大连理工大学本科设计报告

# （计算机原理实验综合设计）

题目： 步进电机控制系统设计

课程名称： 计算机原理实验学院（系）： 电子信息与电气工程学部 专 业： 自 动 化 班 级： 电 自 1603 班学 号： 201682008 学生姓名： 曹 炳 全 成 绩 ：

2018 年 12 月 18 日

# 题目：步进电机控制系统设计

## 设计要求

以 8086 为控制核心，利用定时器 8253 的输出 OUT1 产生定时引发中断，在中断服务程序中对步进电机进行驱动，并通过电位器控制步进电机转速（8 个档位），通过开关改变步进电机转向和实现停转功能，且能通过数码管和 8\*8LED 点阵显示速度和方向。

## 设计分析及系统方案设计

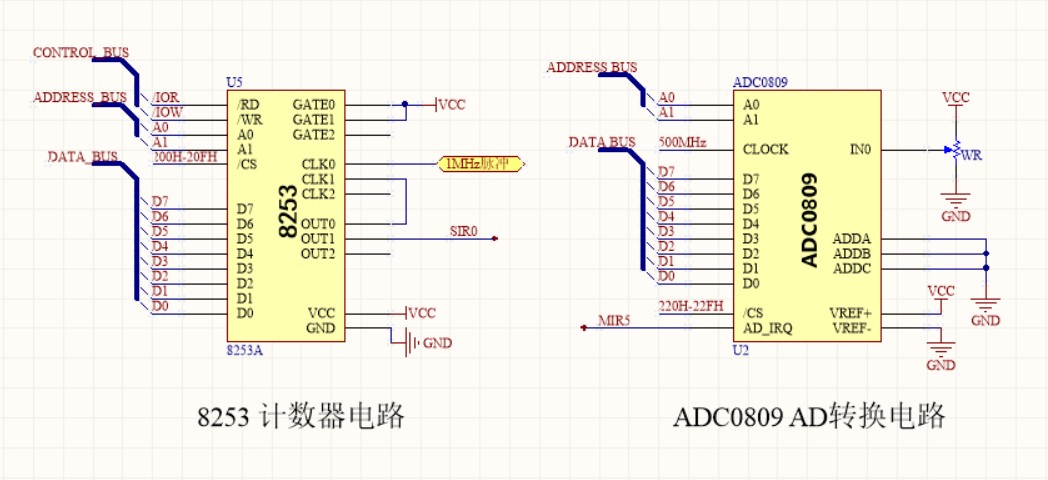
定时器的定时初值决定了定时器溢出的频率，也就是引发中断的频率，而中断服务的操作是输出步进电机新的相序代码，控制着步进电机的转动速度。控制步进电机转动，只需按次序在输入端输入对应相序即可，改变相序输入次序即可改变转向，改变相序切换间隔即可实现调速。为了方便对电机进行控制，此系统采用“双---四拍”的相序信号，将初始相序存放在一个寄存器中（原始相序数据位 33H），然后利用对该寄存器“移位”的方式产生下一个相序，并在内存中保存新的相序，其中对寄存器中的数据移位方向决定着电机旋转的方向。

此系统共用到了两个中断：一个是利用中断服务程序读取 ADC 的数据，并将数据转换为对应的速度档位和计数器写入的初值；另一个是利用 8253 的定时中断通过 8255 的 PC0-PC3 口输出步进电机新的相序代码。而在执行这两个中断服务程序过程中对 PC7 和 PC6 采集到的开关 K7 和 K6 的值进行读取，其中开关 K7 控制电机转动的方向，K7=0 时电机正转（顺时针），K7=1 时电机反转（逆时针）。开关 K6 起停转功能，K6=0 时电机正常运行，K7=1 时电机停止转动，此时速度档位变为零。

