



南 华 大 学
UNIVERSITY OF SOUTH CHINA

毕业设计(论文)

题 目 基于单片机的步进电机多位置控制系统

学院名称 电气工程学院

指导教师 刘冲

职 称 副教授

班 级 自动化 1903 班

学 号 20190520508

学生姓名 李柯烨

2023 年 5 月 21 日

摘 要

本论文旨在研究并实现一种基于 51 单片机的步进电机多位置控制系统。该系统可以通过拨动开关设定转动方向、转动速度及细分，实现对步进电机在多个位置点的精准控制，具有控制精度高、操作简便等优点。本文主要包括以下内容。首先，介绍了步进电机与多位置控制系统的相关背景和理论知识，并分析了步进电机控制的难点和不足之处，提出了采用 51 单片机作为主控单元的设计方案。其次，详细阐述了硬件电路设计和控制程序编写。设计中采用了经典的四线八相驱动方式和 ULN2003A 芯片进行驱动，通过开关设置转动方向、转动速度以及微步进等参数。而程序编写中，主要涉及 51 单片机程序设计，需要实现步进电机的正转、反转、定位控制等功能，并通过显示 lcd1602 实现转动方向、位置指示和转动速度的显示。同时，在控制器电路中引入按键、开关等外设元件，以实现启动/停止控制、4-6 个位置点的定位控制、相应的自检测及报警功能。该系统简便易操作、控制精度高，具有较高的使用价值。

关键词：步进电机；多位置控制；51 单片机；控制精度

Abstract

This paper aims to study and implement a multi-position control system for a stepper motor based on the 51 single-chip microcomputer. The system can achieve precise control of the stepper motor at multiple position points by setting the direction, speed, and subdivision through toggle switches, and has the advantages of high control precision and easy operation. This paper mainly includes the following contents. Firstly, the background and theoretical knowledge of the stepper motor and the multi-position control system are introduced, and the difficulties and deficiencies in the control of the stepper motor are analyzed, and a design scheme using the 51 single-chip microcomputer as the main control unit is proposed. Secondly, the hardware circuit design and control program writing are elaborated. A classic four-wire eight-phase drive method and ULN2003A chip are used for driving, and the direction, speed, and micro-stepping parameters are set through switches. In the program writing, it mainly involves 51 single-chip microcomputer program design, which needs to realize functions such as forward rotation, reverse rotation, and positioning control of the stepper motor, and display the direction, position, and speed of rotation through LCD1602. At the same time, buttons, switches, and other peripheral devices are introduced in the controller circuit to realize start/stop control, positioning control of 4-6 position points, corresponding self-checking and alarm functions. The system is simple to operate, has high control accuracy, and has higher practical value.

Keywords: stepper motor; multi-position control; 51 single-chip microcomputer; control precision

目录

摘要.....	i
Abstract.....	ii
引言.....	1
1 绪论.....	2
1.1 国外研究现状.....	2
1.2 国内研究现状.....	3
1.3 课题研究背景及意义.....	4
2 步进电机控制系统总体设计方案.....	6
3 步进电机控制系统的硬件电路设计.....	8
3.1 STC89C51 单片机最小系统设计.....	8
3.1.1 主控制器.....	8
3.1.2 晶振电路.....	9
3.1.3 复位电路.....	10
3.1.3 12V 转 5V 降压电路.....	11
3.1.5 按键电路.....	12
3.1.6 机械结构设计.....	13
3.2 LCD1602 液晶显示模块.....	17
3.3 电机模块.....	18
4 步进电机控制系统的软件设计.....	22
4.1 主程序设计.....	22
4.2 lcd 液晶显示模块程序设计.....	23
4.3 步进电机定点控制模块程序设计.....	24
4.4 步进电机速度控制程序设计.....	26
5 系统调试与验证.....	28
5.1 开发环境.....	28
5.2 步进电机控制系统的调试与验证.....	29
6 总结.....	32

参考文献:	33
附录 1 元器件清单.....	35
附录 2 电路原理图.....	36
附录 3 电路 pcb.....	37
附录 4 代码展示.....	38
致谢.....	47

引言

步进电机是一种常用在仪表和过程控制中的一种控制元件。步进电机拥有诸多的优点如：结构简单、功能强大、性能稳定、价格低廉等，因此步进电机的应用也十分广泛，其中步进电机在自动化包装、数控机床、电动阀门、医疗设备等诸多领域得到了广泛而深入的应用。步进电机和其他类型的电机相比较而言有许多独特的优点比如：步进电机没有电刷，结构部件少，步进电机能够在较宽范围内实现速度平滑调节，同时负载变化不会影响步进电动机的速度的大小，步进电机还具有快速启停、易于调整、停止时能自锁等特点。因此，步机电设备中应用中最为广泛的一种电机就是步进电机。随着工业控制领域单片机控制技术的发展，步进电机的生产成本变的越来越低，抗干扰能力也越来越强。结合 STC89C51 单片机的特点，对步进电机的控制系统进行研究和设计具有重要的实用意义和价值。通过单片机控制步进电机，可以实现精确的控制，提高生产效率和质量。在设计中，需要考虑到步进电机的特性和工作要求，选用合适的驱动方式和控制算法，同时设计出简单易用、实用可靠的硬件电路和控制程序。总之，步进电机在现代工业和科技领域中具有广泛的应用前景和发展潜力。通过合理地使用单片机进行控制和调节，可以更好地实现其优越的性能特点，进一步推动工业控制和自动化技术的发展。

1、绪论

1.1 国外研究现状

在国外，基于单片机的步进电机多位置控制系统已经得到广泛应用。这种控制系统通常由一个单片机、一个步进电机驱动器和一组位置传感器组成。通过单片机的高精度计算和控制，可以实现步进电机在不同位置的准确调整和控制，改进了生产效率和质量。

在欧美地区，该技术已经被广泛应用于数控机床、自动化包装、医疗设备和电动阀门等领域。例如，在德国，许多工厂已经开始使用这种控制系统来提高机器人和自动化设备的生产效率。在美国，这种控制系统主要应用于汽车制造业，并且已经成为汽车工业中不可或缺的一部分。

此外，在日本，基于单片机的步进电机多位置控制系统也得到广泛应用。在半导体和电子加工行业中，这种控制系统能够帮助企业实现高精度的加工和制造，并且基于单片机的步进电机多位置控制系统被普遍认为是提高生产效率和质量的关键因素。

德国是全球工业自动化技术比较发达的国家之一，因此在基于单片机的步进电机多位置控制系统方面也有着突出的研究成果。例如，德国的西门子公司（Siemens）开发了一种名为 Simotion 的控制器，它可以实现高端控制、运动和机器人技术。同时，该公司还开发了一套名为 Step 7 的程序软件，可以用于开发运动自动化解决方案。此外，德国的贝加莱公司（Berghof Automation）也在基于单片机的步进电机多位置控制系统方面取得了许多成果。

美国在汽车工业、航空航天等领域的自动化技术应用上也非常成熟，因此在基于单片机的步进电机多位置控制系统方面也取得了不错的研究成果。例如，美国的施耐德电气公司（Schneider Electric）针对控制领域提出了一个名为 EcoStruxure 的新概念，该概念可以实现智能、可连续和高效的管理和运营。此外，美国的伊利诺伊工学院（Illinois Institute of Technology）在这方面也有较深入的研究成果。

日本的制造业相对来说比较发达，因此在基于单片机的步进电机多位置控制

系统方面也取得了不错的成果。三菱电机公司（Mitsubishi Electric）开发了 MELSEC-Q 系列的 PLC 控制器，具有高速、多点和高精度等特点。此外，OMRON 公司（Omron Corporation）发布了 MX2 系列卷扬式变速器，在驱动步进电机，并实现步进电机的高精度调节与定位控制中有广泛的应用。

总体而言，基于单片机的步进电机多位置控制系统在国外的应用非常广泛，其优越的性能和高精度的控制能力已经得到了广泛的认可和应用。随着技术的进步和应用领域的不断拓展，这种控制系统未来将继续发挥重要的作用，能够大大降低企业的生产成本，提高企业的生产效率，提高产品的质量。

1.2 国内研究现状

基于单片机的步进电机多位置控制系统是一种应用广泛的智能控制系统，它可以应用于各种不同类型的机器设备中。近年来，随着中国工业自动化水平的不断提高，基于单片机的步进电机多位置控制系统也得到了快速的发展。

在国内，有众多企业在该领域展现出了强大的技术实力和研发能力。其中，华为公司为例，其推出的 Tina 运动控制方案，在步进电机多位置控制系统的开发中具有重要意义。该方案具有快速响应、高度灵活和高精度等优点，并且可以为用户提供完善的技术支持和售后服务，便于企业实现自动化生产。这样的解决方案将会大大推动中国工业自动化和智能制造的发展。

除此之外，国内的一些高校和研究机构在该领域也做出了很大的贡献。例如，北京理工大学机械与车辆学院的研究团队，通过对步进电机控制算法的改进和优化，成功地设计并实现了一种基于单片机的步进电机多位置控制系统，可满足高速运动和高精度定位等要求，可以为有关的行业提供十分强有力的技术支撑。同时这些创新成果的出现也说明，中国在步进电机多位置控制系统领域已具备了一定的自主研发能力。

近年来，在我国工业自动化和智能制造领域，基于单片机的步进电机多位置控制系统得到了广泛应用。基于单片机的步进电机多位置控制系统的执行器件是步进电机，实现对机械设备的运动轨迹控制和精确定位操作，具有精度高、响应快、控制灵活等优点。在实现步进电机多位置控制系统的过程中，需要通过编写相关的软件程序，将控制指令传输给单片机，从而实现电机的精确控制。此外，设计者还需要合理地选择步进电机的型号、控制模式等参数，并考虑如何改进算

法以及降低成本等问题。因此，对于步进电机多位置控制系统的研究和开发，需要较高的技术实力和丰富的实践经验。目前，国内外已经涌现出了许多优秀的步进电机多位置控制系统供应商和解决方案提供商。这些公司和机构不仅在技术和应用方面取得了很大的进展，还为用户提供了一系列完善的售前咨询和售后服务。

此外，国内企业通过技术创新和自主研发，使该领域技术的应用和推广有了极大的提升。例如，深圳市松下电器有限公司成功开发出一款名为 Panasonic MBDKT2210MC1WZL 的步进电机驱动器，该产品具有高效稳定、使用方便等优点，大大提升了步进电机多位置控制系统的应用水平。这种技术的不断成熟和完善，将会促进步进电机多位置控制系统在各个领域的广泛应用。

总的来说，基于单片机的步进电机多位置控制系统在我国得到了快速发展，并且已经开始向智能化、高效性等方向拓展。未来，该领域的发展前景非常广阔，相信会有更多的企业和高校加入到这个领域的研究和开发中，为中国制造业的转型升级做出更为重要的贡献。同时，我国的企业和高校也需要加强技术创新和自主研发，不断提升步进电机多位置控制系统的性能和可靠性，推动中国制造业向着更高的水平迈进。

1.3 课题研究背景及意义

随着科技的不断进步，自动控制系统逐渐走进人们的生活，其中以单片机为核心的自动控制系统成为数字化时代的一个重要成果。这种系统具有实用性强、功能齐全和技术先进等优点，使人们对科学技术的发展更加充满信心。同时，它也让人类深刻认识到数字时代将会如何改变我们的生活，加速科技发展的步伐。步进电机作为过程控制和仪表控制的常用元件之一，在数控机床、自动化包装、电动阀门、医疗设备等领域有着广泛且深入的应用。相较于其他电机，步进电机具有结构简单、功能强大、性能稳定、价格低廉等众多优点。步进电机的速度不受负载变化大小的影响，具有快速启停、容易调整和停止时能够自锁等特点。因此，步进电机在机电设备中应用最为广泛。

