|  |  |
| --- | --- |
| 001 | jsjlogo |
|  |  |

**合肥工业大学**

**计算机与信息学院**

**实验报告**

**课 程： 单片机原理与应用**

**专业班级： 物联网工程19-2班**

**学 号： 2019217819**

**姓 名： 付炎平**

**老 师： 欧 阳 一 鸣**

**2022年5月22日**

目录

**[1 实验一 跑马灯实验 3](#_Toc17634)**

[1.1 实验目的与要求 3](#_Toc25932)

[1.2 实验设备 3](#_Toc23600)

[1.3 实验内容 3](#_Toc13774)

[1.4 实验原理图 3](#_Toc23046)

[1.5 实验步骤 4](#_Toc22105)

[1.6 参考程序 4](#_Toc29148)

[1.7 实验结果 5](#_Toc19343)

**[2 实验二 74HC138译码器实验 6](#_Toc24753)**

[2.1 实验目的与要求 6](#_Toc1487)

[2.2 实验设备 6](#_Toc18262)

[2.3 实验内容 7](#_Toc5324)

[2.4 实验原理图 7](#_Toc18487)

[2.5 实验步骤 7](#_Toc9409)

[2.6 演示程序 7](#_Toc3126)

[2.7 实验结果 8](#_Toc31309)

[2.8 实验扩展及思考 8](#_Toc12182)

**[3 实验三 PWM电压转换实验（选做） 9](#_Toc2629)**

[3.1 实验目的 9](#_Toc24328)

[3.2 实验设备 9](#_Toc29948)

[3.3 实验内容 9](#_Toc28248)

[3.4 实验原理图 10](#_Toc10442)

[3.5 实验步骤 10](#_Toc15695)

[3.6 演示程序 10](#_Toc19937)

[3.7 实验结果 12](#_Toc9403)

[3.8 实验扩展及思考 12](#_Toc28076)

**[4 实验四 8255控制交通灯实验 13](#_Toc13514)**

[4.1 实验目的与要求 13](#_Toc6736)

[4.2 实验设备 13](#_Toc5904)

[4.3 实验内容 13](#_Toc16134)

[4.4 实验原理图 13](#_Toc14837)

[4.5 实验步骤 14](#_Toc18529)

[4.6 演示程序 14](#_Toc22625)

[4.7 实验结果 15](#_Toc3016)

[4.8 实验扩展及思考 16](#_Toc18387)

[4.9 实验总结 16](#_Toc20385)

**[5 实验五 键盘、数码块实验（选做） 17](#_Toc2134)**

[5.1 实验目的与要求 17](#_Toc8713)

[5.2 实验设备 17](#_Toc11499)

[5.3 实验内容 17](#_Toc23047)

[5.4 实验原理图 17](#_Toc10329)

# 

# 实验一 跑马灯实验

## 实验目的与要求

熟悉星研集成环境软件或熟悉Keil C51集成环境软件的使用方法。

熟悉MCS51汇编指令，能自己编写简单的程序，控制硬件。

## 实验设备

SUN系列实验仪一套、PC机一台

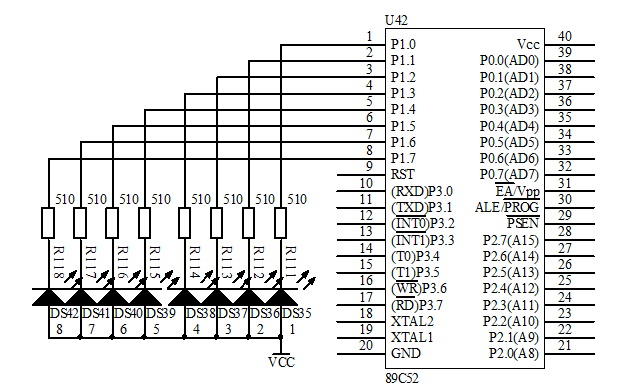
## 实验内容

熟悉星研集成环境软件或熟悉Keil C51集成环境软件的安装和使用方法。

照接线图编写程序：使用P1口控制F5区的8个指示灯，循环点亮，瞬间只有一个灯亮。

观察实验结果，验证程序是否正确。

## 实验原理图



## 实验步骤

连线说明：注意：JP51对应于P1.0，P1.1，……P1.7共用到了8个引脚

|  |  |  |
| --- | --- | --- |
| A3区：JP51 | —— | F5区：JP65 |

编写程序或将参考程序中3个空白的地方填完整。

实验结果：通过F5区的LED指示灯（8个指示灯轮流点亮），观察实验的输出结果是否正确。

## 参考程序

|  |  |  |  |
| --- | --- | --- | --- |
|  | ORG | 0000H |  |
|  | **LJMP** | START | ;在0000H处放一条跳转指令，跳到START处 |
|  |  |  |  |
|  | ORG | 0100H |  |
| START: | MOV | SP,#60H |  |
|  | MOV | A,#0FFH |  |
|  | CLR | C |  |
| START1: | **RL** | A | ;连同符号位C进行循环左移 |
|  | MOV | P1,A |  |
|  | ACALL | Delay |  |
|  | SJMP | START1 |  |
| Delay: | MOV | R5,#2 | ;延时 |
| Delay1: | MOV | R6,#0 |  |
| Delay2: | MOV | R7,#0 |  |
|  | DJNZ | R7,$ |  |
|  | **DJNZ** | R6,Delay2 | ;将R6中的数减1，不为0就跳转 |
|  | DJNZ | R5,Delay1 |  |
|  | RET |  |  |
|  |  |  |  |
|  | END |  |  |

