



# 单片机原理与应用课程实验指导书

周汝雁 编著

上海海洋大学海洋智能信息实验教学示范中心

---

# 目 录

## 第一部分：系统介绍

- 一、实验系统整机介绍
- 二、硬件资源
- 三、调试软件的安装与使用
- 四、实验程序说明

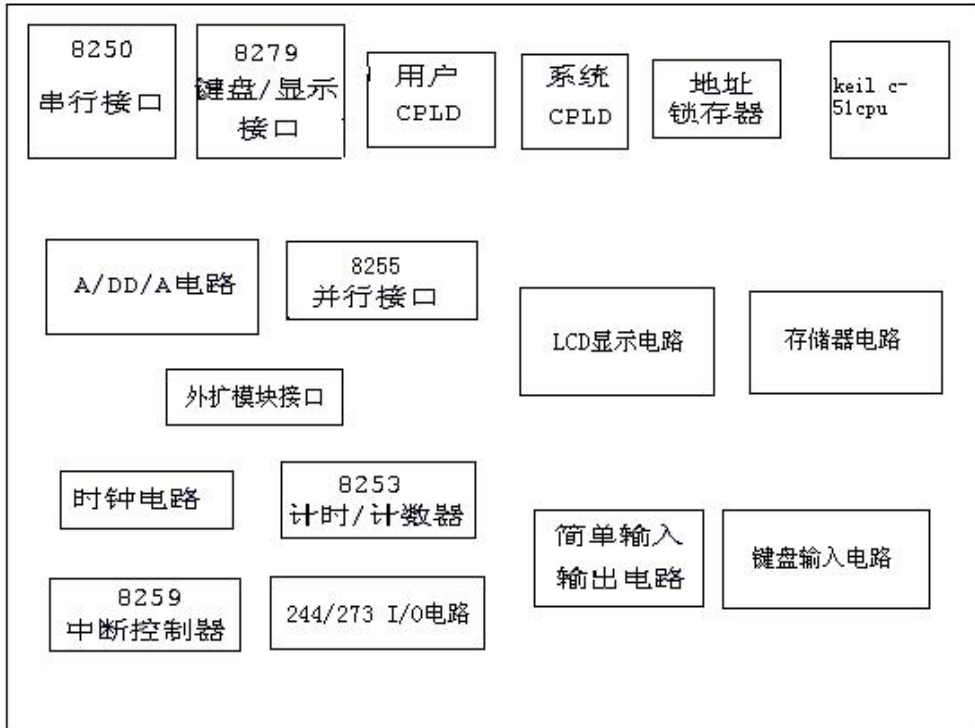
## 第二部分 基本实验

- 实验一 P1 口实验一
- 实验二 P1 口实验二
- 实验三 简单 I/O 口扩展实验一
- 实验四 简单 I/O 口扩展实验二
- 实验五 中断实验
- 实验六 定时器实验
- 实验七 8255A 可编程并行接口实验一
- 实验八 8255A 可编程并行接口实验二
- 实验九 数码显示实验
- 实验十 8279 显示接口实验一
- 实验十一 8279 键盘显示接口实验二
- 实验十二 LED 显示实验

# 第一部分：系统介绍

## 一、实验系统整机介绍

实验系统组成及结构：实验系统由电源、系统板、可扩展的实验模板、微机串口通讯线、JTAG 通讯线及通用连接线组成。系统板的结构简图见下图。



## 二、硬件资源

- 1、可编程并口接口芯片 8255 一片。
- 2、串行接口两个：8250 芯片，系统与主机通讯用。8051 为单片机的通讯端口。
- 3、LED、键盘控制芯片 8279 一片，其地址已被系统固定为 CFE8H、CFE9H。硬件系统要求编码扫描显示。
- 4、独立的六位数码管显示及 128x32 点阵式液晶显示电路。
- 5、独立的 3x8 行列扫描键盘，可用于各种控制电路。
- 6、ADC0809 A/D 转换芯片一片。
- 7、DAC0832 D/A 转换芯片一片，模拟输出可调。
- 8、8 位简单输入接口 74LS244 一个，8 位简单输出接口 74LS273 一个。
- 9、逻辑电平开关，发光二极管显示电路。

- 
- 10、一个可手动产生正、负脉冲的脉冲发生器。
  - 11、一个可自动产生正、负脉冲的脉冲发生器，按基频 24.0MHz 进行 1 分频 (CLK0)、二分频 (CLK1)、四分频 (CLK2)、八分频 (CLK3)、十六分频 (CLK4) 输出方波。
  - 12、一路 0---5V 连续可调模拟量输出 (ANO)。
  - 13、可编程定时器 8253 一个，其地址、三个定时器的门控输入、控制输出。
  - 14、可编程中断控制器 8259 一个，其中断 IRQ 输入、控制输出。
  - 15、2 组总线扩展接口，最多可同时扩展 2 块应用实验板。
  - 16、两块可编程器件 EPM7064/AFT1502，一块被系统占用。另一块供实验用。两块器件皆可通过 JTAG 接口在线编程。
  - 17、电源接口：PC 机电源插座，可与 PC 机电源直接接驳。有外接开关电源，提供所需的+5V，±12V，其输入为 220V 的交流电。

