1 引言

1．1 课题研究的目的和意义

我们对于步进电机的控制是通过步进电机的驱动器来完成的，因此步进电机的驱动器已经被广泛用于控制系统中。步进电动机是用电脉冲信号进行控制，将电脉冲信号转换成相应的角位移或线位移的微电动机，它最突出的优点是可以在宽广的频率范围内通过改变脉冲频率来实现调速，快速起停、正反转控制及制动等。并且用其组成的开环系统既简单、廉价，又非常可行，因此在打印机等办公自动化设备以及各种控制装置等众多领域有着极其广泛的应用。随着微电子和计算机技术的发展，步进电动机的需求量会与日俱增，因此研制步进电机驱动器及其控制系统具有十分重要的意义。

1.2 国内外研究概况

早在 19 世纪 30 年代就出现了步进电机[1]，那时候的步进电机只是一种自由旋转的电磁铁，工作原理如同今天的反应式步进电机。我国早在 1958 年就研制成功了反应式步进电机，并且开始将其应用于数控机床。上世纪 60 年代中期，人们发明了永磁式和混合式步进电机，不断提高效率和降低功耗。

国外方面，1952 年美国麻省理工学院首先研制成三坐标铣床，传统的电动机也已经不能满足要求，为了适应这些要求，研制了一系列新的具备控制功能的电动机系列，其中具有步进电机。

步进电机的发展与计算机工业密切相关。自从步进电机在计算机外围设备中取代了小型直流电动机以后，使其设备的性能提高，很快地促进步进电机的发展。另一方面，微型计算机和数字控制技术的发展，又将作为数控执行部件的步进电机推广应用于其他领域，如加工机床，医疗仪器等。

1.3 主要研究内容

本论文所选的步进电机是四相步进电机，采用的方法是利用单片机控制步进电机的驱动。步进电机是将电脉冲信号转变为角位移或线位移的开环控制元步进电机件。在非超载的情况下，电机的转速、停止的位置只取决于脉冲信号的频率和脉冲数，而不受负载变化的影响，当步进驱动器接收到一个脉冲信号，它就驱动步进电机按设定的方向转动一个固定的角度，称为“步距角”，它的旋转是以固定的角度一步一步运行的。可以

通过控制脉冲个数来控制角位移量，从而达到准确定位的目的；同时可以通过控制脉冲频率来控制电机转动的速度和加速度，从而达到调速的目的。[3]。本次毕业设计就是通过改变脉冲频率来同步调节两台步进电机，结合 CNC,用数字控制技术使步进电机实现分别沿着 X 轴和 Y 轴进行调速和转向的改变。并且通过 LCD12864 液晶显示器来显示姓名和专业、转向和速度等级（2、4、6 等级）、学号、电动机转动圈数[2]。由 IO 口通过驱动芯片 ULN2003 控制双步进电机的运行状态，且两台步进电机能按直线插补算法联动运行，并能绘制一条直线段。另外通过单片机实现它的正反转，步进电机可以作为一种控制用的特种电机，利用其没有积累误差(精度为 100%)的特点，广泛应用于各种开环控制。

2 系统设计方案与器件介绍

2.1 系统总体设计

如图 2-1 所示，该设计完成基于 89C52 单片机的步进电机控制系统，该系统有软件设计和硬件设计相结合来完成。硬件是整个系统的生命，软件赋予了其灵魂。整个系统设计流程分两步：第一步，硬件设计。运用 Protell 软件完成各个外围电路与单片机的接口设计，包括各器件的型号，管脚分配，元器件的封装形式。第二步，软件设计。运用 Keil 集成开发软件平台完成系统的控制功能和显示功能。

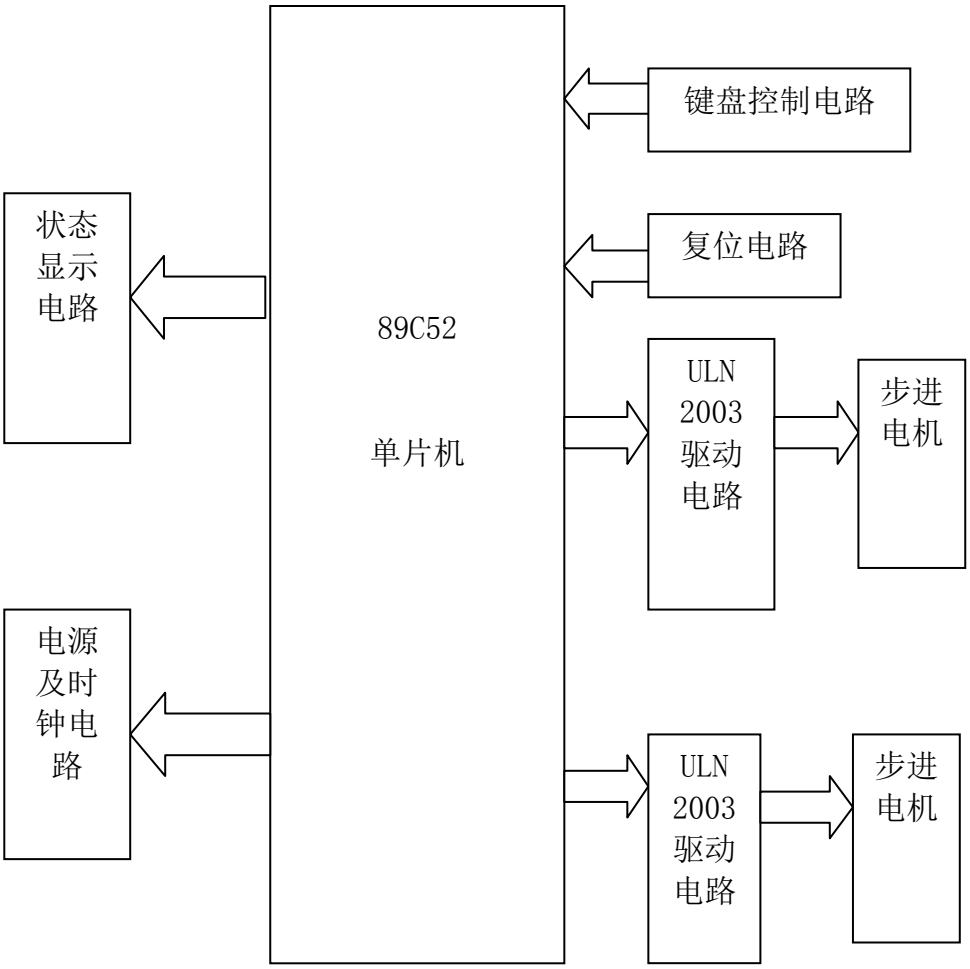


图 2-1 系统总体设计框图

2.2 步进电机

2.2.1 步进电机基本介绍

步进电机是一种感应电机[3]，他的工作原理是利用电子电路，将直流电变成分时供电的，多相时序控制电流，用这种电流为步进电机供电，步进电机才能正常工作，驱动

器就是为步进电机分时供电的，多相时序控制器。如图 2-2 所示。

图 2-2 步进电机

虽然步进电机已被广泛地应用，但步进电机并不能像普通的直流电机，交流电机在常规下使用。它必须由双环形脉冲信号、功率驱动电路等组成控制系统方可使用。因此用好步进电机却非易事，它涉及到机械、电机、电子及计算机等许多专业知识。

作为执行元件，步进电机是机电一体化的关键产品之一，广泛应用在各种自动化控制系统中。随着微电子和计算机技术的发展，步进电机的需求量与日俱增，在各个国民经济领域都有应用。

步进电机是一种将电脉冲转化为角位移的执行机构。通俗一点讲：当步进驱动器接收到一个脉冲信号，它就驱动步进电机按设定的方向转动一个固定的角度（即步进角）。可以通过控制脉冲个数来控制角位移量，从而达到准确定位的目的；同时可以通过控制脉冲频率来控制电机转动的速度和加速度，从而达到调速的目的。

1、静态指标术语

(1)相数：产生不同对极 N、S 磁场的激磁线圈对数。常用 m 表示。

(2)拍数：完成一个磁场周期性变化所需脉冲数或导电状态用 n 表示，或指电机转过一个齿距角所需脉冲数，以四相电机为例，有四相四拍运行方式即 AB-BC-CD-DA-AB,四相八拍运行方式即 A-AB-B-BC-C-CD-D-DA-A。

(3)步距角：对应一个脉冲信号，电机转子转过的角位移用 θ 表示。θ=360 度/（转子齿数*运行拍数），以常规二、四相，转子齿为 50 齿电机为例。四拍运行时步距角为 θ=360 度/（50*4）=1.8 度（俗称整步），八拍运行时步距角为 θ=360 度/（50*8）=0.9 度（俗称半步）。

(4)定位转矩：电机在不通电状态下，电机转子自身的锁定力矩（由磁场齿形的谐波以及机械误差造成的）。

(5)静转矩：电机在额定静态电作用下，电机不作旋转运动时，电机转轴的锁定力矩。此力矩是衡量电机体积的标准，与驱动电压及驱动电源等无关。 虽然静转矩与电磁激磁安匝数成正比，与定齿转子间的气隙有关，但过分采用减小气隙，增加激磁安匝来提高静力矩是不可取的，这样会造成电机的发热及机械噪音。

2、动态术语指标

(1)步距角精度：步进电机每转过一个步距角的实际值与理论值的误差。用百分比表示：误差/步距角*100%。不同运行拍数其值不同，四拍运行时应在 5%之内，八拍运行时应在 15%之内。

(2)失步：电机运转时运转的步数，不等于理论上的步数。称之为失步。

(3)失调角：转子齿轴线偏移定子齿轴线的角度，电机运转必存在失调角，由失调角产生的误差，采用细分驱动是不能解决的。

(4)最大空载起动频率：电机在某种驱动形式、电压及额定电流下，在不加负载的情况下，能够直接起动的最大频率。

(5)最大空载的运行频率：电机在某种驱动形式，电压及额定电流下，电机不带负载的最高转速频率。

(6)运行矩频特性：电机在某种测试条件下测得运行中输出力矩与频率关系的曲线称为运行矩频特性，这是电机诸多动态曲线中最重要的，也是电机选择的根本依据。电机一旦选定，电机的静力矩确定，而动态力矩却不然，电机的动态力矩取决于电机运行时的平均电流（而非静态电流），平均电流越大，电机输出力矩越大，即电机的频率特性越硬。要使平均电流大，尽可能提高驱动电压，采用小电感大电流的电机。

(7)电机的共振点：步进电机均有固定的共振区域，二、四相感应子式的共振区一般在 180-250pps 之间（步距角 1.8 度）或在 400pps 左右（步距角为 0.9 度），电机驱动电压越高，电机电流越大，负载越轻，电机体积越小，则共振区向上偏移，反之亦然，为使电机输出电矩大，不失步和整个系统的噪音降低，一般工作点均应偏移共振区较多。