如果读者使用星研集成环境软件，请考虑以下问题？

1、运行程序前，打开变量窗;

2、使用单步进入命令，运行到第六行后，运行过程中变量窗有何变化？将鼠标停留在A、SP上一秒后，出现什么？,它与变量窗使用场合的区别？

3、第九行是调用延时子程序，如何进入延时子程序（使用单步进入命令）？如何将延时子程序一下子运行完毕（使用单步命令；也可以将光标移到下一行，使用运行到光标处命令；）？单步进入命令与单步命令有何区别？

4、运行几次后，在第十行设置一个断点，使用全速断点命令运行几次，观察运行结果，它与运行到光标处命令有何区别？

5、Delay是一个延时子程序，改变延时常数，使用全速运行命令，显示发生了什么变化？

6、观察寄存器，有哪几种方法？

1）在工作区窗的通用寄存器标签视中；

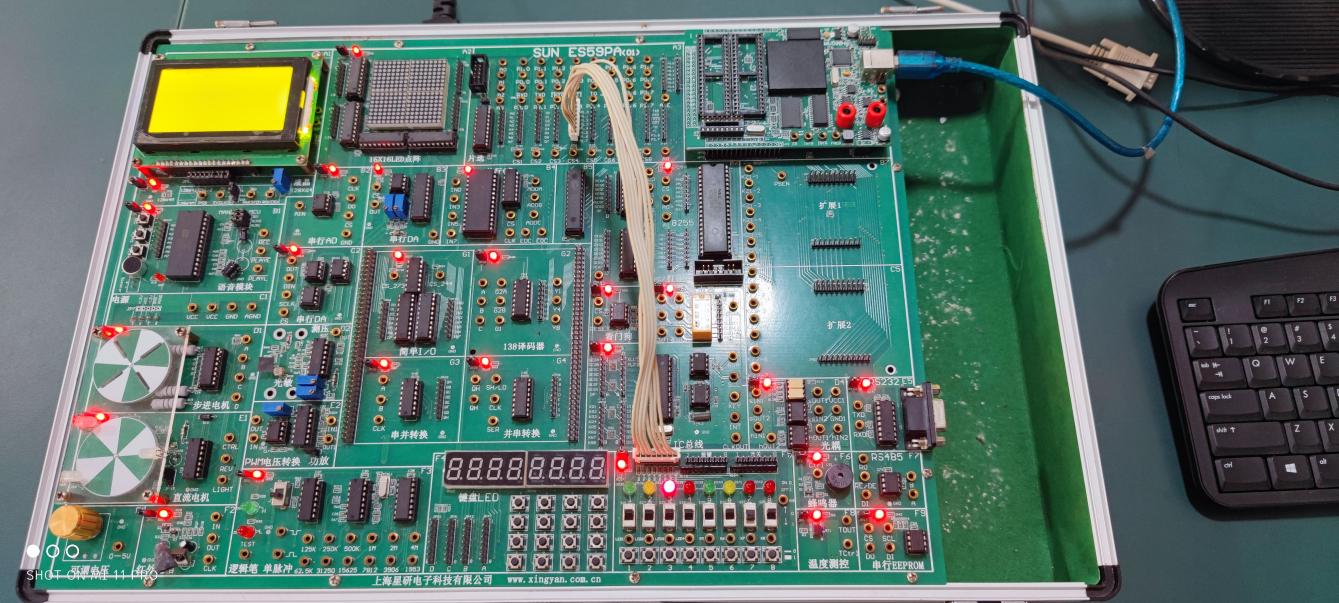
2）变量窗

3）鼠标停留在寄存器上

4）观察窗

5）寄存器窗

## 实验结果



# 实验二 74HC138译码器实验

## 实验目的与要求

掌握74HC138译码器的工作原理，熟悉74HC138译码器的具体运用连接方法，了解74HC138是如何译码的。

认真预习本节实验内容，尝试自行编写程序，填写实验报告

## 实验设备

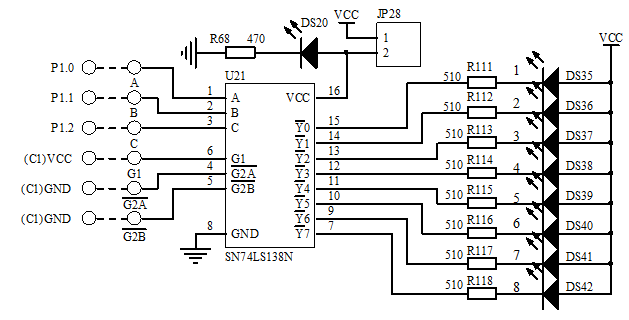
SUN系列实验仪一套、PC机一台

## 实验内容

编写程序：使用单片机的P1.0、P1.1、P1.2控制74HC138的数据输入端，通过译码产生8选1个选通信号，轮流点亮8个LED指示灯。

运行程序，验证译码的正确性。

## 实验原理图



## 实验步骤

连线说明：注意：此处只用到了P1.0，P1.1， P1.2共3个引脚，这和上个实验不同的。

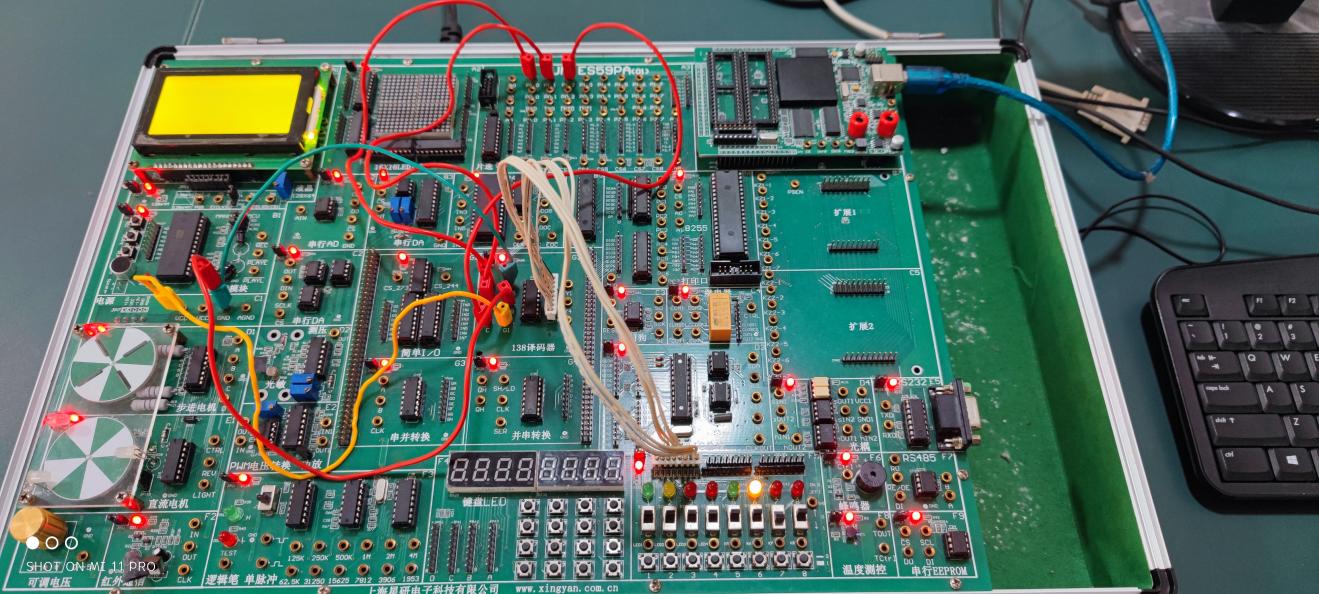
