

计算机控制与应用实验

实验三 实验报告

计算机科学与技术 八班

李天祥 53160815

实验四 步进电机原理及应用

一、实验目的和要求

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。

二、实验设备

单片机基础实验开发平台
步进电机/直流电机实验板
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

参考辅助材料，学习8031 汇编语言使用和步进电机原理

4.2 程序输入

本程序需要使用定时器定时（参见附录三—2.5），并使用中断来同步（参见附录三—2.7）

4.3 定时器中断

在使用定时器时，首先应由外部条件得到要定时的时间长度 t ，如本实验中，就是根据要求的速度计算出的每一步之间的间隔。然后选择适当的定时器工作方式，去计算想要设定的计数器初值 s ，使用如下方程。

$$(2^{\text{定时器最大位数}} - s) \times \text{定时周期} = t$$

定时周期 = $12/\text{CPU 晶振频率}$

得到的 s 需要分成高8 位和低8 位，

分别放入计数器THx 和TLx 中（x 为0 或1）。如果 s

为负数，说明需要的定时时间太长，即使定时器的最大时间也无法满足要求。这种情况下，需

要加入软件循环才能实现。我们可以将需要的定时时间分成 n 份，利用定时器达

到 t/n 的时间

长度，然后在定时器处理程序中，累计某一变量，如果到达 n ，说明总的时间 t 已经达到。

要想使用定时器中断，除了上面的定时器初值设定外，还需要将其他相关的特殊功能寄存

器也都设置好。如果使用方式0 和方式1，不要忘记在计数结束后重新恢复计数器初值。

4.4 程序调试及现象观测

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

4.5实验代码

```
ORG 0000H ;复位起始地址
```

```
LJMP START
```

```
ORG 000BH ;中间地址保留给中断向量表，定时器0中断程序入口地址
```

```
LJMP EINT0
```

```
ORG 0040H ;程序实际起始地址
```

```
START:
```

```
P4 EQU 0C0H ;单片机P4口地址
```

```
P4SW EQU 0BBH
```

```
CLK EQU P4.4 ;移位寄存器的时钟线接单片机模拟串口时钟
```

```
DAT EQU P4.5 ;移位寄存器的数据线接单片机模拟串口数据
```

```
MOV P4SW, #70H
```

```
MOV DPTR, #TAB ;将段码表首地址传给数据指针寄存器，TAB为发光管设置
```

```
LP:
```

```
MOV R3, #0 ;数码管显示用，代表个位
```

```
MOV R4, #0 ;数码管显示用，代表十位
```

```
MOV R5, #0 ;数码管显示用，代表百位
```

```
I1:
```

```
MOV TMOD, #01H ; 选择定时工作方式，即对TMOD赋初值。TMOD是定时器、计数器模式控制寄存器
```

```
MOV IE, #82H ;中断允许寄存器IE，中断总体允许控制位置EA置1，定时器T0溢出中断允许控制位置1
```

```
ORL IP, #2H ;优先级寄存器IP，设置定时器T0中断优先级高
```

```
SETB P1.1 ;步进电机驱动芯片的CE1置高（SETP作用为将端口置1）
```

```
SETB P1.4 ;步进电机驱动芯片的CE2置高
```

```
NEXT:
```

```
JB P3.7, OPP ;单片机的P3.7位接S2开关，如果P3.7等于1，即S2按下，则跳转到OPP
```

```
MOV R0, #00101101B ;松开S2, 步进电机逆时针转动
MOV 20H, R0 ;将步进电机的脉冲时序保存到20H地址中