(8)电机正反转控制：当电机绕组通电时序为 AB-BC-CD-DA 时为正转，通电时序为 DA-CD-BC-AB 时为反转

3、特点特性

（1）一般步进电机的精度为步进角的 3-5%，且不累积。

（2）步进电机外表允许的最高温度。

步进电机温度过高首先会使电机的磁性材料退磁，从而导致力矩下降乃至于失步，因此电机外表允许的最高温度应取决于不同电机磁性材料的退磁点；一般来讲，磁性材料的退磁点都在摄氏 130 度以上，有的甚至高达摄氏 200 度以上，所以步进电机外表温度在摄氏 80-90 度完全正常。

（3）步进电机的力矩会随转速的升高而下降。

当步进电机转动时，电机各相绕组的电感将形成一个反向电动势；频率越高，反向电动势越大。在它的作用下，电机随频率（或速度）的增大而相电流减小，从而导致力矩下降。

（4）步进电机低速时可以正常运转,但若高于一定速度就无法启动,并伴有啸叫声。

步进电机有一个技术参数：空载启动频率，即步进电机在空载情况下能够正常启动的脉冲频率，如果脉冲频率高于该值，电机不能正常启动，可能发生丢步或堵转。在有负载的情况下，启动频率应更低。如果要使电机达到高速转动，脉冲频率应该有加速过程，即启动频率较低，然后按一定加速度升到所希望的高频（电机转速从低速升到高速）。

2.2.2 步进电机的工作原理

通常电机的转子为永磁体，当电流流过定子绕组时，定子绕组产生一矢量磁场[8]。该磁场会带动转子旋转一角度，使得转子的一对磁场方向与定子的磁场方向一致。当定子的矢量磁场旋转一个角度。转子也随着该磁场转一个角度。每输入一个电脉冲，电动机转动一个角度前进一步。它输出的角位移与输入的脉冲数成正比、转速与脉冲频率成正比。改变绕组通电的顺序，电机就会反转。所以可用控制脉冲数量、频率及电动机各相绕组的通电顺序来控制步进电机的转动。

如下所示的步进电机为一四相步进电机，采用单极性直流电源供电。只要对步进电机的各相绕组按合适的时序通电，就能使步进电机步进转动。图 2-3 是该四相反应式步进电机工作原理示意图。

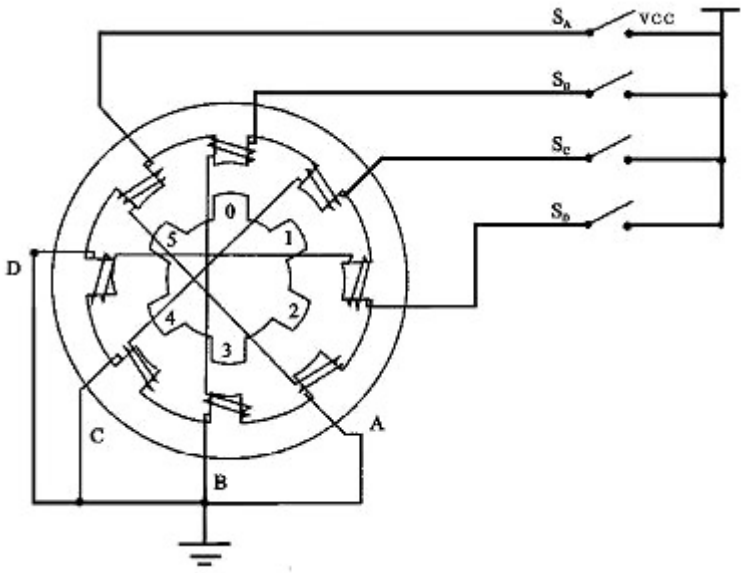


图 2-3 四相步进电机示意图

开始时，开关 SB 接通电源，SA、SC、SD 断开，B 相磁极和转子 0、3 号齿对齐，同时，转子的 1、4 号齿就和 C、D 相绕组磁极产生错齿，2、5 号齿就和 D、A 相绕组磁极产生错齿。

当开关 SC 接通电源，SB、SA、SD 断开时，由于 C 相绕组的磁力线和 1、4 号齿之磁力线的作用，使转子转动，1、4 号齿和 C 相绕组的磁极对齐。而 0、3 号齿和 A、B 相绕组产生错齿，2、5 号齿就和 A、D 相绕组磁极产生错齿。依次类推，A、B、C、D

四相绕组轮流供电，则转子会沿着 A、B、C、D 方向转动。

单四拍、双四拍与八拍工作方式的电源通电时序与波形分别如图 2-4 所示：

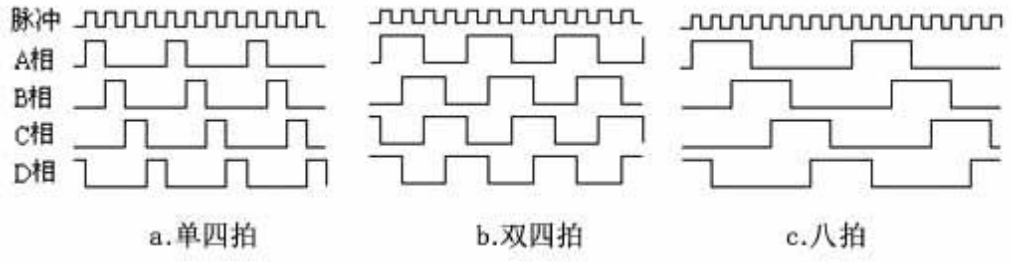


图 2-4 步进电机工作时序波形图

2.2.3 步进电机的分类与选择

步进电机在构造上有三种主要类型：反应式（Variable Reluctance，VR）、永磁式（Permanent Magnet,PM）和混合式（Hybrid Stepping,HS）。

反应式：定子上有绕组、转子由软磁材料组成。结构简单、成本低、步距角小，可

达 1.2°、但动态性能差、效率低、发热大，可靠性难保证。

永磁式：永磁式步进电机的转子用永磁材料制成，转子的极数与定子的极数相同。其特点是动态性能好、输出力矩大，但这种电机精度差，步矩角大（一般为 7.5°或 15°）。

混合式：混合式步进电机综合了反应式和永磁式的优点，其定子上有多相绕组、转子上采用永磁材料，转子和定子上均有多个小齿以提高步矩精度。其特点是输出力矩大、动态性能好，步距角小，但结构复杂、成本相对较高。

步进电机有步距角（涉及到相数）、静转矩、及电流三大要素组成。一旦三大要素确定，步进电机的型号便确定下来了。

1、步距角的选择

电机的步距角取决于负载精度的要求，将负载的最小分辨率（当量）换算到电机轴上，每个当量电机应走多少角度（包括减速）。电机的步距角应等于或小于此角度。目前市场上步进电机的步距角一般有 0.36 度/0.72 度（五相电机）、0.9 度/1.8 度（二、四相电机）、1.5 度/3 度　（三相电机）等。

2、静力矩的选择

步进电机的动态力矩一下子很难确定，我们往往先确定电机的静力矩。静力矩选择的依据是电机工作的负载，而负载可分为惯性负载和摩擦负载二种。单一的惯性负载和单一的摩擦负载是不存在的。直接起动时（一般由低速）时二种负载均要考虑，加速起动时主要考虑惯性负载，恒速运行进只要考虑摩擦负载。一般情况下，静力矩应为摩擦负载的 2-3 倍内好，静力矩一旦选定，电机的机座及长度便能确定下来（几何尺寸）。

3、电流的选择

静力矩一样的电机，由于电流参数不同，其运行特性差别很大，可依据矩频特性曲线图，判断电机的电流（参考驱动电源、及驱动电压）。

2.3　ATM89C52

2.3.1　ATMC89C52 概述

ATM89C52 是一种低功耗、高性能 CMOS8 为微控制器，具有 8K 在系统可编程 Flash 存贮器[4]。使用 Atmel 公司高密度非易失性存贮器技术制造，与工业 80C51 产品指令和引脚完全兼容。片上 Flash 允许程序存储器在系统可编程，亦适于常规编程器。在单芯

片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash，使得 AT89S52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。 AT89S52 具有以下标准功能： 8k 字节 Flash，256 字节 RAM， 32 位 I/O 口线，看门狗定时器，2 个数据指针，三个 16 位 定时器/计数器，一个 6 向量 2 级中断结构，全双工串行口， 片内晶振及时钟电路。另外，AT89S52 可降至 0Hz 静态逻 辑操作，支持 2 种软件可选择节电模式。空闲模式下，CPU 停止工作，允许 RAM、定时器/计数器、串口、中断继续工作。掉电方式保护下，RAM 内容被保存，振荡器呗冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。引脚排列图如图 2-5 所示：

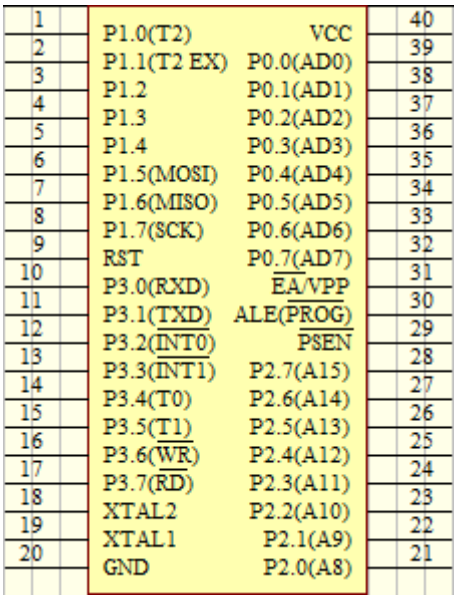


图 2-5 单片机引脚排列图

2.3.2 ATM89C52 主要性能

1、与 MCS-51 单片机产品兼容；　　2、8K 字节在系统可编程 Flash 存储器；

3、1000 次擦写周期；　　4、全静态操作：0Hz-33MHz；

5、三级加密程序存储器；　　6、32 个可编程 I/O 口线；

7、三个 16 位定时器/计数器；　　8、8 个中断源；

9、全双工 UART 串行通道；　　10、低功耗空闲和掉电模式；

11、掉电后中断可唤醒；　　12、看门狗定时器；

13、双数据指针；　　14、掉电标识符。

2.3.3 ATM89C52 引脚说明

1、P0 口

P0 口是一个 8 位漏极开路的双向 I/O 口。作为输出口，每位能驱动 8 个 TTL 逻 辑电平。对 P0 端口写“1”时，引脚用作高阻抗输入。 当访问外部程序和数据存储器时，P0 口也被作为低 8 位地址/数据复用。在这种模式下， P0 不具有内部上拉电阻。 在 flash 编程时，P0 口也用来接收指令字节；在程序校验时，输出指令字节。程序校验 时，需要外部上拉电阻。

2、P1 口

P1 口是一个具有内部上拉电阻的 8 位双向 I/O 口,p1 输出缓冲器能驱动 4 个 TTL 逻辑电平。

此外，P1.0 和 P1.1 分别作定时器/计数器 2 的外部计数输入（P1.0/T2）和定时器/计数器 2 的触发输入（P1.1/T2EX）。 在 flash 编程和校验时，P1 口接收低 8 位地址字节。