|  |  |  |
| --- | --- | --- |
| G2区：A、B、C | —— | A3区：**P1.0、P1.1、P1.2** |
| G2区：G1、G2A、G2B | —— | C1区：VCC、GND、GND |
| G2区：JP35 | —— | F5区：JP65（LED指示灯） |

调试程序，查看运行结果是否正确。

## 演示程序

|  |  |  |  |
| --- | --- | --- | --- |
| 138译码器实验（跑马灯），P1.0--A, P1.1---B, P1.2--C, /G2B--GND, /G2A--GND | | | |
|  | ORG | 0000H |  |
|  | LJMP | START |  |
|  |  |  |  |
|  | ORG | 0100H |  |
| START: | MOV | SP,#60H |  |
|  | **CLR** | A | ;使累加器A清0 |
| START1: | MOV | P1,A | ;初值，第一次0位LED亮 |
|  | **LCALL** | DLTIME | ;调用延迟子程序DLTIME |
|  | INC | A |  |
|  | CLR | **ACC.3** | ;使ACC的第3位为0 |
|  | SJMP | START1 |  |
| DLTIME: | MOV | R5,#20 |  |
| DLTIME1: | MOV | R6,#100 |  |
| DLTIME2: | MOV | R7,#100 |  |
|  | DJNZ | R7,$ |  |
|  | DJNZ | R6,DLTIME2 |  |
|  | DJNZ | R5,DLTIME1 |  |
|  | RET |  |  |
|  |  |  |  |
|  | END |  |  |

