

单片机实验报告（二）

彭泽宇 53160825 8 班

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

三、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。

四、实验步骤

4.1 预习

参考附录二、附录三和 expr/资料/原理的辅助材料，学习 MCS-51 汇编语言使用和步进电机原理，阅读数码显示器的电路图，重点理解步进电机的工作方式和数码管显示方式。

4.2 简单程序录入和调试

MCS51 单片机汇编语言的基本格式比较简单，程序中使用通用寄存器或者内存单元进行计算。另外，单片机的程序没有退出到操作系统的概念，一般都是死循环程序。

一个简单程序举例如下：

ORG 0000H ;复位起始地址

LJMP START ;中间地址保留给中断向量表

ORG 0040H ;程序实际起始地址

START: ;实际程序

MOV 40H, #0H

```

NEXT:
    MOV A, 40H
    INC A
    MOV P0, A ;板上的 P0 口连接到 8 个 LED，可以监视运行状态
    MOV 40H, A
    MOV R6, #0FFH
L2: MOV R7, #0FFH
L1: DJNZ R7, L1
    DJNZ R2, L2 ;延迟一段时间
    LJMP NEXT
END

```

本程序需要使用定时器定时，并使用中断来同步。中断程序的典型例子如下：

```

ORG 0000H
LJMP START
ORG 000BH
LJMP T0IN ;中断向量表
ORG 0040H
START: ...;初始化
...
T0IN: ...;中断程序
...
RETI ;中断返回
END

```

4.3 程序调试

用单步、断点、连续方式调试程序，观察状态指示灯及电机状态，检查运行结果。如果需要，可以将四个输出信号的状态同时输出到 P0 口的某些位上，便于观察。

4.4 编写程序，完成功能

```

ORG 0000H          ;复位起始地址
    LJMP START
ORG 000BH          ;中间地址保留给中断向量表
    LJMP EINT0      ;定时器 0 中断程序入口地址
ORG 0040H          ;程序实际起始地址
START:
    P4 EQU 0C0H
    P4SW EQU 0BBH
    CLK EQU P4.4 ;时钟线
    DAT EQU P4.5 ;数据线
    SW EQU P3.6

    MOV P4SW,#70H
    MOV DPTR,#TAB    ;s = 23869 --5D3E

```

```

LP:
    MOV R3,#0        ;计数
    MOV R4,#0
    MOV R5,#0

```

I1: MOV TMOD,#01H ;选择工作方式，即对 TMOD 赋初值。T0 工作在方式 1，16 位的计数器 GATE(GATE 一门控位，控制定时器的两种启动方式，)等于 0，不受外部控制

MOV IE,#82H ;全局中断，T0 中断允许 中断控制字;直接对中断允许寄存器
IE 和 优先级寄存器 IP 设置
;ORL IP,#2H ;逻辑或，T0 中断优先级高

SETB P1.1 ;CE1 置高
SETB P1.4 ;CE2 置高

NEXT:

JB P3.7,OPP ;如果 P3.7 等于 1 则转移
MOV R0,#00101101B ;按下，顺时针
MOV 20H,R0

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 ;23870 5D3E
LJMP L0

SPD: MOV R2,#1H ;松开，慢速

L0: MOV R1,#4
MOV R0,20H

L1: MOV A,R0
RLC A ;循环左移操作
MOV P3.2,C ;IN1
RLC A
MOV P1.0,C ;IN2
MOV R0,A
LCALL NUM
LCALL TIME
DJNZ R1,L1

LJMP NEXT

TIME:

