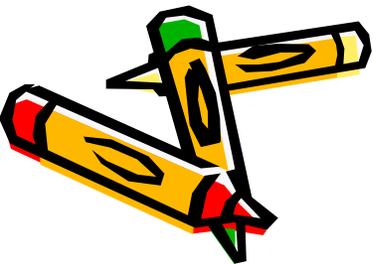


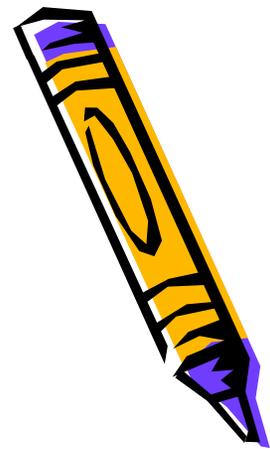
# 单片机原理及应用实验计划

- MCS-51汇编语言程序设计 2学时
- 并行I/O口扩展应用 2学时
- 定时器中断应用 2学时
- 串并转换实验(选) 2学时

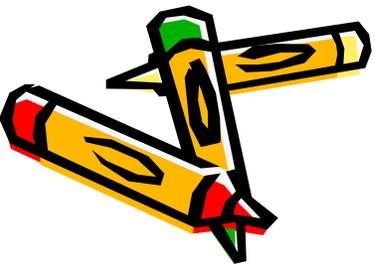
共 6 / 8 学时



# 实验要求：

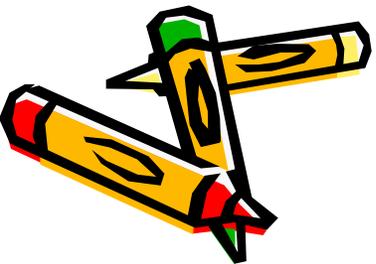
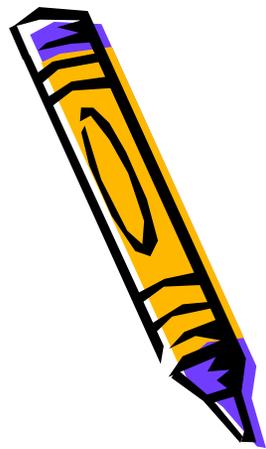


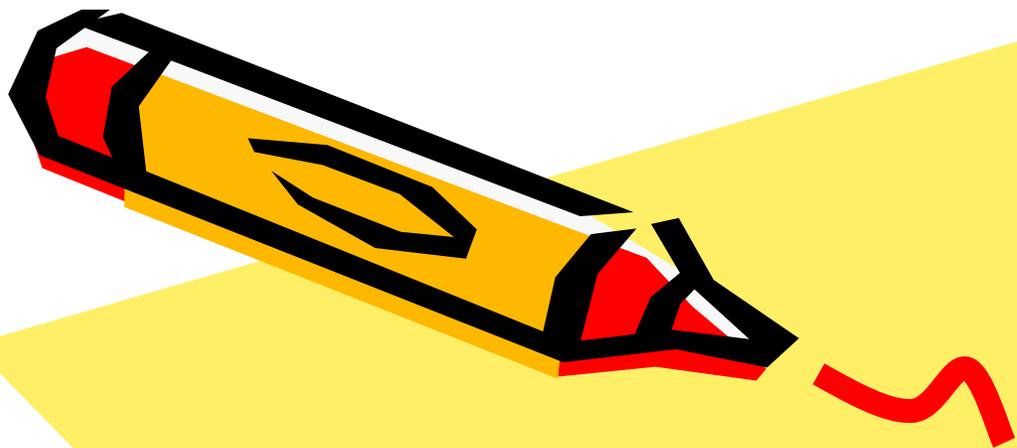
- 实验前预习，提前编写预习程序。
- 实验时按学号对应机号入座，并签到，做完实验请通知老师。
- 实验时不要带电触摸芯片，不可随意拔插芯片。有要求接线的实验，请务必先断电再接线。出现问题及时联系老师。
- 实验完整理好再离开。
- 注意室内卫生。



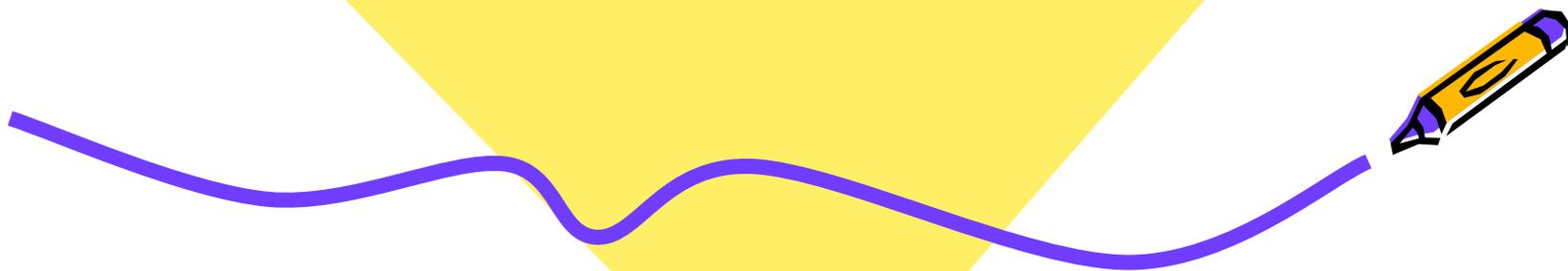
# 实验仪器

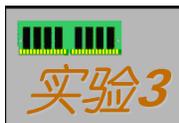
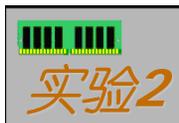
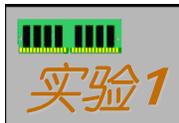
- 计算机一台
- Dais958H实验系统一台



