****

**计算机控制原理与技术**

**课程实验报告**

**学 号 21071003 21071107 21071108**

**姓 名 高立扬 马剑 汤震楚**

**指导教师 宋书瀛**

**提交日期**

**报告成绩评价表**

|  |  |  |  |
| --- | --- | --- | --- |
| 报告结构 | 硬件连线图 | 程序流程图 | 报告成绩 |
| □完全符合要求  □基本符合要求  □有比较多的缺陷  □完全不符合要求 | □符合规范  □基本符合规范  □有一些错误  □完全不正确 | □符合规范  □基本符合规范  □有一些错误  □完全不正确 |  |
| 源代码及注释 | 实验结果 | 实验小结 | 总体评价 |
| □注释完整  □比较完整  □不够完整  □不完整 | □正确符合规范  □有一些错误  □完全不正确  □无 | □比较深刻  □有一定体会  □空泛  □无 |  |

教师签字:

目录

[计算机控制原理与技术 1](#_Toc17639)

[实验一 键盘输入及LED显示输出实验 3](#_Toc25326)

[一、实验目的 3](#_Toc22213)

[二、实验内容 3](#_Toc26096)

[三、硬件连线图 4](#_Toc15116)

[1. 电路图 4](#_Toc11217)

[2. 实物连线图 4](#_Toc24936)

[四、实验程序流程图 4](#_Toc24669)

[1. 基础流程图 4](#_Toc7680)

[2. 实验任务流程图 5](#_Toc10374)

[五、实验源代码及注释 5](#_Toc9682)

[六、实验结果(现象)描述 10](#_Toc25023)

[七、实验小结 10](#_Toc13112)

[1. 遇见的问题： 10](#_Toc3328)

[2. 收获： 11](#_Toc15166)

[实验二 步进电机控制实验 11](#_Toc19252)

[一、实验目的 11](#_Toc8940)

[二、实验内容 11](#_Toc15847)

[三、硬件连线图 11](#_Toc7100)

[1. 电路图 11](#_Toc30302)

[2. 实物连线图 12](#_Toc18633)

[四、实验程序流程图 13](#_Toc11909)

[1. 主程序 13](#_Toc6787)

[2. 加速子程序(左)、匀速子程序(中)、减速子程序(右) 13](#_Toc8803)

[3. 步进一步子程序 14](#_Toc31975)

[五、实验源代码及注释 14](#_Toc21987)

[六、实验结果(现象)描述 16](#_Toc5256)

[七、实验小结 16](#_Toc12856)

[1. 遇见的问题： 16](#_Toc26874)

[2. 收获： 16](#_Toc32559)

[实验三 直流电机控制实验 17](#_Toc3920)

[一、实验目的 17](#_Toc8224)

[二、实验内容 17](#_Toc15820)

[三、硬件连线图 17](#_Toc31073)

[1.电路图 17](#_Toc14850)

[2.实物连线图 18](#_Toc19055)

[四、实验程序流程图 19](#_Toc23787)

[1.初始化及MAIN函数 19](#_Toc27167)

[2.定时器中断处理程序TIME 19](#_Toc1216)

[五、实验源代码及注释 20](#_Toc32148)

[六、实验结果(现象)描述 27](#_Toc19102)

[七、实验小结 27](#_Toc18397)

[1. 遇见的问题： 27](#_Toc16951)

[2. 收获： 27](#_Toc24406)

**实验一 键盘输入及LED显示输出实验**

**一、实验目的**

本实验是“计算机控制原理与技术”后续实验内容的基础，本实验要求学生：

1. 掌握扫描键盘及7段LED显示器的工作原理；

2. 掌握键盘扫描码（键值）和显示输出代码（字型编码）的概念；

3. 掌握51系列单片机汇编程序的编写。

**二、实验内容**

1.编写、调试7段LED显示控制程序，具体要求：

1. 要求将每组同学的班号和学号稳定地显示在LED显示器上；
2. 显示格式为“班号（2 位）+一个同学学号最后2 位+同组另一个同学学号最后2位”，共6位十进制编码。