CJNE R2,#1,QUICK
MOV R6,#6 ;慢速 ;分 6 次来计时

TIM2: MOV TH0,#5DH
MOV TL0,#3EH
SETB TR0
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,60 转/分
MOV TL0,#3EH
SETB TR0

```

        MOV R7,#0H
TIM1: CJNE R7,#1H,TIM1
OUT:
RET
EINT0:
    MOV R7,#1
RETI
NUM:      ;显示已转动的步数，每转动一次显示一个数
S0:  MOV A,R3
      CALL EXP
      MOV A,R4
      CALL EXP
      ;mov r5,0ffh
      MOV A,R5
      CALL EXP

      CJNE R3,#9,S1
      MOV R3,#0
      CJNE R4,#9,S2
      MOV R4,#0
      CJNE R5,#9,S3
      MOV R5,#0

S1:  INC R3
      LJMP STOP
S2:  INC R4
      LJMP STOP
S3:  INC R5
      LJMP STOP
STOP:
RET

EXP:
    MOV     21H,R0
    MOVC A,@A+DPTR
    MOV R0,#8
CLY: CLR CLK ;P4.4 ;时钟线低电平
      RLC A ;累加器 A 的逻辑操作指令
      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

```

五、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。

数码管显示

本开发平台有 3 个数码管，使用串行方式连接在一起，具体电路参见实验原理。要想输出一个字形码，就需要从高位到低位依次向移位寄存器输出 8 个比特。移位寄存器的数据线和时钟线分别接到单片机的 P4.5 和 P4.4 管脚，可以使用 MCS-51 里面的位操作指令进行输出。连续输出 3 个字形，24 个 bit 之后，欲显示的字形将稳定地显示在数码管上，程序可以转而执行其他工作。七段字形的编码方式需要通过实验获得。这些编码作为程序中的常数，使用 DB 命令存放。在程序中，需要将数值转换为相应的字形编码，可以使用 MOVC 指令来完成。

六、 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？

答：每次步进角度是 15 度；

设 $A = in1, B = in2$, (!A)表示 $in1=0$, (!B)表示 $in2=0$

则输出脉冲修改为： $A \rightarrow B \rightarrow (!A) \rightarrow (!B) \rightarrow A$

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？

答：每次步进角度是 7.5 度。

输出脉冲修改为： $A \rightarrow AB \rightarrow B \rightarrow B(!A) \rightarrow (!A) \rightarrow !A!B \rightarrow !B \rightarrow (!B)A$

3. 步进电机的转速取决于那些因素？有没有上、下限？

答：步进电机的转速主要由时钟的周期控制，通过改变输入脉冲的个数决定转过的角度；转速有上限，通过加大控制电压和降低线圈的时间常数可以提高上限；转速无下限。

4. 如何改变步进电机的转向？

答：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？

答：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。

6. 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？

答：（1）工作寄存器组（00H——1FH）

内部 RAM 的 0-1FH 为四组工作寄存器区，每个区有 8 个工作寄存器（R0—R7）。在同一时刻，只能使用一组工作寄存器，这是通过程序状态字 PSW 的位 3，4 位来控制的。例如当此两位为 00 时，使用第 0 组工作寄存器，对应于 00H 到 07H 的内部 RAM 空间。也就是说，这时指令中使用 R0 与直接使用 00 单元是等价的，不过使用工作寄存器的指令简单，且执行快。

（2）可位寻址 RAM 区（20H——2FH）

内部 RAM 的 20H—2FH 为位寻址区域，这 16 个单元的每一位都对应一个位地址，占据位地址空间的 0—7FH，每一位都可以独立置位、清除、取反等操作。

（3）通用的 RAM 区（30H——7FH）

在中断和子程序调用中都需要堆栈。MCS—51 的堆栈理论上可以设置在内部 RAM 的任意区域，但由于 0—1FH 和 20—2FH 区域有上面说的特殊功能，因此一般设置在 30H 以后。

