

实验三 步进电机原理及应用

一、实验目的和要求

初步学习和掌握 MCS-51 的体系结构和汇编语言，了解 Keil 编程环境和程序下载工具的使用方法。

了解步进电机的工作原理，学习用单片机的步进电机控制系统的硬件设计方法，掌握定时器和中断系统的应用，熟悉单片机应用系统的设计与调试方法。

了解数码管输出的原理及编程方式。

二、实验原理

我们使用的单片机系统的频率是 12M；步进电机转动一周需要 24 步。

本步进电机实验板，使用 FAN8200 作为驱动芯片。CPU 通过如下 4 个引脚与 FAN8200 相连，即：

CPU	FAN8200
P1.1	CE1
P1.4	CE2
P3.2	IN1
P1.0	IN2

本实验使用简单的双四拍工作模式即可，这也是 FAN8200 比较方便的工作方式。只要将 CE1 和 CE2 分别置为高，然后 IN1 和 IN2 按照预定的脉冲输出，即 01→11→10→00→01 这个循环构成一个方向旋转的输出脉冲，将此序列翻转，就是相反方向的输出脉冲。

三、实验设备

单片机测控实验系统

步进电机控制实验模块

Keil 开发环境

STC-ISP 程序下载工具

四、实验内容

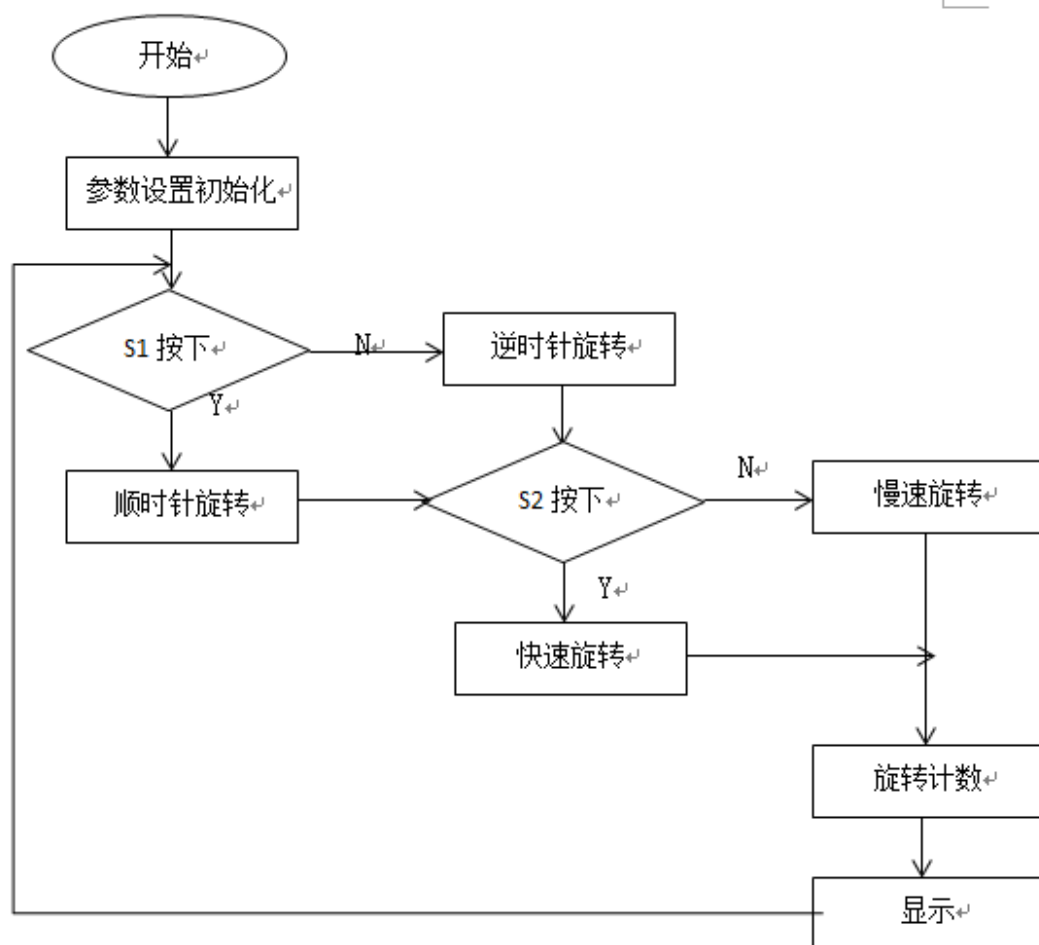
编制 MCS-51 程序使步进电机按照规定的转速和方向进行旋转，并将已转动的步数显示在数码管上。