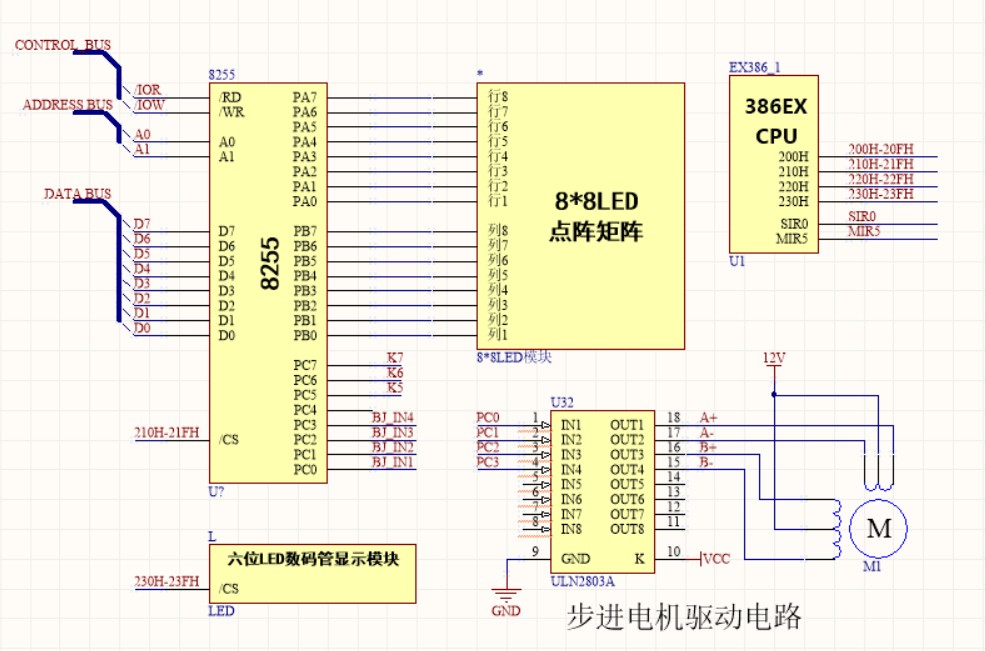
除了两个中断服务程序外，此系统还有两个显示程序，分别驱动数码管和 8\*8LED 点阵显示电机当前的运行情况。在程序运行过程中，数码管通过内部总线传输数据并利用锁存器驱动显示电机的运行数据，若与 8255PC5 口相连的开关 K5=0，则显示转动方向，若开关 K5=1，则显示电机转动的速度档位。且实验台上的 8\*8LED 点阵实时显示电机当前速度。数码管显示程序和 8\*8LED 点阵显示程序均在 ADC 采集完成后进行调用显示，并在最后重新启动 ADC。

## 系统电路图

将 8253\_CS 设定为 200H，ADC\_CS 设定为 220H。具体电路如下图所示：



将 8255\_CS 设定为 210H，LED\_CS 设定为 230H。具体电路如下图所示：



## 外围接口模块硬件电路功能描述

此次实验用到了 386EX 实验仪中的 8259、8255、8254、ADC0809 芯片和六位数码管模块、8\*8LED 点阵显示模块。其中 8259 管理和控制外部中断请求：定时中断和 ADC 采集中断；8255 的 PC0-PC3 口用于输出步进电机相序，PC5-PC7 口用于读取开关状态，PA0-PA7 和 PB0-PB7 口分别驱动 8\*8LED 点阵显示模块的“行线 Ri”和“列线 Ci”；8254 定时中断控制输出相序的时间间隔；ADC0809 用于采集并转换当前电位器的值。数码管模块和 8\*8LED 显示模块用于显示电机的速度和方向。

## 主程序中主要变量说明

|  |  |  |
| --- | --- | --- |
| 变量名称 | 内存单元 | 功能 |
| buf | db | 步进电机“双---四拍”的相序信号 |
| ad\_result | db | 存储 ADC 采集结果 |
| speed\_fre | db | 8253 计数器 1 初值档位 |
| dir | db | 电机转动方向(0-1) |
| led\_dat | db | 电机速度档位(0-8) |
| led\_hex | db | LED 显示编码(0-9) |
| led0-led8 | db | 8\*8LED 点阵显示字形码(0-8) |
| count | dw | 8\*8LED 点阵显示程序查表计数器 |
| bz | dw | 定义一个位扫描码字的空间 |

1. 系统软件中各个子程序的功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 子程序名称 | 入口参数 | 出口参数 | 功能描述 |
| int\_proc\_adc | 无 | led\_dat | 采集 ADC 转换数据并将其  转换成电机相应速度 |
| int\_proc\_pit | 无 | dir | 8253 定时中断，输出步进  电机相序，并控制转向 |
| led\_disp | dir  led\_dat | 无 | 数码管通过开关状态选择  显示电机转动方向或速度 |
| disp\_88 | led\_dat | 无 | 8\*8LED 点阵显示速度 |

1. 主程序程序流程图

程序由主程序、两个中断服务子程序、数码管显示程序和 8\*8LED 点阵显示程序五部分组成。详见下图。



主程序开始

start:

nop;

等待中断

sti开中断

启动ADC 开始AD转换

步进电机的相序

代码buf <- 33H

从8255PC0-PC3口输出

8253初始化

计数器0：方式3，初值100

计数器1：方式2，初值40

8255初始化： A\B\C0-C3口输出C4-C7口输入

设置中断向量表主 片 ：35H 从片：70H

设置386EX主从

8259中断屏蔽字主片：11011011b 从片：11111110b

cli关中断

初始化数据段DS



AD转换完成中断

int\_proc\_adc

al==0?