## 实验结果



## 实验扩展及思考

在单片机系统中，74HC138通常用来产生片选信号，请读者考虑一下，应如何处理？

答：取三根底地址线，接到38译码器的输入端，译码产生，8个使能控制信号。

# 实验三 PWM电压转换实验（选做）

## 实验目的

了解PWM电压转换原理

掌握单片机控制的PWM电压转换

## 实验设备

SUN系列实验仪一套、PC机一台

## 实验内容

PWM电压转换原理：

(1)将一定频率的输入信号转换为直流电；

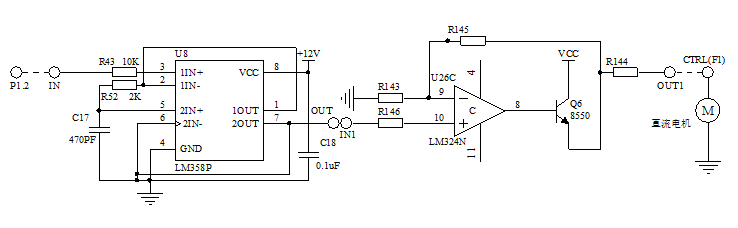
(2)通过调节输入信号占空比调节输出的直流电电压，输出电压随着占空比增大而减小

实验过程

(1)输入15kHZ左右的方波，经LM358进行PWM电压转换，输出直流电，驱动直流电机

(2)通过按键调整占空比来改变PWM输出电压，直流电机的转速会随之变化

## 实验原理图



## 实验步骤

连线说明：

|  |  |  |
| --- | --- | --- |
| E2区：IN | —— | A3区：P1.2，方波输入 |
| E2区：OUT | —— | E2区：IN1 |
| E2区：OUT1 | —— | E1区：CTRL，直流电机电源输入 |
| A3区：JP51 | —— | F5区：JP74 |