在内部 RAM 中，所有的单元都可以作为通用的数据存储器使用，存放输入的数据或计算的中间结果等。也可以作为条件转移的条件使用。

7. 说明 MOVC 指令的使用方法。

答：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 DPTR 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
```

8. MCS51 的指令时序是什么样的，哪类指令的执行时间较长？

答：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。

9. 在本实验环境下，能否控制显示数码的亮度？如何实现？

答：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

提取汉字编码的方法。

二、实验设备单片机测控实验系统

LED 点阵显示器实验模块

Keil 开发环境

STC-ISP 程序下载工具

三、实验内容

了解 16*16 点阵电路的原理，编写汇编语言程序。

编写一行汉字字符（至少三个字）的显示程序。

能够从左到右（或从右到左）循环显示（要求显示过程中字的大小与屏幕尺寸相适应）。

四、实验步骤

1. 掌握点阵式 LED 显示屏的控制方法；
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。

五、实验原理

高亮度 LED 发光管构成点阵，通过编程控制可以显示中英文字符、图形及视频动态图形。所显示字符的点阵数据可以自行编写（即直接点阵画图），也可从标准字库（如 ASC16、HZ16）中提取。后者需要正确掌握字库的编码方法和字符定位的计算。

实验用的 LED 点阵显示屏为 16*16 点阵。

行和列分别使用两个移位寄存器作为输出。

当移位寄存器输出的第 i 行为 0，第 j 列为 1 时点亮点(i,j)。

为了能够显示出一个点阵字型，需要进行循环扫描，也就是每一次只点亮一行，然后在列上输出该列对应的 16 个点阵值。

输出一行后暂停一段时间，输出下一行。为了达到较好的显示效果，整屏总的扫描时间不高于 40ms。

上述过程中行列可以互换。

实验中使用的移位寄存器是 74HC595，它是一个同时具有串行移位和输出锁存驱动功能的器件。

74HC595 是具有 8 位移位寄存器和一个存储器，三态输出功能。移位寄存器和存储器是分别的时钟。

数据在 SRCK（移位寄存器时钟输入）的上升沿输入到移位寄存器中，在 RCK（存储器时钟输入）的上升沿输入到存储寄存器中去。

移位寄存器有一个串行移位输入（行 D_x （P00）、列 D_y (P03)），和一个串行输出（QH），和一个异步的低电平复位，存储寄存器有一个并行 8 位的，具备三态的总线输出，当使能（P02 和 P07 为低电平）时，存储寄存器的数据输出到总线。

在控制 74HC595 时，首先将数据放到串行输入的 SI 端，然后在串行时钟 SRCK 上产生一个脉冲，即可输出一个 bit，重复以上步骤 16 次，输出所有列值。

然后给存储器时钟 RCK 一个脉冲，将串行数据锁存起来。将使能端 输出低电平，驱动到 LED 点阵上。

行的输出每次只移位一次，并重新锁存即可。

其他信息见给定的参考资料。

六 实验代码

```
ORG 0000H
```

```
AJMP START
```

```
ORG 001BH
```

```
LJMP TIME1
```

```
ORG 0030H
```

```
START:
D_X EQU P0.0
CK_X EQU P0.1
CK_XL EQU P0.2
D_Y EQU P0.3
EN_Y EQU P0.4
CK_Y EQU P0.5
CK_YL EQU P0.6
EN_X EQU P0.7
COUNT EQU R0
COUNTH EQU R3
COUNTL EQU R4
MOV  TMOD,#12H
MOV  TH1,#3CH
MOV  TL1,#0B0H
```

```
;????
```

```
SETB  EA
SETB  ET1
SETB  TR1
MOV R0,#0
MOV R1,#8
MOV R2,#16
MOV COUNTH,#0
MOV COUNTL,#0
MOV R5,#33;
CLR EN_X;
CLR EN_Y
MOV  R7,#0
mov R6,#0
```

```

MAIN:

CLR A

MOV COUNTL,A;

CLR A

MOV A,COUNT

MOV COUNTH,A ;

LOOP:

LCALL SCANL;

LCALL SCANH;

CLR CK_YL

NOP

SETB CK_YL

CLR CK_XL

NOP

SETB CK_XL

DJNZ R2,LOOP;

MOV R2,#16

;mov R6,#0      μ±3??ú?aà??3?μ??μ???ò??±?ú?aà???ì?2?3?è?
μ÷ê?ê±?D??à?????ì??'DD2??aμà???'??

