

单片机第三、四次实验报告

53160813

刘国晟

一.实验三

1.1.实验原理

1.1.1 步进电机旋转方向

本实验采用双四拍工作模式，即分别使能 CE1 与 CE2 接口后，将 IN1 与 IN2 按照 01->11->10->00->01 循环输入电信号，即可使步进电机逆时针旋转，若将此循环按照逆序输入电信号，则可使步进电机顺时针旋转。

1.1.2 步进电机旋转速度及定时器中断

对于较快的转速而言，我们可使用定时器中断来控制旋转周期，详细公式为：

$$(2^{\text{定时器最大位数}} - s) \times 12 / \text{CPU 晶振频率} = t$$

其中本实验中晶振频率为 12MHz，定时器所处工作模式最大位数为 16，预定旋转速度为 60 转/分，步进电机转动一周需要 24 步。则可计算出定时器预设值 s 约为 23869，即 0x5D3D。

对于较慢的转速而言，我们可以采用较快转速时定时器中断的预设值，并按照两者旋转周期的倍数关系在程序中设置相应的周期延时。以本次实验为例，较慢转速为 10 转/分，则可令步进电机每 6 个中断周期接收一次脉冲信号以实现目的。

1.1.3 数码管显示

本实验将 3 个数码管串行方式连接，在初始设置参数时使能 P4.4 与 P4.5，并使个位数码管在每次步进电机接收脉冲信号时计数值加一，其中个位数码管由 9 至 0 时使十位数码管加一，十位数码管由 9 至 0 时使百位数码管加一，总计数值为 999 时收到脉冲后将 3 数码管同时清零并重新开始计数。

1.2.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

1.3.程序流程图

图 1.初始化设置流程图

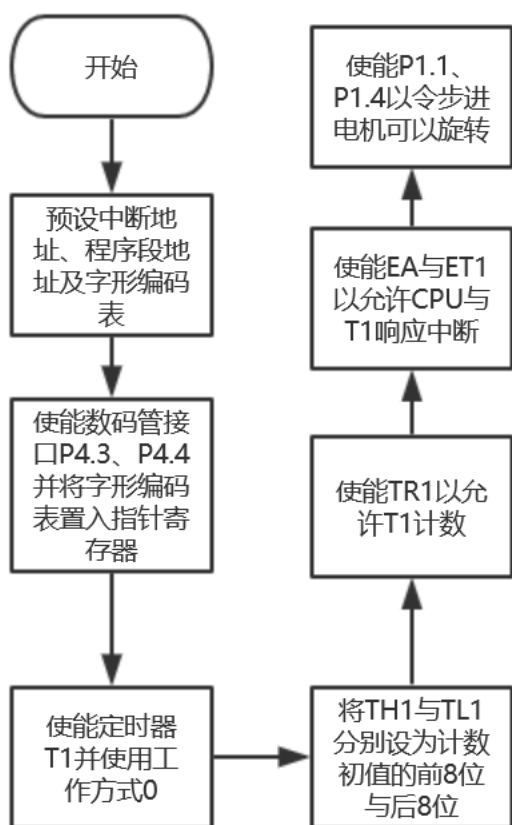
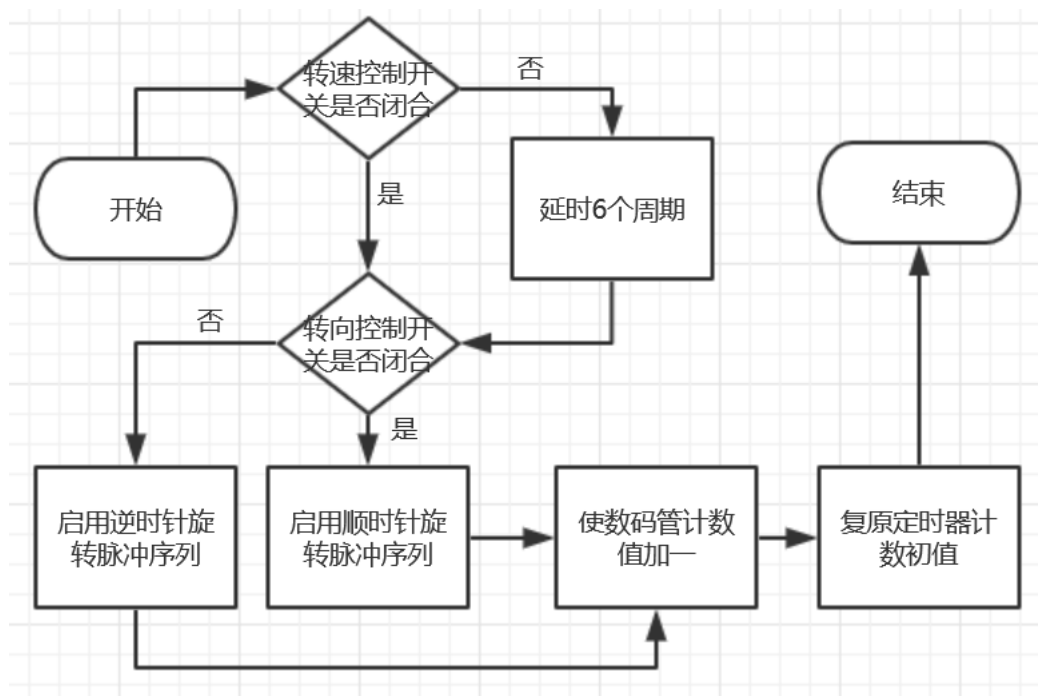