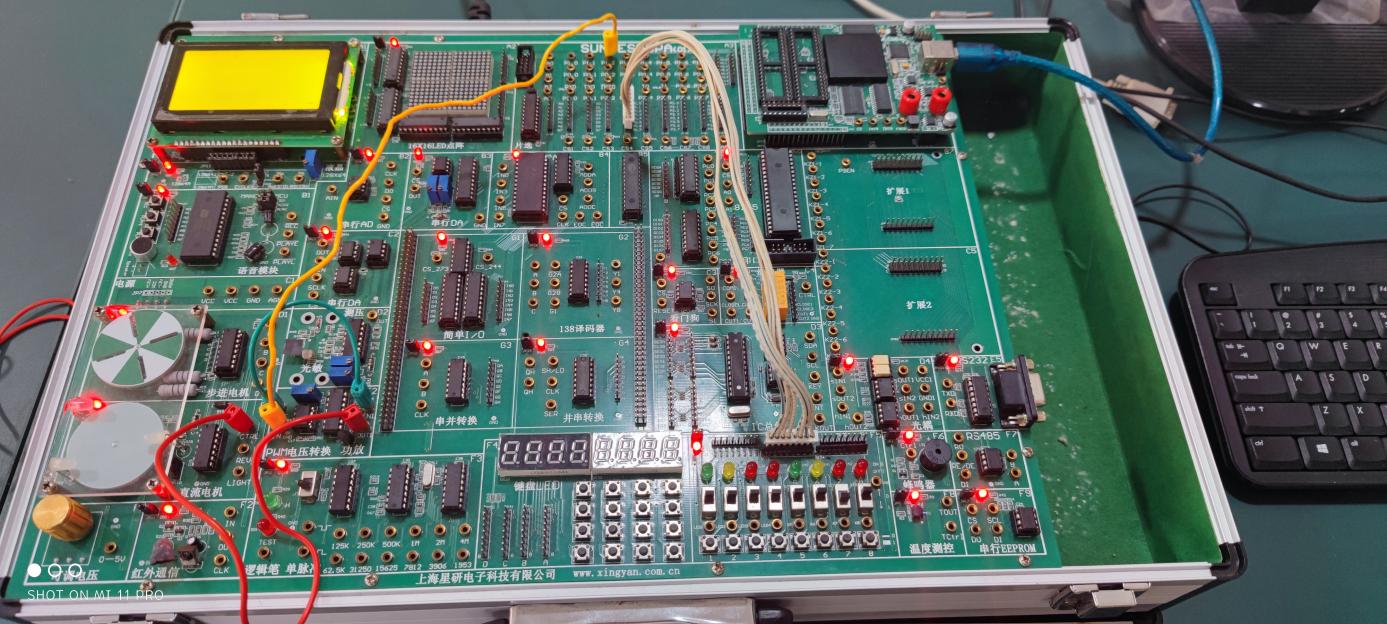
通过F5区的1、2键调整占空比来改变PWM输出电压，直流电机的转速会随之变化：

1号键减少占空比；2号键增加占空比

## 演示程序

|  |  |  |  |
| --- | --- | --- | --- |
| IN | **BIT** | P1.2 | ;PWM方波输入，定义IN代表P1.2 |
| PWM\_LOW | DATA | 30H | ;低电平时间 |
| PWM\_HIGH | DATA | 31H ;高电平时间,控制频率在15kHZ左右 | |
| periods | EQU | 0E0H | ;周期64us |
|  |  |  |  |
|  | ORG | 0000H |  |
|  | LJMP | START |  |
|  |  |  |  |
|  | ORG | 000BH |  |
|  | LJMP | iTIMER0 |  |
|  |  |  |  |
|  | ORG | 0100H |  |
| START: | MOV | SP,#60H |  |
|  | MOV | PWM\_LOW,#periods |  |
|  | MOV | PWM\_HIGH,#periods |  |
|  | MOV | TH0,#periods |  |
|  | MOV | TL0,#periods |  |
|  | MOV | TMOD,#02H |  |
|  | **SETB** | EA | ; 使EA置1 |
|  | SETB | ET0 |  |
|  | SETB | TR0 |  |
| START1: | ACALL | ScanKey |  |
|  | JNZ | Key1 |  |
| Key0: | MOV | A,PWM\_HIGH | ; 增加占空比 |
|  | CJNE | A,#0FBH,Key0\_1 |  |
|  | SJMP | START1 ;大于这个值，对定时中断已反应不过来 | |
| Key0\_1: | INC | PWM\_HIGH |  |
|  | **DEC** | PWM\_LOW | ;低电平时间减1，从而增加占空  ;比 |
|  | SJMP | START1 |  |
| Key1: | MOV | A,PWM\_LOW | ;减少占空比 |
|  | CJNE | A,#0FBH,Key1\_1 |  |
|  | SJMP | START1 ;大于这个值，对定时中断已反应不过来 | |
| Key1\_1: | **INC** | PWM\_LOW | ;低电平时间加1，从而减少占空  ;比 |
|  | DEC | PWM\_HIGH |  |
|  | SJMP | START1 |  |
| iTIMER0: | JBC | IN,iTIMER0\_1 |  |
|  | MOV | TL0,PWM\_HIGH |  |
|  | SETB | IN |  |
|  | RETI |  |  |
| iTIMER0\_1: | MOV | TL0,PWM\_LOW |  |
|  | NOP |  |  |
|  | RETI |  |  |
| ScanKey: | JNB | P1.0, ScanKey1 | ;键扫描 |
|  | JB | P1.1, ScanKey |  |
| ScanKey1: | ACALL | Delay20ms | ;消抖动 |
|  | ACALL | Delay20ms |  |
|  | JNB | P1.0, ScanKey2 |  |
|  | JB | P1.1, ScanKey |  |
|  | MOV | A,#1 |  |
|  | SJMP | ScanKey3 |  |
| ScanKey2: | CLR | A |  |
| ScanKey3: | JNB | P1.0,$ |  |
|  | JNB | P1.1,$ |  |
|  | RET |  |  |
| Delay20ms: | MOV | R6,#10 |  |
| Delay1: | MOV | R7,#100 |  |
|  | DJNZ | R7,$ |  |
|  | DJNZ | R6,Delay1 |  |
|  | RET |  |  |
|  |  |  |  |
|  | END |  |  |