2.编写、调试键盘输入程序，具体要求：

将最后键入的键号永远显示在最右面的显示块上（LED0），前面的显示内容 依次左移1位，循环反复

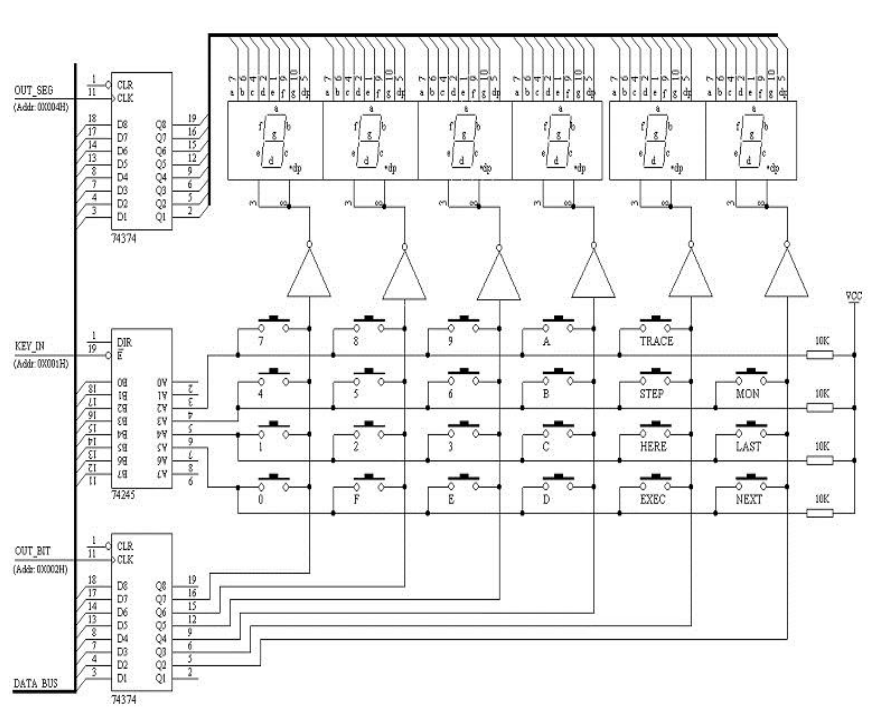
3.编写、调试实时时钟程序，具体要求：

1. 可用键盘键入当前时间：时、分、秒；
2. 1秒钟定时可以采用定时中断的方法，也可以采用软件延时的方法；
3. 按标准进位，正确显示时间。

**三、硬件连线图**

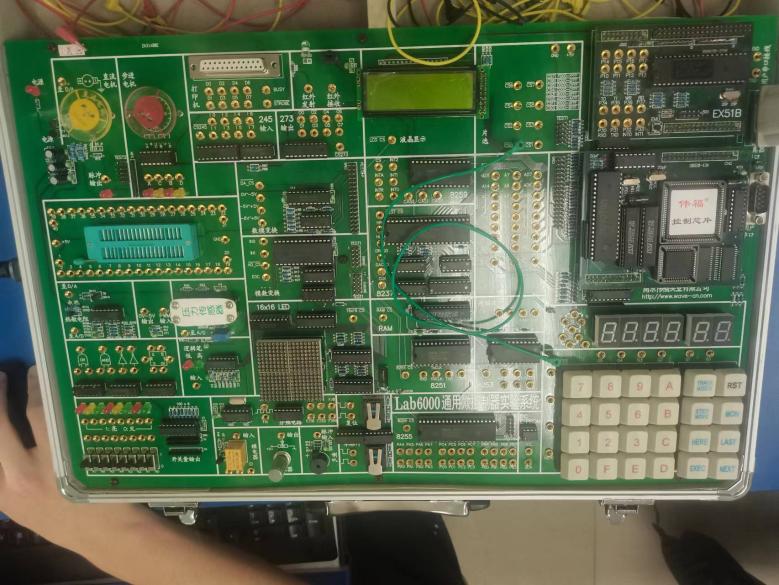
1. 电路图

键盘及LED显示电路如下



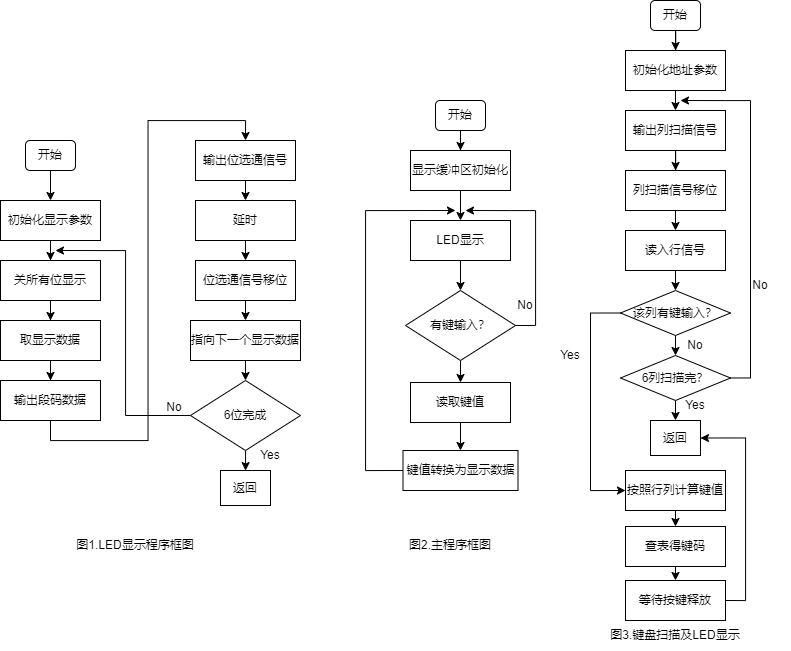
1. 实物连线图

如电路图所示，一共有三个入口，分别是KEY\_IN键盘读入口、OUT\_SEG端控制口和OUT\_BIT位控制口，在实验箱上它是接口，而此三入口需要在汇编程序中设置好偏移量，我们设置的偏移量在8000~8FFF，因此需要接，如下图所示



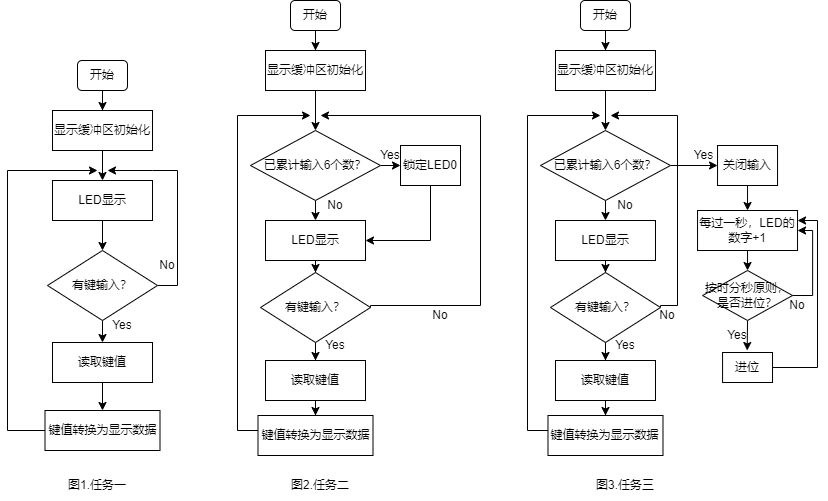
**四、实验程序流程图**

1. 基础流程图



说明：上述三幅流程图是键盘扫描及LED显示的基础流程，根据本实验的三个任务，需要基于上图2进行小修改。

2. 实验任务流程图



说明：三个任务的流程基本一致，略有不同，在代码中稍作修改即可完成。

**五、实验源代码及注释**

OUTBIG EQU 8002H ;位控制口

OUTSEG EQU 8004H ;段控制口

KEYIN EQU 8001H ;键盘读入口

;缓冲器设初值， 赋值到6个数码管

READY: MOV 25H,#08H;8

MOV 24H,#00H;0

MOV 23H,#07H;7

MOV 22H,#00H;0

MOV 21H,#01H;1

MOV 20H,#01H;1

MOV R7,#6

MAIN: ;MOV R7,#6 ;当使按键循环往复时则取消注释该行即可

LCALL DISPLAY ;无键码输入，继续显示当前的数字

LCALL TEST

JZ MAIN ;累加器为0无按键， 继续显示

LCALL SEARCH ;有键值输入?

MOV 20H,21H ;数码管显示出输入的键值

MOV 21H,22H

MOV 22H,23H

MOV 23H,24H

MOV 24H,25H

MOV 25H,A

DJNZ R7, MAIN

MAIN1: MOV R7,#5

LCALL DISPLAY ;无键码输入，继续显示当前的数字

LCALL TEST

JZ MAIN1 ;累加器为0无按键， 继续显示

LCALL SEARCH ;有键值输入?

MOV 20H,21H ;数码管显示出输入的键值

MOV 21H,22H

MOV 22H,23H

MOV 23H,24H

MOV 24H,A

DJNZ R7, MAIN1

CLOCK\_TICK:

LCALL DELONE

LCALL ADD1

SJMP CLOCK\_TICK

DISPLAY: ;显示模块

MOV R0,#20H ;缓冲区从20H处开始

MOV R1,#6 ;共6个八段管

MOV R2,#00100000B ;从左边开始显示灯亮为1，灭为0

LOOP3:

MOV DPTR,#LEDTAB

MOV A,@R0

MOVC A,@A+DPTR

MOV B,A

MOV DPTR,#OUTSEG

MOV A,#0

MOVX @DPTR,A

MOV A,B ;将以20H为首地址的内存值发送到段码输出口8004H

MOVX @DPTR,A

MOV DPTR,#OUTBIG ;输出位选通估号， 使每次只显示一位八段管

MOV A,R2

MOVX @DPTR,A

RR A

MOV R2,A

LCALL DELAY;；延时

INC R0

DJNZ R1,LOOP3;； 显示6次

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELAY: MOV R5,#01H ;延时子程序

DEL1: MOV R6,#00H

DEL2: DJNZ R6,DEL2

DJNZ R5,DEL1

RET

TEST: MOV DPTR,#OUTBIG ;检测有无键值输入

MOV A,#00H

MOVX @DPTR,A ;送出0

MOV DPTR,#KEYIN

MOVX A,@DPTR ;回读行状态

CPL A ;累加器低四位求反

ANL A,#0FH

RET

SEARCH: ;找出键值所在的的位置(行和列)