步进电机的转速分为两档，当按下 S1 开关时，进行快速旋转，速度为 60 转/分。当松开开关时，进行慢速旋转，速度为 10 转/分。当按下 S2 开关时，按照顺时针旋转；当松开时，按照逆时针旋转。

本程序要求使用定时器中断来实现，不准使用程序延时的方式。

五、实验步骤

1. 预习
2. 简单程序录入和调试
3. 程序调试
4. 编写程序，完成功能
5. 流程图



六、实验程序

ORG 0000H

LJMP START

ORG 000BH

LJMP EINT0

ORG 0040H

START:

P4 EQU 0C0H ;p4 作 i/o

P4SW EQU 0BBH ;电复位后设置 P4SW 寄存器

CLK EQU P4.4 ;时钟线

DAT EQU P4.5 ;数据线

SW EQU P3.6 ;低电平有效

MOV P4SW, #70H ;01110000B, P4.4, P4.5, P4.6 作 I/O

MOV DPTR, #TAB ;DPTR 指向 TAB 所在地址

LP:

MOV R3, #0 ;计数

MOV R4, #0

MOV R5, #0

I1: MOV TMOD, #01H ;工作方式

MOV IE, #82H ;全局中断

ORL IP, #2H ;或

SETB P1.1 ;CE1 高

SETB P1.4 ;CE2 高

NEXT: ;转速, 方向

JB P3.7, OPP ;等于 1 转移, 开关二的输入

MOV R0, #00101101B ;按下, 顺时针

MOV 20H, R0 ;R0 复制到 20

LJMP SS1

OPP: MOV R0, #01111000B ;松开, 逆时针

MOV 20H, R0

SS1:

JB P3. 6, SPD

MOV R2, #0H ;按下，快速

LJMP L0

SPD: MOV R2, #1H ;松开，慢速

L0: MOV R1, #4 ;R1 赋 4

MOV R0, 20H

L1: MOV A, R0 ;R0 复制到 A

RLC A ;循环左移

MOV P3. 2, C ;IN1

RLC A

MOV P1. 0, C ;IN2

MOV R0, A

LCALL NUM ;调用 NUM

LCALL TIME

DJNZ R1, L1 ;减一不为 0 不等循环，等于跳转

LJMP NEXT

TIME:

CJNE R2, #1, QUICK

MOV R6, #6 ;慢速，6 次计时

TIM2: MOV TH0, #5DH

MOV TL0, #3EH

SETB TR0 ;启动 t0 定时器

MOV R7, #0H

TIM3: CJNE R7, #1H, TIM3

DJNZ R6, TIM2

LJMP OUT

QUICK:

MOV TH0, #5DH ;定时器 0 启动，快速 60r 每分钟

```

MOV TL0, #3EH
SETB TR0
MOV R7, #0H
TIM1: CJNE R7, #1H, TIM1
OUT:
RET
EINT0:
MOV R7, #1
RETI
NUM: ;显示转动的步数
S0: MOV A, R3
CALL EXP
MOV A, R4
CALL EXP
MOV A, R5
CALL EXP
CJNE R3, #10, S1 ;r3 和 10 不等则执行 s1
MOV R3, #0
CJNE R4, #10, S2
MOV R4, #0
CJNE R5, #10, S3
MOV R5, #0
S1: INC R3 ;R3 自加 1
LJMP STOP
S2: INC R4
LJMP STOP
S3: INC R5
LJMP STOP
STOP:

```

RET

EXP:

MOV 21H, R0

MOVC A, @A+DPTR ;给 DPTR 表的首地址，给 a 数据的转移量，

得到的结果放在 a

MOV R0, #8

CLY: CLR CLK ;P4.4 时钟线低电平

RLC A

MOV DAT, C

SETB CLK ;P4.4 时钟线高电平

DJNZ R0, CLY

MOV R0, 21H

RET

TAB:

DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H

END

七、思考题

1. 如采用单四拍工作模式，每次步进角度是多少，程序要如何修改？

每次步进角度是 15 度。设 A=in1B=in2，(!A)表示 in1=0，(!B)表示 in2=0

输出脉冲修改为：A→B→(!A)→(!B)→A

2. 如采用单双八拍工作模式，每次步进角度是多少，程序要如何修改？

每次步进角度是 7.5 度。

输出脉冲修改为：A→AB→B→B(!A)→(!A)→!A!B→!B→(!B)A

3. 步进电机的转速取决于那些因素？有没有上、下限？

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：当步进电机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（及步进角）。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、电机的电气参数；2、电机与负载的转动惯量；3、电机的工作电源电压。无下限，有上限。步进电机应用于低速场合，每分钟转速不超

过 1000 转，所以适用的最高转速就是 1000 转。

4. 如何改变步进电机的转向？

通过反向 IN1 和 IN2 的输入即可，如将 01→11→10→00→01 改为：
00→10→11→01→00

5. 步进电机有那些规格参数，如何根据需要选择型号？

步进电机的主要参数有最大工作电压、最小启动电压、最大允许功耗和工作频率等。

5. MCS51 中有哪些可存取的单元，存取方式如何？它们之间的区别和联系有哪些？

(1) 工作寄存器组 (00H——1FH)

(2) 可位寻址 RAM 区 (20H——2FH)

(3) 通用的 RAM 区 (30H——7FH)

6. 说明 MOVC 指令的使用方法。

MOVC 用来读取程序存储器；以 16 位的程序计数器 PC 或数据指针 DPT 作为基寄存器，以 8 位的累加器 A 作为变址寄存器，基址寄存器和变址寄存器的内容相加作为 16 位的地址访问程序存储器。如：

MOVC A, @A+PC MOVC A, @A+DPTR

7. MCS51 的指令时序是什么样的，哪类指令的执行时间较长？

一个机器周期包含 6 个状态 (S1-S4)，每个状态分为两个节拍 P1 和 P2，通常，一个机器周期会出现两次高电平 S1P2 和 S4P2，每次持续一个状态 S。乘法及除法指令占 4 个周期，三字节指令均为双周期指令。