## 实验结果



## 实验扩展及思考

改变PWM的输入频率，使用示波器观看LM358的输出，由此加深对PWM电压转换的了解。

# 实验四 8255控制交通灯实验

## 实验目的与要求

了解8255芯片的工作原理，熟悉其初始化编程方法以及输入、输出程序设计技巧。学会使用8255并行接口芯片实现各种控制功能，如本实验（控制交通灯）等。

熟悉8255内部结构和与单片机的接口逻辑，熟悉8255芯片的3种工作方式以及控制字格式。

认真预习本节实验内容，尝试自行编写程序，填写实验报告。

## 实验设备

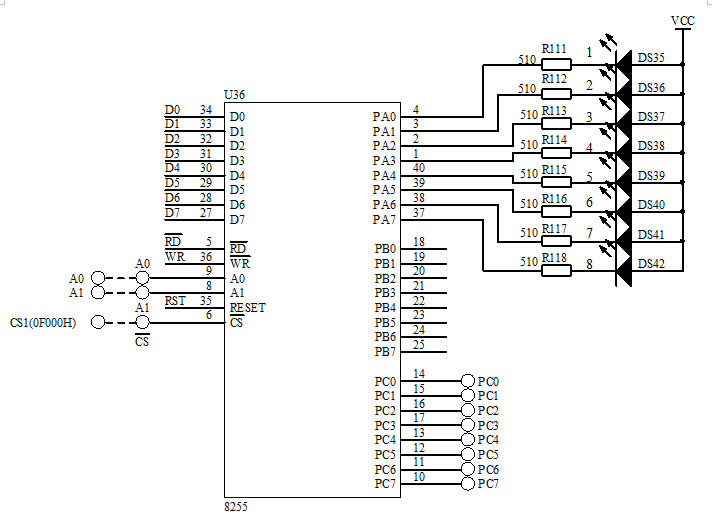
SUN系列实验仪一套、PC机一台

## 实验内容

编写程序：使用8255的PA0..2、PA4..6控制LED指示灯，实现交通灯功能。

连接线路验证8255的功能，熟悉它的使用方法。

## 实验原理图



## 实验步骤

连线说明：

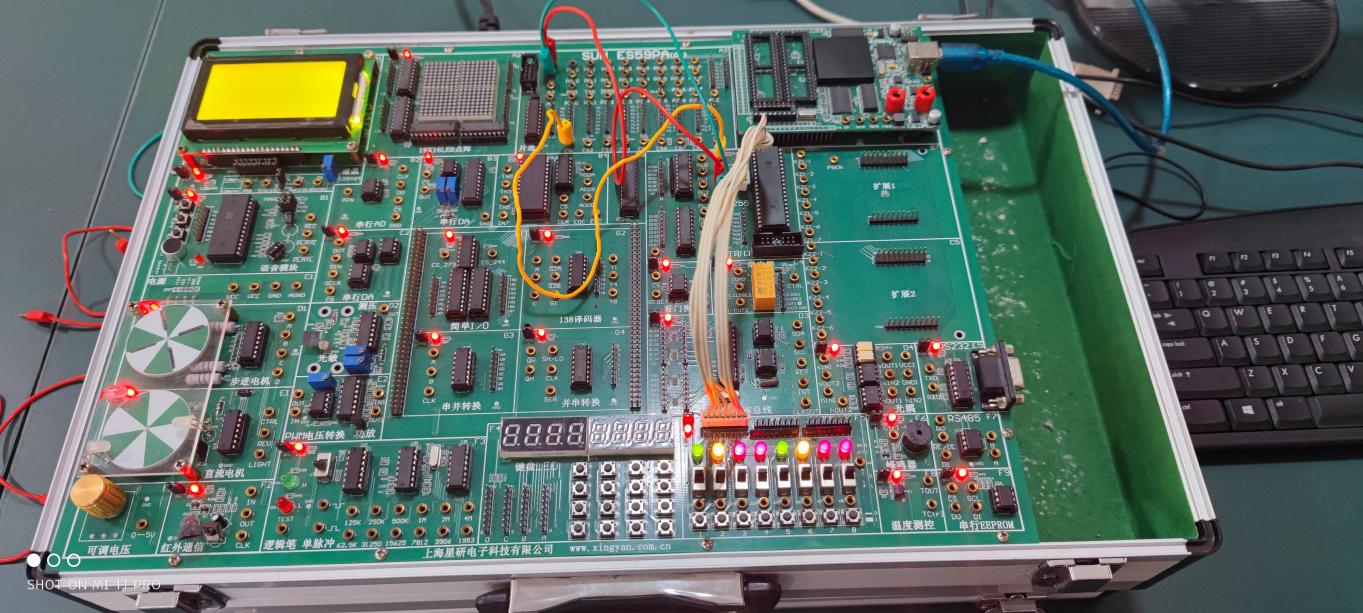
|  |  |  |
| --- | --- | --- |
| B6区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| B6区：JP56（PA口） | —— | F5区：JP65 |

观察实验结果，是否能看到模拟的交通灯控制过程。

## 演示程序

|  |  |  |  |
| --- | --- | --- | --- |
| COM\_ADD | XDATA | 0F003H |  |
| PA\_ADD | XDATA | 0F000H |  |
| PB\_ADD | XDATA | 0F001H |  |
| PC\_ADD | XDATA | 0F002H |  |
|  | ORG | 0000H |  |
|  | LJMP | STAR |  |
|  | ORG | 0100H |  |
| STAR: | MOV | SP,#60H |  |
|  | MOV | DPTR,#COM\_ADD |  |
|  | MOV | A,#80H | ;PA、PB、PC为基本输出模式 |
|  | MOVX | @DPTR,A |  |
|  | MOV | DPTR,#PA\_ADD | ;灯全熄灭 |
|  | MOV | A,#0FFH |  |
|  | **MOVX** | @DPTR,A | ;将累加器A的值送到片外RAM中 |
| START1: | MOV | A,#37H |  |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A | ;东西绿灯，南北红灯 |
|  | **LCALL** | DL5S | ;调用延时5S的子程序 |
|  | MOV | R4,#6 |  |
| START2: | MOV | A,#30H |  |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A | ;东西绿灯闪烁，南北红灯 |
|  | ACALL | DL500ms |  |
|  | MOV | A,#29H |  |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL500ms |  |
|  | DJNZ | R4,START2 |  |
|  | MOV | A,#23H | ;东西黄灯亮，南北红灯 |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL3S |  |
|  | MOV | A,#1EH | ;东西红灯，南北绿灯 |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL5S |  |
|  | MOV | R4,#6 |  |
| START3: | MOV | A,#17H | ;东西红灯，南北绿灯闪烁 |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL500ms |  |
|  | MOV | A,#10H |  |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL500ms |  |
|  | DJNZ | R4,START3 |  |
|  | MOV | A,#0AH | ;东西红灯，南北黄灯亮 |
|  | MOVC | A,@A+PC |  |
|  | MOVX | @DPTR,A |  |
|  | ACALL | DL3S |  |
|  | **SJMP** | START1 | ;跳转到START1处 |
|  | DB | 01111110B | ;东西绿灯，南北红灯 |
|  | DB | 11111110B | ;东西绿灯闪烁，南北红灯 |
|  | DB | 10111110B | ;东西黄灯亮，南北红灯 |
|  | DB | 11011011B | ;东西红灯，南北绿灯 |
|  | DB | 11011111B | ;东西红灯，南北绿灯闪烁 |
|  | DB | 11011101B | ;东西红灯，南北黄灯亮 |
| DL500ms: | MOV | R5,#25 |  |
| DL500ms1: | MOV | R6,#100 |  |
| DL500ms2: | MOV | R7,#100 |  |
|  | DJNZ | R7,$ |  |
|  | **DJNZ** | R6,DL500ms2 | ;将R6减1，不为0就跳转。 |
|  | DJNZ | R5,DL500ms1 |  |
|  | RET |  |  |
| DL3S: | MOV | R4,#6 |  |
| DL3S1: | LCALL | DL500ms |  |
|  | DJNZ | R4,DL5S1 |  |
|  | RET |  |  |
| DL5S: | MOV | R4,#10 |  |
| DL5S1: | LCALL | DL500ms |  |
|  | DJNZ | R4,DL5S1 |  |
|  | RET |  |  |
|  |  |  |  |
|  | END |  |  |