引脚号第二功能：

P1.0 T2（定时器/计数器 T2 的外部计数输入），时钟输出

P1.1 T2EX（定时器/计数器 T2 的捕捉/重载触发信号和方向控制）

P1.5 MOSI（在系统编程用）

P1.6 MISO（在系统编程用）

P1.7 SCK（在系统编程用）

3、P2 口

P2 口是一个具有内部上拉电阻的 8 为双向 I/O 口，P2 输出缓冲器能驱动 4 个 TTL 逻辑电平。对 P2 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入 口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流（IIL）。 在访问外部程序存储器或用 16 位地址读取外部数据存储器（例如执行 MOVX @DPTR） 时，P2 口送出高八位地址。在这种应用中，P2 口使用很强的内部上拉发送 1。在使用 8 位地址（如 MOVX @RI）访问外部数据存储器时，P2 口输出 P2 锁存器的内容。 在 flash 编程和校验

时，P2 口也接收高 8 位地址字节和一些控制信号。

4、P3 口

P3 口是一个具有内部上拉电阻的 8 位双向 I/O 口,p3 输出缓冲器能驱动 4 个 TTL 逻辑电平。 P3 口亦作为 AT89S52 特殊功能（第二功能）使用，如下表所示。 在 flash 编程和校验时，P3 口也接收一些控制信号。

端口引脚第二功能：

P3.0 RXD(串行输入口)

P3.1 TXD(串行输出口)

P3.2 INT0(外中断 0)

P3.3 INT1(外中断 1)

P3.4 T0(定时/计数器 0)

P3.5 T1(定时/计数器 1)

P3.6 WR(外部数据存储器写选通)

P3.7 RD(外部数据存储器读选通)

此外，P3 口还接收一些用于 FLASH 闪存编程和程序校验的控制信号。

5、RST

复位输入。当振荡器工作时,RST 引脚出现两个机器周期以上高电平将是单片机复位。

6、ALE/PROG

当访问外部程序存储器或数据存储器时，ALE（地址锁存允许）输出脉冲用于锁存地址的低 8 位字节。一般情况下，ALE 仍以时钟振荡频率的 1/6 输出固定的脉冲信号，因此它可对外输出时钟或用于定时目的。要注意的是：每当访问外部数据存储器时将跳过一个 ALE 脉冲。对 FLASH 存储器编程期间，该引脚还用于输入编程脉冲（PROG）。如有必要，可通过对特殊功能寄存器（SFR）区中的 8EH 单元的 D0 位置位，可禁止 ALE 操作。该位置位后，只有一条 MOVX 和 MOVC 指令才能将 ALE 激活。此外，该引脚会被微弱拉高，单片机执行外部程序时，应设置 ALE 禁止位无效。

7、PSEN

程序储存允许（PSEN）输出是外部程序存储器的读选通信号，当 AT89S52 由外部程序存储器取指令（或数据）时，每个机器周期两次 PSEN 有效，即输出两个脉冲，在此期间，当访问外部数据存储器，将跳过两次 PSEN 信号。

8、EA/VPP

外部访问允许，欲使 CPU 仅访问外部程序存储器（地址为 0000H-FFFFH），EA 端必须保持低电平（接地）。需注意的是：如果加密位 LB1 被编程，复位时内部会锁存 EA 端状态。如 EA 端为高电平（接 VCC 端），CPU 则执行内部程序存储器的指令。FLASH 存储器编程时，该引脚加上+12V 的编程允许电源 VPP，当然这必须是该器件是使用 12V 编程电压 VPP。

9、XTAL1

振荡器反相放大器和内部时钟发生电路的输入端。

10、XTAL2

振荡器反相放大器的输出端

2.4 LCD12864 液晶显示器

2.4.1 LCD12864 概述

带中文字库的 128X64 是一种具有 4 位/8 位并行、2 线或 3 线串行多种接口方式，内部含有国标一级、二级简体中文字库的点阵图形液晶显示模块;其显示分辨率为 128×64，内置 8192 个 16*16 点汉字，和 128 个 16*8 点 ASCII 字符集.利用该模块灵活的接口方式和简单、方便的操作指令，可构成全中文人机交互图形界面[5]。可以显示 8×4 行 16×16 点 阵的汉字.也可完成图形显示.低电压低功耗是其又一显著特点。由该模块构成的液晶显示方案与同类型的图形点阵液晶显示模块相比，不论硬件电路结构或显示程序都要简洁得多，且该模块的价格也略低于相同点阵的图形液晶模块。

2.4.2 LCD12864 基本特性

1、低电源电压（VDD:+3.0--+5.5V）；　　2、显示分辨率:128×64 点 ；

3、内置汉字字库，提供 8192 个 16×16 点阵汉字(简繁体可选)；

4、内置 128 个 16×8 点阵字符 ；　　5、2MHZ 时钟频率 ；

6、显示方式：STN、半透、正显 ；　　7、驱动方式：1/32DUTY，1/5BIAS ；

8、视角方向：6 点 ；

9、背光方式：侧部高亮白色 LED，功耗仅为普通 LED 的 1/5—1/10；

10、通讯方式：串行、并口可选；　　11、内置 DC-DC 转换电路，无需外加负压；

12、无需片选信号，简化软件设计；

13、工作温度：0℃ － +55℃ ,存储温度：-20℃ － +60℃.

2.4.3 LCD12864 引脚说明

引脚说明如表 2-1 所示：

表 2-1 引脚功能

引脚	名称	电平	说明	引脚	名称	电平	说明
1	VSS	0	电源地	11	DB4	H/L	三态数据线
2	VCC	3.0-5V	电源正	12	DB5	H/L	三态数据线
3	V0	-	对比度（亮度）调整	13	DB6	H/L	三态数据线
4	RS(CS)	H/L	RS=“H”，表示 DB7——DB0 为显示数据 RS=“L”，表示 DB7——DB0 为显示指令数据	14	DB7	H/L	三态数据线
5	R/W(SID)	H/L	R/W=“H”,E=“H”,数据被读到 DB7——DB0 R/W=“L”,E=“H→L”，DB7——DB0 的数据被写到 IR 或 DR	15	PSB	H/L	H：8 位或 4 位并口方式，L：串口方式（见注释 1）
6	E(SCLK)	H/L	使能信号	16	NC	-	空脚
7	DB0	H/L	三态数据线	17	RESET	H/L	复位端，低电平有效（见注释 2）
8	DB1	H/L	三态数据线	18	VOUT	-	LCD 驱动电压输出端
9	DB2	H/L	三态数据线	19	A	VDD	背光源正端（+5V）（见注释 3）
10	DB3	H/L	三态数据线	20	K	VSS	背光源负端（见注释 3）

注释 1：如在实际应用中仅使用并口通讯模式，可将 PSB 接固定高电平，也可以将模块上的 J8 和“VCC”用焊锡短接。

注释 2：模块内部接有上电复位电路，因此在不需要经常复位的场合可将该端悬空。

注释 3：如背光和模块共用一个电源，可以将模块上的 JA、JK 用焊锡短接。

2.4.4 LCD12864 模块主要硬件构成说明

1、控制器接口信号说明

（1）RS，R/W 的配合选择决定控制界面的 4 种模式，如表 2-2 所示：

表 2-2 RS、R/W 控制界面模式

RS	R/W	功能说明
L	L	MPU 写指令到指令暂存器（IR）
L	H	读出忙标志（BF）及地址记数器（AC）的状态
H	L	MPU 写入数据到数据暂存器（DR）
H	H	MPU 从数据暂存器（DR）中读出数据

（2）E 信号

E 信号状态结果如表 2-3 所示：

表 2-3 E 信号状态

E 状态	执行动作	结果
高——>低	I/O 缓冲——>DR	配合/W 进行写数据或指令
高	DR——>I/O 缓冲	配合 R 进行读数据或指令
低/低——>高	无动作	

2、忙标志:BF

BF 标志提供内部工作情况.BF=1 表示模块在进行内部操作,此时模块不接受外部指令和数据.BF=0 时,模块为准备状态,随时可接受外部指令和数据.利用 STATUS RD 指令,可以将 BF 读到 DB7 总线,从而检验模块之工作状态。

3、字型产生 ROM（CGROM）

字型产生 ROM（CGROM）提供 8192 个此触发器是用于模块屏幕显示开和关的控制。DFF=1 为开显示（DISPLAY ON),DDRAM 的内容就显示在屏幕上，DFF=0 为关显示（DISPLAY OFF)。DFF 的状态是指令 DISPLAY ON/OFF 和 RST 信号控制的。

4、显示数据 RAM（DDRAM）

模块内部显示数据 RAM 提供 64×2 个位元组的空间,最多可控制 4 行 16 字(64 个字）的中文字型显示，当写入显示数据 RAM 时，可分别显示 CGROM 与 CGRAM 的字型；此模块可显示三种字型，分别是半角英数字型(16*8)、CGRAM 字型及 CGROM 的中文字型，三种字型的选择,由在 DDRAM 中写入的编码选择,在 0000H—0006H 的编码中(其代码分别是 0000、0002、0004、0006 共 4 个）将选择 CGRAM 的自定义字型，02H—7FH 的编码中将选择半角英数字的字型，至于 A1 以上的编码将自动的结合下一个位元组，组成两个位元组的编码形成中文字型的编码 BIG5（A140—D75F），GB（A1A0-F7FFH）。

5、字型产生 RAM(CGRAM)

字型产生 RAM 提供图象定义(造字)功能, 可以提供四组 16×16 点的自定义图象空间，使用者可以将内部字型没有提供的图象字型自行定义到 CGRAM 中，便可和 CGROM 中的定义一样地通过 DDRAM 显示在屏幕中。

6、地址计数器 AC

地址计数器是用来贮存 DDRAM/CGRAM 之一的地址,它可由设定指令暂存器来改变，之后只要读取或是写入 DDRAM/CGRAM 的值时，地址计数器的值就会自动加一，当 RS 为“0”时而 R/W 为“1”时，地址计数器的值会被读取到 DB6——DB0 中。

7、光标/闪烁控制电路

此模块提供硬体光标及闪烁控制电路，由地址计数器的值来指定 DDRAM 中的光标或闪烁位置。

2.5 达林顿 2003

2.5.1 ULN2003 概述

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 COMS 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[6]。

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 内部还集成了一个消线圈反电动势的二极管，可用来驱动继电器。它是双列 16 脚封装，NPN 晶体管矩阵，最大驱动电压=50V，电流=500mA，输入电压=5V，适用于 TTL COMS，由达林顿管组成驱动电路。ULN 是集成达林顿管 IC，内部还集成了一个消线圈反电动势的二极管，它的输出端允许通过电流为 200mA，饱和压降 VCE 约 1V 左右，耐压 BVCEO 约为 36V。用户输出口的外接负载可根据以上参数估算。采用集电极开路输出，输出电流大。故可直接驱动继电器或固体继电器，也可直接驱动低压灯泡。通常单片机驱动 ULN2003 时，上拉 2K 的电阻较为合适，同时，COM 引脚应该悬空或接电源。