MOV R1,#00100000B ;初始列，从左边开始

MOV R2,#06H ;找6列

MOV R5,#00H ;记录当前所在列数

MOV R4,#03H ;已经跳过了多少列

LSEARCH: ;找出列所在位置后

MOV A,R1

CPL A

MOV DPTR,#OUTBIG

MOVX @DPTR,A

CPL A

RR A ;通过判断循环几次进位为1来实现计数的功能

MOV R1,A

MOV DPTR,#KEYIN

MOVX A,@DPTR

CPL A

ANL A,#0FH

JNZ HSEARCH ;A为零就找到所在列， 不为零继续寻找行所在位置

INC R5 ;列数加一继续查找

DJNZ R2,LSEARCH

HSEARCH:

MOV R6,#04H;；一共有四列

LOOP2: RRC A

JC GET

DEC R4

DJNZ R6,LOOP2

GET: ;确定输入的键码值

MOV A,R4 ;键值=行\*6+列,R4\*6+R5->A

MOV B,#6H

MUL AB

ADD A,R5

MOV DPTR,#KEYTAB;取出键码所在位置

MOVC A,@A+DPTR

MOV 26H,A ;将取出键值暂时保存在内存中,防止数据丢失

WAIT: MOV DPTR,#OUTBIG ;等键释放

CLR A

MOVX @DPTR,A

LCALL DELAY

LCALL TEST

JNZ WAIT

MOV A,26H ;释放出键盘输入值

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADD1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ADD1: ;20h ---25h

;秒低位

MOV A,25H

INC A

MOV 25H,A

CJNE A,#0AH,QUIT;秒低位不等于10则返回

MOV 25H,#00H

;秒高位

MOV A,24H

INC A

MOV 24H,A

CJNE A,#06H,QUIT;秒高位不等于6则返回

MOV 24H,#00H

;分低位

MOV A,23H

INC A

MOV 23H,A

CJNE A,#0AH,QUIT;分低位不等于10则返回

MOV 23H,#00H

;分高位

MOV A,22H

INC A

MOV 22H,A

CJNE A,#06H,QUIT;分高位不等于6则返回

MOV 22H,#00H

;时低位,特殊处理0~19,20~24

MOV A,20H

CJNE A,#02H,HOUR019

JMP HOUR2024

;时高位

HOUR019:

MOV A,21H

INC A

MOV 21H,A

CJNE A,#0AH,QUIT;时低位不等于10则返回

MOV 21H,#00H

;时高位小于等于1，直接对时高位加一处理

MOV A,20H

INC A

MOV 20H,A

JMP QUIT

HOUR2024:

MOV A,21H

INC A

MOV 21H,A

CJNE A,#04H,QUIT;时低位不等于4则返回

MOV 21H,#00H

;此时时高位将大于2，直接对时高位置0处理

MOV 20H,#00H

JMP QUIT

QUIT:

RET

;\*\*\*\*\*\*\*\*\*\*\*\*delay 1 second\*\*\*\*\*\*\*\*\*\*\*

DELONE:

MOV 27H,#3

L3: MOV 28H,#10

L2: MOV 29H,#10

L1: LCALL DISPLAY

DJNZ 29H,L1

DJNZ 28H,L2

DJNZ 27H,L3

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LEDTAB: ;八段管显示码

DB 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h

DB 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

KEYTAB: ;键盘码

DB 07h, 08h, 09h, 0ah, 10h, 0ffh,

DB 04h, 05h, 06h, 0bh, 11h, 14h,

DB 01h, 02h, 03h, 0ch, 12h, 15h,

DB 00h, 0fh,0eh, 0dh, 13h, 16h,

**六、实验结果(现象)描述**

任务一顺利完成，可以从下图看出班号（711班的“11”）和两位711班同学的学号（“07”和“08”）。

任务二顺利完成，在输入六个数字后，最右边的LED0锁定，如下图所示

任务三顺利完成，在输入时分秒后，停止读取键盘输入，LED的数字每1秒后就+1，秒到60会进位给分，如下图所示

**七、实验小结**

1. 遇见的问题：

在实验中我们遇见了两个问题：任务二在测试过程中，欲通过修改MAIN函数实现LED0的锁定，但是实际测试却无法锁定LED0；针对此问题，我们重新写了一个MAIN1函数，在MAIN函数结束后由于不发生跳转，MAIN1函数被自动执行，无需添加额外的判断逻辑，顺利完成了任务。

其二，则是每次按RST键之后，LED会显示乱码或错误地显示一些数字，这会导致汇编程序再次输入到实验箱后，会错误执行。起初我们不知所措，以为是程序问题或实验箱问题，只能关闭电源重启，后来我们发现，只需要再次按下RST，直至LED灯显示8051即可。

2. 收获：

本实验虽然是实验一，但是我们做完实验二才开始做实验一。在本次实验中，我们熟悉了键盘扫描、键盘输入和LED显示，学习了其控制代码的写法，这为我们完成第三个实验打下了基础。