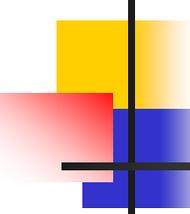


请签到







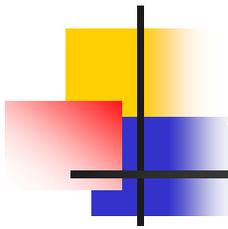


# 实验一：汇编语言程序设计及调试

---

## 一、实验目的

- 1.了解单片机各部件的工作原理；
- 2.熟悉Dais集成调试软件的使用；
- 3.掌握MCS-51汇编语言指令系统；
- 4.学习程序的调试步骤。



## 二、实验内容

---

- 1. 修改存放在E:\MCS51中文件EX1.ASM。  
该程序含有多处语法错误，要求将其语法错误处修改正确。
- 2. 编写并调试一查找程序：
  - 从内部RAM 50H~59H十个单元中查找与5AH存放数相等的单元个数，并存入60H中。
  - (如时间不够，可下次课调试)

## 三、实验步骤

1. 开启计算机进入98系统，点击桌面图标



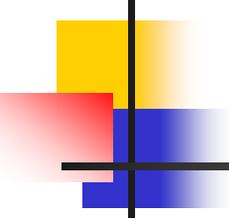
选择取消（本次实验不要求连接实验系统），进入Dais集成调试软件的主界面。

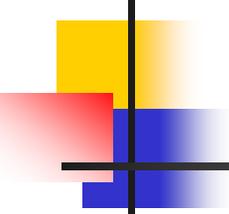
菜单列表

## 2. 打开MCS-51程序 (E:\MCS51\EX1.ASM)



菜 单	用 途
文件菜单	建立、打开、保存源文件，关闭当前窗口，输出列表表文件，退出系统。
编辑菜单	对活动窗口有关内容进行编辑。
编译菜单	源程序编译、连接、装载，浏览模块，设置编译选项。
调试菜单	单步、断点、连续运行方式，增加变量，程序复位，程序暂停，设置 PC。
工程菜单	建立、打开、编辑、关闭工程，打开工程文件列表。
设置菜单	设置仿真模式及通讯端口。
视图菜单	查看调试、寄存器、存储器、变量、编译信息窗口。
窗口菜单	层叠、平铺及关闭所有窗口，选择已打开的窗口。
帮助菜单	操作使用说明、实验指导、显示关于 Dais-IDE 的版本信息。

- 
3. 用编译菜单中“编译 / 文件编译、连接”命令对该文件进行编译。
- 由于源程序存在着若干个语法错误，那么编译错误信息便出现在“编译连接信息”窗口中；
  - 将光标放在程序窗中，双击错误信息行，光标便自动跳转到源文件编辑窗口的相应出错行；
  - 按正确语法要求进行修改。
  - 同学边修改边进行记录，填入如下格式中。