图 2.中断程序流程图



进行电气检查后结果如下：

1.4.思考题及解答

(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：步进角度为 15 度；程序中可使 IN2 空置，IN1 脉冲序列循环设为 1->1->0->0->1 循环。

(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：步进角度为 7.5 度；程序中使 IN1 与 IN2 以如下脉冲序列循环运作(若仅一电平则 IN2 空置)：1->11->1->10->0->00->0->01->1

(3)步进电机的转速取决于哪些因素？有没有上、下限？

答：取决于定时器中断预设初值、延时周期长度、步进电机硬件承载能力等因素；其上限应为步进电机硬件所能承受的最大限度，下限转速为 0。

(4)如何改变步进电机的转向？

答：将正向脉冲序列倒置后输入即可。

(5)步进电机有哪些规格参数，如何根据需要进行选择型号？

答：静扭矩、转速、安装规格、定位精度(振动情况)；转速要求高时，应选相电流较大、电感较小的电机，以增加功率输入。且在选择驱动器时采用较高供电电压。负载大时，需采用大力矩电机。

(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：64K 字节程序存储器空间 (0—0FFFFH)、64K 外部数据存储器空间 (0—0FFFFH)、256 字节内部 RAM 空间 (0—0FFH)、256 位寻址空间 (0—0FFH)、工作寄存器区；

(7)说明 MOVC 指令的使用方法。

答：具体格式如下：MOVC A, @A+PC MOVC A, @A+DPTR；上述两条指令通常作为查表指令来使用，以 DPTR 或 PC 作为基址寄存器，A 的内容和两者之一相加作为程序存储器的地址，此地址内容送到 A。这种查表指令比较方便使用，表格可以存放到任意地址，可在多处使用，但表格的大小不能超过 256 字节。

(8)MCS51 的指令时序是什么样的，哪类指令的执行时间较长？

答：可分为以下几种：单字节指令单机器周期、单字节指令双机器周期、双字节指令单机器周期、双字节指令双机器周期、三字节指令双机器周期、单字节指令四机器周期；单字节乘除指令执行时间较长

(9)在本实验环境下，能否控制显示数码的亮度？如何实现？

答：可以；调整数码管显示程序段中的延时长度即可控制数码管亮度。

二.实验四

2.1.实验原理

2.1.1 硬件接口部分

本实验需用到 74HC595 器件，根据幻灯片中原理图所示，其中 DX/Y 接口接 SER 口，即数据输入引脚；ENX/Y 接使能引脚(接 GND 口)，电平为低时使能 74HC595；CKX/Y 接 SCK 口，即移位寄存器时钟引脚，上升沿时，移位寄存器中的 bit 数据整体后移，并接受新的 bit(从 SER 口输入)；CKX/YL 接 RCK 口，即存储寄存器时钟输入引脚。上升沿时，数据从移位寄存器转存带存储寄存器。