在传统的步进电机控制方法中，是利用触发器产生控制的脉冲信号来对步进电机进行控制。然而，这种控制方法的缺点是当步进电机的参数发生变化时，就需要对控制器进行重新设计除此之外，由传统触发器构成的控制系统控制电路复杂、控制精度低、生产成本低。与以触发器为核心的传统控制器相比，基于单片

机的步进电机多位置控制系统的控制核心是微电子芯片，执行机构是电力电子功率变换器，因此，这种控制系统能够通过控制步进电机的转速或转矩以此来实现生产机械或运动部件按照人们所希望的规律运动。这种控制系统不仅克服了传统控制器的缺点，而且能够很好的契合工业生产新的控制要求，具有更大的优点和实用价值，因此在数字控制系统中具有十分广泛的应用。如今，在各种各样的行业当中都可以看到步进电机的身影，因此，高精度、实时监控的步进电机控制系统具有十分重要的实用价值和意义。通过该系统的研发，能够实现机器自动化程度的提升，大大提高生产效率，降低生产成本。除此之外，步进电机控制系统还展现出了广阔的应用前景，可以为全球经济发展做出巨大的贡献。

总之，随着技术的迅速发展和自动化程度的加深，步进电机多位置控制系统在自动化领域当中具有重要的地位。步进电机多位置控制系统的优点在于结构简单、功能强大、性能稳定、价格低廉等，并且在数控机床、自动化包装、电动阀门、医疗设备等领域中有广泛的应用。通过该系统的研发，能够实现机器自动化程度的提升，从而大大提高生产效率，降低生产成本。步进电机控制系统展现着广阔的应用前景，将为全球经济发展做出巨大的贡献。

2、步进电机控制系统总体设计方案

方案一：基于 AT89C51 单片机的步进电机控制系统方案是一种利用硬件法和软件法相结合的方式，通过分配脉冲信号来控制电机的通电、换相和变换控制。该方案主要由 AT89C51 单片机、液晶显示模块、键盘模块、时钟电路和步进电机模块等部件构成。在硬件方面，通过 AT89C51 单片机控制步进电机的驱动电路，将序列脉冲信号分配到四个驱动器中以控制电机的转动。同时，在按键复位电路和时钟电路中增加系统的耦合性，使系统与外围电路更好的结合在一起。在软件方面，在 keil 开发环境中通过编写 C 语言源代码对脉冲的分配进行调控，并且控制脉冲的上升沿和下降沿的通断时间，从而完成对步进电机的控制。该方案使用编程语言，优点是：大大提高了步进电机控制的准确性和有效性，同时也避免了系统震荡和失步等问题。为了能够实时监测步进电机的运行状态，该方案还增加了 lcd1602 液晶显示模块。通过液晶显示模块，可以实时显示步进电机的正反转情况，转速，定点控制等信息，并且根据实际工程的需要，增加了按键模块来满足电机的启停、正反转和调速等其他功能。整个控制系统的框架如图 2.1 所示。可以看出，该方案通过硬件法和软件法相结合，实现了对步进电机的精确控制，并且具有实时监测电机状态和丰富的功能扩展性等优点。在实际应用中，需要注意系统的稳定性和安全性，避免数据传输的干扰和攻击等情况。同时，需要根据实际需求选择合适的方案，并且合理设计硬件和软件，以达到最佳的控制效果。图 2.1 为控制系统框图。

方案二：主要由 Arduino 开发板、步进电机驱动模块、步进电机、脉冲发生器等部件构成。可以实时监测电机的位置信息，通过控制脉冲频率和方向，实现电机沿着多个位置进行精确移动。该方案适用于需要高精度定位和实时监测的场合。同时，Arduino 开发板拥有良好的开源社区支持，可以快速获取相关代码和库文件。基于 Arduino 开发的步进电机控制系统，响应速度较慢，无法满足高频快速控制的需求；此外，Arduino 开发板的成本较高，不太适用于大规模应用场景。

方案三：基于无线通信技术的远程控制步进电机系统，主要由远程控制设备、步进电机控制器、步进电机、无线通信模块等部件构成。用户可以通过远程控制

设备，向步进电机控制器发送指令，实现对步进电机的远程控制。该方案适合需要实现远程控制的场合，可以大大提高生产效率和工作灵活性。同时，无线通信技术的不断发展，也为该方案提供了更为广阔的应用前景。缺点：方案三实现远程控制的过程中，需要考虑数据传输的稳定性和安全性问题；此外，无线通信技术可能会遭受干扰和攻击，导致控制失灵。因此，需要一定的技术支持和风险评估。

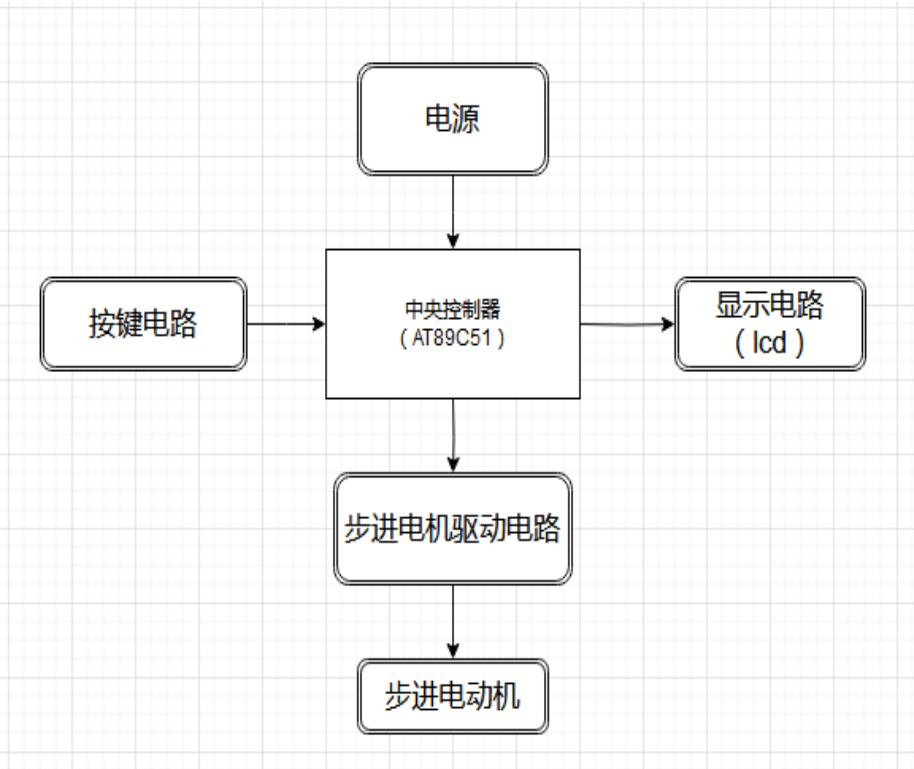


图 2.1 控制系统流程图

综上三个方案，本次设计采用方案一：基于 51 单片机的步进电机多位置控制系统，成本相对适中，调试更为简单，能很好地满足本次设计的步进电机控制需求。

3、步进电机控制系统的硬件电路设计

3.1 STC89C51 单片机最小系统设计

3.1.1 主控制器

STC89C51 单片机是由杭州STC微电子有限公司生产的一款 51 系列单片机，STC89C51 是基于Intel 8051 的架构，使用CMOS工艺制造，其中，STC89C51 包含了四组通用的I/O口，分别是P0,P1,P2,P3 和五个中断源，分别是外部中断 0（INT0/P3.2）：中断服务程序入口地址为 0003H，外部中断 1（INT0/P3.3）：中断服务程序入口地址为 0013H，定时计数器中断 0（P3.4）：中断服务程序入口地址为 000BH，定时计数器中断 1（P3.5）：中断服务程序入口地址为 001BH，以及串口中断：中断服务程序入口地址为 001BH，并且STC89C51 还具备 512B的内部RAM 和 64KB的闪存程序存储器。STC89C51 单片机通过高性能的CPU和硬件乘法器实现了高效、高速的处理能力，能够满足多种控制应用的需求。其主要的特性如下：

（1）高性能的CPU：STC89C51 单片机采用了高性能的CPU，主频可以达到 33MHz，可以通过硬件乘除法器可以实现比较高效的数字计算。

（2）容量较大的存储器：STC89C51 单片机的存储器分为两部分分别为：可编程ROM和RAM，可以提供 64KB的程序存储空间和 512B的数据存储空间，基本可以满足大多数应用的需求。

（3）十分丰富的通信接口：STC89C51 单片机内部的通信接口集成了UART、SPI、I2C等多种通信接口，可以实现与外部设备进行数据通信。

（4）低功耗设计：STC89C51 单片机的低功耗设计可以使其在待机状态下功耗非常低，以适应多种不同的应用场景。

（5）可靠性高：STC89C51 单片机采用的是工业级温度范围，并通过了众多的国际认证，确保其稳定可靠。

STC89C51 硬件连接和引脚如图 3.1 所示。

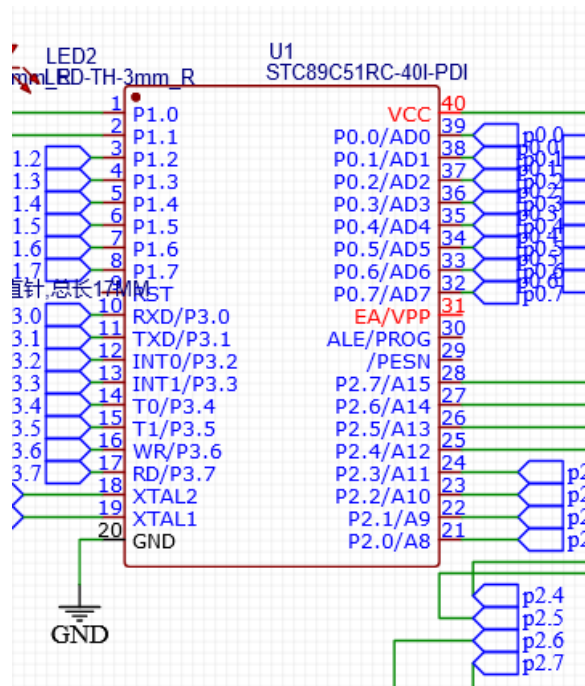


图 3.1 STC89C51 硬件连接和引脚图

3.1.2 晶振电路

众所周知晶体振荡器是硬件电路的核心所在，因为它能产生稳定的时钟信号，整个硬件系统都是围绕晶振进行运作的，单片机中常说的单周期指令、双周期指令、四周期指令等都是建立在由晶振电路所产生的时钟信号上，如果它不工作，整个电路就会停止运转，因此，它是决定数字电路能否正常工作的前提。

本文设计的 STC89C51 采用 11.0592MHz 的晶振提供时钟信号，STC89C51 单片机最小系统的晶振模块是整个电路中至关重要的一个部分，它主要用于为单片机提供基准时钟信号。在实际应用中，采用 11.0592MHz 的晶振，其可准确地得到波特率为 9600 和 115200 等时序关系，可以满足大多数应用的需要。STC89C51 单片机使用的晶振为外部晶振，通过电容 C1 和 C2 来帮助无源晶振起振，并维持晶振的振荡。C1 和 C2 的作用是帮助晶振保持稳定的振荡，降低干扰和噪声等，并且可以调节晶振频率的稳定性。