- 
- 记录格式为：发生错误行号、原语句、修改后语句。

发生错误行号	原语句	修改后语句

- 实验完当堂交给指导教师。

4.调试程序，从内部RAM 50H~59H十个单元中查找与5AH存放数相等的单元个数，并存入60H中。

(1) 新建源程序文件

建议存放在F盘或G盘中，文件名中的类型为 (.ASM)，文件名须小于8个字符，只允许数字或字母。

如dq1.asm

(2) 编译源程序文件

用编译菜单，编译源程序；程序有语法错误必须改正。成功弹出如下信息：

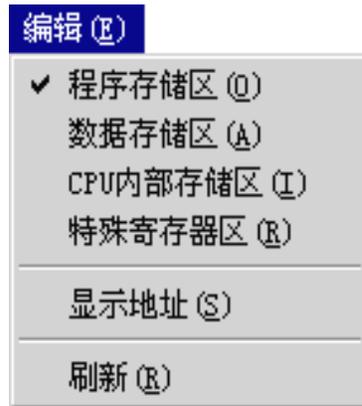


### (3) 设置仿真模式

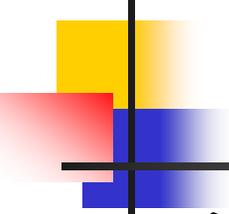


- 按上图设置好，点击确认

(4) 打开“视图”中相关存储区窗口，输入实验数据



- 本题中使用内部RAM 50H~5AH、60H单元，所以打开CPU内部存储区，输入数据。



(5) 下载程序，选择“编译+连接+下载”  菜单项

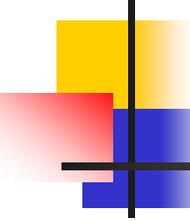
(6) 运行程序，可选择“运行到光标处”  菜单项

(7) 查看CPU内部存储区内容，并分析60H中内容正确？

## 四、实验报告

- 写出本次实验中内容2的源程序及流程框图。
- 编写排序程序，将内部RAM 50H~59H中的十个单字节无符号的正整数，按从小到大的次序重新排列。



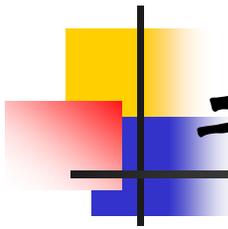


# 实验二：并行I/O接口应用

---

## 一、实验目的

- 1.掌握可编程I/O接口芯片8255的接口原理；
- 2.熟悉对8255初始化编程和实现I/O设计方法。



## 二、实验内容

---

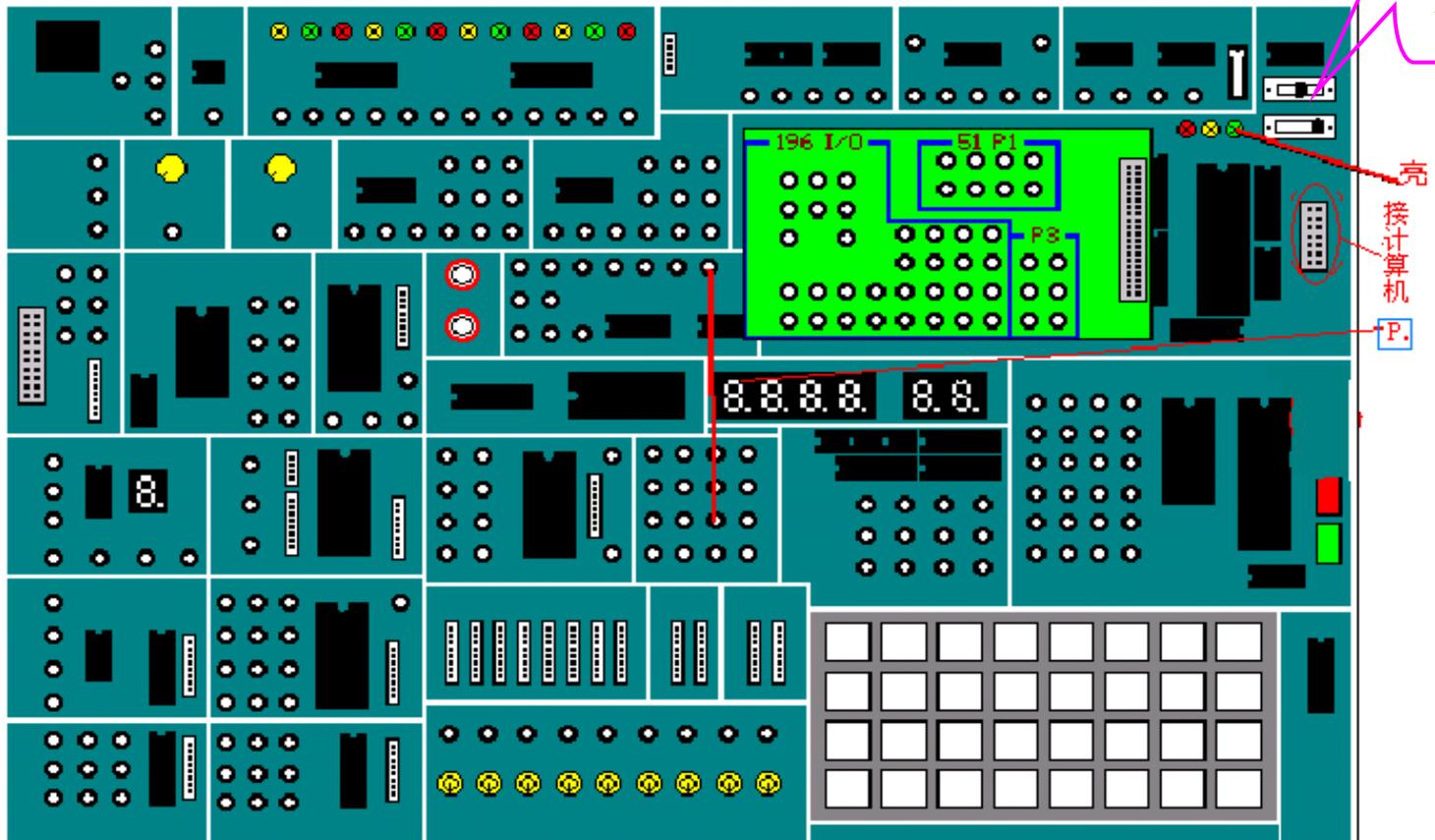
利用8255的PB口为输入，PA口为输出。

实现：

- (1) 当输入开关状态为偶数个1时，输出口对应开关状态；
- (2) 当输入开关状态为奇数个1时，输出口为全0，指示灯全亮。

# 三、实验步骤

## 1 实验系统

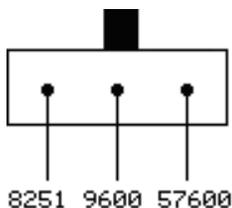


确保实验箱在断电情况下连接串口通信线!