Y

N

中断服务子程序

int\_proc\_adc结束

sti开中断;

iret

发送中断结束命令主片8259清中断标志

调用led\_disp程序

进行数码管显示

调用disp\_88程序 进行8\*8LED点阵显示

将计数器1初值档位1-8 先乘30再加10进行计算将结果写入8253计数器1

将计数器1初值档位1-8 对应转换为速度档位8-1 并存入led\_dat中

停转，速度为零

led\_dat <- 0

将0-255范围内的

AD值转换为1-8档

并存入speed\_fre中

读取AD值到al

并存储到内存中

读入8255C口数据取PC6（K6）数据



8253定时中断

int\_proc\_pit

N

al==0?

Y

Y

al==0?

N

中断服务子程序

int\_proc\_pit结束

sti开中断; iret

相应寄存器出栈

恢复现场

发送中断结束命令从片8259清中断标志主片8259清中断标志

8255PC0-PC3口输出

步进电机的相序代码并将新值存入内存中

反转:方向1存入Dir

相序代码循环右移

正转:方向0存入Dir 相序代码循环左移

读入8255C口数据取PC7（K7）数据

读入8255C口数据

取PC6（K6）数据

相应寄存器入栈保护现场



LED显示程序

led\_disp

N

al==0?

Y

LED显示程序

led\_disp结束

ret

返回

01H送位扫描锁存器显示

dx指向LED的数据口输出字形码

al <- [si+bx]

[si+bx]指向显示

数据的LED字型码

取led\_hex的段内

偏移地址存入bx

ah <- 0 si <- ax

al <- led\_dat

取led\_dat的数据到al

al <- dir

取dir的数据到al

读入8255C口数据取PC5（K5）数据



8\*8LED点阵显示

程序disp\_88

扫描码=0?

N

Y

8\*8LED点阵显示

程序disp\_88结束

ret

返回

扫描码左移1位

shl bh,1

关闭扫描信号

（熄灭LED）

计数器count+1

延时

从bz中获取扫描码送PB口输出

查表得字型码mov bl,byte ptr[di] 送PA口输出

关闭扫描信号

（熄灭LED）

计数器count=0

设定最高位扫描码

01H -> bh

al乘8结果存入ax di <- di+ax

取led0的段内

偏移地址存入di

取速度档位数值

al <- led\_dat

* 1. 主程序流程图
  2. 中断服务子程序

int\_proc\_adc流程图

* 1. 中断服务子程序

int\_proc\_pit流程图

* 1. LED显示程序

led\_disp流程图

* 1. 8\*8LED点阵显示程序disp\_88流程图

**图 6.1** 程序流程图

## 程序清单

CS\_8253 equ 200h CS\_8255 equ 210h CS\_0809 equ 220h LED\_IO equ 230h data segment

buf db 0 ;“双---四拍”的相序信号

ad\_result db 0 ;ADC 采集结果

speed\_fre db 0 ;数值越大转的越慢

dir db 0 ;转动方向：0 正转,顺时针；1 反转，逆时针

led\_dat db 0 ;电机速度档位 1-8,1 最慢，8 最快

led\_hex db 3fh,06h,5bh,4fh,66h,6dh,7dh,07h,7fh,67h ;0-9，LED 显示编码

led0 db 00h,3eh,51h,49h,45h,3eh,00h,00h ;0 的字形码led1 db 00h,00h,42h,7fh,40h,00h,00h,00h ;1 的字形码led2 db 00h,42h,61h,51h,49h,46h,00h,00h ;2 的字形码led3 db 00h,21h,41h,45h,4bh,31h,00h,00h ;3 的字形码led4 db 00h,18h,14h,12h,7fh,10h,00h,00h ;4 的字形码led5 db 00h,27h,45h,45h,45h,39h,00h,00h ;5 的字形码led6 db 00h,3ch,4ah,49h,49h,30h,00h,00h ;6 的字形码led7 db 00h,00h,01h,79h,05h,03h,00h,00h ;7 的字形码led8 db 00h,36h,49h,49h,49h,36h,00h,00h ;8 的字形码count dw 00h ;查表计数器