8. 在本实验环境下，能否控制显示数码的亮度？如何实现？

能, 通过修改刷新频率

实验四 LED 点阵显示屏

一、实验目的和要求

了解 LED 点阵显示的基本原理和实现方法。掌握点阵汉字库的编码和从标准字库中提取汉字编码的方法。

二、实验原理

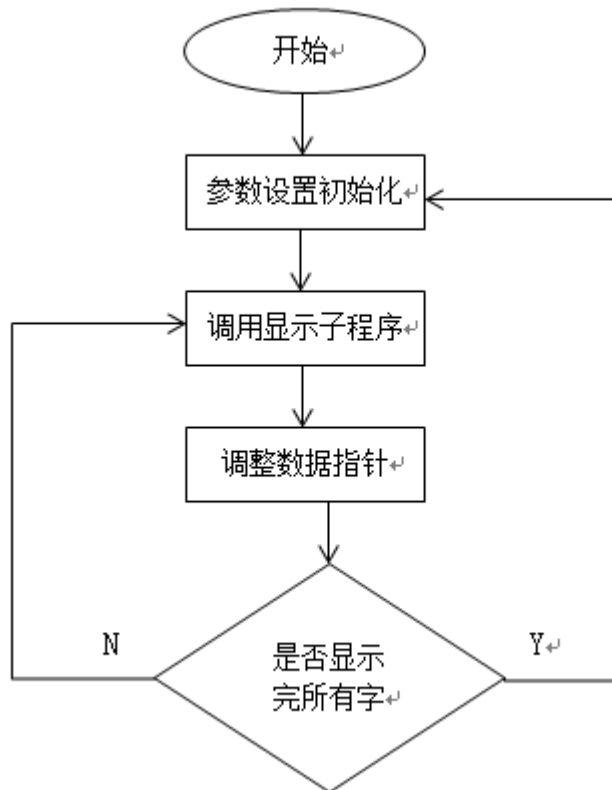
2. 使用 MCS-51 汇编语言, 使用 LED 点阵显示器显示出正确的汉字字符及动

Title		LED点阵显示器 LED(16x16)-V2.0		2011.11	
Size	Number			Revision	
A4					
Date	24-Dec-2013			Sheet of	
Doc	LED_V2.0			24-Dec-2013	

态效果；

3. 将编译后的程序下载到 51 单片机，观察 LED 显示屏的显示结果。

4. 流程图



六、实验代码

循环显示的主要思想是一个虚拟的视窗，可以把所有要显示的字符看作一个 $16 \times N$ 的数组，而这个虚拟的视窗大小为 16×16 （即点阵的大小，视窗从数组的左端开始向右滑动，每次把视窗内的数位输出到点阵中，若视窗内的数位超出了数组的右端，则超出的数位从左端开始计算。

```
ORG 000H
```

```
LJMP START
```

```
ORG 0040H
```

```
START:
```

```
DX EQU P0.0
```

```
DY EQU P0.3
```

```
CLKYWX EQU P0.1
```

```
CLKYWY EQU P0.5
```

CLKCCX EQU P0.2

CLKCCY EQU P0.6

OUTX EQU P0.7

OUTY EQU P0.4

MOV R3, #0FEH

MOV R7, #0

TB:

INC R3

INC R3

CJNE R3, #0, TTT

TTT:

MOV B, R3

PUSH B

MOV R6, #0FH

SM:

POP B

PUSH B

MOV R3, B

MOV B, R6

PUSH B

MOV B, R3

MOV R0, #0

MOV R1, #1

MOV R4, #1

MOV R5, #0

MOV R3, #16

SM16:

SETB OUTX

SETB OUTY

```
CLR  CLKCCX
MOV  DPTR, #TABLE1
MOV  A, R0
MOVC A, @A+DPTR
MOV  R6, #8
```

YW1:

```
CLR  CLKYWX
RLC  A
MOV  DX, C
SETB CLKYWX
DJNZ R6, YW1
MOV  A, R1
MOVC A, @A+DPTR
MOV  R6, #8
```

YW0: CLR CLKYWX

```
RLC  A
MOV  DX, C
SETB CLKYWX
DJNZ R6, YW0
SETB CLKCCX
CLR  OUTX
LCALL DELAY1
CLR  CLKCCY
MOV  DPTR, #TABLE
MOV  A, R4
ADD  A, B
MOVC A, @A+DPTR
MOV  R6, #8
```

YW3:

```
CLR CLKYWY
RRC A
MOV DY, C
SETB CLKYWY
DJNZ R6, YW3
MOV A, R5
ADD A, B
MOVC A, @A+DPTR
MOV R6, #8
```

YW2:

```
CLR CLKYWY
RRC A
MOV DY, C
SETB CLKYWY
DJNZ R6, YW2
SETB CLKCCY
CLR OUTY
LCALL DELAY1
INC R0
INC R0
INC R1
INC R1
INC R4
INC R4
INC R5
INC R5
DJNZ R3, SM16
POP B
MOV R6, B
```

```
DJNZ R6, SM
POP B
MOV R3, B
INC R7
CJNE R7, #32, TEST
JMP START
```