晶振电路如图 3.2 所示

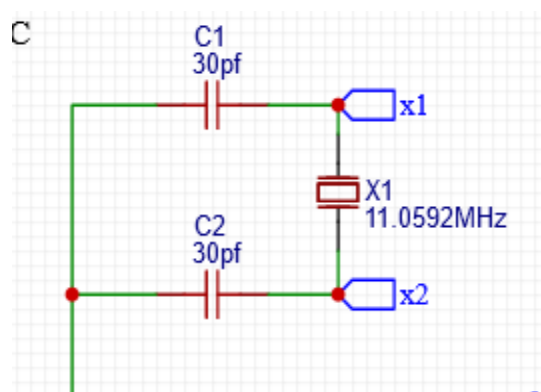


图 3.2 晶振电路

3.1.3 复位电路

在电路运行过程中，会不可避免的出现一些错误，这个时候就需要使用复位电路将电路恢复到最初始的状态，以便程序重新开始执行。目前常见的复位方式有三种：供电后立即进行重置、手动复位和程序自动复位。而本设计采用了低电平手动复位电路，复位过程经历两个阶段。

首先，在系统启动的时候，会进行一次重置。由于电容短路，导致 ret 电平为 0。随着时间的推移，电容缓慢充电并逐渐变成高电平。因此，在上电过程中会发生从低电平到高电平的变化，也就是说，上电时会进行一次复位。其次，当我们按下按键时，系统会再次进行一次复位。这个时候，开关导通，电容被短路。由于回路的作用，电容开始放电。当电容电压达到某一阈值以下时，ret 引脚由高电平变为低电平，从而使单片机进入复位状态。在该设计中，R 选用的是 1K 电阻，C 选用的是 10uF 电容。当按下按键时，ret 引脚被 GND 拉低，以此来实现手动复位操作。

复位电路如图 3.3 所示

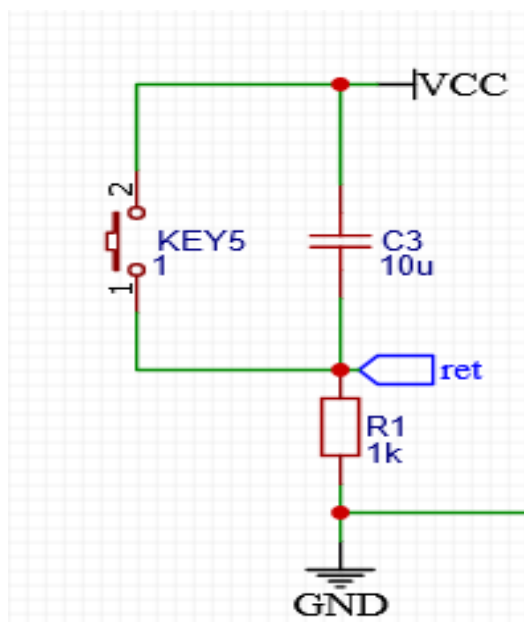


图 3.3 复位电路

3.1.3 12V 转 5V 降压电路

12V 转 5V 的降压电路采用的芯片是 LM2596T-12 降压芯片，LM2596T-12 芯片采用开关稳压器技术，能够将较高的输入电压稳定升压或降压至所需的输出电压。它能够在输入电压为 4V 至 40V 的范围内工作，并且能够输出 1.5A 的电流，同时还具有很高的效率和低的线性温度特性，在实际应用中，LM2596T-12 芯片广泛用于各种不同的领域，例如：工业控制、汽车电子、LED 照明、通讯设备等。它在降压逆变器中被广泛应用，可以承受较高的输入电压和大的负载，因此在应用中具有广泛的用途。

本设计电路的电源接口接 12V 的锂电池，电流会先经过 LM2596T-12 芯片将其降压为 5V，用 5V 电压为整个电路供电。电路中还设置了一个电源指示灯，当有电源接入时，电路导通，此时指示灯会亮起来。电源电路中的电容起到滤波作用。

12V 转 5V 降压电路图如图 3.4 所示

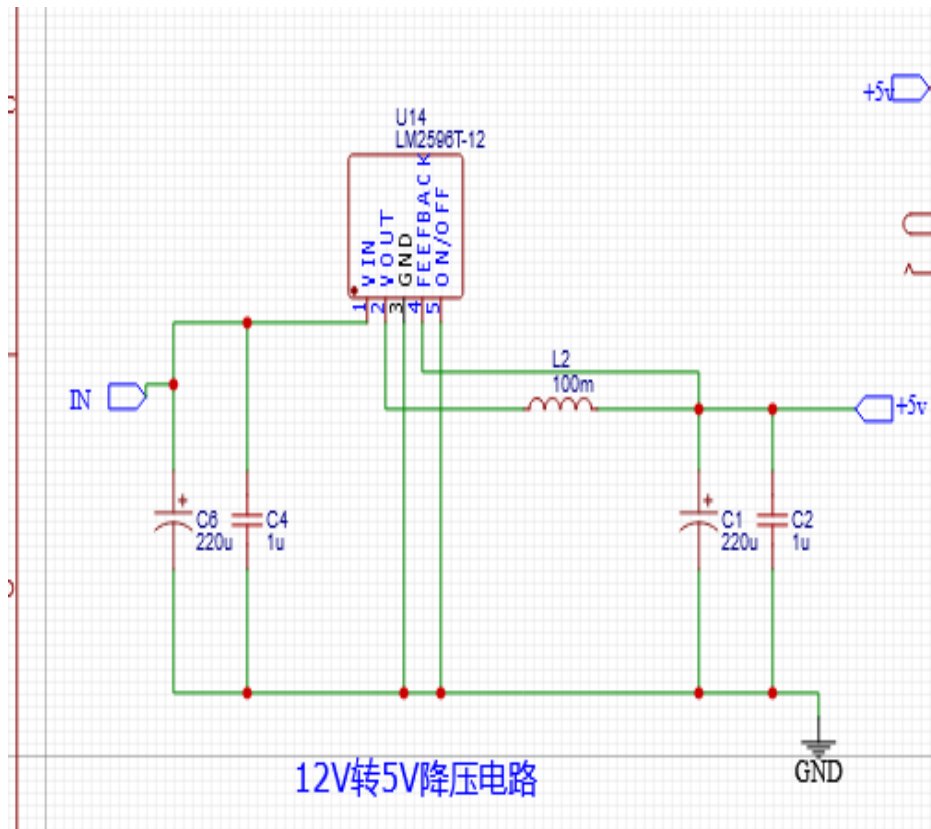


图 3.4 12V 转 5V 降压电路图

3.1.5 按键电路

按键电路主要是为了检测用户的输入信号以此对步进电机实现相应的控制功能，根据设计的需求，步进电机所要实现的控制功能分为三个部分：1 控制步进电机的启动与停止，以及控制步进电机的正转和反转。2 控制步进电机的速度，根据系统设计的要求，将步进电机的速度分为三个档次。3 实现步进电机的定点控制，本设计的系统能够实现对步进电机四个位置的定点控制。

基于以上的控制功能，按键电路共包含了 9 个按键，其中，按键采用的是拨码开关，将拨码开关的公共端接控制端口，其余两端接电源和地。

按键电路图如图 3.5 所示

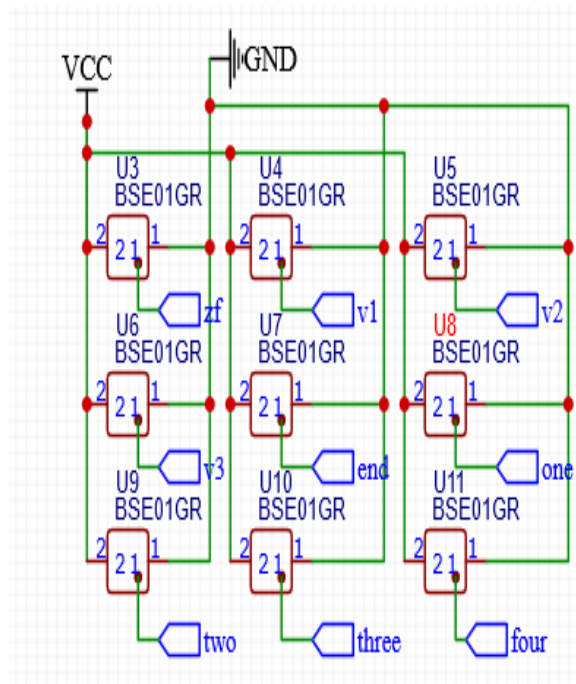


图 3.5 按键电路

3.1.6 机械结构设计

根据定点控制的要求，需要对步进电机转动的轴承以及圆盘进行机械结构的设计，在步进电机的机械结构设计中，需要遵循以下原则：机械结构应具有足够的稳定性和刚度，以确保电机的稳定性和准确性；机械结构应简单、易于制造和维护，成本尽可能低廉；底盘应具有足够的空间和支撑面积，以便安装电机和其他必要的元件；底盘上应安装原点复位开关，确保电机在每次运动前始终处于同一初始位置。

基于这些设计原则，本项目采用了以下机械结构设计方案：

- （1） 底盘采用适中大小的扇形设计，底面平整，材料为 ABS 塑料，使用 3D 打印技术制造；
- （2） 底盘上固定了一个原点复位开关，用于检测电机旋转的起始位置；
- （3） 电机与旋转轴之间增加凸轮结构，以实现更加平稳的转动和更高的准确性。