bz dw ? ;定义一个位扫描码字的空间data ends

code segment

assume cs:code,ds:data start:

cli

mov ax,data mov ds,ax

;设置 386EX 主从 8259 中断屏蔽字

in al,21h

and al,11011011b ;主片 MIR5 out 21h,al

in al,0a1h

and al,11111110b ;从片 SIR0

out 0a1h,al

;设置中断向量表push ds

mov ax,0 mov ds,ax

lea ax,cs:int\_proc\_adc ;主片 MIR5 为 AD 转换完成中断

mov si,35h add si,si add si,si

mov ds:[si],ax push cs

pop ax

mov ds:[si+2],ax mov ax,0

mov ds,ax

lea ax,cs:int\_proc\_pit ;从片 SIR0 为定时中断

mov si,70h add si,si add si,si

mov ds:[si],ax push cs

pop ax

mov ds:[si+2],ax pop ds

;8255 初始化

mov dx,CS\_8255+3 ;指向 8255 的控制口

mov al,10001000b ;A\B\C0-C3 口输出，C4-C7 口输入

out dx,al

;8253 初始化

mov dx,CS\_8253+3

mov al,36h ;设定 8253 计数器 0 为方式 3 out dx,al

mov dx,CS\_8253

mov ax,100 ;写入计数器 0 初值 100 out dx,al

mov al,ah

out dx,al

mov dx,CS\_8253+3

mov al,74h ;设定 8253 计数器 1 为方式 2 out dx,al

mov dx,CS\_8253+1

mov ax,40 ;写入计数器 1 初值 40（必须大于 35） out dx,al

mov al,ah out dx,al

;输出步进电机相序代码

mov buf,00110011b ;buf=33H，初始相序 0011B mov al,buf

mov dx,CS\_8255+2

out dx,al ;8255PC0-PC3 口输出步进电机的相序代码

;启动 ADC

mov dx,CS\_0809 ;ADC 启动转换地址

out dx,al ;启动 ADC sti

;等待中断

lll: nop jmp lll

;中断服务子程序

int\_proc\_adc proc far ;AD 转换完成中断，转换成相应速度

push ax push dx

mov dx,CS\_8255+2

in al,dx ;读入 8255C 口数据

and al,40h ;取 PC6（K6）数据，0：正常转动，1：停转cmp al,0

jz stop1

mov led\_dat,0 ;停转，将速度 0 存入led\_dat jmp final

stop1:

mov dx,CS\_0809+1 ;读 AD in al,dx

lea bx,ad\_result ;存 AD 值

mov [bx],al ;AD 值范围 0-255 shr al,1

shr al,1

shr al,1 ;除以 8，取值 0-31

add al,4 ;加上 4，取值 4-35 shr al,1

shr al,1 ;除以 4，取值 1-8

lea bx,speed\_fre ;将计数器 1 初值档位存入 speed\_fre，数值越大速度越小mov [bx],al

mov ah,9 sub ah,al

mov al,ah ;9 减去 1-8，取值 8-1

lea bx,led\_dat ;存入速度档位 led\_dat mov [bx],al

jmp final final:

mov dx,CS\_8253+1;指向 8253 计数器 1

mov al,speed\_fre mov bl,30

mul bl

add al,10 ;初值为 speed\_fre\*30+10

out dx,al ;写入计数器 1 初值，低 8 位

mov al,ah

out dx,al ;写入计数器 1 初值，高 8 位

call disp\_88 ;调用 8\*8LED 点阵（字符）显示程序

call led\_disp ;调用 LED 显示程序mov dx,CS\_0809 ;ADC 启动转换地址out dx,al ;启动 ADC

mov al,20h ;发送中断结束命令

out 20h,al ;向 386EX 主 8259 pop dx

pop ax sti

iret int\_proc\_adc endp

int\_proc\_pit proc far ;8253 定时中断,输出相序

push ax push cx

mov dx,CS\_8255+2;读入 8255C 口数据

in al,dx

and al,40h ;取 PC6（K6）数据，0：正常转动，1：停转cmp al,0

jnz stop ;若不为 0，则直接跳到最后，不再变换相序，电机停转mov dx,CS\_8255+2;读入 8255C 口数据

in al,dx

and al,80h ;取 PC7（K7）数据，0：正转，顺时针，1：反转，逆时针cmp al,0

jz normal

unnormal: ;反转，逆时针mov al,1

mov dir,al ;将方向 1 存入 dir mov al,buf

ror al,1 ;相序代码循环右移jmp nx1

normal: ;正转，顺时针mov al,0

mov dir,al ;将方向 0 存入 dir mov al,buf

rol al,1 ;相序代码循环左移nx1:mov dx,CS\_8255+2

out dx,al ;8255PC0-PC3 口输出步进电机的相序代码

mov buf,al ;将移动后的相序代码存入内存stop:mov al,20h ;从片 8259 清中断标志

out 0a0h,al

mov al,20h ;主片 8259 清中断标志

out 20h,al pop ax

pop cx sti

iret int\_proc\_pit endp

led\_disp proc ;LED 显示程序

push ax push bx push cx push dx

mov dx,CS\_8255+2

in al,dx ;读入 8255C 口数据

and al,20h ;取 PC5（K5）数据，0：数码管显示方向，1：数码管显示速度cmp al,0

jnz display\_speed lea bx,dir

mov al,[bx] ;取内存中方向 dir 数据到 al 寄存器

jmp ffff display\_speed:

lea bx,led\_dat

mov al,[bx] ;取内存中速度 led\_dat 数据到 al 寄存器

ffff:

mov ah,0

mov si,ax ;将取到的 dir/led\_dat 数据数值存入 si lea bx,led\_hex

mov dx,LED\_IO ;指向 LED 的数据口

mov al,[si+bx] ;si+bx 为显示数据的 LED 字型码

out dx,al ;输出字形码mov dx,LED\_IO+1

mov al,01h ;扫描码 01H 送扫描锁存器

out dx,al ;显示pop dx

pop cx pop bx pop ax ret

led\_disp endp

disp\_88 proc ;8\*8LED 点阵（字符）显示程序

push bx push ax push dx

mov al,led\_dat ;取速度档位数值mov di,offset led0

mov bl,8

mul bl ;速度值乘 8，结果存入 ax

add di,ax ;速度值乘 8 再加上 di 得到显示字符代码在内存中偏移量

mov cx,03h ;显示某一字符的时间loop5:

mov count,0000h ;计数器初始=0 mov bh,01h ;产生最高位扫描码

mov al,0 ;扫描码消失（熄灭） mov dx,CS\_8255+1;指向b 口

out dx,al ;输出扫描码熄灭数码管

l88:

mov byte ptr bz,bh;将位扫描码字节送存储空间 bz push di ;保存变量指针

add di,count ;修改变量数据指针（基地址+偏移量） mov bl,byte ptr[di] ;查表得到变量数据送 bl

pop di ;恢复变量指针

mov al,bl ;扫描码消失（熄灭） mov dx,CS\_8255 ;指向 a 口

out dx,al ;输出扫描码熄灭数码管mov al,byte ptr bz ;使相应的数码管亮 mov dx,CS\_8255+1

out dx,al ;显示字形inc count

push cx mov cx,50h

delay88:

loop delay88 ;延时pop cx

mov al,0 ;扫描码消失（熄灭） mov dx,CS\_8255+1;指向b 口

out dx,al ;输出扫描码熄灭数码管mov bh,byte ptr bz;取出扫描码

shl bh,1 ;将扫描码右移一位=01H

jnz l88 ;如果不等于 0 则转 l88

loop loop5 pop dx pop ax pop bx

ret disp\_88 endp code ends end start

## 系统调试运行结果说明、分析所出现得问题，设计体会与建议

实验系统的硬件为 386EX 试验仪，采用 32 位嵌入式模块 CPU386EX 模块，通过 USB 与 PC 机通讯。PC 机运行 386EX 集成开发调试软件 HQFC，可以实现具有 32 位汇编语言的编辑、编译、链接、源语句符号调试功能。

运行结果：在开关均为 0 的情况下，旋转电位器，步进电机速度与电位器同步变化；使用开关

K7 可以改变步进电机转向，K7=0 顺时针转动，K7=1 电机逆时针转动；将开关 K6 拨到高电位可以将电机转速调为零。在程序运行过程中，数码管显示电机转向，8\*8LED 点阵显示转速档位（0-8 档），同时可以通过开关 K5 切换数码管显示内容（方向/速度）。实验结果符合预期要求。

## 结论及设计体会

此次设计实验较为综合，开放程度很高，更多的时候需要我们自己去思考去查阅相关资料，这在一定程度上提高了我们独立解决问题的能力。

## 参考文献

[1] 秦晓梅,巢明. 计算机原理综合实验教程[M]. 大连理工大学电工电子实验中心 2017.10