2.1.2.LED 显示部分

获得 LED 显示字形的预设值后，我们可以将其置入 DPTR 指针寄存器中，再通过循环读取其中数据以实现字段在 LED 灯上的循环显示；由于字形数据生成时采用的是逐行扫描法，所以设置 X 轴显示为内层循环，Y 轴显示为外层循环，依次顺序读入数据。

2.2.实验内容

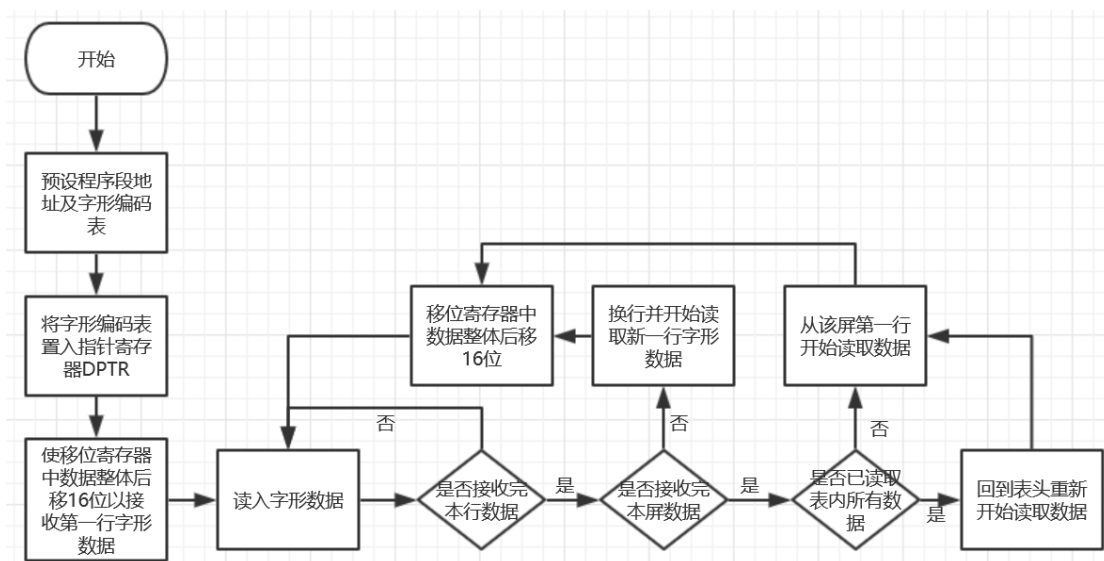
了解 16*16 点阵电路的原理，编写汇编语言程序。

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）。

2.3.程序流程图

图 3.LED 读取流程图



2.4.思考题及解答

(1)如何使用软件调整和控制 LED 点阵的亮度。

答：通过控制延时时间；延时越长则越暗，延时越短则越亮。

(2)如何尽量避免显示过程中的闪烁。

答：使延时尽量保持在一个可清晰看到 LED 显示字段的区间内并尽量增加延时长度。

(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：改变程序段中行数与列数的设定初值并多次实验以保证 LED 亮度与闪烁程度都处于一个可接受的最佳值。

三.实验中遇到的问题

3.1.步进电机中断程序段冗余延时

根据程序中中断程序段的长短，程序所执行的时间与预定时间也有一定的差别。本次实验三中由于中断程序段过长，程序执行的时间也存在着一一定的误差，下面以数码管计数值与相应时间的关系来探究误差大小(肉眼观测，秒表计时，数据本身存在一定误差)。

表 1.慢速状态下计数次数与经过时间长度数据

计数次数	100	250	500	750	1000	1600	1800	2000
时间(s)	25.13	62.57	125.62	188.14	251.57	401.47	450.64	501.53