LJMP SS1; LJMP为长距离跳
OPP:
MOV R0, #01111000B ;按下S2, 步进电机顺时针转动
MOV 20H, R0 ;将步进电机的脉冲时序保存到20H地址中
```

```
SS1:
JB P3.6, SPD ;单片机的P3.6接S1开关, 如果P3.6等于1, 即S1按下, 则跳转
MOV R2, #0H ;松开S1, 步进电机慢速旋转;23870 5D3E
LJMP L0
SPD:
MOV R2, #1H ;按下S1, 步进电机快速旋转
```

```
L0:
MOV R1, #4 ;根据步进电机的双四拍工作模式, 将对应的循环次数4保存到R1中
MOV R0, 20H ;取出步进电机的脉冲时序(顺时针/逆时针), 存入R0中
L1:
MOV A, R0 ;将R0保存的脉冲时序(顺时针/逆时针), 存入累加器A中
RLC A ;累加器A循环左移, 进位位C受到影响
MOV P3.2, C ;将脉冲的高一位送入步进电机驱动芯片与单片机P3.2相连的IN1中
RLC A ;累加器A循环左移, 进位位C受到影响
MOV P1.0, C ;将脉冲的低一位送入步进电机驱动芯片与单片机P1.0相连的IN2中
MOV R0, A ;将累加器A循环左移两次后的结果保存到R0中
LCALL NUM ;调用LED显示器的子程序, 显示步进电机已转动的次数
LCALL TIME ;调用定时器的子程序
DJNZ R1, L1 ;R1先减1, 若不等于0, 则跳转, 否则顺次执行下一条指令
```

```
LJMP NEXT ;重新判断开关是否按下, 这是一个死循环
```

```
;定时器子程序
```

```
TIME:
CJNE R2, #1, SLOW ;R2表示步进电机转速, 等于0慢速, 跳转到SLOW; 等于1快速,
顺次执行下一条指令
MOV TH0, #5DH ;快速, 60转/分, TH0为高八位寄存器
MOV TL0, #3EH; TL0为低八位寄存器
SETB TR0 ;TR0置1, 定时器0启动, 定时器0启动跳转至中断地址
MOV R7, #0H ;R7为中断判断标志, 置0
TIM1:
CJNE R7, #1H, TIM1 ;R7若小于1则跳转, 等待定时器中断产生; 若等于0则顺次执行
下一条指令
LJMP OUT
```

```
SLOW:
```

```

MOV R6, #6 ;慢速, 10转/分, 设置嵌套循环6次
TIM2:
MOV TH0, #5DH
MOV TL0, #3EH
SETB TR0 ;TR0置1, 定时器0启动
MOV R7, #0H ;R7为中断判断标志, 置0
TIM3:
CJNE R7, #1H, TIM3 ;R7若小于1则跳转, 等待定时器中断产生; 若等于0则顺次执行下一条指令
DJNZ R6, TIM2 ;先将R6减1, 若不为0则跳转, 否则顺次执行下一条指令

OUT:
RET ;返回主程序处

;定时器0中断处理程序
EINT0:
MOV R7, #1 ;中断标志置1
RETI

;调用LED显示器的子程序, 显示步进电机已转动的次数
NUM:
S0:
MOV A, R3 ;将R3表示的个位传给累加器A
CALL EXP ;调用数码管显示子程序, 显示个位
MOV A, R4 ;将R4表示的十位传给累加器A
CALL EXP ;调用数码管显示子程序, 显示十位
MOV A, R5 ;将R5表示的百位传给累加器A
CALL EXP ;调用数码管显示子程序, 显示百位

CJNE R3, #9, S1 ;R3表示的个位若等于9, 表示要进位, 则顺次执行下一条指令, 否则跳转
MOV R3, #0 ;个位清零
CJNE R4, #9, S2 ;R4表示的十位若等于9, 表示要进位, 则顺次执行下一条指令, 否则跳转
MOV R4, #0 ;十位清零
CJNE R5, #9, S3 ;R5表示的百位若等于9, 表示要进位, 则顺次执行下一条指令, 否则跳转
MOV R5, #0 ;百位清零

S1:
INC R3
LJMP STOP
S2: INC R4
LJMP STOP