**实验二 步进电机控制实验**

**一、实验目的**

1. 掌握步进电机的内部结构及工作原理。

2. 掌握开环控制系统的工作原理。

3. 掌握4相步进电机的计算机控制原理。

**二、实验内容**

1. 根据自行设计的方案进行硬件连线。

2. 编写并调试4相步进电机控制程序，具体要求如下：

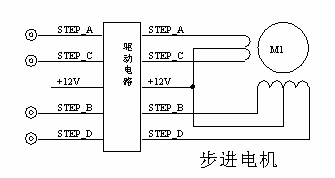
(1) 控制电机按照四相单4拍方式工作。

(2) 按指定的步数、方向、速度运转，即正转加速100步、匀速100步、减速100 步；反转加速100步、匀速100步、减速100步；循环反复。

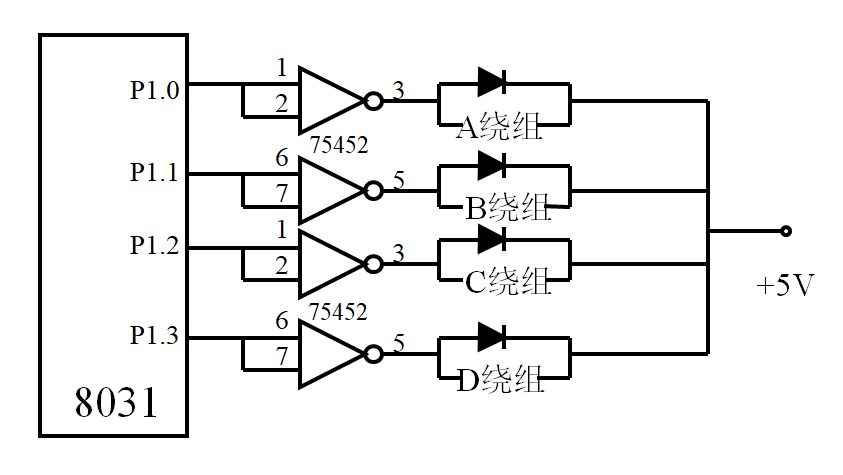
**三、硬件连线图**

1. 电路图

步进电机驱动电路如下：



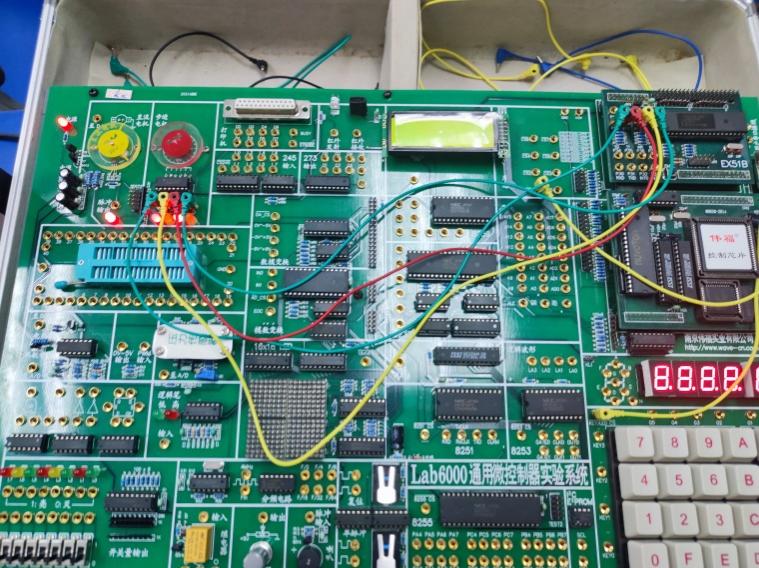
单片机四相小功率步进电机控制系统如下：

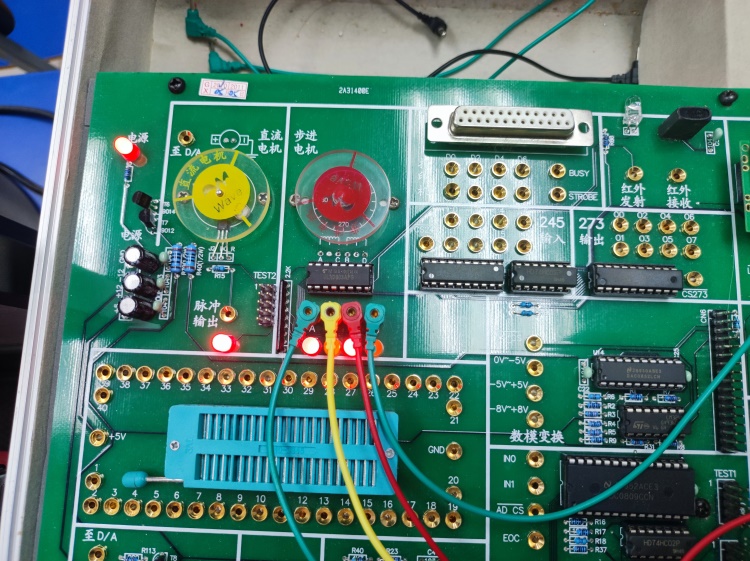


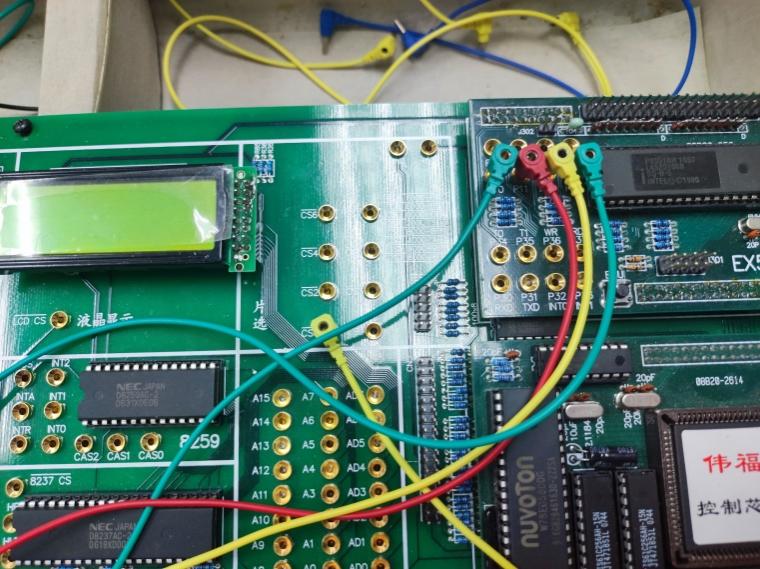
综合上面2图可知，只需要将P1.0，P1.1，P1.2，P1.3分别连接A，B，C，D口即可。