返回

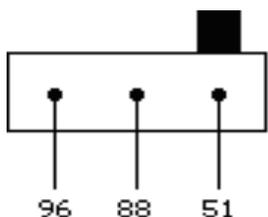
## 开关设置

### ■ ① 串行口及通信波特率选择开关



如图，其中最左边表示选择8251串行通信，右边两位为系统与计算机通信的波特率选择位。一般选择9600波特率。

### ■ ② CPU选择开关

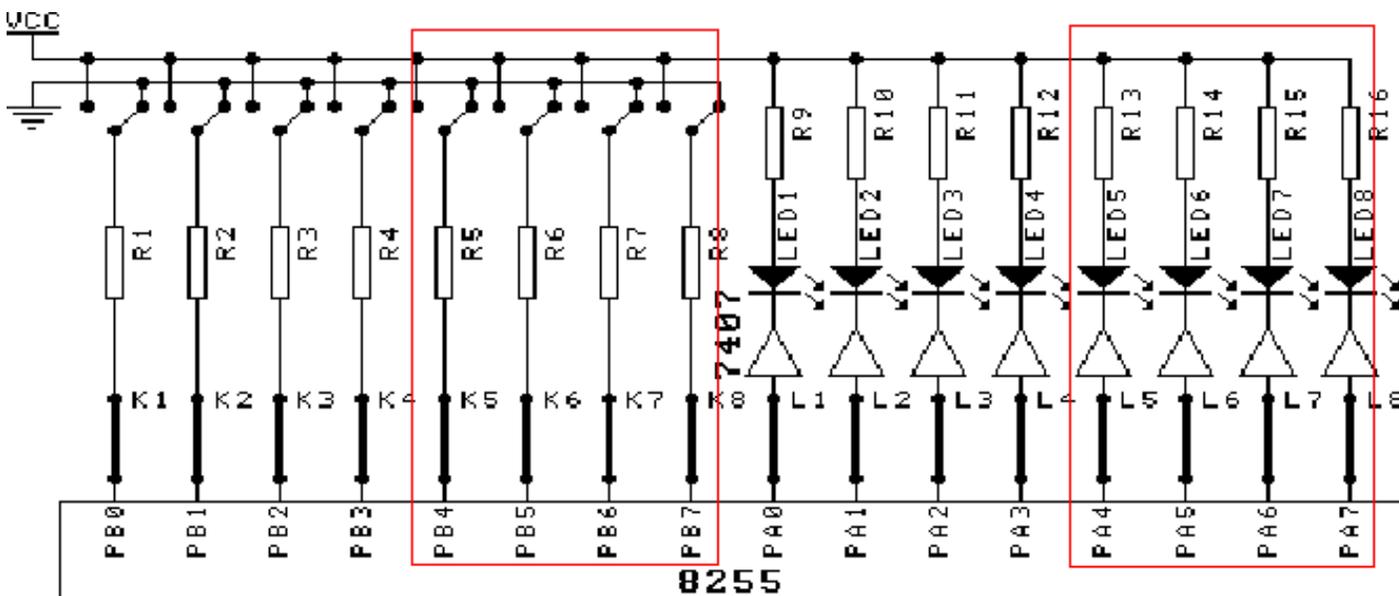


如图，在开机前选择使用CPU类型。对应指示灯会亮，选择51，绿灯亮。

- 开启电源开关后，数码管最高位显示P.，若无按动右边的复位按钮。

## 2 连接实验线

将8255PB口低四位或高四位接K1-K4，PA口对应位接L1-L4，无须连接CS片选信号线。



8255芯片的A口地址为0FFD8H、B口地址为0FFD9H、C口地址为0FFDAH、控制口地址为0FFDBH。

连接好实验线，再打开实验系统电源开关（在后面）

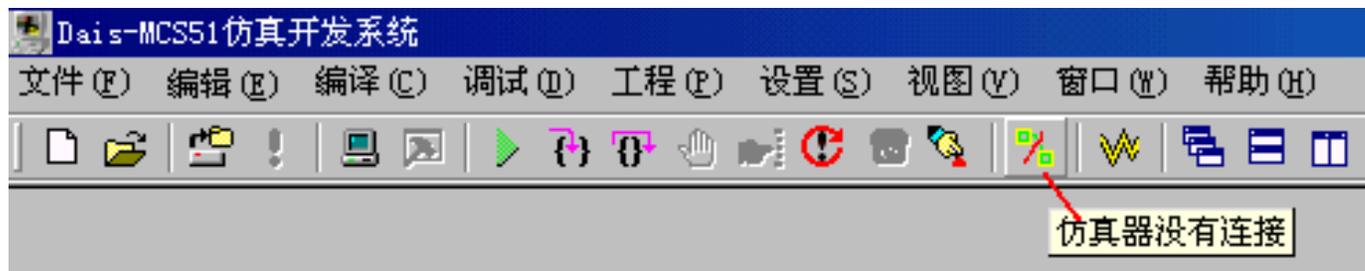
点击桌面图标

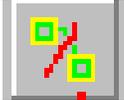


进入软件开发环境



点击“确认”后联机。

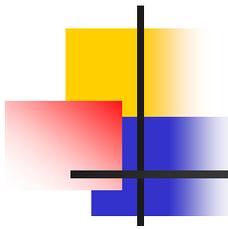


当图标  变灰色表示联机成功

## 设置仿真模式

仿真模式

<b>型号</b> <input checked="" type="radio"/> Dais-52 <input type="radio"/> Dais-52L <input type="radio"/> Dais-52M <input type="radio"/> Dais-52H	<b>片外RAM</b> <input type="radio"/> 用户 <input checked="" type="radio"/> 系统	<b>片外ROM</b> <input type="radio"/> 用户 <input checked="" type="radio"/> 系统	<input type="button" value="确认"/>
<b>I/O</b> <input type="checkbox"/> P2.0 <input type="checkbox"/> P2.1 <input type="checkbox"/> P2.2 <input type="checkbox"/> P2.3 <input type="checkbox"/> P2.4 <input type="checkbox"/> P2.5 <input type="checkbox"/> P2.6 <input type="checkbox"/> P2.7	<b>触发方式</b> <input type="radio"/> 下降沿 <input checked="" type="radio"/> 上升沿	<b>晶振源</b> <input type="radio"/> 外部 <input checked="" type="radio"/> 内部	<input type="button" value="取消"/>
<input type="checkbox"/> 存储区映射	<b>频率</b> <input type="radio"/> 24MHz <input checked="" type="radio"/> 12MHz <input type="radio"/> 6MHz	<b>片内ROM</b> <input checked="" type="radio"/> 0000H--0FFFH <input type="radio"/> 0000H--1FFFH <input type="radio"/> 0000H--3FFFH <input type="radio"/> 0000H--7FFFH	
	<input type="checkbox"/> P0口 勾=I/O		