慢速状态下转速为 10 转/分，即 4 次(脉冲)/秒，则通过以上数据可得误差平均值约为 0.9103 毫秒/次(脉冲)，可见过长的中断程序会带来一定的时间误差。

3.2.LED 点阵字形生成

由于一开始实验时没有注意到使用正向字形逐列生成时，字形数据生成软件会从点阵中央(第一列第九行)开始向上至顶，而后从第二列自底向上生成字形，导致实验进度拖延了很长时间。随后使用软件逆向逐行扫描左右相反的字形解决了该问题。

正常生成字形后发现每显示一遍字段后会产生一段显示乱码，而后循环往复。在老师的提醒下我在字段初值预设中将原有的第一字的字形编码重复添加到了第三字后进行调试解决了该问题。(例如：原有字段为“刘国晟”输入产生乱码，重设为“刘国晟刘”解决该问题)

四.代码附录

4.1.实验三代码

```
P4 DATA 0C0H
```

```
P4SW DATA 0BBH
```

```
ORG 0000H
```

```
LJMP START
```

```
ORG 001BH
```

```
LJMP TINT
```

```
ORG 0040H
```

```
TABLE:DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H;
```

```
START:
```

```
MOV P4SW,#30H
```

```
MOV DPTR,#TABLE
```

```
MOV R1,#00H
```

```
MOV R2,#00H
```

```
MOV R3, #00H
```

MOV R0, #00H

MOV R4, #00H

MOV TMOD,#10H

MOV TH1,#5DH

MOV TL1,#3DH

SETB ET1

SETB EA

SETB TR1

SETB P1.1

SETB P1.4

LOOP : AJMP LOOP

TINT:

MOV C,P3.6

JNC TT0

INC R4

CJNE R4,#6,RETURN

MOV R4,#00H

TT0:

MOV C,P3.7

JNC TT1

ACALL ROTATE1

AJMP RETURN

TT1:

ACALL ROTATE2

RETURN:

MOV TH1,#5DH

MOV TL1,#3DH

ROTATE1:

CLR P1.1

CLR P1.4

CJNE R0,#0,A10

AJMP L10

A10:

CJNE R0,#1,A11

AJMP L11

A11:

CJNE R0,#2,A12

AJMP L12

A12:

AJMP L13

L10:

CLR P3.2 ;

SETB P1.0

INC R0

AJMP EXIT1

L11:

SETB P3.2

SETB P1.0

INC R0

AJMP EXIT1

L12:

SETB P3.2

CLR P1.0

INC R0

AJMP EXIT1

L13:

CLR P3.2

CLR P1.0

MOV R0,#00H

EXIT1:

SETB P1.1

SETB P1.4

ACALL SHOWSTEPS

RET

ROTATE2:

CJNE R0,#0,A20

AJMP L20

A20:

CJNE R0,#1,A21

AJMP L21

A21:

CJNE R0,#2,A22

AJMP L22

A22:

AJMP L23

L20:

CLR P3.2

SETB P1.0

INC R0

AJMP EXIT2

L21:

CLR P3.2

CLR P1.0

INC R0

AJMP EXIT2

L22:

SETB P3.2

CLR P1.0

INC R0

AJMP EXIT2

L23:

SETB P3.2

SETB P1.0

MOV R0,#00H

EXIT2:

ACALL SHOWSTEPS

RET

;???????

SHOWSTEPS:

CJNE R1,#9H,LL1

CJNE R2,#9H,LL2

CJNE R3,#9H,LL3

MOV R1,#0

MOV R2,#0

MOV R3,#0

AJMP EXIT3

LL1:

INC R1

AJMP EXIT3

LL2:

MOV R1,#0

INC R2

AJMP EXIT3

LL3:

MOV R1,#0

MOV R2,#0

INC R3

EXIT3:

MOV A,R1

MOVC A,@A+DPTR

ACALL SHOW

MOV A,R2

MOVC A,@A+DPTR

ACALL SHOW

MOV A,R3

MOVC A,@A+DPTR

ACALL SHOW

RET

SHOW:

MOV R7,#8

LL4:

RLC A;move to the left

MOV P4.5,C

CLR P4.4

```
SETB P4.4
DJNZ R7,LL4
RET
```

```
END
```

4.2 实验四代码

```
ORG 0000H
```

```
    LJMPL    START
```

```
ORG 0040H
```

```
    TAB:
```

```
DB 00H,00H,00H,00H,0FFH,7FH,00H,80H,00H,40H,0F8H,0FH,00H,00H,08H,00H;
```

```
DB 08H,00H,78H,60H,88H,19H,0EH,06H,89H,09H,48H,10H,28H,20H,08H,40H;"刘",0
```

```
DB 00H,00H,00H,00H,0FEH,0FFH,02H,40H,12H,48H,92H,4CH,92H,4AH,92H,48H;
```

```
DB 0F2H,4FH,92H,48H,92H,48H,92H,48H,12H,48H,02H,40H,0FEH,0FFH,00H,00H;"国",1
```

```
DB 00H,00H,00H,00H,40H,0F0H,60H,43H,5FH,2CH,55H,10H,55H,2CH,0F5H,43H;
```

```
DB 55H,40H,55H,9EH,55H,0A2H,55H,12H,5FH,02H,0C0H,1FH,00H,60H,00H,80H;"晟",2
```

```
DB 00H,00H,00H,00H,0FFH,7FH,00H,80H,00H,40H,0F8H,0FH,00H,00H,08H,00H;
```

```
DB 08H,00H,78H,60H,88H,19H,0EH,06H,89H,09H,48H,10H,28H,20H,08H,40H;"刘",0
```

```
START:
```

```
    DX EQU P0.0    ;简介：DX/Y->数据输入引脚
```

CKX EQU P0.1 ; ENX/Y->使能引脚(接 GND)，电平为低时使能
74HC595

CKXL EQU P0.2 ; CKX/Y->接 SCK，移位寄存器时钟引脚，上升沿时，
移位寄存器中的 bit 数据整体后移，并接受新的 bit（从 SER 输入）。

DY EQU P0.3 ; CKX/YL->接 RCK，存储寄存器时钟输入引脚。上升沿
时，数据从移位寄存器转存带存储寄存器。

ENY EQU P0.4

CKY EQU P0.5

CKYL EQU P0.6

ENX EQU P0.7

S0:

MOV R3,#30H

MOV DPTR,#TAB

S1:

MOV R0,#32

CLR ENX

CLR ENY

CLR CKYL

MOV R2,#16 ;由于一次读入 16 位数据，所以需要使 74HC595 移位寄存器
中数据整体后移 16 次。

SETB C

ROW1:

CLR CKY

MOV DY,C

SETB CKY

DJNZ R2,ROW1

CLR C

CLR CKY

MOV DY,C

SETB CKY

SETB CKYL

LOOP0:

MOV A,#32

CLR C

SUBB A,R0

MOVC A,@A+DPTR

MOV R2,#8

CLR CKXL

SETB ENX

LOOP1:

RRC A

CLR CKX

MOV DX,C

SETB CKX

DJNZ R2,LOOP1

DEC R0

MOV A,#32

CLR C

SUBB A,R0

MOVC A,@A+DPTR

MOV R2,#8

LOOP2:

RRC A

CLR CKX

MOV DX,C

SETB CKX

DJNZ R2,LOOP2

SETB CKXL

CLR ENX

DEALY:

MOV R5,#80

TIME:

MOV R6,#45H

DJNZ R6,\$

DJNZ R5,TIME ;总延时=80x45H=5520(个)脉冲

COPLAN:

SETB C

CLR CKYL

CLR CKY

MOV DY,C

SETB CKY

SETB CKYL

DJNZ R0,LOOP0

INC DPTR

INC DPTR ;2x30H=96，即 TAB 表中总数据条数

DJNZ R3,S1 ;若未读完表则继续读表

JMP S0 ;若已读完表则重新读表

END