2. 实物连线图

该实验的连线有一个易出错的点：控制步进电机的端口从左到右的顺序是ACBD，而不是ABCD。

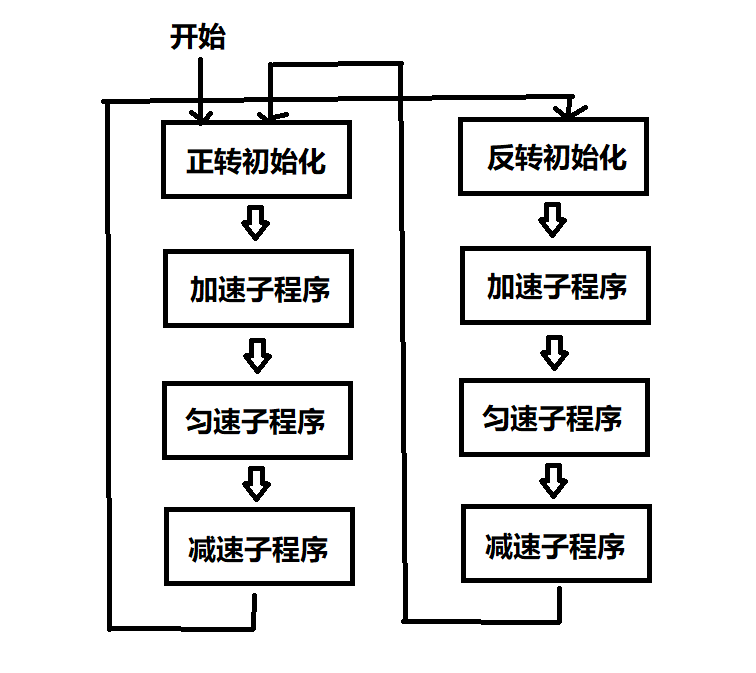






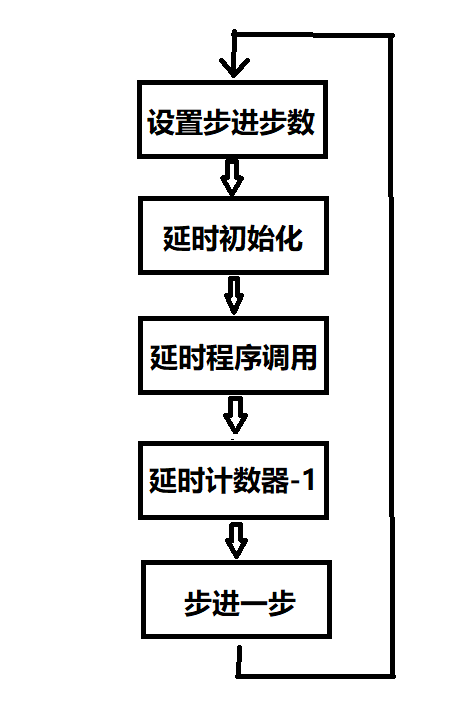
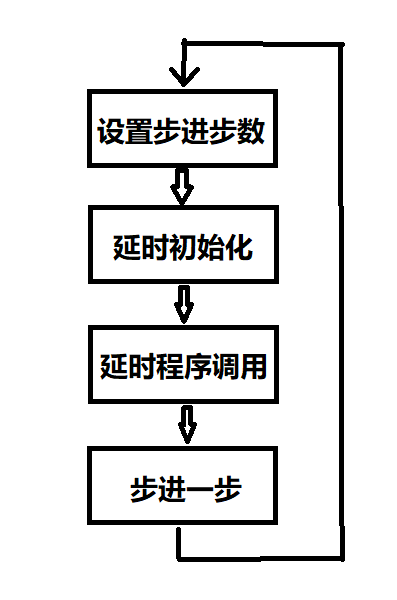
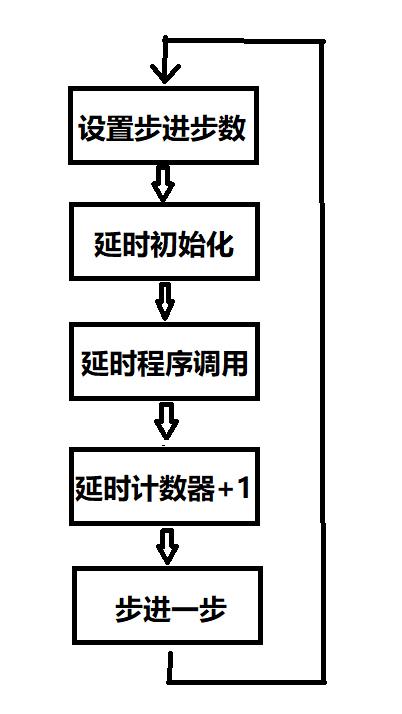
**四、实验程序流程图**

1. 主程序



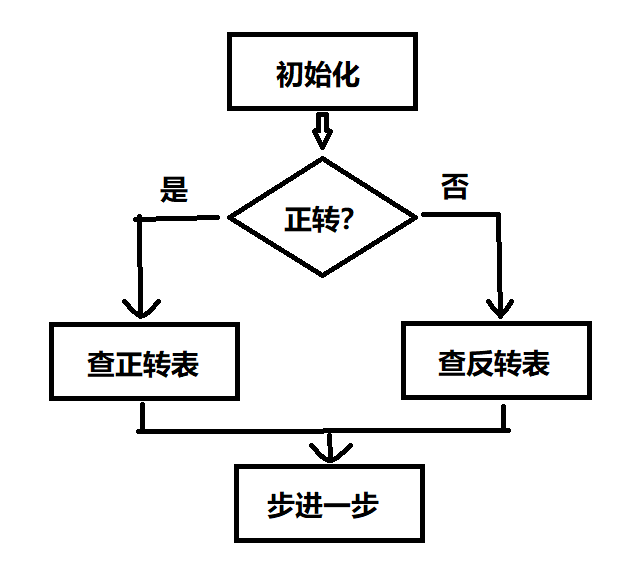
说明：按照正转加速100步、匀速100步、减速100 步；反转加速100步、匀速100步、减速100步；循环反复。

2. 加速子程序(左)、匀速子程序(中)、减速子程序(右)

说明：三者唯一的区别在于延时计数器的设置上。若加速，则计数器-1；若减速，则计数器+1；若匀速，则计数器不变(不设置计数器)。

3. 步进一步子程序



说明：步进一步子程序本质上是一个分支查表程序和步进一步的组合。步进一步通过通电状态字的加载进行控制。

**五、实验源代码及注释**

STRT:MOV 20H, #00H ;20H的D7置0正转,D1D0地址偏移量

MOV P1, #01H ;步进电机初始化7,P1给#01H,A相通电

MOV 42H, #0A0H ;将100送入延迟计数器42H

acall speedup ;调用加速子程序

MOV 20H, #00H ;20H的D7置0正转,D1D0地址偏移量

acall speed ;调用匀速子程序

MOV 20H, #00H ;20H的D7置0正转,D1D0地址偏移量

acall speedlow ;调用减速子程序

MOV 20H, #80H ;20H的D7置1反转,D1D0地址偏移量

MOV P1, #01H ;步进电机初始化,P1给#01H,A相通电

MOV 42H, #0A0H ;100送入延迟计数器42H

acall speedup ;调用加速子程序

MOV 20H, #80H ;20H的D7置1反转,D1D0地址偏移量

acall speed ;调用匀速子程序

MOV 20H, #80H ;20H的D7置1反转,D1D0地址偏移量

acall speedlow ;调用减速子程序

sjmp strt ;循环反复整个过程

;加速步进程序

speedup: MOV R7, #64H ;将100送入步进计数器R7

MLP0: MOV R6, 42H ;延时

MLPX0: LCALL DEL ;调用短延时子程序

DJNZ R6, MLPX0 ;延时

DEC 42H ;将延迟计数器42H中的值减一

LCALL STEPS ;步进一步子程序

DJNZ R7, MLP0 ;步进步数减1,不为零继续步进

ret

;匀速步进程序

speed: MOV R7, #64H ;100送入步进计数器R7

MLP1: MOV R6, 42H ;延时

MLPX1: LCALL DEL ;调用短延时子程序

DJNZ R6, MLPX1 ;延时

LCALL STEPS ;调步进一步子程序

DJNZ R7, MLP1 ;步进步数减1,不为零继续步进

ret

;减速步进程序

speedlow: MOV R7, #64H ;100送入步进计数器R7

MLP2: MOV R6, 42H ;延时

MLPX2: LCALL DEL ;调用短延时子程序

DJNZ R6, MLPX2 ;延时

Inc 42H ;将延迟计数器42H中的值加一

LCALL STEPS ;步进一步子程序

DJNZ R7, MLP2 ;步进步数减1,不为零继续步进

ret

;步进一步子程序

STEPS: INC 20H ;地址偏移量加1

ANL 20H, #83H ;保留20H的D7、D1、D0位状态

JB 7, STPSC ;20H的D7=1反转, D7=0正转

MOV DPTR, #FTAB ;正转表首址送DPTR

SJMP STPW ;跳转到查表程序

STPSC: MOV DPTR, #CTAB ;反转表首址送DPTR

STPW: MOV A, 20H ;查表程序

ANL A, #03H ;取地址偏移量

MOVC A, @A + DPTR ;取通电状态字

MOV P1, A ;输出通电状态字,步进电机走一步

RET

FTAB: DB 01H, 02H, 04H, 08H ;正转表

CTAB: DB 01H, 08H, 04H, 02H ;反转表

;短延时子程序

DEL: MOV R5, #40H ;1T

DEL0: DJNZ R5, DEL0 ;2T

RET ;1T

END

**六、实验结果(现象)描述**

步进电机按指定的步数、方向、速度运转：正转加速100步、匀速100步、减速100 步；反转加速100步、匀速100步、减速100步；循环反复。

符合实验要求。

**七、实验小结**

1. 遇见的问题：

在实验中我们遇见了一个小问题：加速过程和匀速过程极为明显，但减速过程几乎看不见；最终我们通过调节减速过程的所转圈数为200(原先为100)，延长了减速过程的时间，最终观测到了检测过程。