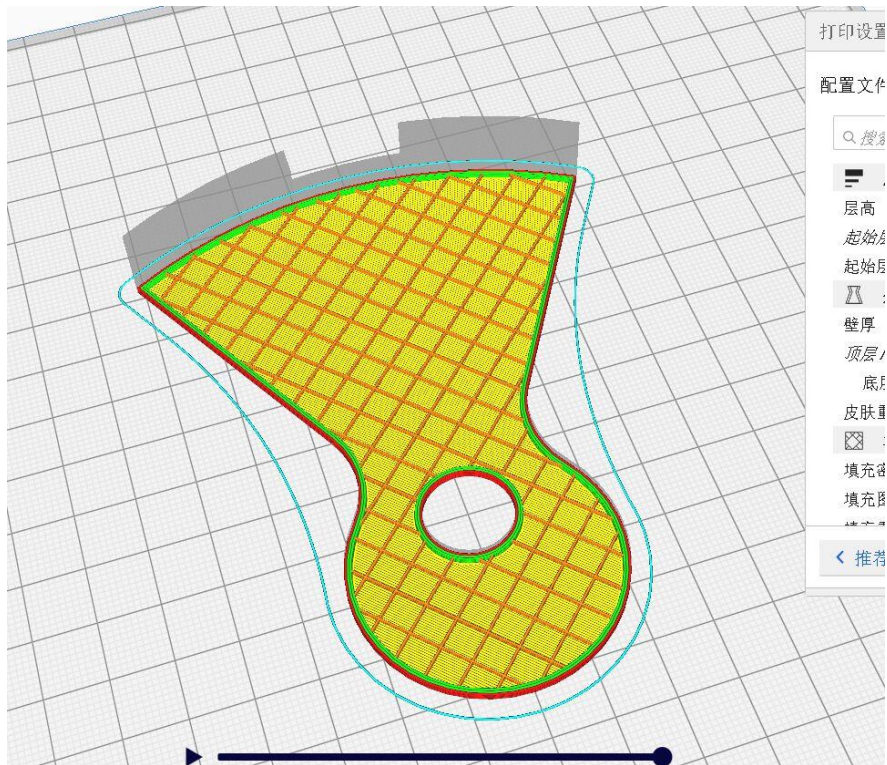


图 3.6 底座 3D 打印切片图

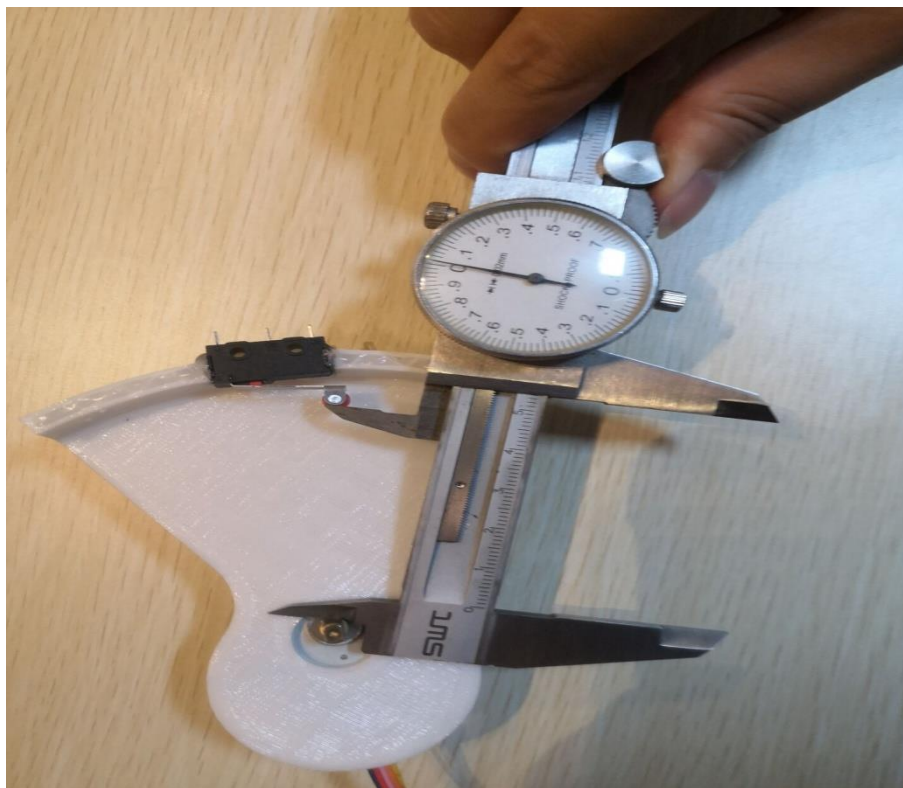


图 3.7 测量步进电机旋转轴长度

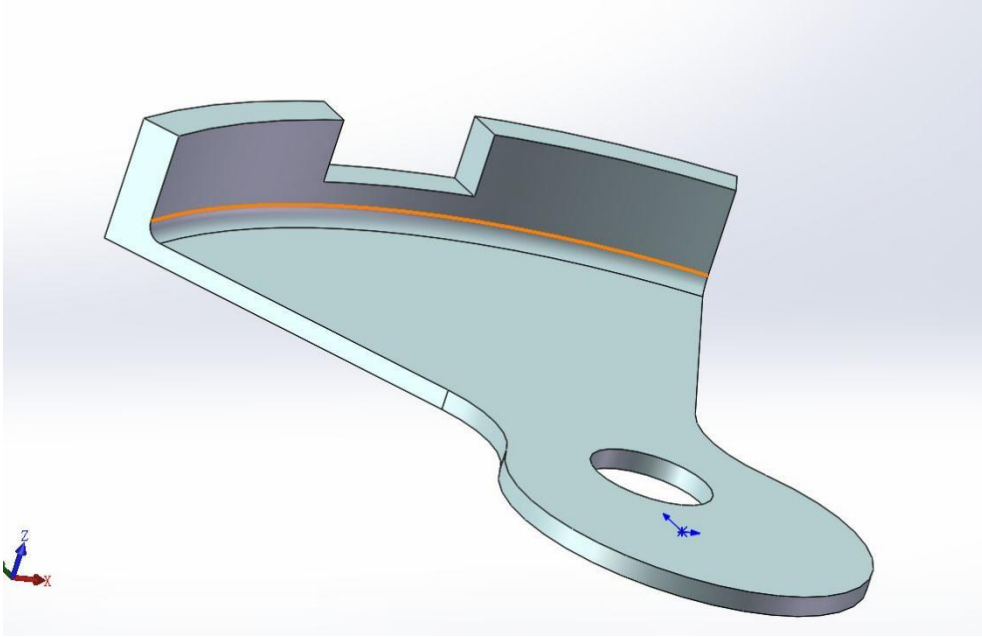


图 3.8solidwork 底座建模图



图 3.93d 打印图

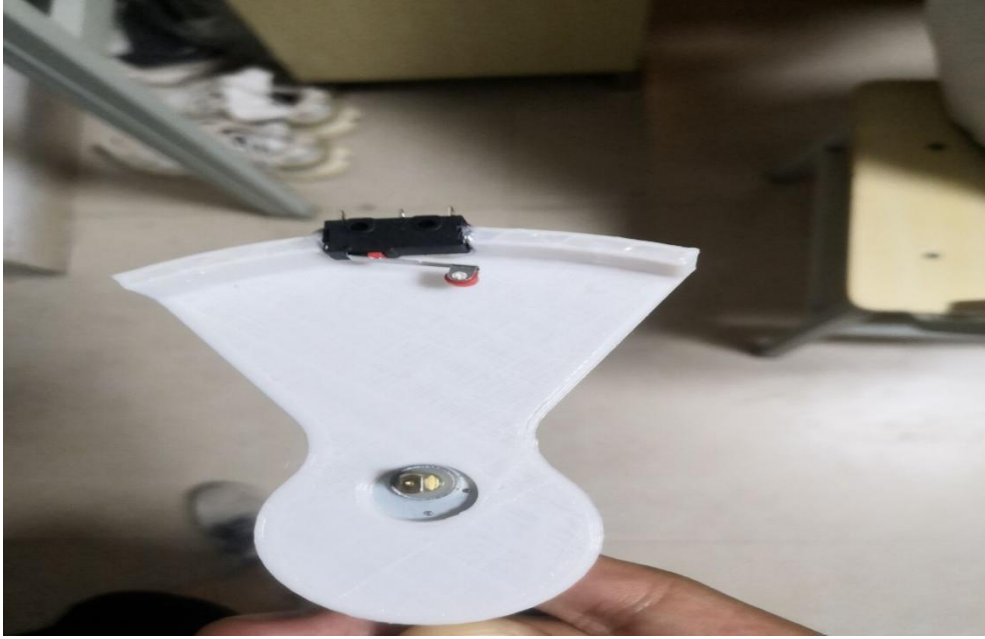


图 3.10 3D 打印成品图

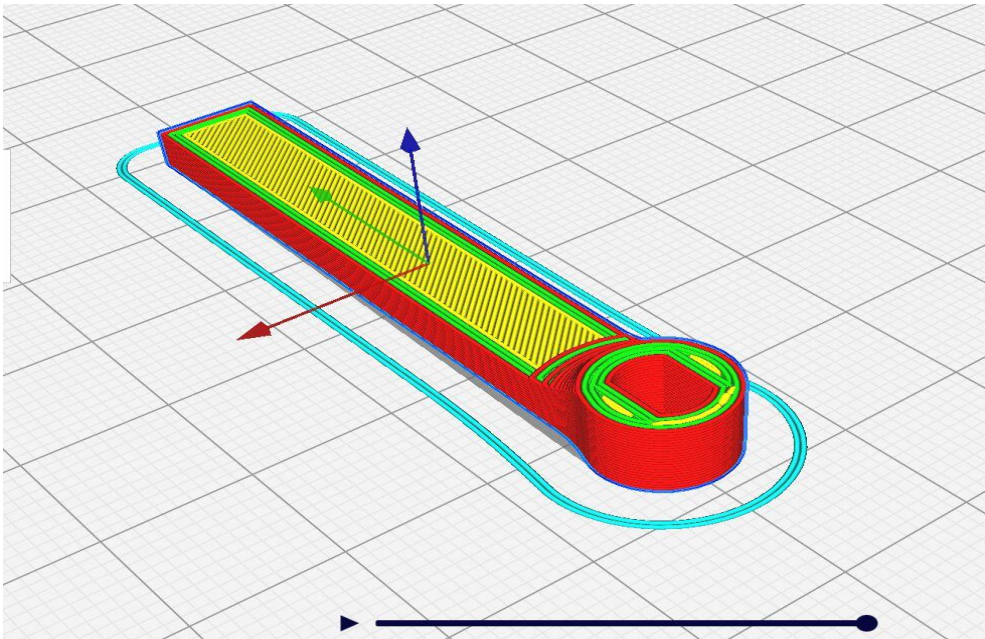


图 3.11 旋转轴 3D 打印切片图

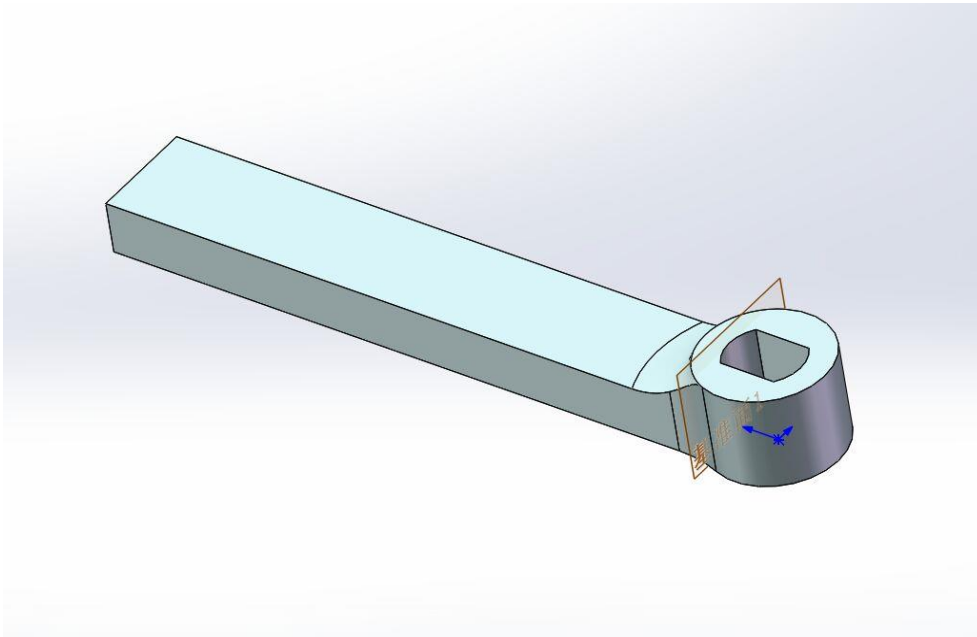


图 3.12solidwork 旋转轴建模图

3.2 LCD1602 液晶显示模块

LCD1602 液晶显示模块是一种点阵型液晶模块，也被称为 1602 字符型液晶模块。LCD1602 液晶显示模块由多个 5x7 的点阵字符位组成，可以显示字母、数字和符号等内容，一共可以显示 16 行 2 列的字符。并且每个字符之间有一个点距以及行间距，以适当的间隔展示字符。LCD1602 液晶无法显示图像。

LCD1602 液晶模块共有 16 个管脚，各个管脚的功能介绍如下：

(1) 管脚 1 (VSS)：连接 5V 电源的负极（即 GND）。

(2) 管脚 2 (VDD)：连接 5V 电源的正极。

(3) 管脚 3 (VL)：液晶的显示偏压信号，用于调整液晶显示的对比度。当接电源正极时，对比度最弱；当接地时，对比度最高。为避免“鬼影”现象，可通过 10k Ω 电位器调节。