TEST:

```
JMP TB
```

DELAY1:

```
MOV R6, #20
DEL1: MOV R2, #20
DEL2: DJNZ R2, DEL2
      DJNZ R6, DEL1
RET
```

TABLE:

```
DB
OFFH, OFFH, OFFH, OFDH, OBEH, OFDH, OBEH, OFDH, OBEH, OFDH, OBEH, OFDH, 80H, 01H, 0
BEH, OFDH;
```

```
DB
OBEH, OFDH, OBEH, OFDH, OBFH, OFDH, OFFH, OFDH, OCEH, 7FH, OBFH, 81H, 0F9H, 0DFH, 0
F7H, 0DFH; "汪", 0
```

```
DB
OFFH, OFFH, OFFH, OFDH, OBFH, OFDH, OBEH, OFDH, OBEH, OFDH, OBEH, OFDH, OBEH, OFDH
, OBEH, OFDH;
```

```
DB
80H, 01H, OBFH, OFDH, OBFH, OFDH, OBFH, OFDH, 0BCH, 01H, OBFH, OFDH, OBFH, OFDH, OF
FH, OFDH; "正", 1
```

DB

0FFH, 0FFH, 0FDH, 0BFH, 0DDH, 0BFH, 0D5H, 0BFH, 0D4H, 01H, 0D5H, 0BEH, 05H, 0B5H, 0D0H, 0AFH;

DB

0D5H, 3FH, 0D5H, 0CFH, 0DDH, 0F7H, 0FFH, 0FBH, 0CEH, 7FH, 0BFH, 81H, 0F9H, 0DFH, 0F7H, 0DFH;“涛”, 2

TABLE1:

DB 80H, 00H

DB 40H, 00H

DB 20H, 00H

DB 10H, 00H

DB 08H, 00H

DB 04H, 00H

DB 02H, 00H

DB 01H, 00H

DB 00H, 80H

DB 00H, 40H

DB 00H, 20H

DB 00H, 10H

DB 00H, 08H

DB 00H, 04H

DB 00H, 02H

DB 00H, 01H

END

七、思考题

1. 如何使用软件调整和控制 LED 点阵的亮度

通过 PWM 波开启和关闭 LED 来改变正向电流的导通时间，以达到亮度调节的效果。该方法基于人眼对亮度闪烁不够敏感的特性，使负载 LED 时亮时暗。如果亮暗的频率超过 100 Hz，人眼看到的就是平均亮度，而不是 LED 在闪烁。

2. 如何尽量避免显示过程中的闪烁

扫描频率要 50HZ 以上肉眼看起来就不闪烁,就拿 50HZ 阈值,那么一个点阵显示的时间为 $1S/50=20mS$,即需要 20mS 后再对它进行扫描。

3. 如何将本实验的软硬件推广到多行多列的 LED 显示屏 (如 64*1280)

循环显示的主要思想是一个虚拟的视窗,可以把所有要显示的字符看作一个 $16 \times N$ 的数组,而这个虚拟的视窗大小为 16×16 (即点阵的大小,视窗从数组的左端开始向右滑动,每次把视窗内的数位输出到点阵中,若视窗内的数位超出了数组的右端,则超出的数位从左端开始计算。

八、实验的体会与收获

在三周的时间完成了步进电机以及 LED 点阵显示屏的分析与设计,经过这两个实验的学习与联系,我基本上掌握了单片机测控实验系统以及 Keil 开发环境这个软件的一些基本操作,了解了如何运用 Keil 编写代码,如何在单片机测控实验系统中运行自己设计的程序,观察程序的运行状态,分析代码的正确与否。经过这两次实验的学习和训练,我明白了理论和实践确实是两种不一样的形式,只有将理论运用到了实践上,才能真正算掌握了知识。