2. 收获：

步进电机虽是实验二，但是是我们做的第一个实验。本次实验中，我们熟悉了伟福6000开发环境以及实验箱的使用。同时，我们掌握了4相步进电机的计算机控制原理，并学习了其控制代码的写法。此外，我们遇到了上述的小问题并使用开发环境进行了调试工作，这让我们熟悉了debug的方法，为后面的实验设计和调试打下了基础。

**实验三 直流电机控制实验**

**一、实验目的**

1. 掌握步进电机的内部结构及工作原理；

2. 掌握开环控制系统的工作原理

3. 掌握 4 相步进电机的计算机控制原理。

**二、实验内容**

1. 根据自行设计的方案进行硬件连线；

2. 编写、调试直流电机转速控制程序，具体要求：

(1) 从键盘输入期望转速值（单位：转/秒），显示在显示器的左边 2 位上；

(2) 采用定时中断的方法，根据单位时间内记取的脉冲数，折算出实际转速，

并显示在 LED 显示器上；

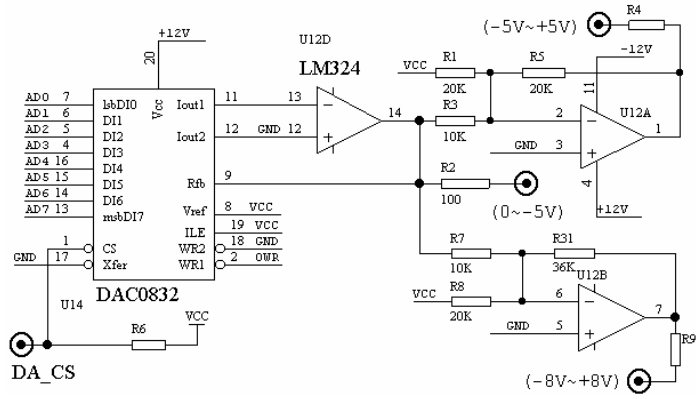
(3) 比较实际转速与期望值之间的关系，如果实际转速小于期望值则加大输

出，反之则减小输出，使得实际转速与期望值相吻合

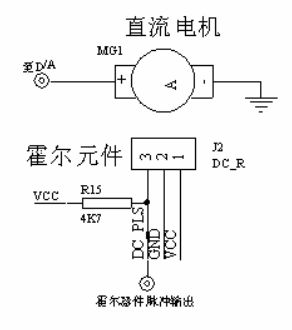
**三、硬件连线图**

## 1.电路图

D/A转换电路图如下：



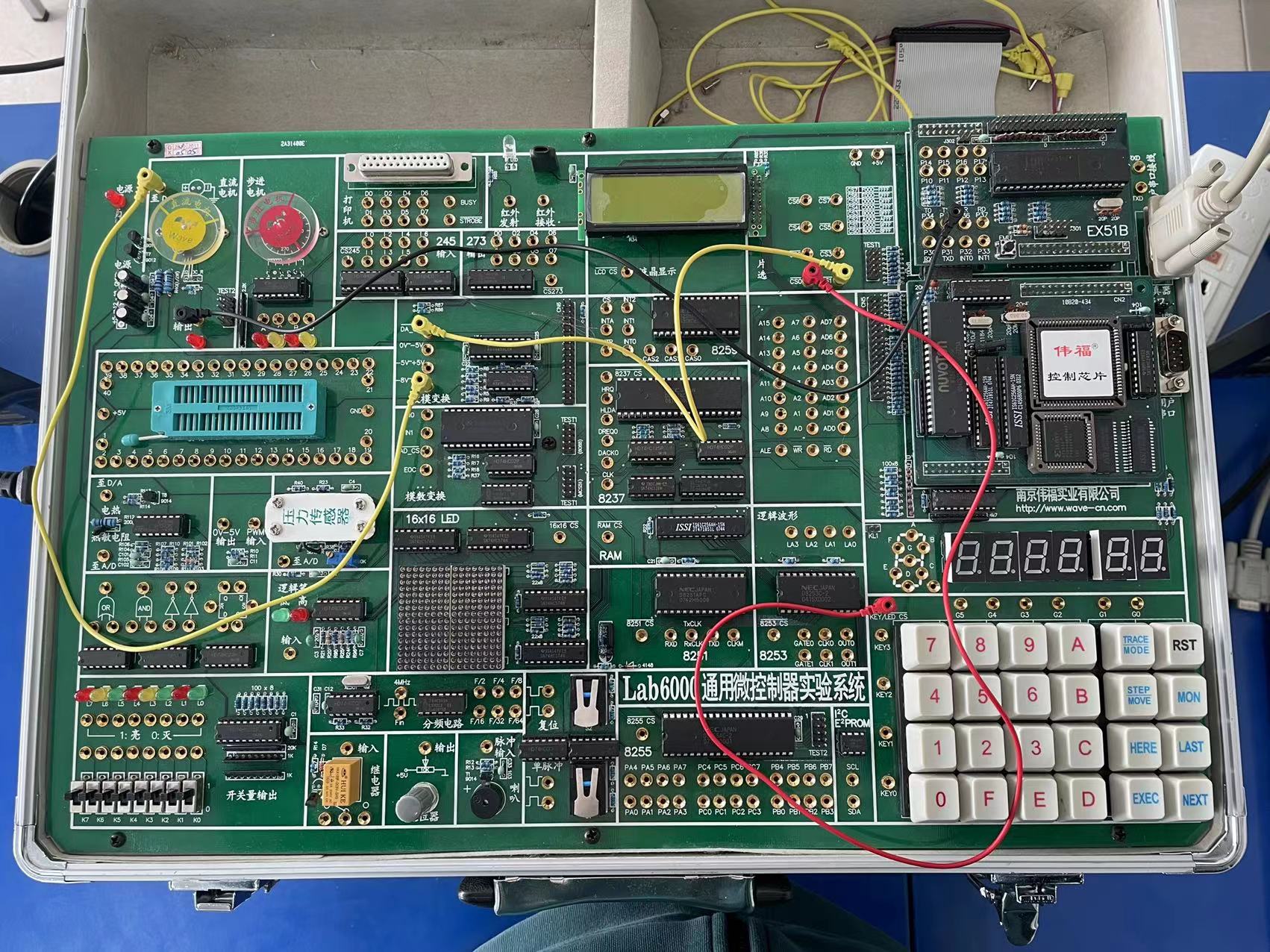
直流电机/霍尔器件电路图如下：



## 2.实物连线图

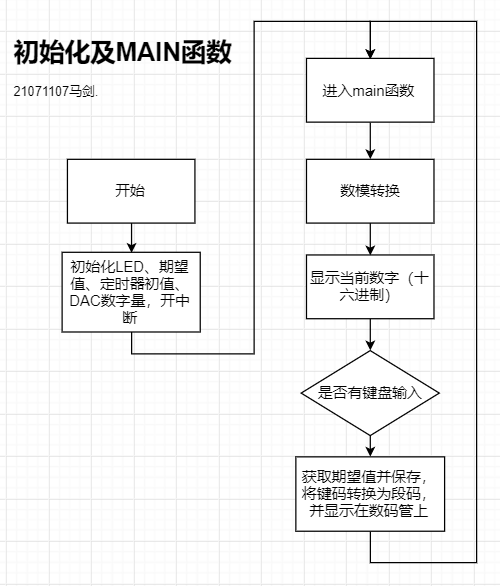
本实验采用控制DAC0832的模拟量输出来控制电机的转速，所以将直流电机输入连到DAC0832的-8V至+8V端，通过单位时间脉冲数计算直流电机转速，所以将霍尔元件脉冲输出输出连到计数器TIMER1的P3.5端口，根据代码中设置的偏移量，将DAC0832的DA\_CS端口连到CS1，KEY/LED 的CS端口连到CS0。