CJNE  R6,#1,MAIN

mov R6,#0

MOV   R1,#8

INC   COUNT

INC   COUNT


DJNZ  R5,MAIN

AJMP  START


SCANL:

MOV DPTR,#TAB_L

```

```
CLR A
MOV A,COUNTL
MOVC A,@A+DPTR
MOV R1,#8
```

LP_L1:

```
RLC A;
    CLR CK_Y;
MOV D_Y,C;
SETB CK_Y;
    DJNZ R1,LP_L1 ;
```

```
MOV R1,#8
INC COUNTL;
```

```
CLR A
MOV A,COUNTL
MOVC A,@A+DPTR
```

LP_L2:

```
RLC A
CLR CK_Y
MOV D_Y,C
SETB CK_Y
    DJNZ R1,LP_L2
```

```
MOV R1,#8
INC COUNTL
```

```
CLR EN_Y;
RET
```

SCANH:

MOV DPTR,#TAB_H

CLR A

MOV A,COUNTN

MOVC A,@A+DPTR

MOV R1,#8

LP_H1:

RLC A;

CLR CK_X;

MOV D_X,C;

SETB CK_X;

DJNZ R1,LP_H1 ;

MOV R1,#8

INC COUNTN;

CLR A

MOV A,COUNTN

MOVC A,@A+DPTR

LP_H2:

RLC A

CLR CK_X

MOV D_X,C

SETB CK_X

DJNZ R1,LP_H2

MOV R1,#8

INC COUNTH

CLR EN_X;

RET

TIME1:

PUSH ACC

PUSH PSW

INC R7 ;R1??,??

CJNE R7,#20,RETURN1 ;???20(??20)??return1,TIME1??20?,??1??

mov R7,#0

MOV R6,#1

RETURN1:

MOV TH1,#3CH

MOV TL1,#0B0H

pop psw

pop acc

RETI

TAB_H:

DB

00H,02H,00H,0CH,3FH,0F0H,22H,02H,2AH,0C4H,2AH,0A8H,2AH,92H,0AAH,81H;

DB

7FH,0FEH,2AH,0A0H,2AH,90H,2AH,88H,2FH,94H,22H,22H,22H,02H,00H,00H;"彭",0

DB

00H,80H,00H,80H,10H,80H,10H,80H,10H,80H,10H,80H,0FFH,0FFH,10H,80H;

DB

10H,80H,10H,80H,9FH,84H,40H,82H,20H,84H,00H,0F8H,00H,00H,00H,00H;"泽",1

```
DB
00H,00H,1FH,0F0H,12H,20H,0FFH,0FFH,12H,20H,1FH,0F0H,00H,00H,42H,22
H;
```

```
DB
46H,44H,4BH,88H,52H,32H,63H,0C1H,42H,02H,03H,0FCH,00H,00H,00H,00H;
"宇",2
```

TAB_L:

```
DB
~80H,0FFH,~40H,0FFH,~20H,0FFH,~10H,0FFH,~08H,0FFH,~04H,0FFH,~02
H,0FFH,~01H,0FFH
```

```
DB
0FFH,~80H,0FFH,~40H,0FFH,~20H,0FFH,~10H,0FFH,~08H,0FFH,~04H,0FFH
,~02H,0FFH,~01H
```

END

七、思考题

1. 如何使用软件调整和控制 LED 点阵的亮度

答：可以通过控制行显示延时调整亮度。延时越短，LED 点阵越亮；延时越长，LED 点阵越暗。

2. 如何尽量避免显示过程中的闪烁

答：增加每一屏显示次数。

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）

答：可将 64*1280 看做由 4*80 也即 320 块 16*16 显示屏组成的，显示方法与本实验相同。