### 3 程序设计说明

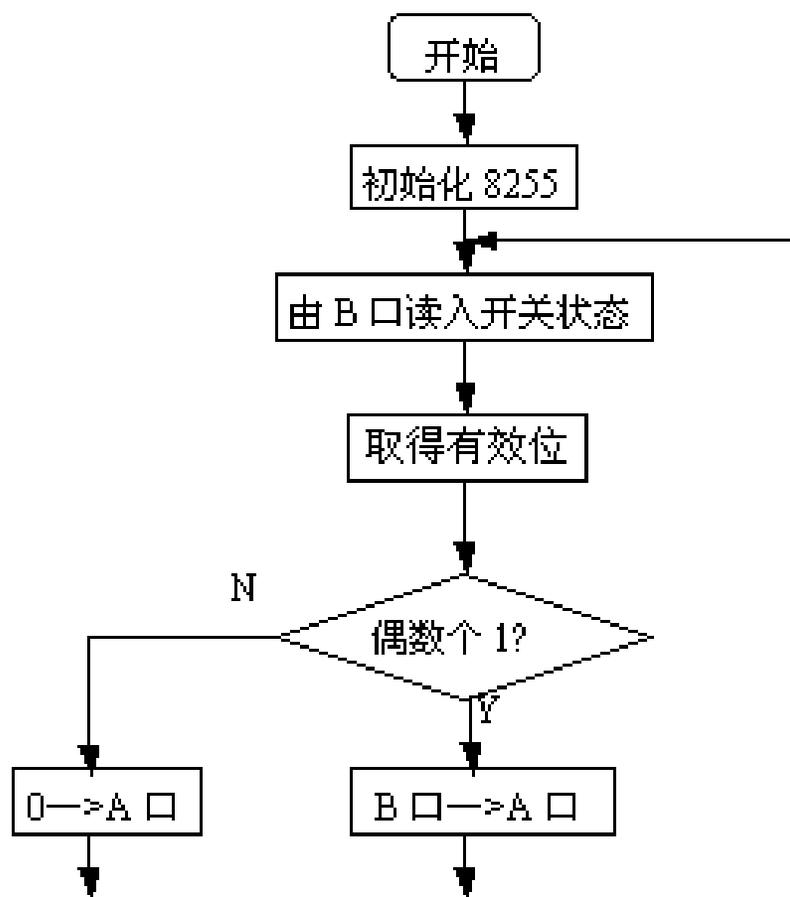
---

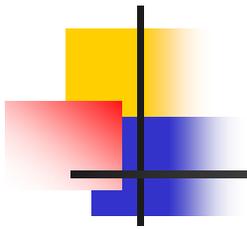
- 8255A是可编程外围通用并行接口芯片，它具有 A、B、C 三个并行接口，能在三种方式下工作：
  - 方式 0：基本输入/输出式；
  - 方式 1：选通输入/输出式；
  - 方式 2：双向选通工作方式。

工作方式控制字：用来设定通道的工作方式及数据的传送方向的。

工作方式控制字放在程序的开始部分（即初始化 8255）。

### 3 程序设计说明





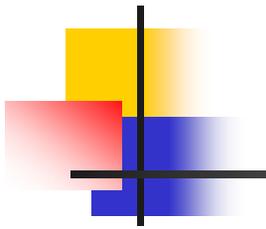
4 程序输入、编译、连接、装载。

5 程序运行 ，拨动K1-K8，观察L1-L8发光二极管变化是否对应实验要求。

## 四、实验报告

- 写出实验1中内容2的源程序及流程框图。
- 8255接口应用实验程序清单。





# 实验三:中断应用实验

## ■ 一、实验目的

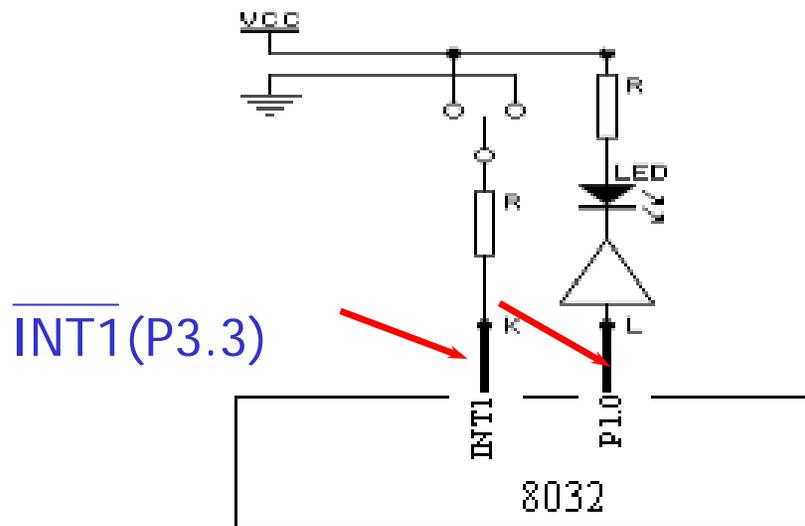
- 1.熟悉中断方式的设置及中断服务的编制。
- 2.掌握多级中断的应用及程序设计方法。
- 3.了解定时器的设计方法。

## ■ 二、实验内容

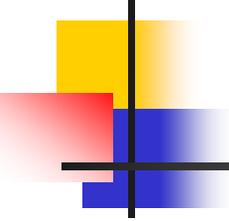
用单片机内部的两级中断来实现白天灯不亮、夜晚灯闪亮控制。

要求用外部中断1接开关,控制白天或夜晚;用定时器1进行定时,控制灯每2秒变换亮灭。

### 三、实验原理



设计说明：当开关为低电平（夜晚），产生外部中断，进入外部中断服务程序；再启动定时器1的工作，定时灯亮2S、灭2S，反复工作。当开关为高电平（白天），结束外部中断程序，并关闭定时器，灯灭。



## 设计说明

- CPU8032内有6个中断源，CPU响应了某一中断，将会转到对应中断服务程序的入口地址处执行跳转指令，以跳到用户设计的中断处理程序入口。
- **外部中断1**的中断服务程序入口地址是**0013H**,
- **定时器T1**的中断服务程序入口地址是**001BH**。
- 实验系统的晶振频率为6MHz。定时器1选用定时方式0，它的最大定时达不到2S，假设定时常量设定为10ms，需对它计数200次才能实现时间2秒。

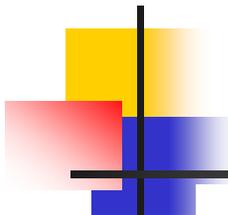
## 程序框架：

```
ORG 0000H
AJMP MAIN
ORG 0013H
AJMP WINT1
ORG 001BH
AJMP TIME1
ORG 0050H
```

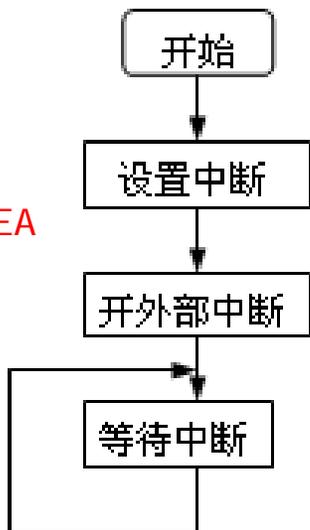
```
MAIN:  ;
      ;
WINT1: ;
      ;
TIME1: ;
      ;
```