2.5.2 ULN2003 参数

参数如表 2-4 所示：

表 2-4 ULN2003 参数表

参数名称	符号	测试条件	最小	典型	最大	单位
输出漏电流	Icex	Vce=50V，Tamb=25℃			50	uA
		Vce=50V，Tamb=70℃			100	
饱和压降	Vce(sat)	Ic=100mA，Is=250uA		0.9	1.1	V
		Ic=200mA，Is=350uA		1.1	1.3	
		Ic=350mA，Is=500uA		1.3	1.6	
输入电流	Iin(on)	Vin=3.85V		0.93	1.35	mA
		Ic=500uA，Tamb=70℃	50	65		uA
输入电压	Vin(on)	Vce=2.0V，Ic=200mA			2.4	V
		Vce=2.0V，Ic=250mA			2.7	
		Vce=2.0V，Ic=300mA			3.0	
输入电容	Cin			15	25	pF
上升时间	Tplh	0.5 Ein to 0.5 Eout		0.25	1.0	uS
下降时间	Tphl	0.5 Ein to 0.5 Eout		0.25	1.0	uS
钳位二极管漏电流	Ir	Vr=50V，Tamb=25℃			50	uA
		Vr=50V，Tamb=70℃			100	

芯片引脚如图 2-5 所示：

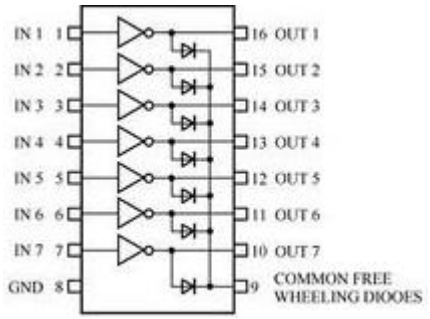


图 2-5 ULN 芯片引脚图

(1)引脚 1-7：CPU 脉冲输入端

(2)引脚 8：接地

(3)引脚 9：该脚是内部 7 个续流二极管负极的公共端，各二极管的正极分别接各达林顿管的集电极。用于感性负载时，该脚接负载电源正极，实现续流作用。如果该脚接地，实际上就是达林顿管的集电极对地接通

(4)引脚 10：脉冲信号输出端，对应 7 脚信号输入端

(5)引脚 11：脉冲信号输出端，对应 6 脚信号输入端

(6)引脚 12：脉冲信号输出端，对应 5 脚信号输入端

(7)引脚 13：脉冲信号输出端，对应 4 脚信号输入端

(8)引脚 14：脉冲信号输出端，对应 3 脚信号输入端

(9)引脚 15：脉冲信号输出端，对应 2 脚信号输入端

(10)引脚 16：脉冲信号输出端，对应 1 脚信号输入端

3 CNC 的应用

3.1 CNC 的介绍

CNC(数控机床)是计算机数字控制机床(Computer numerical control)的简称，是一种由程序控制的自动化机床。该控制系统能够逻辑地处理具有控制编码或其他符号指令规定的程序，通过计算机将其译码，从而使机床执行规定好了的动作，通过刀具切削将毛坯料加工成半成品成品零件。

数控加工智能逆向仿真系统 Virtual CNC， 是一套通过逆向后置处理器和虚拟机床来模拟实际 CNC 控制器和机床，并在电脑端进行检验 CNC 加工过程的软件。它根据机器、刀具、毛坯和夹具信息，来模拟加工 CNC 程序，并能鉴定加工过程中存在的错误。

3.2 数字控制基础

数字控制主要用于数控机床、线切割机、焊接机、气割机以及低速小型数字绘图仪等[6]。采用数字程序控制的机床成为数控机床。数控机床可以加工形状复杂的零件，具有加工精度高、生产效率高、便于改变加工零件品种等众多优点，是机床实现自动化的发展方向。对于不同的设备，其控制系统有所不同，但基本的数字控制原理是相同的。

3.2.1 数字控制的基本原理

平面曲线图形如图5-1所示，下面分析如何利用计算机在绘图仪或加工装置上重现，下面分三步。

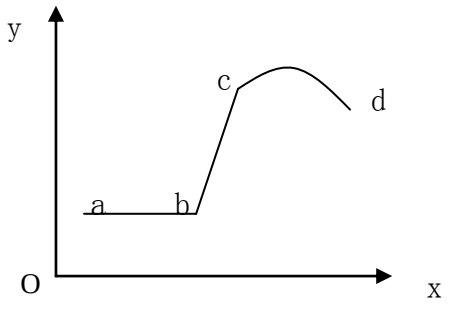


图3-1 曲线分段

1、将此曲线分割成若干线段，可以是直线段，也可是曲线段，把该图分割成3段，即 $\overline{ab}$， $\overline{bc}$ 和弧线，然后把a,b,c,d四点坐标记下来并送给计算机。

2、当给定a,b,c,d各点坐标和x,y值之后，如何确定各坐标值之间的中间值？求得这

些中间值的数值计算方法称为插值或插补。插补计算的宗旨是通过给定的基点，以一定的速度连续定出一系列中间点，而这些中间点的坐标值是以一定的精度逼近给定的线段的。

3、把插补运算过程中定出的各中间点，以脉冲信号形式去控制x,y方向上的步进电机，带动画笔、刀具或线电机运动，从而绘出图形或加工出符合要求的轮廓。这里的每一个脉冲信号代表步进电机走一步，即画笔或刀具在x方向或y方向移动一个位置。我们把对应于每个脉冲移动的相对位置称为脉冲当量，又称为步长，常用 Δx 和 Δy 来表示，并且总是取 $\Delta x=\Delta y$ 。

如图3-2是一段用折线逼近直线的直线插补线段，其中（ x_0 ， y_0 ）代表该线段的起点坐标值，（ x_e ， y_e ）代表终点坐标值，则*x*方向和*y*方向应移动的总步数 N_X 和 N_Y 分别为

$$N_x=\frac{x_e-x_0}{\Delta x} \qquad\qquad N_y=\frac{y_e-y_0}{\Delta y}$$

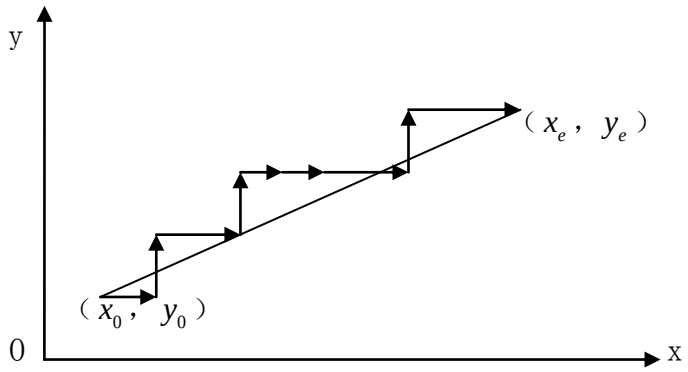


图3-2 用折线逼近直线段

如果把 Δx 和 Δy 约定为坐标增量值，即 X_0，Y_0， X_e， Y_e 均是以脉冲当量定义的坐标值，则

$$N_X=X_e-X_0 \qquad\qquad N_Y=Y_e-Y_0$$

3.2.2 数字控制方式

数控系统按控制方式来分类，可以分为点位控制、直线切削控制和轮廓切削控制，以上三种控制方式均是运动的轨迹控制[7]。

1．点位控制

在一个点位控制系统中，只要求控制刀具行程终点的坐标值，即工件加工点准确定位，至于刀具从一个加工点移到下一个加工点走什么路径、移动的速度、沿哪个方向都

无需规定，并在移动过程中不做任何加工，只是在准确到达指定位置后才开始加工。在机床加工行业，钻床、冲床、镗床采用这类控制。

2．直线切削控制

这种控制也主要是控制行程的终点坐标值，不过还要求刀具相对于工件平行某一直角坐标轴作直线运动，且在运动过程中进行切削加工。

3．轮廓的切削控制

这类控制的特点是能够控制刀具沿工件轮廓曲线不断地运动，并在运动过程中将工件加工成某一形状。这种方式是借助于插补器进行的，插补器根据加工的工件轮廓向每一坐标轴分配速度指令，以获得图纸坐标点之间的中间点。

3.2.3 数字控制系统

计算机数控系统主要分为开环数字控制和闭环数字控制两大类，由于它们的控制原理不同，因此其系统结构差异很大。

1．闭环数字控制

闭环数字控制的结构图如图3-3所示。这种结构的执行机构多采用直流电压（小惯量伺服电机和宽调速力矩电机）作为驱动元件，反馈测量元件采用光电编码器（码盘）、光栅、感应同步器等，该控制方式主要用于大型精密加工机床，但其结构复杂，难于调整和维护，一些常规的数控系统很少采用。

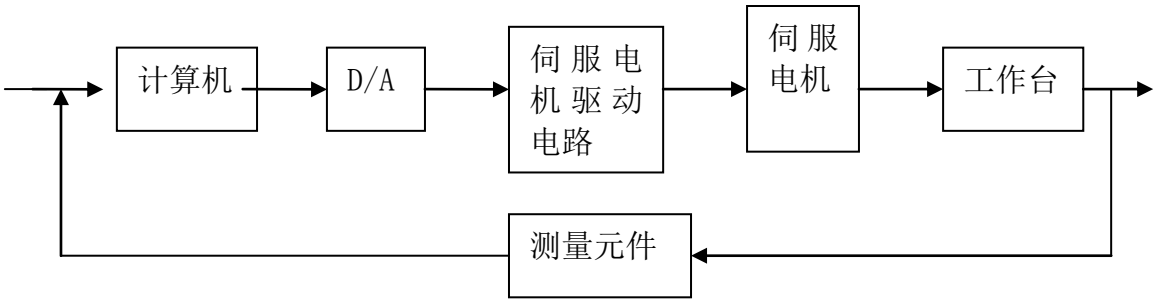


图3-3 闭环数字控制

2．开环数字控制

开环数字控制的结构如图3-4所示，这种控制结构没有反馈检测元件，工作台由步进电机驱动。步进电机接收步进电机驱动电路发来的指令脉冲作相应的旋转，把刀具移动到与指令脉冲相当的位置，至于刀具是否到达了指令脉冲规定的位置，那是不受任何检

查的，因此这种控制的可靠性和精度基本上由步进电机和传动装置来决定。

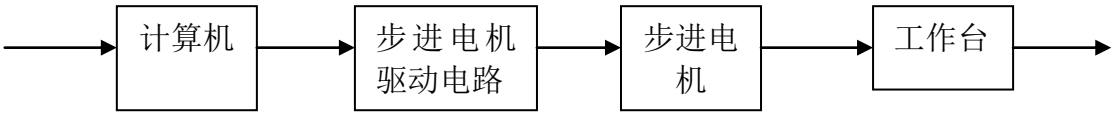


图3-4 开环数字控制

3.3 逐点比较法插补原理

3.3.1 逐点比较法插补介绍

所谓逐点比较法插补，就是刀具或绘图笔每走一步都要和给定轨迹上的坐标值进行比较，看这点在给定轨迹的上方或下方，或是给定轨迹的里面或外面，从而决定下一步的进给方向。

逐点比较法是以阶梯折线来逼近直线或圆弧等曲线的，它与规定的加工直线或圆弧之间的最大误差为一个脉冲当量，因此只要把脉冲当量（每走一步的距离即步长）取得足够小，就可达到加工精度的要求。本设计主要运用第一象限内的直线插补，因此下面主要介绍其插补原理。