## 实验结果



## 实验扩展及思考

如何对8255的PC口进行位操作？

答:8255控制字D7=1时，D6~DO为口模式控制，D7=O时控制字为PC口的位操作模式，如控制字=00H，PCO复位；控制字=01H，PCO置位;控制字=OEH，PC7复位;控制字=OFH，PC7置位；其中D6~D4没定义，D3~D1从000B~111B分别指定PCO~PC7脚，DO位为相应的电平高低，0为低电平，1为高电平。

## 实验总结

通过本次试验，我了解了8255芯片的工作原理，熟悉其初始化编程方法以及输入、输出程序设计技巧。学会使用8255并行接口芯片实现控制交通灯。熟悉8255内部结构和与8088的接口逻辑，熟悉8255芯片的3种工作方式以及控制字格式。

# 实验五 键盘、数码块实验（选做）

## 实验目的与要求

进一步掌握8255的设计、编程方法。

掌握矩阵键盘的扫描方法

掌握动态扫描数码块的方法

认真预习，做好实验前的准备工作，填写实验报告

## 实验设备

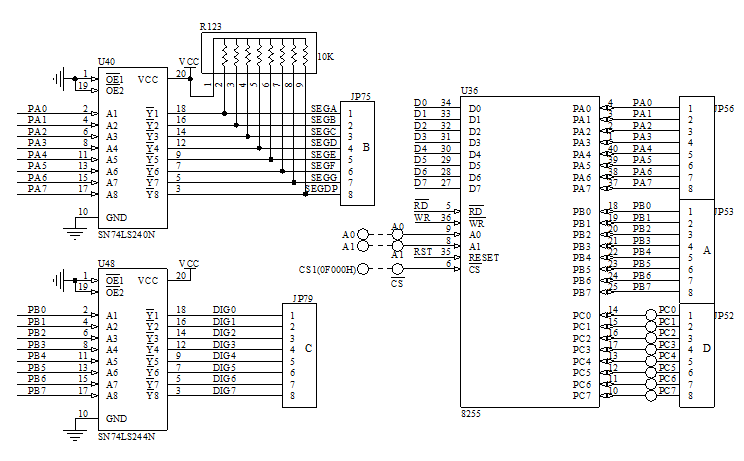
SUN系列实验仪一套、PC机一台

## 实验内容

编写程序：扫描键盘，如有按键，键号显示于数码管。

连接线路，验证8255的功能，熟悉它的使用方法。

## 实验原理图



## 实验步骤

连线说明：

|  |  |  |
| --- | --- | --- |
| B6区：CS、A0、A1 | —— | A3区：CS1、A0、A1 |
| B6区：JP53(PB口)、JP75(B)、JP79(C)、JP52(PC口) | —— | F4区：A、B、C、D |

运行程序，观察实验结果（任意按下F4区4X4键盘几个键，它上面的8个LED显示器会将按键的编码从左至右依次显示出来），可依此验证对程序的正确性。

## 演示程序

#define u8 unsigned char

xdata u8 COM\_8255 \_at\_ 0xF003;

xdata u8 PA\_8255 \_at\_ 0xF000;

xdata u8 PB\_8255 \_at\_ 0xF001;

xdata u8 PC\_8255 \_at\_ 0xF002;

u8 buffer[8]; //8个字节显示缓冲区

void DL1()

{

u8 i,j;

i = 0x2;

do

{

j = 250;

while(--j)

{;}

}while(--i);

}

code const u8 SegArray[] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e,0xff};

void DIR()

{

u8 i = 0xfe;

u8\* pBuffer = buffer;

while(i != 0xff)

{

PA\_8255 = SegArray[\*pBuffer++]; //段数据->8255 PA口

PB\_8255 = i; //扫描模式->8255 PB口

DL1(); //延迟1ms

i = ((i << 1) | 0x1);

PB\_8255 = 0xff;

}

}

u8 AllKey()

{

PB\_8255 = 0x0; //全"0"->扫描口

return ~PC\_8255 & 0x3; //读键状态, 取低二位

}

u8 keyi()

{

u8 i,j;

while (1)

{

if (AllKey() == 0) //调用判有无闭合键子程序

{

DIR();

DIR(); //调用显示子程序,延迟6ms

continue;

}

DIR();

DIR();

if (AllKey() == 0) //调用判有无闭合键子程序

{

DIR();

continue;