END

返回

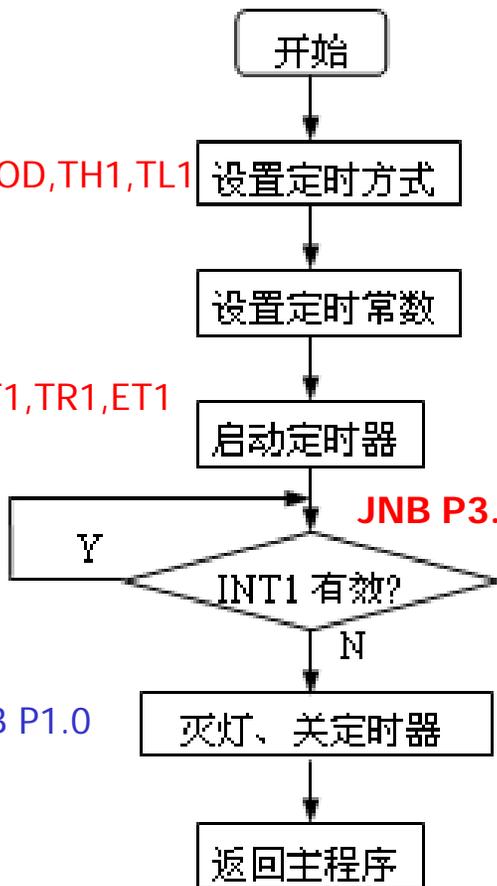


### 主程序



IT1,PX1,EX1,EA

### INT1 中断服务程序



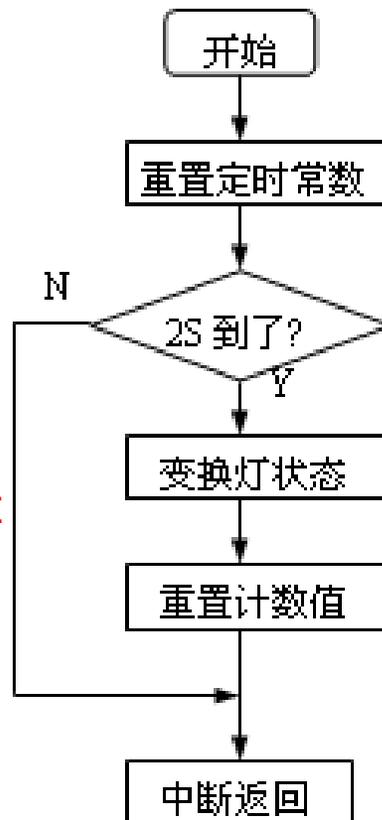
TMOD,TH1,TL1

PT1,TR1,ET1

JNB P3.3,HERE

SETB P1.0

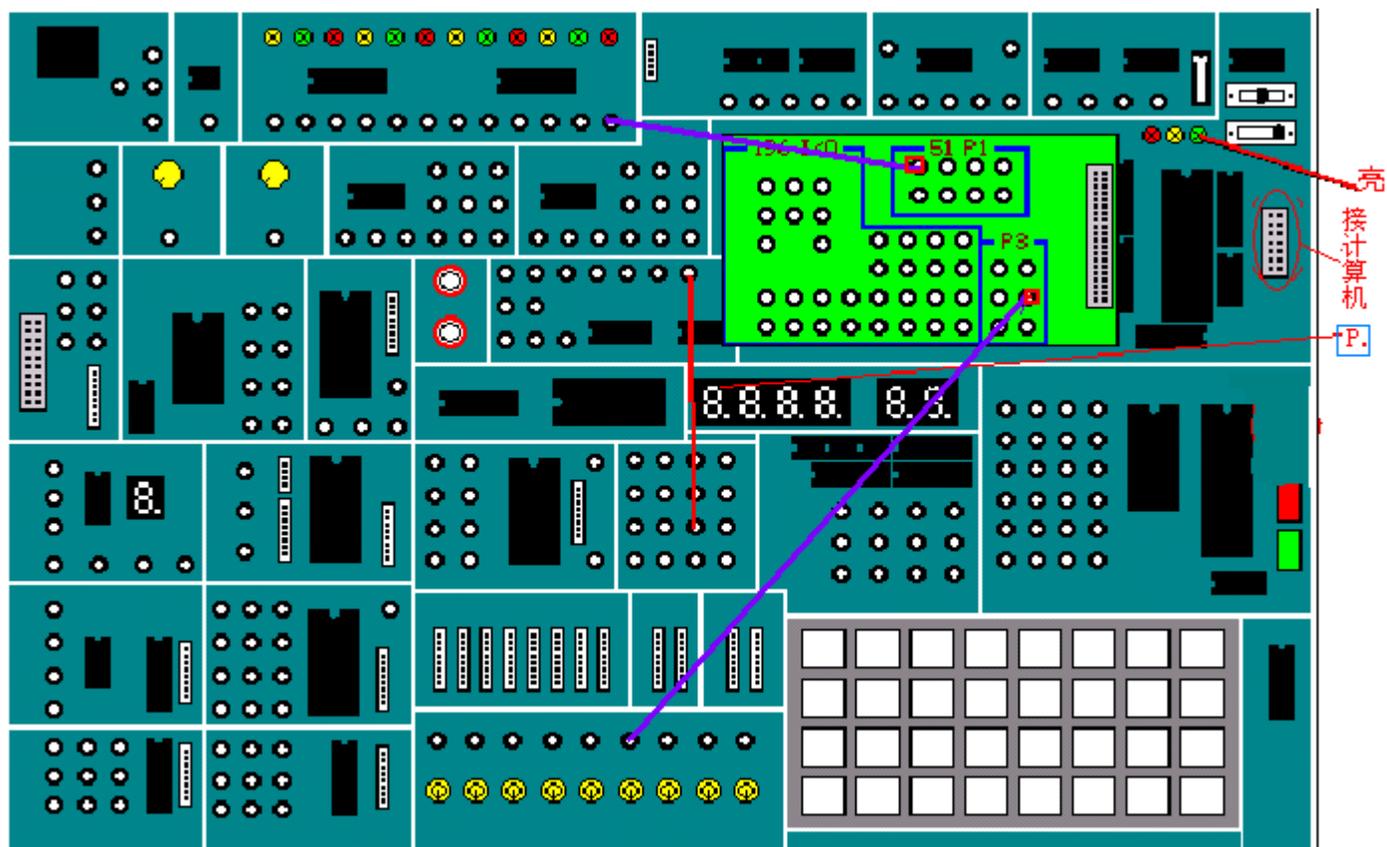
### 定时器 1 中断服务程序

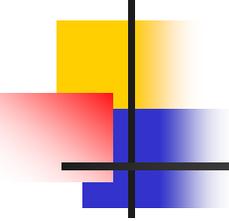


RETI

## 五、实验步骤

1. 连接串口线，P1.0接灯，P3.3接开关，打开电源

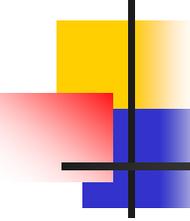