3.3.2 第一象限内的直线插补

1、直线插补计算原理

假设加工的轨迹为第一象限中的一条直线*OA*，如图3-5所示。

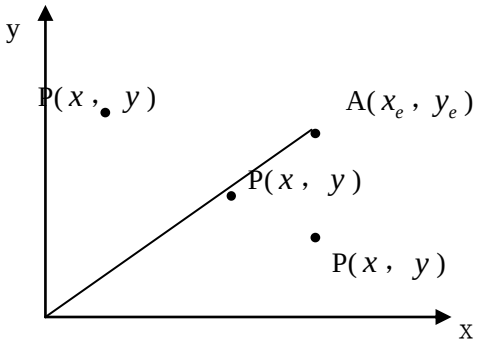


图3-5 第一象限直线插补

在图3-5中，坐标起点为O(0，0)，坐标终点为A(x_e，y_e)。设刀具位于点P(x，y)，则有下述三种情况：

（1）P点在直线OA上，则OP与OA重合，它们的斜率相等，有

$$y / x = y_e / x_e$$

可改写为：

$$y\ x_e = y_e\ x$$

得

$$y\ x_e - y_e\ x = 0 \qquad (3\text{-}1)$$

（2）P点在直线OA上方，OP的斜率大于OA，有

$$y / x > y_e / x_e$$

即

$$y\ x_e > y_e\ x$$

可改写为：

$$y\ x_e - y_e\ x > 0 \qquad (3\text{-}2)$$

（3）*P*点在直线*OA*下方，则有

$$y / x < y_e / x_e$$

即

$$y\ x_e < y_e\ x$$

可改写为：

$$y\ x_e - y_e\ x < 0 \qquad (3\text{-}3)$$

现用F来表示P点的偏差量，定义

$$F = y\ x_e - y_e\ x \qquad (3\text{-}4)$$

则当F=0时P点位于直线OA上；

F>0时P点位于直线OA的上方；

F<0时P点位于直线OA的下方。

这样，根据*F*值的大小，就可以控制刀具的进给方向，如图3-6所示。

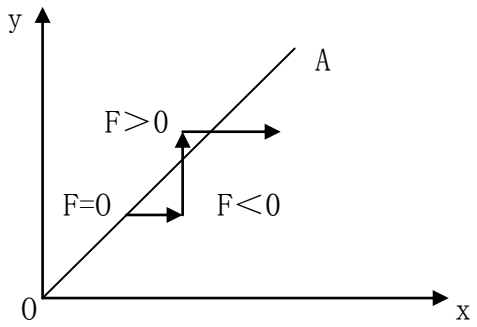


图3-6 进给方向的规定

当F=0时，可以向$+x$方向走一步，也可向$+y$方向走一步，通常规定为向$+x$方向走一步。

当F>0时，控制刀具向$+x$方向走一步

当F<0时，控制刀具向$+y$方向走一步。

刀具每进给一步后，将刀具新的坐标值代入式（3-4），求出新的F值，以确定下一步进给方向。如此反复下去，即可完成直线插补。但是在完成上述计算中，需要进行两次乘法运算和一次减法运算，这种计算方法将直接影响插补速度，为了简化运算对式（3-4）作一些变换。

在图3-6中，当F≥0时，沿+x方向走一步，到达点(x+1, y),令新的加工偏差为F'，则由式（3-4）可得

$$F'=yx_e-y_e(x+1)=yx_e-y_ex-y_e=F-y_e \tag{3-5}$$

当F<0时，刀具向+y方向进给一步，到达点(x, y+1)，令新的加工偏差为F'，同样可得

$$F'=(y+1)x_e-y_ex=yx_e+x_e-y_ex=F+x_e \tag{3-6}$$

式（3-5）、式（3-6）是简化后的偏差计算公式。走完+x后，用式（3-5）；走完+y后，用式（3-6）。即在原偏差值上减一个y_e或加一个x_e，用以求得新的偏差值，作为下一步进给方向的判别依据。这种利用前一加工点的偏差，递推出新的加工点的偏差的方法，称为递推法。

刀具到达终点时自动停止进给。最常用的终点判别方法是设置一个长度计数器，其计数长度为两个方向进给步数之和。无论x轴还是y轴，每发一个进给脉冲，计数长度减1，

当计数长度减到零时，表示到达终点，插补结束。

2、直线插补工作过程

综上所述，逐点比较法直线插补工作过程可归纳为以下四步：

（1）偏差判别，即判断上一步进给后的偏差值是F≥0还是F<0。

（2）进给，即根据偏差判别的结果和插补所在象限决定在什么方向上进给一步。

（3）偏差运算，即计算出进给一步后的新偏差值，作为下一步进给的判别依据。

（4）终点判别，看是否已到终点，若已到达终点，就停止插补，若未到达终点，则重复一至四步工作。

3、直线插补计算举例

假设对第一象限直线OA进行插补，OA的起点坐标为(0,0)，终点坐标为(5,4)。

直线插补的起点与OA的起点重合，此时的偏差值F=0。计数长度$l=x_e+y_e=5+4=9$，即x方向走5步，y方向走4步，共9步。插补过程如图3-7和表3-1所示:

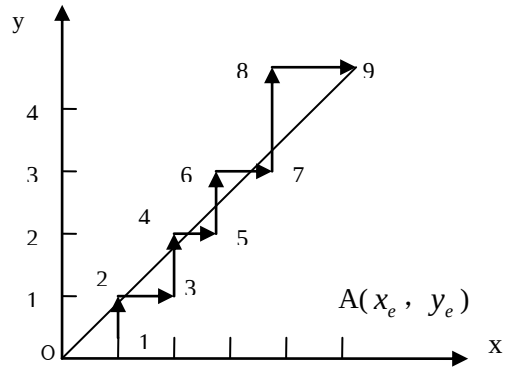


图3-7 直线插补走步轨迹图

表3-1 直线插补过程

序号	偏差判别	进给	偏差计算	终点判别
0			$F_0=0$	$l=9$
1	$F_0=0$	$+x$	$F_1=F_0-y_e=-4$	$l=8$
2	$F_1=-4<0$	$+y$	$F_2=F_1+x_e=1$	$l=7$

3	F_2=1＞0	$+x$	$F_3=F_2-y_e$=−3	l=6
4	F_3=−3＜0	$+y$	$F_4=F_3+x_e$=3	l=5
5	F_4=3＞0	$+x$	$F_5=F_4-y_e$=−2	l=4
6	F_5=−2＜0	$+y$	$F_6=F_5-y_e$=3	l=3
7	F_6=3＞0	$+x$	$F_7=F_6-y_e$=−1	l=2
8	F_7=−1＜0	$+y$	$F_8=F_7+x_e$=4	l=1
9	F_8=4＞0	$+x$	$F_9=F_8-y_e$=0	l=0

4、直线插补流程图

第一象限逐点比较法直线插补流程图，如图3-8所示。初始化主要包括读入终点坐标值 x_e， y_e，求出计数长度 $l=x_e+y_e$，设置初始偏差值 F_0=0等。

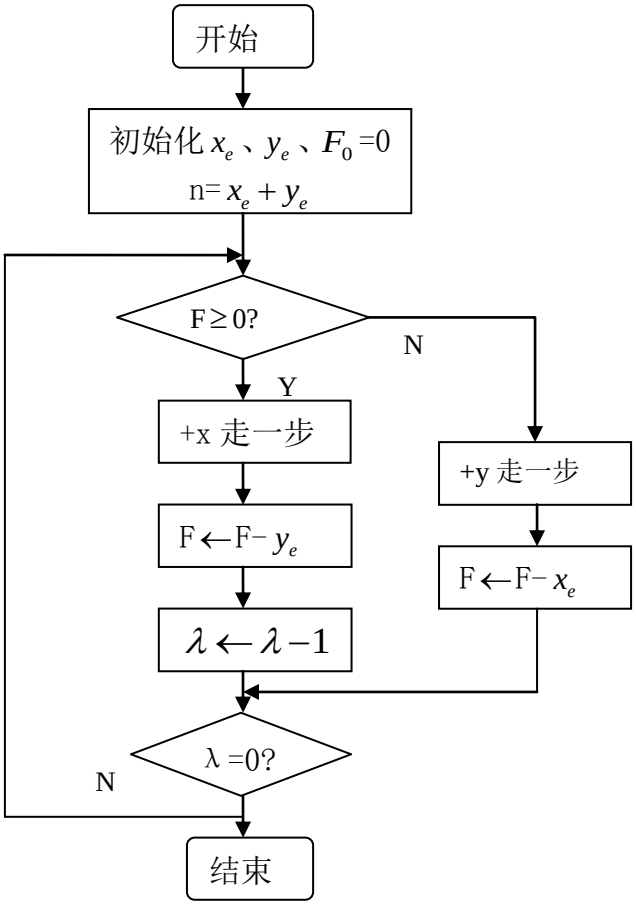


图3-8 第一象限直线插补流程图

4 硬件设计

根据本次设计的要求，选用四相八拍步进电机 28BYJ48，选用单片机 ATM89C52 作为控制器，选用 LCD12864 用于显示基本信息，选用 ULN2003 作为步进电机的驱动芯片，还有若干小键盘控制步进电机运转。电路原理图如图 4-1 所示：