}

i = 0xfe;

j = 0;

while(i != 0xff)

{

PB\_8255 = i;

if ((PC\_8255 & 0x1) == 0)

{ //0行有键闭合

break;

}

else if ((PC\_8255 & 0x2) == 0)

{ //1行有键闭合

j += 8;

break;

}

j++; //列计数器加1

i = ((i<<1) | 1);

}

if (i == 0xff) //完成一次扫描,没有键按下

continue;

do

{

DIR();

}while(AllKey() != 0); //判断释放否

return j; //键号

}

}

void main()

{

u8 i;

COM\_8255 = 0x89; //PA、PB输出，PC输入

for (i = 0; i < 8; i++)

buffer[i] = 0x10; //0x10-消隐

DIR();

while(1)

{

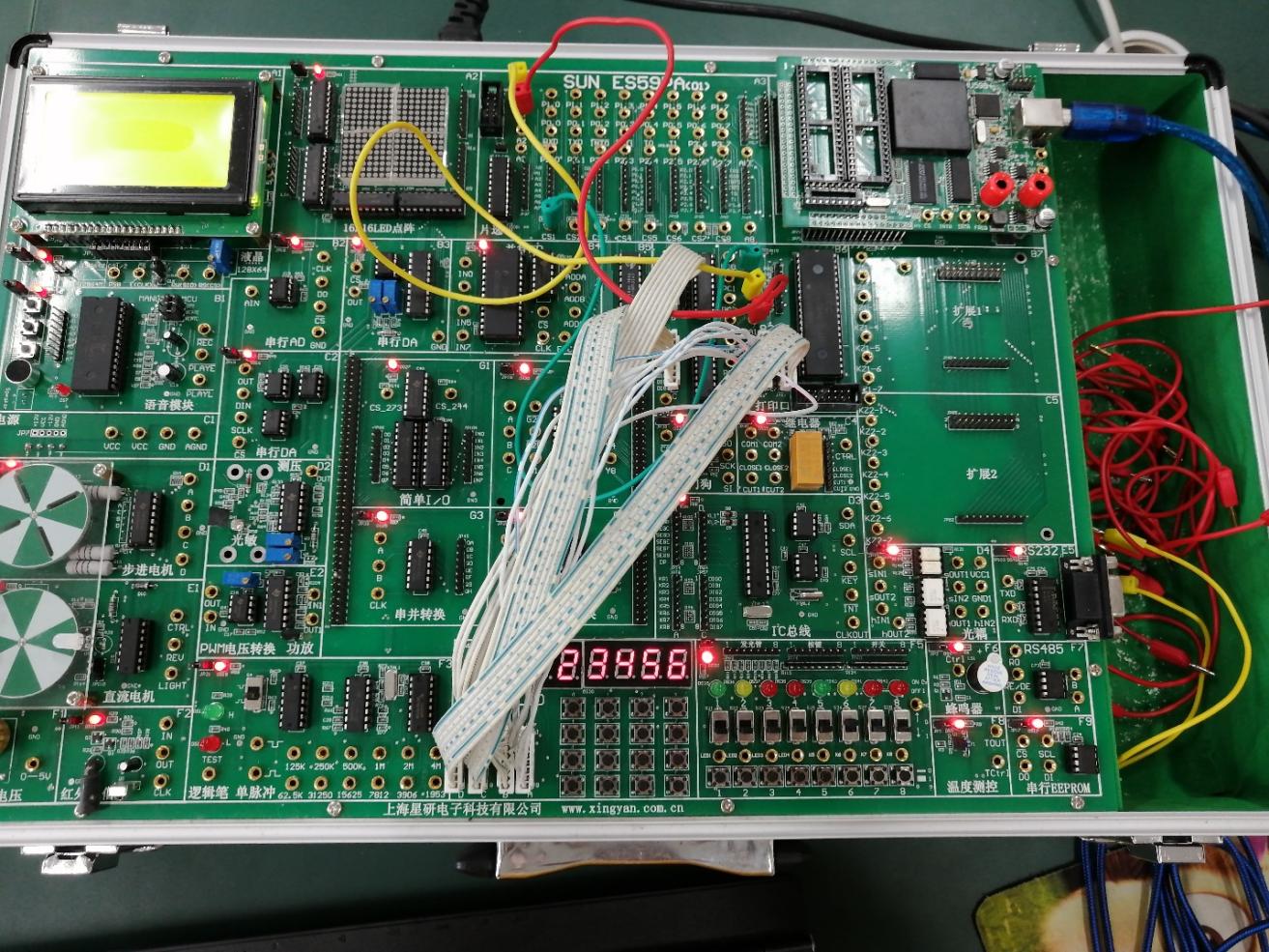
for (i = 0; i < 8; i++)

buffer[i] = keyi();

}

}

## 实验结果



## 实验扩展及思考

显示程序中延时函数起什么作用？如何调节数码块亮度？

答：在每段程序后面，都有一个延时和消影。消影的主要作用是避免下段代码受到干扰，而延时的作用是区分显示和消影，避免无法观察的数字的现象。

调节数码块亮度：增大电压。