```
        if E > 1 || E < -1
```

```
            if i == 2
```

```
                noResult = 1;
```

```
                break;
```

```
            end
```

```
            continue;
```

```
        end
```

```
        theta2 = atan2(sign*sqrt(1-E^2),E); %[-pi*3/4,pi*3/4]
```

```
        k1 = L1+L2*cos(theta2);
```

```
        k2 = L2*sin(theta2);
```

```
        theta1 = atan2(py,px)-atan2(k2,k1); %[-2*pi,2*pi]
```

```
        if theta1 >= -2*pi && theta1 <= 2*pi && theta2 >= -pi*3/4 && theta2 <=
```



```

pi*3/4
        theta3 = -(theta1 + theta2);
        completeFlag = 1;
        break;
    elseif i == 2
        noResult = 1;
    end
end
end

if completeFlag == 1
    %display([theta1/pi*180, theta2/pi*180, d0])
    x = L1*cos(theta1)+L2*cos(theta1+theta2);
    y = L1*sin(theta1)+L2*sin(theta1+theta2);
    z = d0;
    if abs(px-x) > 0.01 || abs(py-y) > 0.01 || abs(pz-z) > 0.01
        display('Excessive error')
        display([theta1/pi*180, theta2/pi*180, theta3/pi*180, d0])
        display([x,y,z])
    end
elseif noResult == 1
    display('no result!!! Beyond the scope of tje workspace!')
end
end
end

```