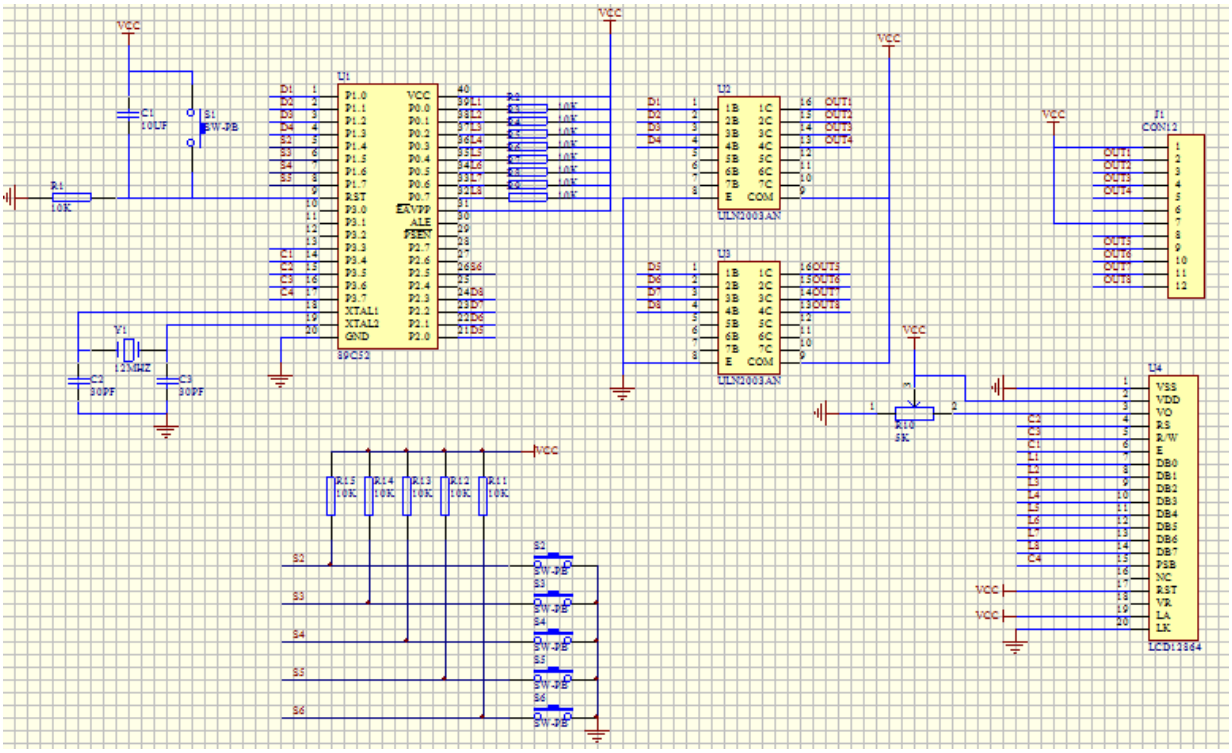


图 4-1 电路原理图

具体的接线连接：

(1)LCD12864 数据端接单片机 P0 口

其 RS 端接单片机 P3.5 口，EN 端接单片机 P3.4 口，R/W 端接单片机 P3.6 口,PSB 端接单片机 P3.7 口

(2)五个独立按键接单片机 P1.3-P1.7 和 P2.5 口

(3)第一个 ULN2003 接单片机 P1.0-P1.3，第二个 ULN2003 接单片机 P2.0-P2.3

(4)步进电机接 ULN2003 的 13-16 引脚

4.1 驱动电路

如图 4-2 所示，是 ULN2003 达林顿管驱动步进电机的电路图[10]。对步进电机和 ULN 的功能介绍见第二章 2.2 和 2.5 所示。在本设计中，28BYJ48 型四相八拍电机，电压为 DC5V—DC12V，相序表如表 4-1 所示，其中 D C B A 对应单片机输出值，接线指示为 A（橙）、B(黄)、C（蓝）、D（灰）、E（红，中点接+5V）。步进电机工作方式为单双八拍：A-AB-B-BC-C-CD-D-DA （即一个脉冲，转 3.75 度）八拍运行时步距角为 θ =360 度/(50*8)=0.9 度(俗称半步)。

表 4-1 四相八拍电机相序表

序号	D	C	B	A	十六进
1	0	0	0	1	01H
2	0	0	1	1	03H
3	0	0	1	0	02H
4	0	1	1	0	06H
5	0	1	0	0	04H
6	1	1	0	0	0CH
7	1	0	0	0	08H
8	1	0	0	1	09H

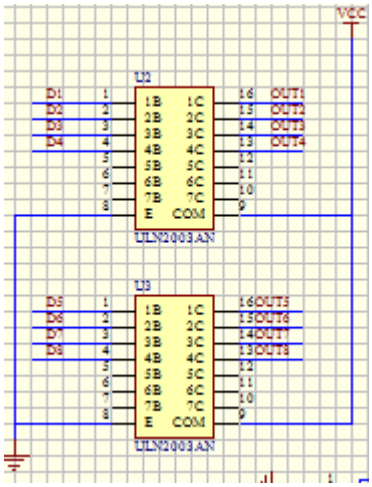


图 4-2 ULN2003 达林顿管驱动步进电机的电路图

4.2 单片机系统

对于单片机的功能介绍见第二章 2.3。如图 4-3 所示，有 89C52 单片机及其 12M 晶振接至 XTAL1（19）、XTAL2（18）引脚组成外部时钟电路和手动按键复位电路 RST（9）引脚组成的 52 单片机最小系统。

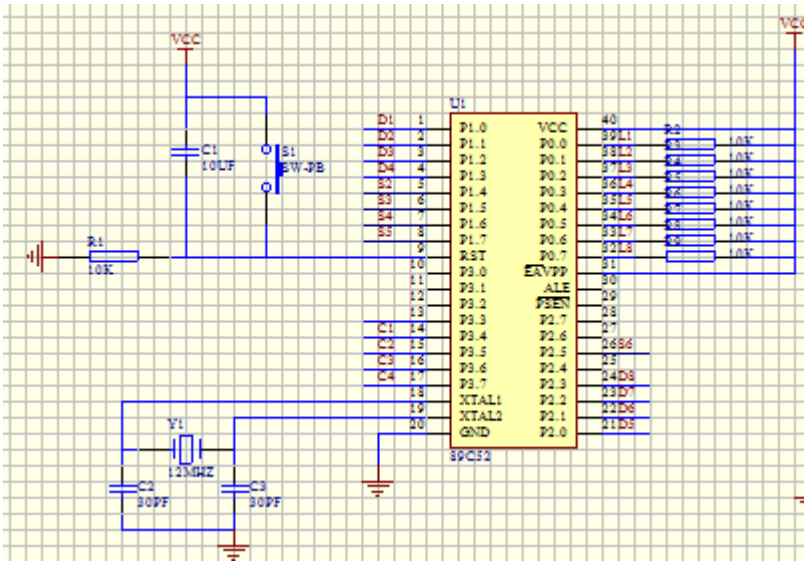


图 4-3 单片机最小系统

4.3 LCD12864 液晶显示电路

对于 LCD12864 的介绍见第二章 2.4.如图 4-4 所示，单片机的 P0 口作为 LCD12864 液晶显示单元的数据接口[11]，P3.4、P3.5、P3.6 作为控制使能端。

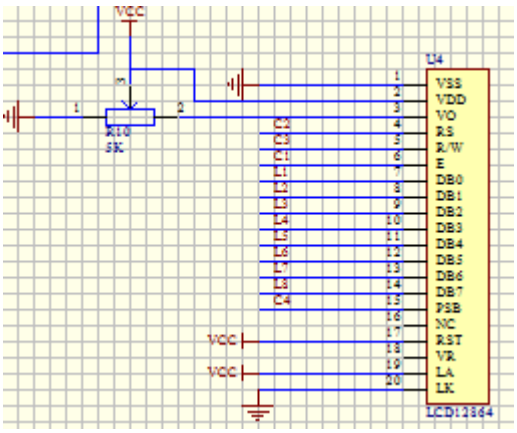


图 4-4 LCD12864 接线原理图

4.4 键盘接口

键盘接口电路图如图 4-5 所示。在程序中使用软件延时来消抖，一共有五个独立按键[12]。单片机的 P1.3-P1.7 和 P2.5 口作为键盘输入信号端，分别对应的功能为顺时针、逆时针、调速、旋转圈数和停止。公共端接电源地，按键为常开触点，按键闭合时接口逻辑电平置“0”。

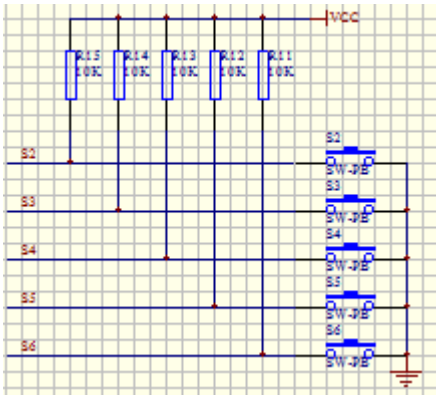


图 4-5 按键接线原理图

5 软件设计

软件设计主要工作是控制功能和显示功能的实现。在硬件介绍当中要实现的功能是用单片机来驱动两台步进电机沿着 X 轴和 Y 轴来转动。因为在软件当中实现有一定的困难性，所以本设计的软件部分就研究一台步进电机。本章节内容是运用 Keil C51 集成开发环境进行程序设计，并采用 C 语言编写软件程序[9]。程序流程图清晰的描述了该系统运行时的全过程，本章节从程序中筛选一些关键的程序段进行说明。完整程序见附录。

5.1 主程序设计

Main 函数中按照不同的独立按键来操作步进电机运动。控制模块中，有顺时针、逆时针、调速、旋转圈数和停止。显示模块中显示姓名、专业、转向和速度等级（2、4、6等级）、学号和电动机转动圈数。一些主要的程序如下。流程图如图 5-1 所示：

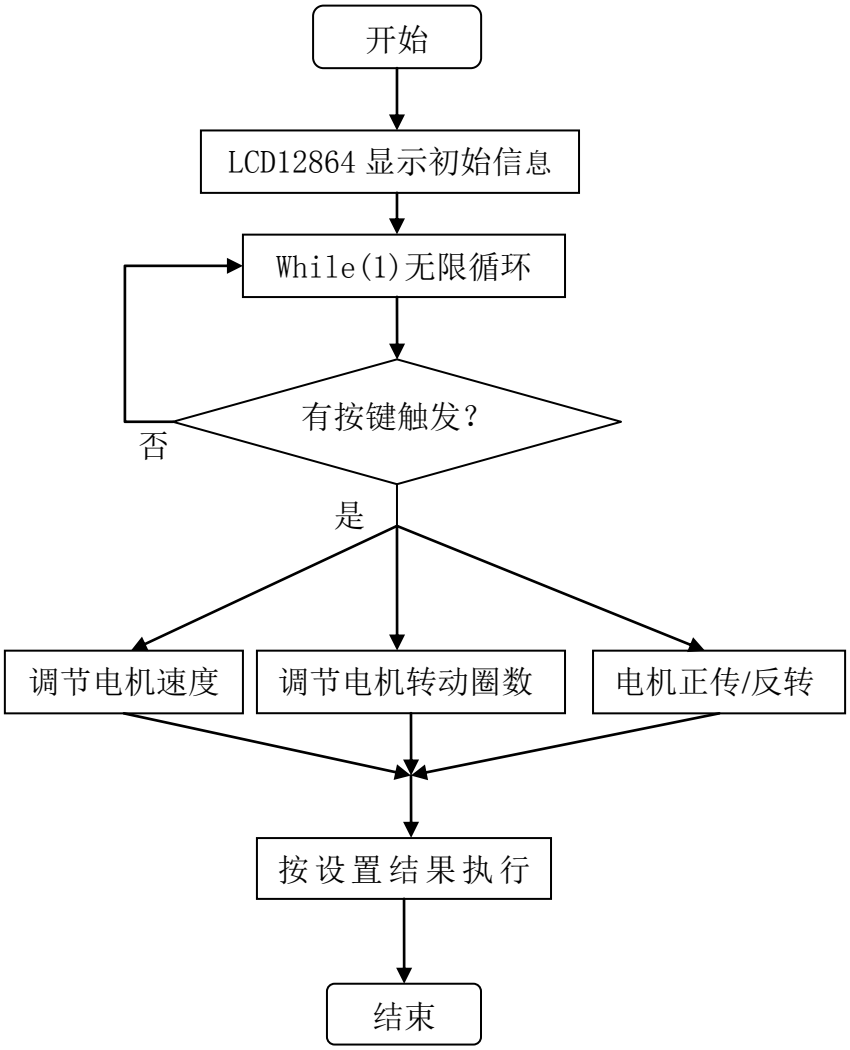


图 5-1 主程序设计

调节速度和显示：

```
while(1)
{
    if(KEY1==0)                 //三级调速按键
    {
        delayms(10);
        if(KEY1==0)
        {
            Beep();
            Speed_time=Speed_time-2;
            if(Speed_time<2)
                Speed_time=6;
            if(Speed_time==6)
            {
                Speed_flag='1';         //LCD 上显示的速度级别
            }
            else if(Speed_time==4)
            {
                Speed_flag='2';
            }
            else if(Speed_time==2)
            {
                Speed_flag='3';
            }
```