如下图所示

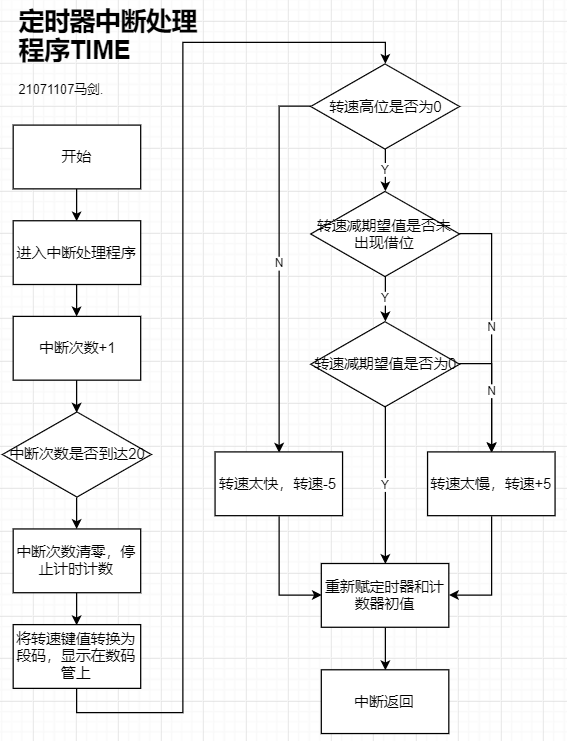


**四、实验程序流程图**

1.初始化及MAIN函数

****

## 2.定时器中断处理程序TIME

****

**五、实验源代码及注释**

以下是实验源代码及注释，在一些实验三中的关键内容进行了补充说明：

; LED显示16进制数

ORG 0000H ; 程序从ROM的0000开始存放程序

OUTBIG EQU 8002H ; 位选信号控制地址

OUTSEG EQU 8004H ; 段选信号控制地址

KEYIN EQU 8001H ; 键盘行选信号控制地址

DAC EQU 9000H ; D/A转换器数字量放置位置, D/A选通信号为CS1

AJMP INIT ; 初始化

ORG 000BH ; T0的中断服务程序地址TIME

AJMP TIME

在这里解释一下ORG与AJMP的作用。当T0定时器中断产生时，默认内部中断的入口是000BH，这时候ORG 000BH告诉了编译器，从000BH地址开始执行下面的语句，也就是AJMP TIME，绝对跳转至我们编写的中断处理程序TIME，完成转速控制的功能。

; T1计数, T0定时(晶振认为是12MHz, 实际是11.0592MHz)

INIT:

MOV SP, #60H ; 设置堆栈指针

; 设置LED显示初值

MOV 20H, #3fH ; 0

MOV 21H, #3fH ; 0

MOV 22H, #3fH ; 0

MOV 23H, #3fH ; 0

MOV 24H, #3fH ; 0

MOV 25H, #3fH ; 0

MOV 26H, #00H ; 存放期望值

MOV 27H, #00H ; 记录中断次数

MOV 28H, #01H ; 记录期望值低位

MOV 29H, #20H ; 29H存放的是数字量, 用于D/A转换,(越大转速越慢，越小转速越快)

MOV TMOD, #51H ; (0101\_0001)B, T1工作于方式1, T0工作于方式1

T0和T1都是由两个8位的计数器所构成的16位计数器，将工作方式设置为1即为使用16位定时器的功能，下面T0的定时常数意为设置定时器中断的周期，也相当于设置了转速更新的频率。

若晶振为12MHZ，则机器周期=12÷12MHZ=1uS，中断周期为（65536-15536）×1uS=50000us=0.05s,所以每20次中断为一秒，每秒调整一次转速。

MOV TL0, #0B0H

MOV TH0, #3CH ; 设置T0定时常数为15536, 若晶振为12MHz, 则每20次中断为1秒

T1用来计霍尔元件传来的脉冲数，配合T0的使用，达成了计算转速的功能。

MOV TL1, #00H

MOV TH1, #00H ; 设置T1计数器计数初值为0

SETB是单片机的位操作命令，在这里意为将该位置1，EA（Enable All）、ET（Enable Timer）寄存器用于控制中断的开关和定时器中断的开关，他们是IE寄存器中的两位。TR0和TR1是定时器T0和T1的运行控制位，他们是TCON寄存器的两位。他们都是置1有效的。

SETB EA ; 允许总中断

SETB ET0 ; 允许T0中断

SETB TR0 ; 启动计时

SETB TR1 ; 启动计数

MAIN:

LCALL DACHANGE ; D/A转换

LCALL DISPLAY ; 无键码输入, 继续显示当前的数字(十六进制)

LCALL TEST ; 测试是否有键盘输入

JZ MAIN ; 累加器为0,无按键,继续显示

LCALL SEARCH ; 有键值输入, 则获取键值, 存在A中

MOV 20H, 21H ; 左侧2位的数码管显示数（期望值）左移

MOV B, 28H ; 期望值高位放在B

MOV 28H, A ; 期望值低位放在A

MOV DPTR, #LEDTAB

MOVC A, @A + DPTR ; 将键值转换为段码

MOV 21H, A ; 将新键值的段码放入左数第二位LED显示

MOV A, #10H ; 将乘数16放入A

MUL AB ; 高位在B, 低位在A

ADD A, 28H ; 期望值结果放A

MOV 26H, A ; 期望值放26H

LCALL DACHANGE ; D/A转换

SJMP MAIN

; D/A转换

DACHANGE:

MOV DPTR, #DAC

MOV A, 29H

MOVX @DPTR, A

RET

; 显示模块

DISPLAY:

MOV R0, #20H ; 缓冲区从20H处开始

MOV R1, #6 ; 共6个八段管

MOV R2, #00100000B ; 从左边开始显示灯亮为1, 灭为0

LOOP:

MOV DPTR, #OUTSEG

MOV A, #0

MOVX @DPTR, A ; 段码设定为0

MOV A, @R0 ; 将以20H为首地址的内存值发送到段码输出口8004H

MOVX @DPTR, A

MOV DPTR, #OUTBIG

MOV A, R2

MOVX @DPTR, A ; 输出位选通信号， 使每次只显示一位八段管

RR A

MOV R2, A

LCALL DELAY ; 延时

INC R0

DJNZ R1, LOOP ; 显示6次

RET

; 延时子程序

DELAY:

MOV R7, #02H

DEL1:

MOV R6, #0ffH

DEL2:

DJNZ R6, DEL2

DJNZ R7, DEL1

RET

; 检测有无键盘输入

TEST:

MOV DPTR, #OUTBIG

MOV A, #00H

MOVX @DPTR, A

MOV DPTR, #KEYIN

MOVX A, @DPTR ; 回读行状态

CPL A ; 累加器A各位求反

ANL A, #0FH ; 取A低4位

RET

; 获取按下的键信息

SEARCH:

MOV R1, #00100000B ; 初始列, 从左边开始

MOV R2, #06H ; 找6列

MOV R5, #00H ; 记录当前所在列

MOV R4, #03H ; 已经跳过了多少列

; 找出有效列和行

LSEARCH:

MOV A, R1

CPL A ; 键盘列扫描低有效，LED位选高有效

MOV DPTR, #OUTBIG

MOVX @DPTR, A

CPL A

RR A ; 始终将LED位选通信号右移，存入R1

MOV R1, A

MOV DPTR, #KEYIN

MOVX A, @DPTR ; 读出键盘行扫描信号(低有效)

CPL A

ANL A, #0FH ; 取低4位

JNZ HSEARCH ; A不为零就找到所在列和行(列存在R1中, 行存在A中)

INC R5 ; A为全零, 则列数加一继续查找

DJNZ R2, LSEARCH ; 至多6次可以找到

; 将行从二进制位表示转化为标号的形式

HSEARCH:

MOV R7, #04H ; 一共有四行

LOOP2:

RRC A

JC GET

DEC R4

DJNZ R7, LOOP2

; 确定输入的键码值

GET:

MOV A, R4 ; 键值在键盘上的偏移量 = 行号 \* 6 + 列号

MOV B, #6H

MUL AB

ADD A, R5

MOV DPTR, #KEYTAB

MOVC A, @A + DPTR

MOV 26H, A ; 将取出键值暂时保存在内存中, 防止数据丢失

WAIT:

MOV DPTR, #OUTBIG ; 等键释放

CLR A

MOVX @DPTR, A

LCALL DELAY

LCALL TEST

JNZ WAIT

MOV A, 26H ; 将键值存入A

RET

; T0的中断处理程序TIME

TIME函数是处理T0中断的函数，一旦T0定时器产生中断，该函数就会启动。该函数的功能如下：首先保存中断现场，接着取出中断次数并读取期望值，之后借助T1计数器的计数，将转速显示到LED上，然后通过对比当前转速和期望值，来选择当前转速应当增加还是减少，最后重置T0和T1的初值，开启中断并启动计时计数，还原中断现场后中断返回。下面是time函数：

TIME:

PUSH PSW

PUSH ACC

PUSH B

PUSH DPL

PUSH DPH

SETB RS0 ; 选择寄存器组1

CLR RS1

MOV R0, 27H ; 取出中断次数

INC R0 ; 中断次数加一

MOV 27H, R0 ; 存回中断次数

CJNE R0, #14H, END1 ; 不是20直接返回

MOV R0, #00H

MOV 27H, R0 ; 存回中断次数

CLR TR0 ; 停止计时

CLR TR1 ; 停止计数

MOV 30H, TL1 ; 计数低8位送入30H

MOV 31H, TH1 ; 计数高8位送入31H

MOV A, 30H ; 被除数存A

MOV B, #10H ; 除数存B

DIV AB ; 结果的商存A, 余数存B

MOV DPTR, #LEDTAB ; 存入显示缓冲区

MOVC A, @A + DPTR ; 将键值转换为段码

MOV 24H, A

MOV A, B

MOVC A, @A + DPTR ; 将键值转换为段码

MOV 25H, A

; 处理高8位

MOV A, 31H ; 被除数存A

MOV B, #10H ; 除数存B

DIV AB ; 结果的商存A, 余数存B

MOV DPTR, #LEDTAB ; 存入显示缓冲区

MOVC A, @A + DPTR ; 将键值转换为段码

MOV 22H, A

MOV A, B

MOVC A, @A + DPTR ; 将键值转换为段码

MOV 23H, A

; 转速控制

MOV A, 31H ; 取转速高位

JNZ SPEEDDOWN ; 高位不是0(转速太快)

MOV A, 30H ; 取转速低位

CLR C ; 清除C位

SUBB A, 26H ; 转速-期望值, 结果存在A

JB CY, SPEEDUP ; 出现借位, 转速偏低, 加速

JZ SUBEND ; 若A全零, 转速和期望值相同则跳转

SJMP SPEEDDOWN ; 否则, 转速太快, 减速

SPEEDDOWN:

MOV A, 29H ; 数字量放入A

ADDC A, #5

MOV 29H, A ; 转速-5

SJMP SUBEND

SPEEDUP:

MOV A, 29H ; 数字量放入A

SUBB A, #5

MOV 29H, A ; 转速+5

SJMP SUBEND

; 重新赋初值然后启动计时计数

SUBEND:

MOV TL0, #0B0H

MOV TH0, #3CH ; 设置T0定时常数为15536, 若晶振为12MHz, 则每20次中断为1秒

MOV TL1, #00H

MOV TH1, #00H ; 设置T1计数器计数初值为0

SETB ET0 ; 允许T0中断

SETB EA ; 允许总中断

SETB TR0 ; 启动计时

SETB TR1 ; 启动计数

END1:

POP DPH

POP DPL

POP B

POP ACC

POP PSW

RETI ; 中断返回只能用RETI

; 段码与数值

;八段管显示码

LEDTAB:

DB 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h

DB 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

;键盘码

KEYTAB:

DB 07h, 08h, 09h, 0ah, 10h, 0ffh

DB 04h, 05h, 06h, 0bh, 11h, 14h

DB 01h, 02h, 03h, 0ch, 12h, 15h

DB 00h, 0fh,0eh, 0dh, 13h, 16h

**六、实验结果(现象)描述**

从键盘输入了期望转速值，并成功显示在显示器的左边 2 位上；

在 LED 显示器上，有实际的转速显示；

在转速小于时期望值，直流电机的转速提升，反之则直流电机转速减小，最终成功令转速与期望值吻合。

**七、实验小结**

1. 遇见的问题：

在设计该实验的过程中，我们发现了一些困难之处。一开始，我们不知道如何初始化和控制T0和T1完成定时和计时的功能，同时不太理解晶振的相关知识导致不会编写time处理程序。后来我们在网上查阅了一些资料并配合着实验手册很快地理解了这些知识，并解决了这些问题。

2. 收获：

直流电机实验感觉与前面的键盘扫描和步进电机控制实验很不一样。它包含了数码管显示、键盘扫描、定时器初始化与控制等诸多模块，属于对一个闭环系统的整体性设计

通过这次实验，我们较为深入的了解了定时器和计数器的相关知识，然后学会了初始化它们和控制它们的相关方法。同时也熟悉了直流电机这种闭环反馈系统与D/A转换电路的综合应用。我们收获颇丰。