- 
2. 输入程序，设置仿真模式，编译连接下载，运行程序。
  3. 拨动开关，观察现象并分析。

## 六、实验报告

1. 写出程序清单。
2. 试分析程序中的中断优先级及响应时序。





# 实验四:串并转换实验

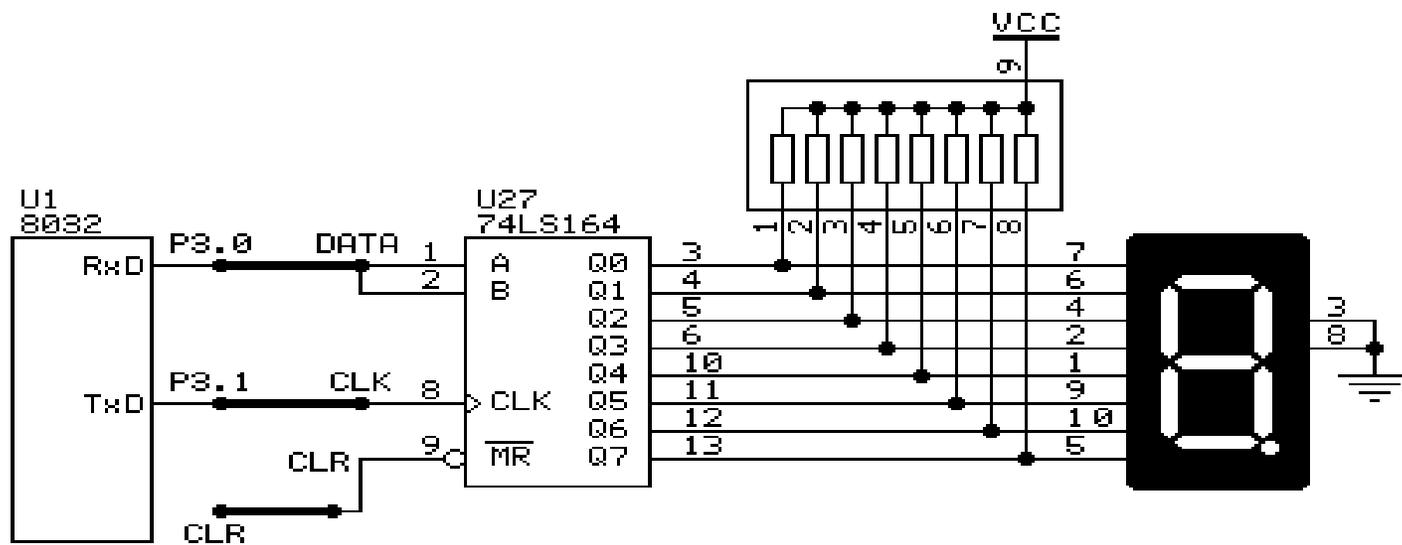
## ■ 一、实验目的

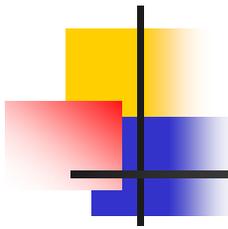
- 1.掌握8032串行口方式0工作方式及编程方法。
- 2.掌握利用串行口扩展I/O通道的方法。

## ■ 二、实验内容

利用8032串行口和串行输入并行输出移位寄存器74LS164, 扩展一个8位输出通道, 用于驱动一个数码显示管, 在数码显示管上循环显示8032串行口输出的0~9这十个数字。