顺时针和显示：

```
else if (KEY2==0)                   //顺时针按键
 {
          delayms(10);
          if(KEY2==0)
          {
              Change(Status, 0);                    //LCD 显示顺时针

              for(r=0;r<N*(Circle_num-'0');r++)  //步进电机旋转 Circle_num 周
              {
                  Motor_forward();
                  if(KEY3==0)
                  {
                      delayms(5);
                      if(KEY3==0)
                      {
                          break;
                      }
```

旋转圈数和显示：

```
else if(KEY5==0)                        //旋转几周
      {
         delayms(10);
         if(KEY5==0)
         {
            Beep();
            Circle_num++;
            if(Circle_num>'8')
               Circle_num='1';
            Change_Circle(Circle_num);          //LCD 显示步进电机旋转几周
         }
      }
```

5.2 控制功能实现

这里使用到五个独立按键：步进电机顺时针、逆时针、调速、旋转圈数和停止按键。实时监测独立按键，通过控制脉冲输出顺序和脉冲个数实现上述功能，单片机系统经过初始化后，进入安检扫描程序，等待用户操作。当按下顺时针独立按键时去调用步进电机的正转相序表，反之，则去调用步进电机的反转相序表。相序表如表 4-1 所示。步进电机正转时，单片机 P1 和 P2 口输出的驱动脉冲序列为从上到下，将单片机反向输出改组序列即可实现步进电机的反转。控制功能流程图如图 5-2 所示：

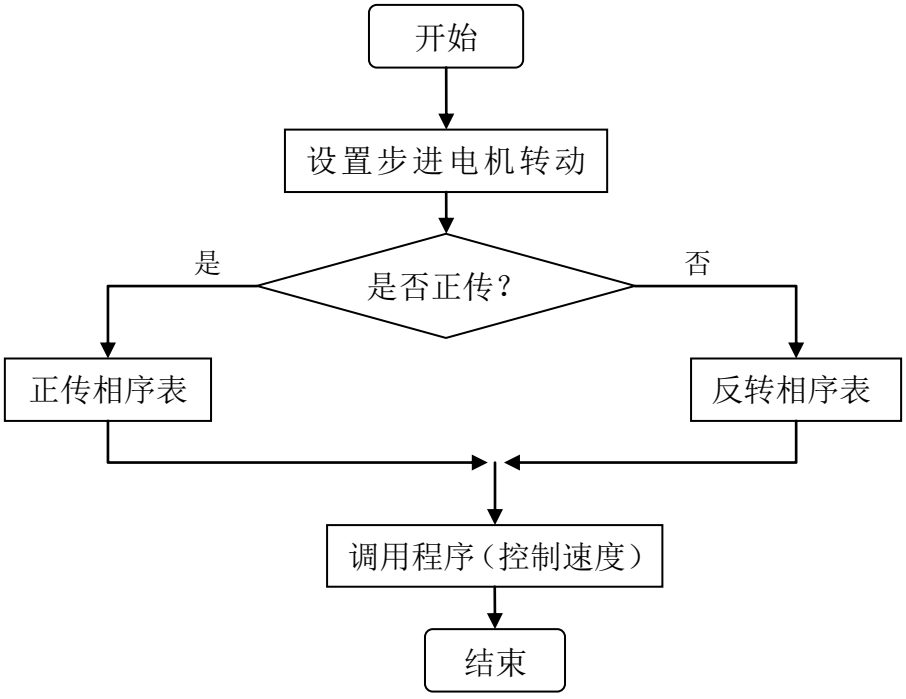


图 5-2 控制功能流程图

5.3 显示功能实现

显示功能室本系统的重要组成部分。首先，程序开始运行时对 LCD12864 进行初始化设置。然后，载入定时器初值。其次，判断 LCD12864 是否可以写入，不忙则进行下一步，显示运行速度，返回进入下一个循环周期。显示中，第一行显示姓名和专业，第二行显示转向和速度等级（2、4、6 等级），第三行显示学号，第四行显示电动机转动圈数。程序流程图如图 5-3 所示：

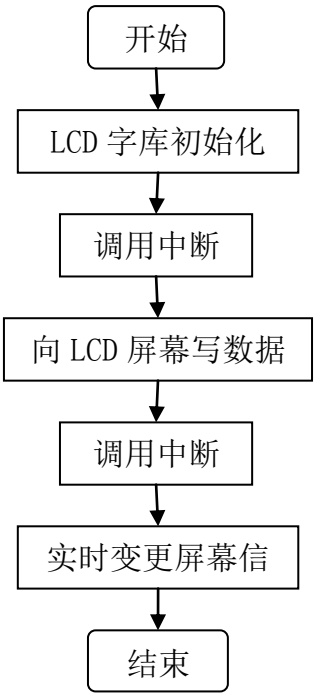


图 5-3 显示功能程序流程图

6 结 论

本毕业设计从 3 月份就开始做，现在也即将完成。我再这段时间里通过查阅相关文献，写开题报告，翻译英文文献等一系列毕业设计的前期工作，我学到了如何开始设计一样属于自己的东西。通过这几个月的磨练学习，把所有学过的知识进行了一次全面而系统的综合，并融会贯通应用到实际中去，不但完善了自己的知识结构，同时对所学过的各种理论知识与专业知识进行了一次全面的梳理。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在系统的设计中做了如下的工作。

1、查阅相关资料。根据课题要求查阅资料，然后在老师的指导下，有针对性地学习相关知识，对资料进行消化和吸收[13]。

2、根据系统的要求确定控制系统的总体设计方案。系统以 AT89C52 单片机为控制核心，结合直线插补运算运用到数控机床当中，并设计相应的接口电路，键盘电路、显示电路、步进电机的驱动电路等。

3、编写应用程序。软件是根据控制系统的要求设计的，包括主程序的设计、键盘显示程序设计、调速程序设计等程序的设计。

4、用 keil 软件完成程序的编写和调试。

5、用 Protell 画原理图[14]。

论文采用单片机技术，完成对控制系统的硬件电路和应用软件设计。论文虽然完成了系统的设计，但由于开发经验不足，系统一定存在不妥之处，尤其是步进电机的应用方面有待进一步的研究和探讨。论文中的不足之处敬请老师批评指正。

7 经济分析报告

本设计是一个开环系统，它既简单、廉价，又非常可行。各种元器件用的都是很广泛，很容易购买。具体的价钱如表 8-1 所示：

表 8-1 元器件清单价格

器件名称	个数（个）	单价（元）	单个总价(元)	总价（元）
10K 电阻	15	0.03	0.45	53.44
5K 滑动变阻器	1	0.6	0.6	
四角按键	6	0.1	0.1	
极性电容电容 10uF	1	0.04	0.04	
电容 30pF	2	0.05	0.1	
12MHz 晶振	1	0.25	0.25	
ATM89C52	1	8.0	8.0	
ULN2003	2	0.35	0.7	
四相八拍步进电机	2	4．1	8.2	
LCD12864	1	35	35	

有表格可知，对于这样的设计是非常经济实用的。

致　谢

随着毕业设计的逐渐完成，离毕业离校的日子也就越来越近了。在这大学四年里，我非常感谢各位老师四年来对我的栽培，以及各位同学朋友对我的帮助，正是由于他们在学习和生活上给予我无微不至的关怀，才使我在这四年期间不仅学到了扎实的专业知识，还学到了很多为人处事的原则。在论文完成之际，谨向给予我指导和支持的老师、同学表示我最真挚的谢意！

通过这次最后的毕业设计，使我学到了很多专业知识，实验技能也有了很大的提高。其实，有些事并非我们想的那样难，世上无难事，只怕有心人，只要我们真正用心去做了，你就会发现一切就变得很容易了。虽然感觉自己做的还不是很好，但毕竟自己亲自去完成了论文的全过程，所以还是有成就感的。这也是我在大学最后的一次论文，我会非常珍惜这次的论文。

在整个设计、修改过程中，邵龙安老师悉心地给予了我许多指导和帮助，对我的毕业设计旅程起到了非常好的导航作用，在此我要衷心地感谢。通过老师的指导，使我少走了很多弯路。由衷的感谢他们！

最后，我还要感谢我的母校——景德镇陶瓷学院科技艺术学院，感谢“她”为我提供了一个良好的学习和生活环境。如果没有“她”也就没有今天的我们。即将离开学校，还真有点舍不得，在以后的日子里我会时常想起这里的一幕幕，因为在这里曾经给我留下了太多的喜、怒、哀、乐，有太多的人和事值得我去怀念了。

在这即将离开之即，祝所有的老师和同学朋友身体健康，工作顺利，愿我的母校明天更加美好！

参 考 文 献

[1]庄渊昭.步进电机控制系统设计[J].电子世界.2010（9）:43-46.

[2]郭改枝，张鹏举.单片机控制液晶屏接口电路的设计与实现[J].内蒙古师范大学学报.2010.39(4):425-427

[3]未知.步进电机.

http://baike.baidu.com/link?url=tL6ZoVxVtn3yxtDBsx6ri7ymxjLILc_De2C1IsvYMO_hsdgPHbgcIM4SkkFOkFOG.2014-4-15