(4) 管脚 4 (RS)：液晶的寄存器选择脚。当该管脚为高电平时，选择数据寄存器；当为低电平时，选择指令寄存器。

(5) 管脚 5 (R/W)：读/写选择管脚。当该管脚为高电平时，进行读操作；当为低电平时，进行写操作。当 RS 和 R/W 管脚都为低电平时，可写入指令或显示地址。当 RS 管脚为低电平，R/W 管脚为高电平时，可读取忙信号。当 RS 管脚为高电平，R/W 管脚为低电平时，可写入数据。

步进电机 28BYJ-48 名称含义：

28：表示步进电机的有效最大外径为 28 毫米

B：表示步进电机“步”字汉语拼音首字母

Y：表示永磁式“永”字汉语拼音首字母

J：表示减速型“减”字汉语拼音首字母

BYJ：组合即为永磁式减速步进电机

48：表示四相八拍

5V：表示额定电压为 5V，且为直流电压

步进角：5.625 度，就是 64 个脉冲信号转 5.625 度，64*64 个信号转 360 度。

减速比：1/64，电机壳里边的电机转 64 圈，电机壳外边的部分转 1 圈。

四相：ABCD 四相(电机定子上有 8 个齿，相对的 2 个齿是 1 相)，

八拍：(A-AB-B-BC-C-CD-D-DA-A)。

一拍就是一个脉冲信号，完成一个循环用 8 个脉冲信号。

当通电状态的改变完成一个循环时，转子转过一个齿距。转 8 个齿距就是一圈， $8 \times 8 = 64$

64 拍，64*64 个脉冲信号转一圈 360 度。

步进电机模块接线图如图 3.10 所示

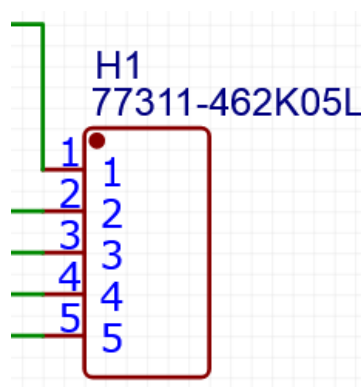


图 3.14 步进电机模块接线路

(2) 步进电机驱动模块

ULN2003 是高耐压、大电流复合晶体管阵列，由七个硅 NPN 复合晶体管组成。一般采用 DIP—16 或 SOP—16 塑料封装。

ULN2003 的主要特点：

1 ULN2003 的每一对达林顿都串联一个 2.7K 的基极电阻，在 5V 的工作电压下

它能与 TTL 和 CMOS 电路直接相连,可以直接处理原先需要标准逻辑缓冲器来处理的数据。

2 ULN2003 工作电压高,工作电流大,灌电流可达 500mA,并且能够在关态时承受 50V 的电压,输出还可以在高负载电流并行运行。

ULN2003 的作用:

ULN2003 是大电流驱动阵列,多用于单片机、智能仪表、PLC、数字量输出卡等控制电路中,可直接驱动继电器等负载。输入 5VTTL 电平,输出可达 500mA/50V。简单地说,ULN2003 其实就是用来放大电流的,增加驱动能力。比如说单片机输出引脚一般输出就几 mA,是无法驱动电机、继电器或者电磁阀的,像要让直流电机转需要 500mA,而用 ULN2003 放大后,可以通过单片机的输出引脚直接控制这些设备。

ULN2003 的接法:

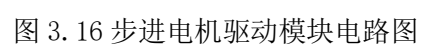
输入端: 引脚 1 到引脚 7 分别对应输入 1 到输入 7,拥有达林顿对的 7 个输入引脚,其中每一个引脚都是连接到晶体管的基极,且可以用+5V 触发。

输出端: 引脚 10 到引脚 16 分别对应输出 1 到输出 7,只有当其各自的输入引脚为高电平时,每个输出引脚才会接地。

公共端: com 端是用做测试引脚用的,是可以选的。

引脚	引脚名称	描述
1 到 7	输入 1 至输入 7	达林顿对的七个输入引脚,每个引脚都连接到晶体管的基极,可以使用+5V触发
8	接地	接地参考电压 0V
9	COM引脚	用作测试引脚或电压抑制引脚(可选)
10 至 16	输出 1 至输出 7	七个输入引脚的各自输出。只有当其各自的输入引脚为高电平(+5V)时,每个输出引脚

图 3.15 ULN2003 芯片引脚功能图



4、步进电机控制系统的软件设计

4.1 主程序设计

图 4.1 为步进电机控制系统的软件结构图。当电路开始运行时，先对所有的模块进行初始化；然后按键模块在不断检测输入的信号值，单片机根据按键检测到的值，执行对应的功能函数，以对步进电机进行相应的控制；然后打开 LCD1602 液晶显示屏，显示步进电机的实时控制状况。

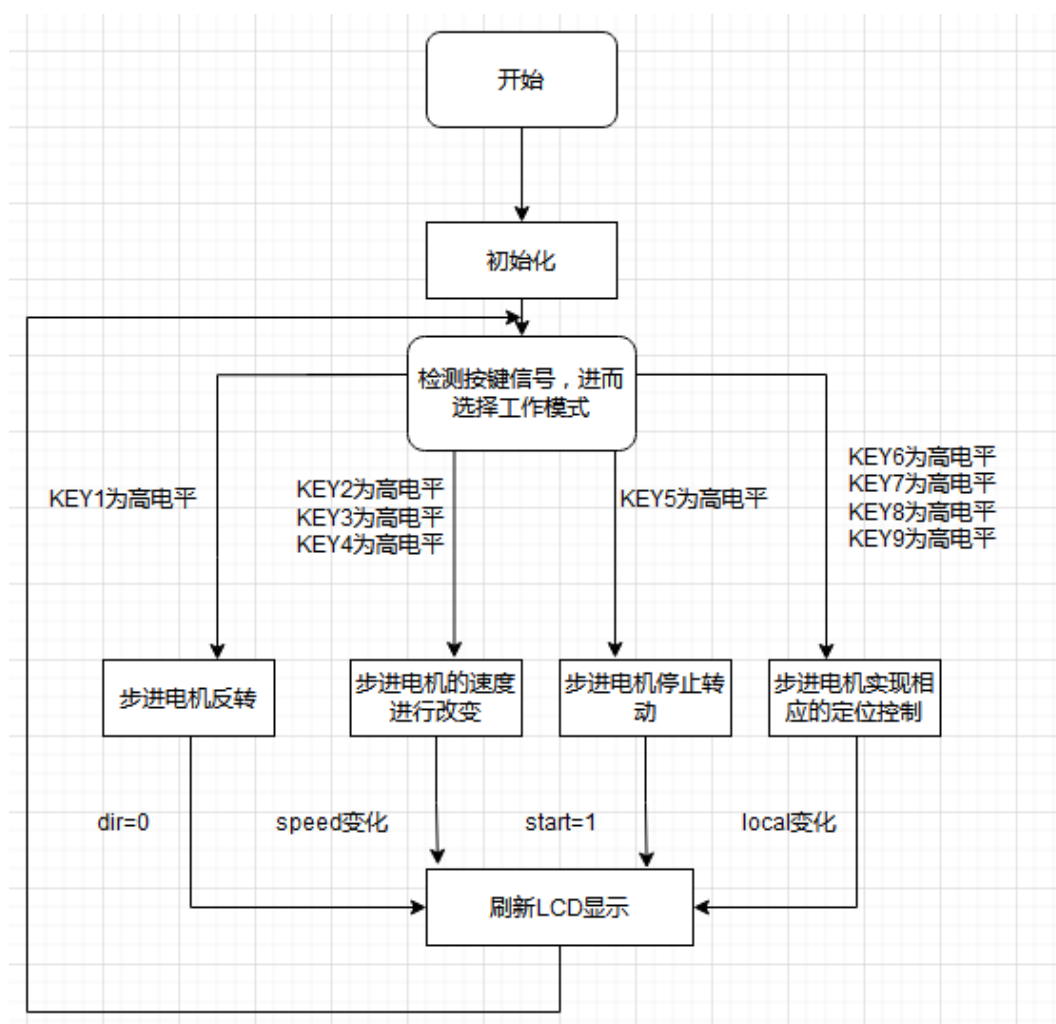


图 4.1 软件结构图

程序是使用 STC89C51 单片机实现步进电机多位置控制系统。程序使用的是 28BYJ-48 步进电机，本程序利用 if else 语句实现按键检测功能，通过检测按键是否按下，以此来实现对应的步进电机控制功能，如：步进电机正转，反转控制，步进电机速度的控制，以及步进电机的定点控制。然后利用 LCD1602 来显示步进电机的运行状态，如：正反转的情况，运行速度的情况以及定点控制的情况。

4.2lcd 液晶显示模块程序设计

首先，要初始化液晶显示模块。然后在初始化过程中，需要将 RS、RW、E 引脚置为高电平，然后给液晶显示模块发送初始化命令。其中，这些命令包括设置显示模式、设置光标位置、设置字符显示模式等等。需要注意的是在发送命令之前，我们需要确保液晶显示模块处于空闲状态，避免出现数据传输错误的情况。

在完成液晶显示模块初始化后，开始向其发送数据。在向液晶显示模块发送数据前我们需要先将 RS 引脚置高电平，然后将数据发送到数据线上，最后将 E 引脚置高电平。对于需要在液晶显示模块上显示的字符，然后在将字符的 ASCII 码转换成对应的字符模式，并将其发送到液晶显示模块上。

除了发送字符外，还可以向液晶显示模块上的特定位置写入数据，这些数据包括字符、数字、符号等等。为了实现在液晶显示模块上的特定位置写入数据，我们需要先将光标移动到需要写入数据的位置，并向该位置发送数据。

图 4.2 为 LCD1602 液晶显示模块程序流程图。

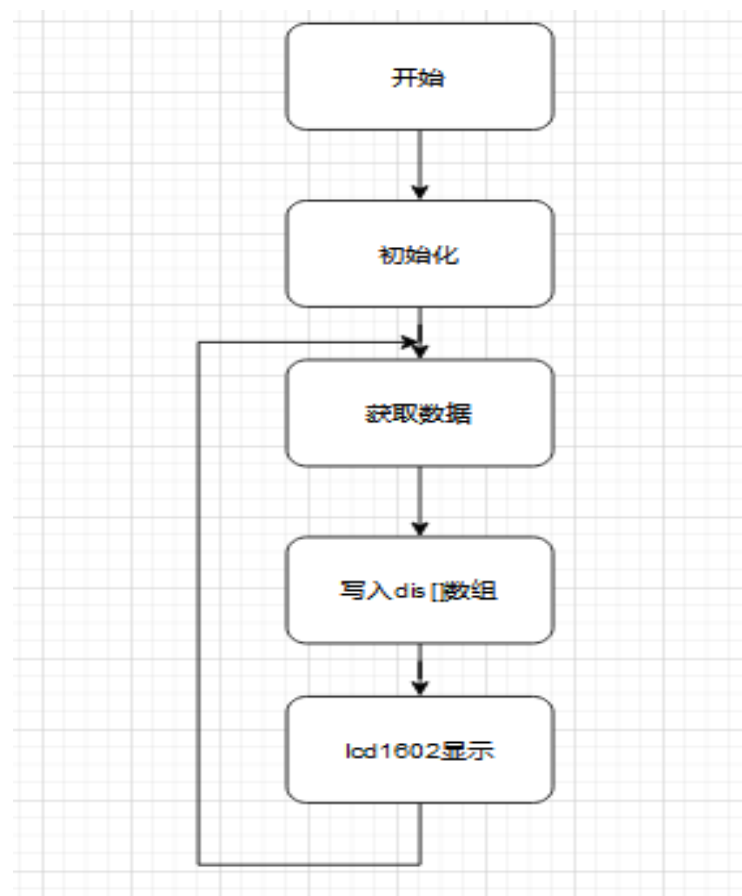


图 4.2LCD1602 液晶显示模块程序流程图

4.3 步进电机定点控制模块程序设计

图 4.3 为步进电机定点控制模块流程图。首先将步进电机转动到初始位置，然后通过按键检测到的位置信号执行对应的步进电机控制函数，以此来实现对步进电机的定点控制，其中位置 1 为以原点为基准，顺时针转动六十度，位置 2 为以原点为基准，顺时针转动一百二十度，位置 3 为以原点为基准，顺时针转动一百八十度，，位置 4 为以原点为基准，顺时针转动二百七十度。