### 三、实验接线

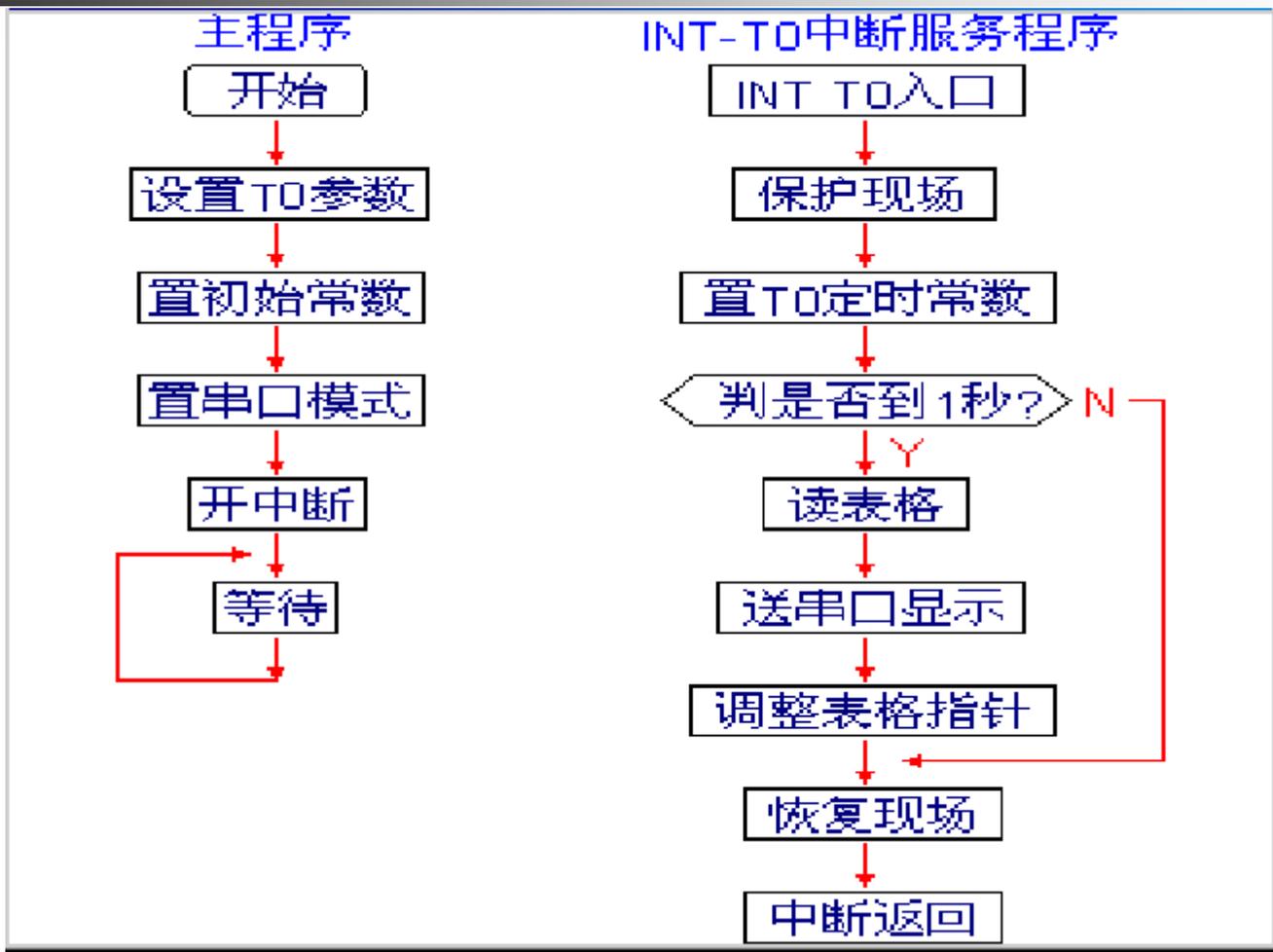


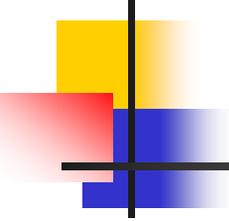


## 四、设计说明及流程

---

串行口工作在方式0时，可通过外接移位存储器实现串并行转换。在这种方式下，数据为8位，只能从RXD端输入输出，TXD端总是输出移位同步时钟信号，其波特率固定为晶振频率1/12。由软件置位串行控制寄存器(SCON)的REN后才能启动串行接收，在CPU将数据写入SBUF寄存器后，立即启动发送。待8位数据输完后，硬件将SCON寄存器的TI位置1，TI必须由软件清零。

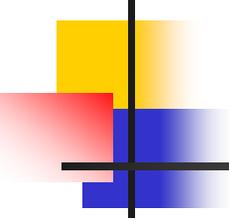




## 五、实验参考程序

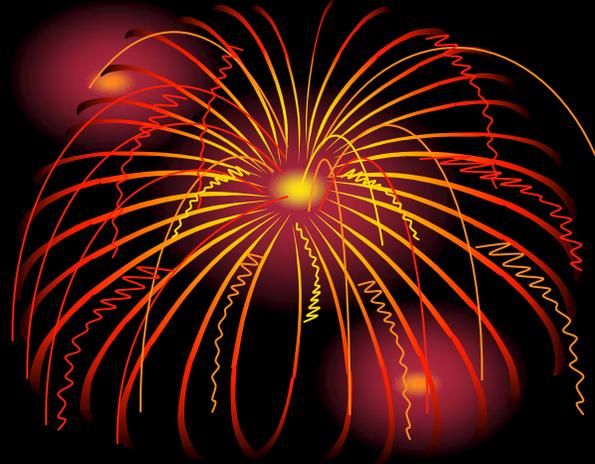
---

- ORG 0000h
- LJMP main
- ORG 000BH
- LJMP INT\_T0
- ORG 0050h
- main: MOV SP,#53H
- MOV TMOD,#01H
- MOV TL0,#00H
- MOV TH0,#4BH
- MOV R0,#00H
- MOV R1,#20H
- MOV SCON,#00H
- CLR TI
- CLR RI
- SETB TR0
- SETB ET0
- SETB EA
- SJMP \$



- INT\_T0: PUSH ACC
- PUSH PSW
- CLR EA
- CLR TR0
- MOV TLO,#00H
- MOV TH0,#4BH
- SETB TR0
- DJNZ R1,EXIT
- MOV R1,#20H
- MOV DPTR,#CDATA
- MOV A,R0
- MOVC A,@A+DPTR
- CLR TI
- CPL A
- MOV SBUF,A
- INC R0
- CJNE R0,#0AH,EXIT
- MOV R0,#00H
- EXIT: SETB EA
- POP PSW
- POP ACC
- RETI
- CDATA: DB 03H,9FH,25H,0DH,99H,49H,41H,1FH,01H,09H
- END





谢 谢