[4]未知. ATM89C52.http://baike.baidu.com/view/2251929.htm?fr=Aladdin.2014-4-1

[5]未知.LCD12864.http://baike.baidu.com/view/6297556.htm?fr=Aladdin.2014-4-8

[6]何雪明，吴晓光，刘有余.数控技术.武汉：华中科技大学出版社，2010

[7]王书峰，谭建豪.计算机控制技术.武汉：华中科技大学出版社，2011

[8]邵世凡.电机与拖动.杭州：浙江大学出版社，2008

[9]杨忠宝，董晓明.C 语言程序设计.北京：北京大学出版社，2010

[10]张润和.电力电子技术及应用.北京：北京大学出版社，2008

[11]韩璞.自动化专业概论.北京：中国电力出版社，2007

[12]张彬宏，吴青萍.模拟电子技术.北京：北京理工大学出版社，2008

[13]伊利民、尹全英. 电气制图与读图. 北京：机械工业出版社

[14]郑阿奇.Protell 实用教程.北京：电子工业出版社，2010

附　录

```
#include<reg52.h>
#include<intrins.h>
#include<string.h>
#define uchar unsigned char
#define uint unsigned int

//逆时针旋转相序表
uchar code Reversal[8]={0x09，0x08，0x0c，0x04，0x06，0x02，0x03，0x01};
//正时针旋转相序表
uchar code Forward[8]={0x01，0x03，0x02，0x06，0x04，0x0c，0x08，0x09};

sbit KEY1=P1.4;        //调速按键
sbit KEY2=P1.5;        //顺时针按键
sbit KEY3=P2.5;        //停止按键
sbit KEY4=P1.6;        //逆时针按键
sbit KEY5=P1.7;        //步进电机旋转圈数按键

sbit RS =P3.5;                 //LCD12864
sbit WRD=P3.6;
sbit E    =P3.4;
sbit PSB=P3.7;

//传送数据或者命令，当 DI=0 是，传送命令，当 DI=1，传送数据
void Transfer(uchar data1，bit DI);
void Change(uchar *adder，uint j);          //LCD 显示顺时针，逆时针
void Change_speed(uchar adder);             //LCD 显示速度级别
void Change_Circle(uchar adder);             //LCD 显示步进电机的圈数

void delayms(uint i);                        //毫秒级延时
void delay(uint m);


void Lcd_mesg(uchar *adder);                //显示 LCD 基本信息
uchar Status[]={"顺逆"};                     //LCD 显示
uint Speed_time=6;                           //步进电机转速 6，4，2 三级
uchar Speed_flag='1';                        //步进电机默认转速，最慢-->第一级
uchar Circle_num='1';

uint r;
uint N=64;                                   //因为步进电机是减速步进电机，减速比的 1/64 ，
                                             //所以 N=64 时，步进电机主轴转一圈
uchar LCD_DATA[]={
"王道梅　自动化　　"
"201030455141　　"
"顺时针　速度 1　"
"电机转动圈子 1　"
};
//line 1，3，2，4

void Initinal(void)                  //LCD 字库初始化程序
{
    delay(40);                       //大于 40MS 的延时程序
```

```
    PSB=1;                          //设置为 8BIT 并口工作模式
    delay(1);                       //延时
    RES=0;                          //复位
    delay(1);                       //延时
    RES=1;                          //复位置高
    delay(10);
    Transfer(0x30，0);              //Extended Function Set ：8BIT 设置，RE=0：
                                    //basic instruction set， G=0 ： graphic display OFF
    delay(100);                     //大于 100uS 的延时程序
    Transfer(0x30，0);              //Function Set
    delay(37);                      //大于 37uS 的延时程序
    Transfer(0x08，0);              //Display on Control
    delay(100);                     //大于 100uS 的延时程序
    Transfer(0x10，0);              //Cursor Display Control 光标设置
    delay(100);                     //大于 100uS 的延时程序
    Transfer(0x0C，0);              //Display Control，D=1，显示开
    delay(100);                     //大于 100uS 的延时程序
    Transfer(0x01，0);              //Display Clear
    delay(10);                      //大于 10mS 的延时程序
    Transfer(0x06，0);              //Enry Mode Set，光标从右向左加 1 位移动
    delay(100);                     //大于 100uS 的延时程序
}

void Motor_reversal(void)           //逆时针转动
{
    uchar i，j;
for(j=0;j<8;j++)                    //电机旋转一周，不是外面所看到的一周，是里
                                    //面的传动轮转了一周
    {
        if(KEY3==0)                 //停止按键
        {

            break;                  //如果 KEY3 按下，退出此循环
        }

        for(i=0;i<8;i++)            //旋转 45 度
        {
            P2=Reversal[i];
            delayms(Speed_time);    //调节转速
        }
    }
}

void Motor_forward(void)            //顺时针转动
{
    uchar i，j;

    for(j=0;j<8;j++)                //如果 KEY3 按下，退出此循环
    {
        if(KEY3==0)                 //停止按键
        {
break;
        }
```

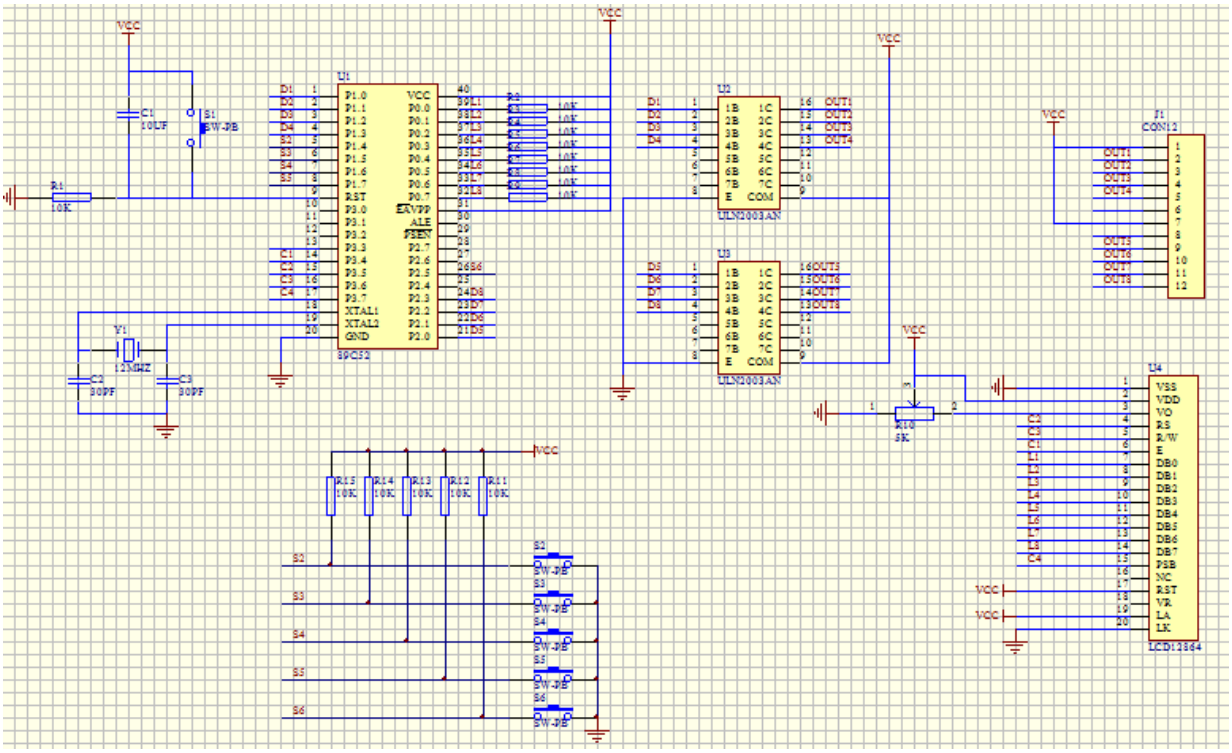
```
        for(i=0;i<8;i++)                    //旋转 45 度
        {
            P2=Forward[i];
            delayms(Speed_time);        //调节转速
        }
    }
}

void Main(void)
{
     Initinal();                            //调用 LCD 字库初始化程序
     delay(100);                            //大于 100uS 的延时程序
     Lcd_mesg(LCD_DATA);                //LCD 显示基本信息
     delay(1000);

    while(1)
    {
        if(KEY1==0)                         //三级调速按键
        {
            delayms(10);
            if(KEY1==0)
            {
                Beep();
                Speed_time=Speed_time-2;
                if(Speed_time<2)
                    Speed_time=6;
                if(Speed_time==6)
                {
                    Speed_flag='1';             //LCD 上显示的速度级别
                }
                else if(Speed_time==4)
                {
                    Speed_flag='2';
                }
                else if(Speed_time==2)
                {
                    Speed_flag='3';
                }
                Change_speed(Speed_flag);     //LCD 显示速度级别
            }
        }

        else if (KEY2==0)                           //顺时针按键
{
            delayms(10);
            if(KEY2==0)
            {

                Change(Status，0);                          //LCD 显示顺时针

                for(r=0;r<N*(Circle_num-'0');r++)       //步进电机旋转 Circle_num 周
                {
                    Motor_forward();
                    if(KEY3==0)
                    {
```

```
                        delayms(5);
                        if(KEY3==0)
                        {

                            break;
                        }
                    }
                }
            }
        }
        else if(KEY4==0)                        //逆时针
        {
            delayms(10);
            if(KEY4==0)
            {

                Change(Status，2);              //LCD 显示顺时针

                //Change_speed(Speed_flag);     //LCD 显示速度级别
                for(r=0;r<N*(Circle_num-'0');r++)   //步进电机旋转 Circle_num 周
                {
                    Motor_reversal();
                    //Change(Status，2);
                    //Change_speed(Speed_flag);
if(KEY3==0)
                    {
                        delayms(5);
                        if(KEY3==0)
                        {
                            break;
                        }
                    }
                }
            }
        }
        else if(KEY5==0)                        //旋转几周
        {
            delayms(10);
            if(KEY5==0)
            {
                Beep();
                Circle_num++;
                if(Circle_num>'8')
                    Circle_num='1';
                Change_Circle(Circle_num);      //LCD 显示步进电机旋转几周
            }
        }
        else
        P2=0x0f;

    }
}

void Lcd_mesg(uchar *adder)                     //显示 LCD 信息
{
    uchar i;
```

```
    Transfer(0x80，0);
    delay(100);

    for(i=0;i<32;i++)
    {
        Transfer(*adder，1);
        adder++;
}

  Transfer(0x90，0);
  delay(100);

  for(i=32;i<64;i++)
    {
        Transfer(*adder，1);
        adder++;
    }
}

void Change(uchar *adder，uint j)                    //LCD 显示顺时针，逆时针
{
    uint i;
    Transfer(0x90，0);
    delay(100);

    for(i=j;i<j+2;i++)
    {
        Transfer(adder[i]，1);
    }
}

void Change_speed(uchar adder)                       //LCD 显示速度级别
{
    Transfer(0x96，0);
    delay(100);
    Transfer(adder，1);
}

void Change_Circle(uchar adder)                  //LCD 显示步进电机的圈数
{
    Transfer(0x9E，0);
    delay(100);
    Transfer(adder，1);
}
void Transfer(uchar data1，bit DI)               //传送数据或者命令，当 DI=0 是，传
                                                 //送命令，当 DI=1，传送数据
{
    WRD=0;
    RS=DI;
    delay(1);
    P1=data1;

    E=1;
delay(1);
    E=0;
}
```

```
void delay(uint m)                          //延时程序
{
    uint i，j;
    for(i=0;i<m;i++)
        for(j=0;j<10;j++)
        {
            ;
        }
}

void delayms(uint i)                         //1ms 基准延时程序
{
    uchar j;
    while(i--)
    {
        for(j=0;j<115;j++)
        {
            ;
        }
    }
}
```

原理图


VCC
C1
10UF
S1
SW-PB
R1
10K
U1
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
RST
P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.6
P3.7
XTAL1
XTAL2
GND
VCC
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
EA/VPP
ALE
PSEN
P2.7
P2.6
P2.5
P2.4
P2.3
P2.2
P2.1
P2.0
89C52
Y1
12MHZ
C2
30PF
C3
30PF
U2
U3
ULN2003AN
J1
CON12
OUT1
OUT2
OUT3
OUT4
OUT5
OUT6
OUT7
OUT8
R16
5K
U4
VSS
VDD
VO
RS
R/W
E
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
PSB
NC
RST
VR
LA
LK
LCD12864
R13
10K
R14
10K
R15
10K
R12
10K
R11
10K
S2
S3
S4
S5
S6