部分代码如下：

```
if(start==1) {}  
    else{  
  
        key_scan();//按键信号检测  
        byj(KEY1, speed);//电机控制  
    }  
    if(KEY5) {start=1;} //停止转动  
    if (KEY5==0) {start=0;};  
  
    if (KEY6==1)  
    {  
        start=1;  
        for (j=0; j<8*8*64/6; j++)  
        {  
            if (i<0) i=7;  
            P1=bjright[i--];  
            delay100us(10)  
        }  
    }  
    if (KEY7==1)  
    {  
        start=1;
```

```

        for(j=0;j<8*8*64/3;j++)
        {
            if(i<0) i=7;
            P1=bjright[i--];
            delay100us(10);
        }
    }
    if(KEY8==1)
    {
        start=1;
        for(j=0;j<8*8*64*3/4;j++)
        {
            if(i<0) i=7;
            P1=bjright[i--];
            delay100us(10);
        }
    }
    if(KEY9==1)
    {
        start=1;
        for(j=0;j<8*8*64/2;j++)
        {
            if(i<0) i=7;
            P1=bjright[i--];
            delay100us(10);
        }
    }
}

```

```
if(KEY10==1&&KEY11==1){start=1;};
```

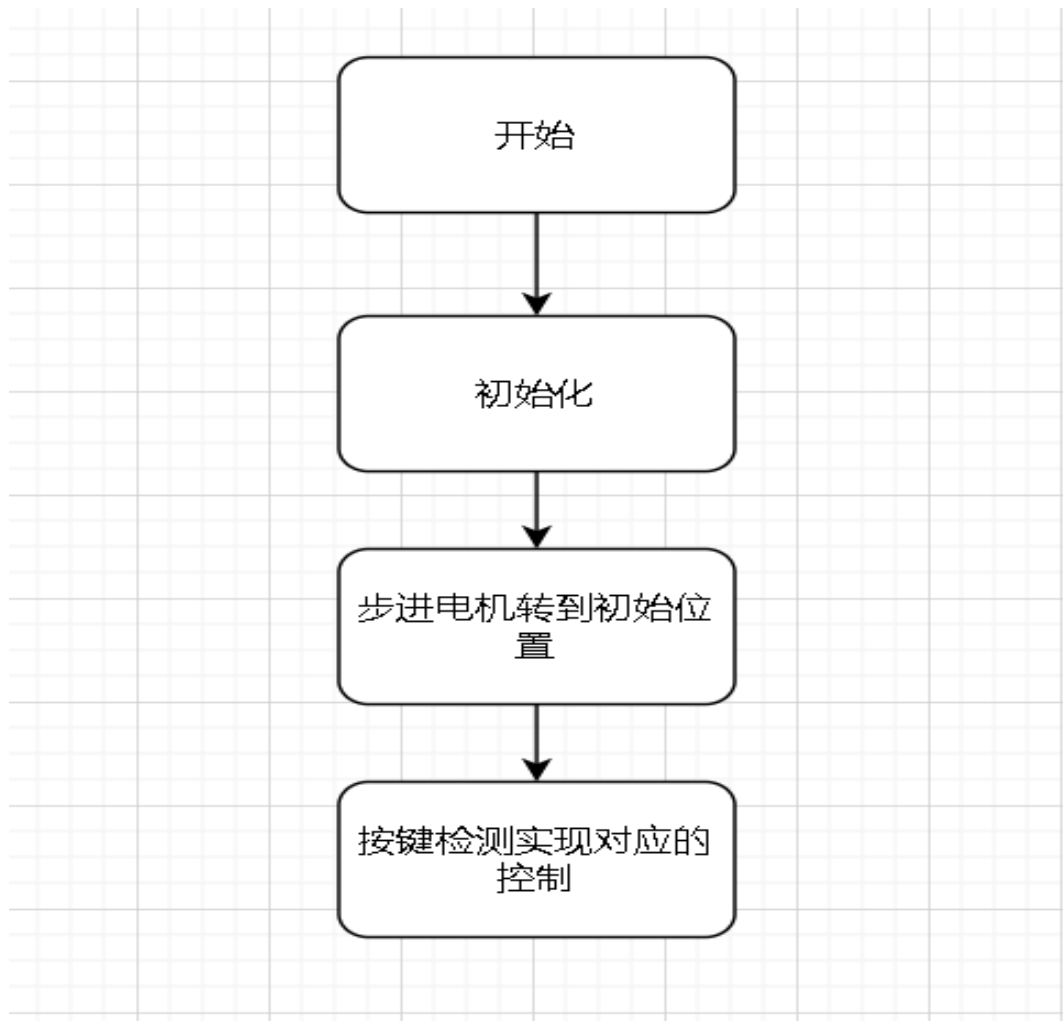


图 4.3 步进电机定点控制模块程序流程图

4.4 步进电机速度控制程序设计

步进电机的速度控制是通过控制延时函数的延时时间长短来控制刷新的速度，以此来控制步进电机的转动速度，部分代码如下：

```
if(off==1){  
}else{  
    //key=key_scan(0);  
  
    if(KEY1==0){dir=0;Disp[4]='+';}//正转
```

```

if(KEY1==1) {dir=1;Disp[4]='-' ;} //反转
if(KEY2==1&&KEY3==0&&KEY4==0) {speed=200;Disp[15]=' 1' ;} //1
if(KEY2==0&&KEY3==1&&KEY4==0) {speed=100;Disp[15]=' 2' ;} //2
if(KEY2==0&&KEY3==0&&KEY4==1) {speed=50;Disp[15]=' 3' ;} //3
step_motor_28BYJ48_send_pulse(step++,dir);
if(step==7)step=0;

```

图 4.4 为步进电机速度控制模块程序流程图。

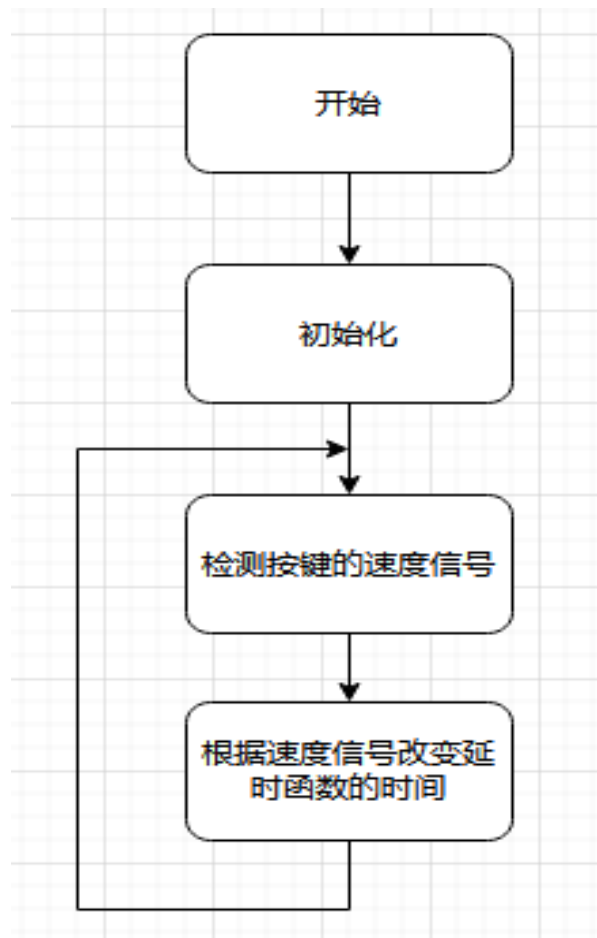


图 4.4 步进电机速度控制模块程序流程图

5、系统调试与验证

5.1 开发环境

KeilVision5 是目前单片机开发的主流软件，现属于 ARM 公司。它功能强大，有十分丰富健全的开发方案，KeilVision5 除了能够完成 C 语言和汇编语言的开发和编译的工作，还内含一个连接器，这个连接器可以对单片机进行在线仿真调试。KeilVision5 使用的界面是市面上主流的 Windows 界面，操作的流程在很大程度上是模仿微软 VC++ 这一个软件，因此，对于初次接触单片机调试的初学者来说是十分友好的，上手简单，虽然 KeilVision5 并不支持中文，但我们可以通过补丁的方式进行汉化。具体的单片机开发过程如下：

首先我们需要建立一个新的工程项目，在 MCU 类型下找到相应型号的芯片（STC89C51），然后再建立一个 C 源文件文本这个源文件文本是用于编辑代码的，将其添加到工程项目文件中。代码编辑完成后直接进行编译，若编译成功没有报错，就可以将其下载到所连接的单片机中。

整个开发流程非常简单易懂，甚至常用的 C 语言开发软件如出一辙，它包含市面上所能购买的到的众多品牌的芯片，如 ST、Atmel、NXP、TI 等。因此在调试单片机时，往往会选择 Keil 这一开发环境。

KeilVision5 界面如图 5.1 所示。

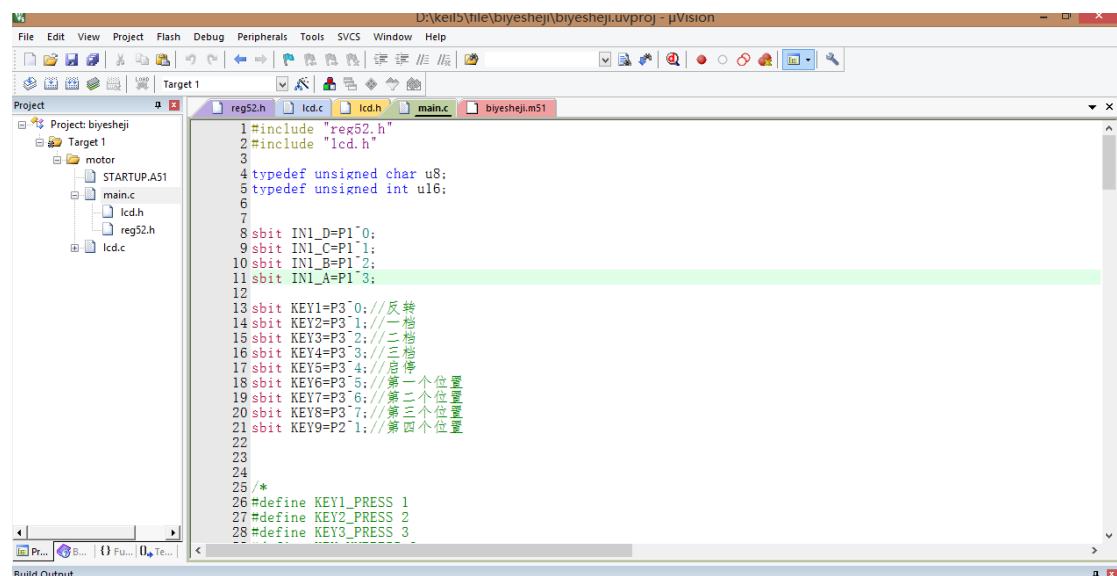


图 5.1KeilVision5 界面

5.2 步进电机控制系统的调试与验证

系统调试的思路：先一一调试验证每个模块的功能，保证各个模块在单独运行的时候能够正常运行。然后在进行系统的联合调试，验证系统的各个功能。不断完善系统的功能以及修改系统的 bug。