```

```
S3: INC R5
LJMP STOP
STOP:
RET
```

EXP: ; 显示程序

```
MOV 21H, R0; 相当于压栈, 保存调用该子程序前R0的值, 防止修改R0的值
MOVC A, @A+DPTR; 变址寻址方式, 将累加器(变址寄存器)与数据指针寄存器(基址寄存器)保存的数相加, 并将该数(地址)的内容传给累加器A, 得到要显示在某一发光管上的字段码
MOV R0, #8; 设置循环8次, 从高位到低位依次向移位寄存器输出8个比特
```

CLY:

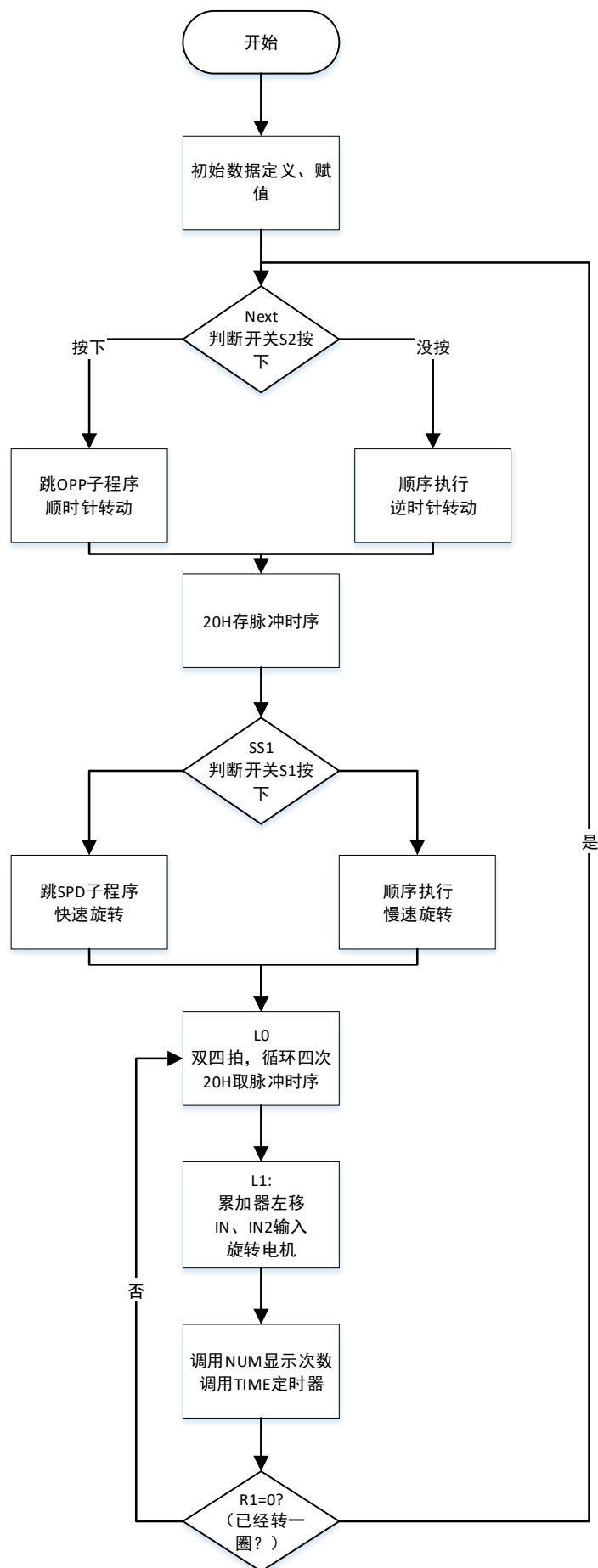
```
CLR CLK ;P4.4 ;CLR清0, 移位寄存器的时钟线给低电平
RLC A ;累加器A的逻辑左移, C进位位置0或1
MOV DAT, C;P4.5位, 即将C所表示的0或1送入移位寄存器中(给相应的字段码低电平或高电平, 即相应的数码管不亮或亮)
SETB CLK ;P4.4位置1, 即时钟线给高电平, 产生上升沿, 移位寄存器向低位右移一位(当8位都输入后, 74HC164并行输出到相应数码管)
```

```
DJNZ R0, CLY; R0先减1, 若不等于0, 则跳转
MOV R0, 21H; 相当于出栈, 还原R0原来的值
RET; 返回主程序
```

TAB:

```
DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90; 共阴极发光二极管0到9的段选码
END
```

4.6流程图:



五、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？★

每次步进角度为15度。用A代表 往IN1口送数据1，用B代表往IN2口送数据1，(!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？

每次步进角度为7.5度。A、B含义同题1，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？★

步进电机的转动速度的控制是通过改变时钟的周期，即改变换相的频率来实现的；它转过的角度由输入的脉冲个数决定。当然，由于各种物理因素（包括摩擦、机械惯性、响应时间等），步进电机的最高转速有限制。

4. 如何改变步进电机的转向？★

改变脉冲顺序

5. 步进电机有那些规格参数，如何根据需要进行选择型号？

步进电机的主要参数有电机的最大工作电压、最小启动电压、最大允许功耗及工作频率等。

六、遇到的问题

步进电机无法顺利转动一周，有时候只是在小范围来回抖动，发现是代码编写的问题。

计算机控制与应用实验

实验四 实验报告

计算机科学与技术 八班

李天祥 53160815

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R（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。

实验代码：

```
ORG 0000H
```

```
LJMP START
```

```
ORG 000BH
```

```
LJMP EINT0
```

```
ORG 0040H
```

```
START:
```

```
OE1 EQU P0.7 ;存储器x方向使能端口
```

```
OE2 EQU P0.4 ;存储器y方向使能端口
```

```
SER1 EQU P0.0 ;移位寄存器串行行输入口Dx
```

```
SER2 EQU P0.3 ;移位寄存器串行列输入口Dy
```

```
SRCK1 EQU P0.1 ;移位寄存器x方向时钟输入口
```

```
SRCK2 EQU P0.5 ;移位寄存器y方向时钟输入口
```

```
RCK1 EQU P0.2 ;存储器x方向时钟输入口
```

```
RCK2 EQU P0.6 ;存储器y方向时钟输入口
```

```
MOV DPTR, #TABLE1 ;表首地址
```

```
MOV TMOD, #01H ;定时器控制字，选择定时器工作方式为定时
```

```
MOV IE, #82H ;中断允许屏蔽控制字，中断设置，定时器0允许中断
```

```
L0:
```

```
MOV R4, #0 ;偏移
```

L2:

MOV R3, #01FH ;时间控制，一个屏幕的时间<40ms

S0:

MOV R0, #15 ;行控制

S00:

CLR SRCK1 ;行清零

CLR SER1 ;行数据清零

SETB SRCK1 ;上升沿读入数据

DJNZ R0, S00 ;自减一，循环

S1:

MOV R0, #0

CLR SRCK1 ;从第一开始选中

SETB SER1

SETB SRCK1

S2:

MOV A, R0 ;计数器

ADD A, R4

MOVC A, @A+DPTR ;寻址，将R0, R4控制的地址取出

MOV R1, #8 ;前8位，循环控制

S3:

CLR SRCK2

RRC A ;向右移

MOV SER2, C ;溢出一位给C

SETB SRCK2 ;上升沿输出数据

DJNZ R1, S3 ;R1! =0, 则跳转S3

INC R0

;重复第二组8位

MOV A, R0

ADD A, R4

MOVC A, @A+DPTR

MOV R1, #8

S4:

CLR SRCK2

RRC A

MOV SER2, C

SETB SRCK2

DJNZ R1, S4

INC R0

S5:

CLR RCK1

SETB RCK1

CLR RCK2

SETB RCK2

CLR OE1 ;使能段有效

CLR OE2

LCALL TIME ;长跳转

SETB OE1

SETB OE2

CLR SRCK1;下一行

CLR SER1

SETB SRCK1

CJNE R0, #32, S2 ;不相等跳转，一个屏幕的32个8

DJNZ R3, S0 ;不等于零跳转窗口大小固定

MOV A, R4

ADD A, #2

MOV R4, A

CJNE R4, #128, L2 ;四个字， 4×32 ，32个8位是一个字，结束之后，循环回到开始。

LJMP L0

TIME: ;延迟实现

MOV TH0, #0FFH

MOV TL0, #3CH ;0.2ms

SETB TR0

MOV R7, #0H

T11:

CJNE R7, #1H, T11

RET

;定时器0中断处理程序

EINT0:

MOV R7, #1H

RETI

TABLE1 :

DB 20H, 00H, 24H, 00H, 24H, 0FCH, 24H, 88H, 24H, 88H, 24H, 88H, 0FCH, 88H;

DB 24H, 88H, 24H, 88H, 24H, 88H, 24H, 0FCH, 24H, 00H, 20H, 00H, 00H, 00H;"?", 0

DB 10H, 40H, 11H, 80H, 16H, 00H, 0FFH, 0FCH, 12H, 00H, 11H, 00H, 00H, 20H;

DB 10H, 0C0H, 13H, 00H, 0FFH, 0FCH, 13H, 00H, 10H, 0C0H, 10H, 20H, 00H, 00H;"?", 1

DB 08H, 04H, 08H, 04H, 08H, 08H, 08H, 10H, 08H, 60H, 0BH, 80H, 0FCH, 00H;

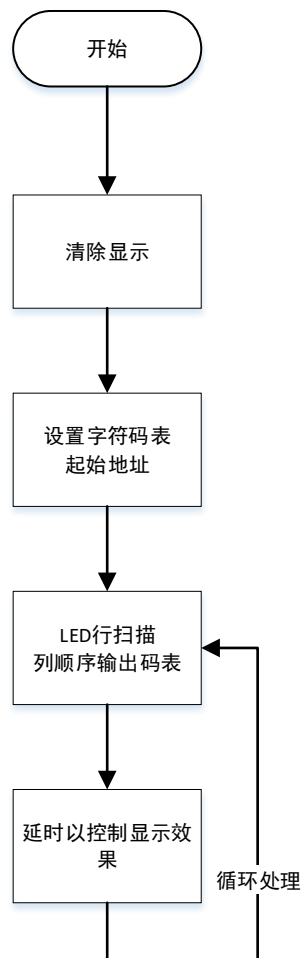
DB 0BH, 80H, 08H, 60H, 08H, 10H, 08H, 08H, 08H, 04H, 08H, 04H, 00H, 00H;"?", 2

DB 1CH, 40H, 10H, 40H, 92H, 40H, 52H, 40H, 12H, 44H, 92H, 44H, 52H, 0FCH;

DB 12H, 0C0H, 33H, 40H, 52H, 40H, 90H, 40H, 10H, 40H, 1CH, 40H, 00H, 00H;"?", 3

END

流程图:



六、思考题

1. 如何使用软件调整和控制LED点阵的亮度

调整扫描频率

2. 如何尽量避免显示过程中的闪烁

提高扫描频率，根据人眼的停留效应，一般刷新频率提高到24HZ以上时基本可以消除闪烁。

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）

硬件方面可以通过添加新的LED发光管，以及74HC595器件实现；软件方面，将控制行扫描的16为数字0FFFEH修改为64位数0FFFF FFFF FFFF FFFEh.