首先将各个模块与单片机按照硬件电路设计连接起来。将代码进行编译，代码编译通过之后在下载单片机。打开电源，将系统进行初始化，lcd1602 液晶显示屏显示小车状态。

经过实际的测试，当按下正反转控制按键时，步进电机将进行对应的正转和反转，当按下步进电机速度控制按键时，步进电机将进行对应的速度改变，当按下步进电机的定点控制按键时，步进电机将转动到对应的位置。如图 5.2，图 5.3，图 5.4，图 5.5 和图 5.6 所示。



图 5.2 第一个定点 60 度



图 5.3 步进电机回到原点



图 5.4 第二个定点 120 度

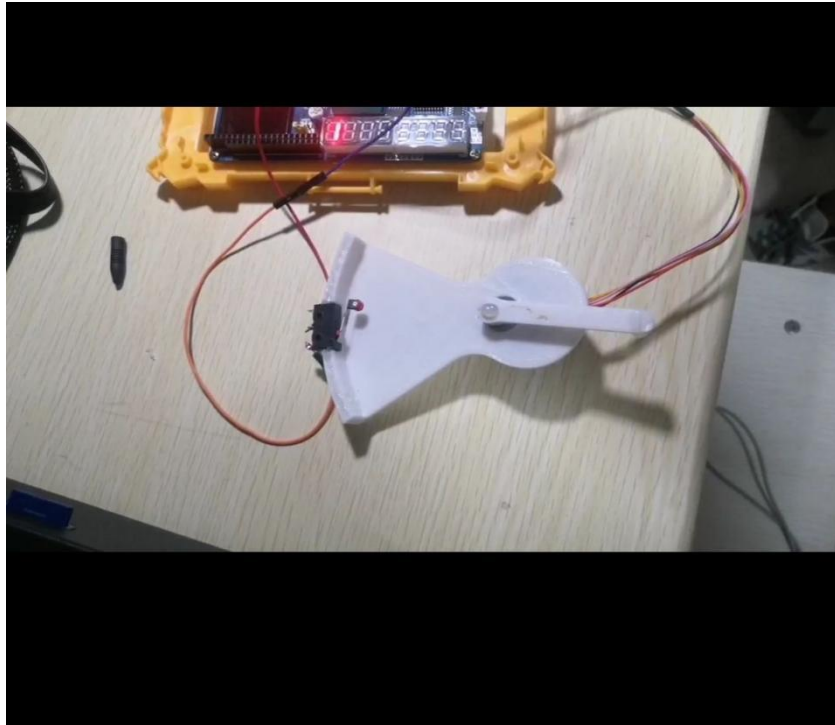


图 5.5 第三个定点 180 度

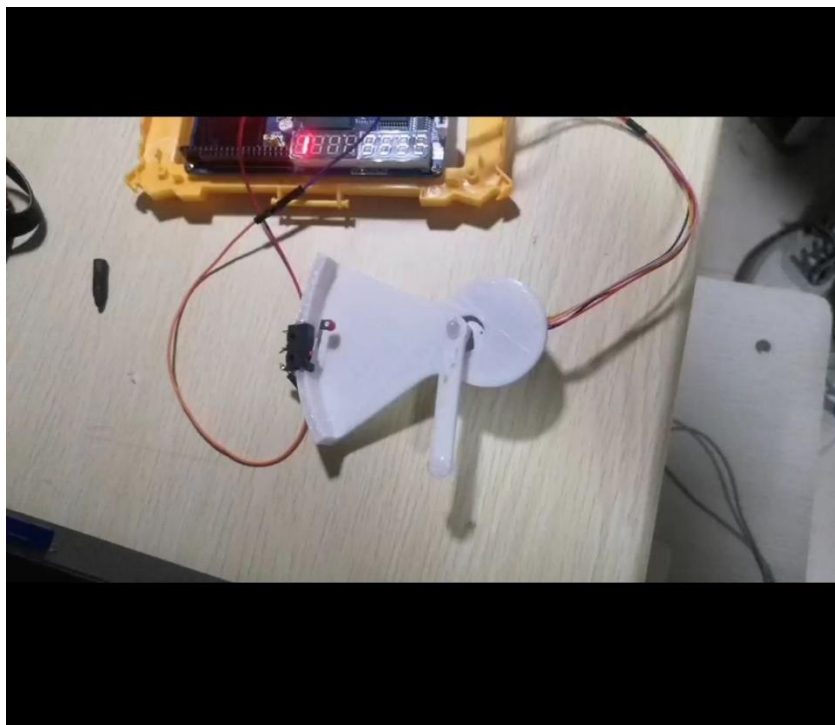


图 5.6 第四个定点 270 度

6、总结

基于 51 单片机的步进电机多位置控制系统使用的是步进电机，这种电机可以根据输入的脉冲信号旋转一定角度，从而实现精确的位置控制。本系统可以控制步进电机旋转多个不同的位置，控制步进电机的转动速度以及控制步进电机的正转和反转，并且可以根据需要进行调整。这种系统在许多应用中都有很大的作用，比如在机器人、自动化生产线等方面都有广泛的应用。

本系统的优点在于，可以实现精确的位置控制，具有较高的稳定性和可靠性。同时，由于采用了 51 单片机作为控制器，成本较低，适合在一些小型应用中使用。不足之处在于，由于步进电机的特性，速度较慢，不适合用于一些需要高速旋转的应用。总之，本文介绍的基于 51 单片机的步进电机多位置控制系统具有广泛的应用前景，可以在机器人、自动化生产线等领域发挥重要作用。在实际应用中，还需要进一步完善和优化，以满足不同的需求。

参考文献

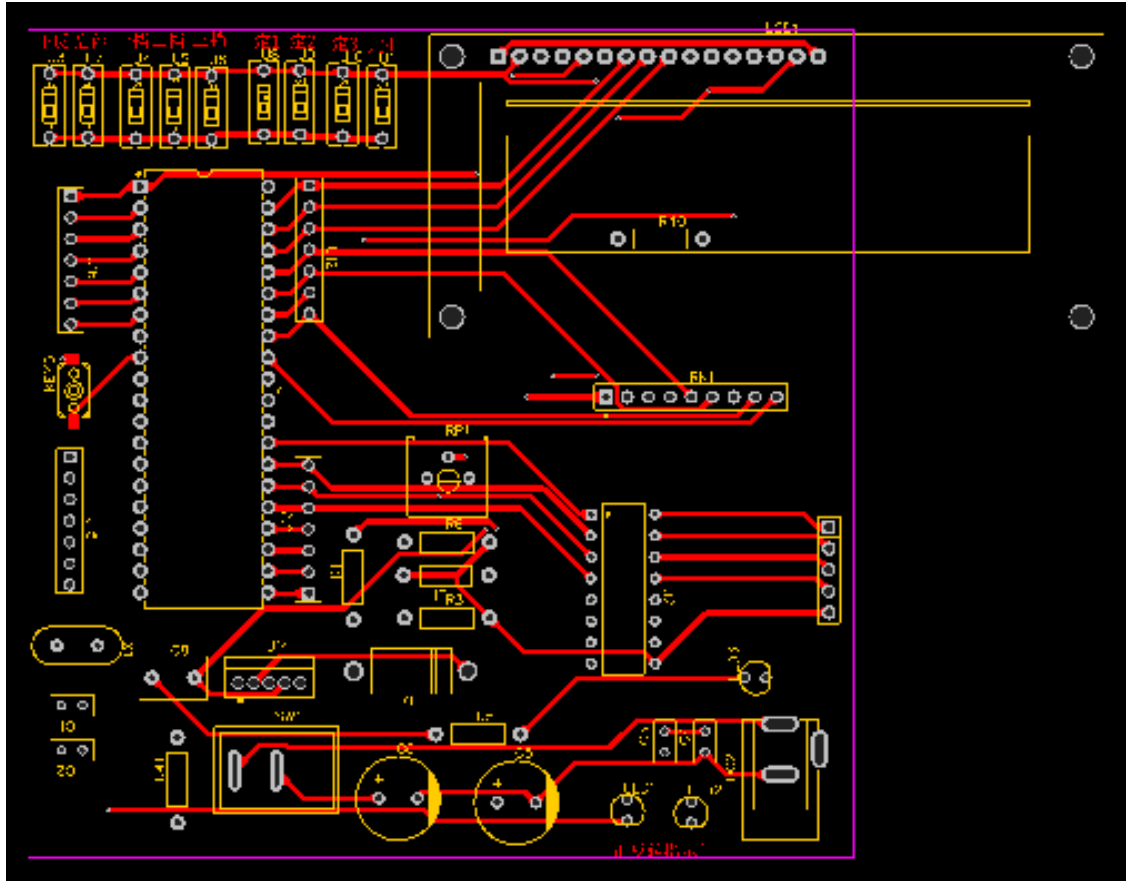
- [1] 王福瑞,等. 单片机微机测控系统设计大全[M]. 北京: 北京航空航天大学出版社, 2000.
- [2] 李冰. 基于单片机的煤矿步进电机驱动控制系统设计[J]. 煤矿机械, 2014, 35(5): 226-228.
- [3] 何冲, 王淑红, 侯胜伟, 慧文. 基于 AT89C52 单片机的步进电机控制系统研究[J]. 电气技术, 2012, (4): 5-8.
- [4] 袁红. 浅谈单片机的步进电机控制系统[J]. 科技展望, 2016, 31: 56-56.
- [5] 古志坚. 基于单片机的步进电机控制系统研究[D]. 华南理工大学硕士学位论文, 2013.
- [6] 牛慧佳. 基于单片机的步进电机控制系统设计研究[J]. 数字技术与应用, 2018, 36(7): 11-12.
- [7] 田聪, 苏畅. 基于 51 单片机的步进电机控制系统设计与实现[J]. 电子测试, 2017(22): 5-6, 19.
- [8] 曹旻罡. 基于单片机的步进电机控制系统研究[J]. 数字技术与应用, 2017(6): 20, 22.
- [9] 张秀辉, 苏娱. 基于单片机的步进电机控制系统研究[J]. 电脑知识与技术, 2016, 12(12): 279-280, 286.
- [10] 杨永清. 基于 STM32 和 FPGA 的多通道步进电机控制系统设计[D]. 成都: 西南交通大学, 2017.
- [11] 付艺豪, 许建明, 李忠. 基于单片机的步进电机系统设计[J]. 电子世界, 2016(4): 149-150.
- [12] 陈理壁. 步进电机及其应用[M]. 上海: 上海科学技术出版社, 1989.
- [13] 刘保延, 等. 步进电机及其驱动控制系统[M]. 哈尔滨: 哈尔滨工业大学出版社, 2011.
- [14] 周向红. 51 系列单片机应用与实践教程[M]. 北京: 航天航空大学, 2021. 5.
- [15] 郭天祥. 51 单片机 C 语言教程[M]. 北京: 电子工业出版社, 2021, 12.
- [16] 谭浩强. C 语言设计[M]. 北京: 清华大学出版社, 2005. 7.

- [17] 建领. 51 单片机开发宝典[M]. 北京: 电子工业出版社, 2021.
- [18] 徐益民. 步进电机的单片机控制系统的设计[D]. 煤矿机械, 2005.
- [19] 马忠梅, 籍顺心. 单片机应用程序设计[M]. 北京: 北京航空航天大学出版社, 2006: 92-93.
- [20] 马忠梅, 等. 单片机的 C 语言应用程序设计(第 3 版)[M]. 北京: 北京航空航天大学出版社, 2003.
- [21] 刘振栋. 步进电机接口电路的设计[J]. 电测与仪表, 1999(9):55-65.
- [22] 李广弟, 朱月秀, 王秀山. 单片机基础(修订本)[M]. 北京: 北京航空航天大学出版社.
- [23] 何福庆, 曹养书, 罗小兵. 可预置步进电机驱动器[J]. 成都大学学报(自然科学版), 2002, 21(1): 17~20.
- [24] 素行. 模拟电子技术简明教程[M]. 北京: 高等教育, 2021.7.
- [25] Baluta, Gheorghe. "Microstepping mode for stepper motor control." 2007 International Symposium on Signals, Circuits and Systems. Vol. 2. IEEE, 2007.
- [26] Zribi, M., and J. Chiasson. "Position control of a PM stepper motor by exact linearization." IEEE Transactions on automatic control 36.5 (1991): 620-625.
- [27] Virgala, Ivan, et al. "Control of stepper motor by microcontroller." Journal of Automation and Control 3.3 (2015): 131-134.
- [28] Stoianovici, Dan, et al. "A new type of motor: pneumatic step motor." IEEE/ASME Transactions On Mechatronics 12.1 (2007): 98-106.
- [29] Condit, Reston, and Douglas W. Jones. "Stepping motors fundamentals." Microchip Inc. Publication AN907 (2004): 1-22.

附录 1 元器件清单

序号	元器件名称	型号或参数	数量
1	单片机	stc89c51	1 个
2	液晶显示屏	Lcd1602	1 个
3	步进电机	28YBJ-4828BYJ48	1 个
4	降压芯片	Lm2596	1 个
5	色环电感	33uh	1 个
6	圆孔 ic 插座	40p	1 个
7	排针	2.54mm	20 个
8	电解电容	470u	1 个
9	电解电容	220u	1 个
10	直针座	5p	2 个
11	排阻	1k	1 个
12	排母	2.54mm	2 个
13	拨码开关	1p	10 个
14	二极管	Sr5200	2 个
15	晶振	11.0592mhz	1 个
16	电阻	1KΩ	3 个
17	电阻	3.3KΩ	1 个
18	电阻	10KΩ	2 个
19	电感	33uh	1 个
20	晶振	36.725KHz	1 个
21	电机驱动	ULn2003	1 个
22	电源接口		1 个
23	LED		5 个
24	12V 锂电池		1 个
25	按键		1 个

附录 3 电路 pcb



附录 4 完整代码

```
#include "reg52.h"

typedef unsigned char u8;
typedef unsigned int ul6;

sbit e_1602=P2^7;
sbit rw_1602=P2^5;
sbit rs_1602=P2^6;

sbit IN1_D=P1^0;
sbit IN1_C=P1^1;
sbit IN1_B=P1^2;
sbit IN1_A=P1^3;

sbit KEY1=P3^0;//反转
sbit KEY2=P2^1;//一档
sbit KEY3=P2^2;//二档
sbit KEY4=P2^3;//三档
sbit KEY5=P3^4;//启停
sbit KEY6=P3^5;//第一个位置
sbit KEY7=P3^6;//第二个位置
sbit KEY8=P3^7;//第三个位置
sbit KEY9=P3^1;//第四个位置
sbit KEY10=P3^3;//复位触发
sbit KEY11=P3^2;//复位检测

char speed=30;//速度控制
char local=0;//角度控制
char start=0;//启停标志位
unsigned int int_counter = 0;//中断计数器

unsigned char bjrigh[] = {0x0e, 0x0c, 0x0d, 0x09, 0x0b, 0x03, 0x07, 0x06};

//void ext_interrupt_init()
//{
//    // 设置外部中断 0 的触发方式和使能
```

```

//    IT0 = 0;    // 设置外部中断 0 为上升沿触发
//    EX0 = 1;    // 使能外部中断 0

////    // 设置外部中断 1 的触发方式和使能
////    IT1 = 0;    // 设置外部中断 1 为上升沿触发
////    EX1 = 1;    // 使能外部中断 1

//    // 允许中断总开关
//    EA = 1;
//}

//复位检测的中断函数
void ext_interrupt() interrupt 0
{
    if(KEY10==1) start=1;    // 步进电机停止运行
    if(KEY10==0) start=0;
}

void delay100us(unsigned char b)
{
    unsigned char a;
    for(;b>0;b--)
        for(a=43;a>0;a--);
}

void delay(u16 ms)
{
    u16 i, j;
    for(i=ms;i>0;i--)
        for(j=114;j>0;j--);
}

void write_1602(unsigned char order_data, char i)
//i==1:data, i==0:order    //写 数据 1 命令 0
{
    e_1602=0;                                //write_1602(0x80, 0);指针
(光标)指向一行一个(一行 80----80+27;;二行 80+40-----40+27)
    rs_1602=i;                                //0x01 清屏 0
    P0=order_data;
    e_1602=1;                                //' 1' ' 2' ' d' ' b' 1 显
示相应字符,其他的查看手册
    delay(1);
    e_1602=0;

```

```

}
void init_1602() //初始化
{
    rw_1602=0;          //写
    write_1602(0x38,0);  //液晶设置不判忙模式
    delay(15);
    write_1602(0x38,0);
    delay(5);
    write_1602(0x0c,0);  //光标设置 0000 1dcb d:1:开 0:关 c:1:开光标
                          0:关光标 b:1:闪烁 0:不闪烁
    write_1602(0x06,0);  //0000 0lns n:1:每次指针加一（光标后移）0:
                          减一 s:整屏左移（n=1）
}

```

```

void byj(unsigned char right,unsigned char delaybai)
{

    char i;

    if(right==1&&KEY6==0&&KEY7==0&&KEY8==0&&KEY9==0)
    {
        if(i>7) i=0;
        P1=bjright[i++];
    }
    else if(right==0&&KEY6==0&&KEY7==0&&KEY8==0&&KEY9==0)
    {
        if(i<0) i=7;
        P1=bjright[i--];
    }

    delay100us(delaybai);
}

```

```

// 定义中断计数器
volatile unsigned int timer_counter = 0;

```

```

// 定义中断响应函数
void timer_interrupt() interrupt 3
{
    TH0 = (65536-1000)/256;    // 设置定时器初值高 8 位

```

```

    TL0 = (65536-1000)%256;
    timer_counter++;    // 计数器加 1

    //lcd1602 进行显示
    if(timer_counter==200){
        write_1602(0x81,0);
        write_1602('Q',1);
        write_1602(':',1);
        if(KEY1)write_1602('1',1);
        else    write_1602('0',1);
        write_1602('-',1);
        write_1602('S',1);
        write_1602(':',1);
        write_1602(0x30+speed/10,1);
        write_1602(0x30+speed%10,1);
        timer_counter=0;
    }

}

void timer_init()
{
    // 设置定时器模式和计数初值
    //TMOD &= 0xF0;    // 清零定时器模式位
    TMOD |= 0x01;    // 设置定时器为模式 1
    TH0 = (65536-1000)/256;    // 设置定时器初值高 8 位
    TL0 = (65536-1000)%256;    // 设置定时器初值低 8 位

    // 设置定时器中断使能和优先级
    EA = 1;    // 允许中断总开关
    ET1 = 1;    // 允许定时器 1 中断
    // 启动定时器
    TR1 = 1;    // 启动定时器 1
}

void key_scan(void)//按键检测函数
{
    if(KEY2)    speed=60;
    if(KEY3)    speed=25;
    if(KEY4)    speed=10;

    if(KEY6==1||KEY7==1||KEY8==1||KEY9==1) { start=1;} //位置信息检测
    // if(KEY6==0&&KEY7==0&&KEY7==0&&KEY7==0) { local=0; }
}

```

```

void main()
{
    u16 j=0;
    char i;
    init_1602();
    P3=0x00;
    P2=0x00;
    timer_init();
    //ext_interrupt_init(); //外部中断初始化

    while(1)
    {
        if(start==1) {}
        else{

            key_scan(); //按键信号检测
            byj(KEY1, speed); //电机控制
        }
        if(KEY5) {start=1;} //停止转动
        if(KEY5==0) {start=0;};

        //演示代码

        if(KEY10==1&&KEY11==1)
        {
            start=1; //停止转动
            delay100us(40000); //等待两秒

            //转动到第一个位置，60 度
            for(j=0; j<8*8*64/6; j++)
            {
                if(i<0) i=7;
                P1=bjright[i--];
                delay100us(10);
            }
            delay(2000); //等待两秒

            //转动到第二个位置，120 度
            for(j=0; j<8*8*64/6; j++)
            {
                if(i<0) i=7;
                P1=bjright[i--];

```

```

delay100us(10);

    }
delay(2000); //等待两秒

//转动到第三个位置, 180 度
for(j=0; j<8*8*64/6; j++)
    {
        if(i<0) i=7;
        P1=bjright[i--];
        delay100us(10);

    }
delay(2000); //等待两秒

//转动到第四个位置, 270 度
for(j=0; j<8*8*64/4; j++)
    {
        if(i<0) i=7;
        P1=bjright[i--];
        delay100us(10);

    }
delay(2000); //等待两秒

//回到原点
for(j=0; j<8*8*64/4; j++)
    {
        if(i<0) i=7;
        P1=bjright[i--];
        delay100us(10);

    }
delay(2000); //等待两秒

//转动到第三个位置, 180 度
for(j=0; j<8*8*64/2; j++)
    {
        if(i<0) i=7;
        P1=bjright[i--];
        delay100us(10);

    }
delay(2000); //等待两秒

```



```

//转动到第二个位置，120 度
for(j=0;j<8*8*64/6;j++)
{
    if(i>7)    i=0;
    P1=bjright[i++];
    delay100us(10);

}
delay(2000); //等待两秒

```

```

//转动到第一个位置，60 度
for(j=0;j<8*8*64/6;j++)
{
    if(i>7)    i=0;
    P1=bjright[i++];
    delay100us(10);

}
delay(2000); //等待两秒

```

```

//回到原点
for(j=0;j<8*8*64/6;j++)
{
    if(i>7)    i=0;
    P1=bjright[i++];
    delay100us(10);

}

delay(6000); //等待两秒

```

```

}

```

```

//      if(KEY6==1)
//      {
//
//          start=1;
//          for(j=0;j<8*8*64/6;j++)
//          {
//              if(i<0) i=7;
//              P1=bjright[i--];
//
//          }
//      }

```

```

//                                delay100us(10);
//
//                                }
//
//                                }
//
//                                if(KEY7==1)
//                                {
//                                    start=1;
//                                    for(j=0;j<8*8*64/3;j++)
//                                    {
//                                        if(i<0) i=7;
//                                        P1=bjright[i--];
//                                        delay100us(10);
//                                    }
//
//
//                                }
//                                if(KEY8==1)
//                                {
//                                    start=1;
//                                    for(j=0;j<8*8*64*3/4;j++)
//                                    {
//                                        if(i<0) i=7;
//                                        P1=bjright[i--];
//                                        delay100us(10);
//                                    }
//
//                                }
//                                if(KEY9==1)
//                                {
//                                    start=1;
//                                    for(j=0;j<8*8*64/2;j++)
//                                    {
//                                        if(i<0) i=7;
//                                        P1=bjright[i--];
//                                        delay100us(10);
//                                    }
//
//                                }
//                                if(KEY10==1&&KEY11==1){start=1;};

```

}

致谢

行文至此，我的大学生涯也算告一段落了，回想大学四年，自己一直处于浮躁的状态，一度迷茫到怀疑自己，幸运的是自己的运气还算可以，遇到了一群很好的朋友，感谢一直以来支持帮助我的家人，同学，朋友，路很长慢慢走，我相信总有一天会到达终点的，不管走到哪一步希望自己归来仍少年！