其中，可扩展实验板 Keil-c51cpu 组成：

- 1、微处理器：8051，它的 P1 口、T0、T1、EX0、EX1、RD、WR 皆可使用。
- 2、时钟频率：24.0MHz
- 3、存储器：程序存储器与数据存储器统一编址，最多可达 64k，板载 ROM(监控程序 27C256)16k；RAM(程序存储器 6264)8k 供下载实验程序，可扩展达 32k；RAM(数据存储器 6264)8k 供程序使用，可扩展达 32k。

在程序存储器中：

- 0000H---2FFFH 为监控程序存储器区，不可用。
- 3000H---3FFFH 为数据存储器区。
- 4000H---7FFFH 为实验程序存储器区，供下载实验程序
- 8000H---CF9FH, CFF0H-----FFFFH 为用户 CPLD 试验区段，可在此段空间编程。
- CFA0H---CFDFH 系统 I/O 区，可用但不可更改。

**注意：KEILC-51 编译是从 0 地址开始仿真。**

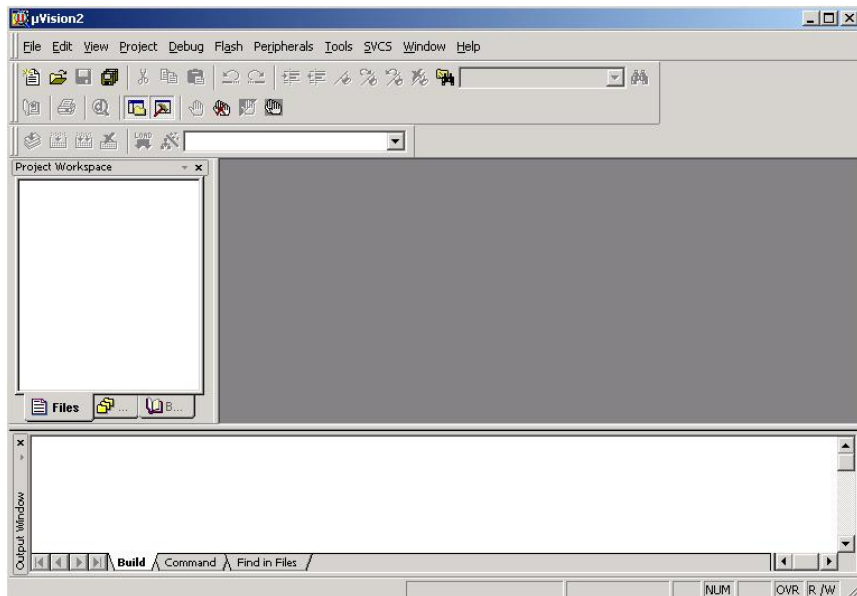
### 三、调试软件的安装与使用

#### (一) 硬件准备


- 1、把扩展实验板 Keil-c51cpu 插入实验箱的 Keil-c51cpu 插座，位置见系统板的结构简图。
- 2、将交叉串口电缆的一端（针形口）与 CPU 模块左侧的串行插口（孔形口）插座连接，另一端（孔形口）与 PC 机的 COM1 连接。
- 3、打实验箱电源开关，通电。

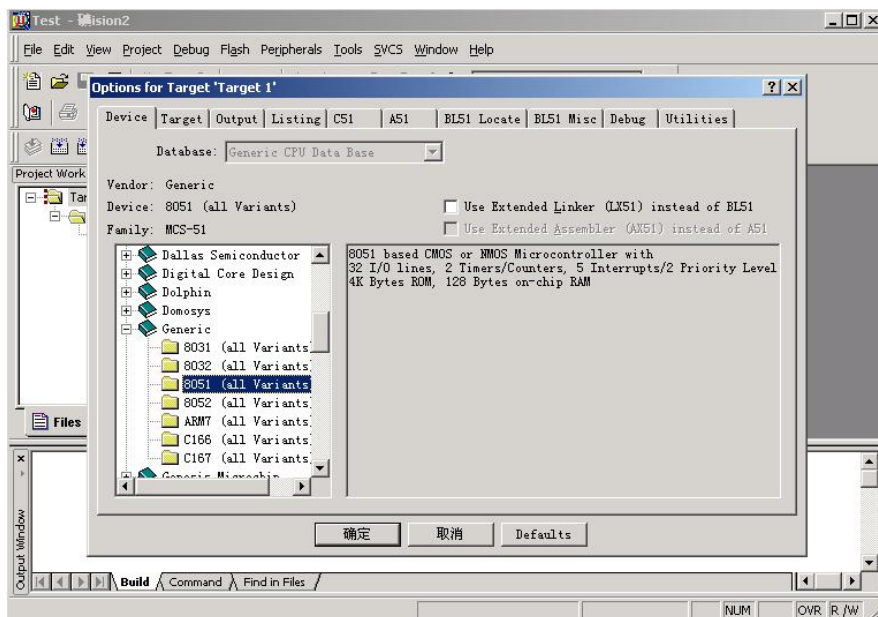
#### (二) 软件设置

1. 打开 Keil C 环境。



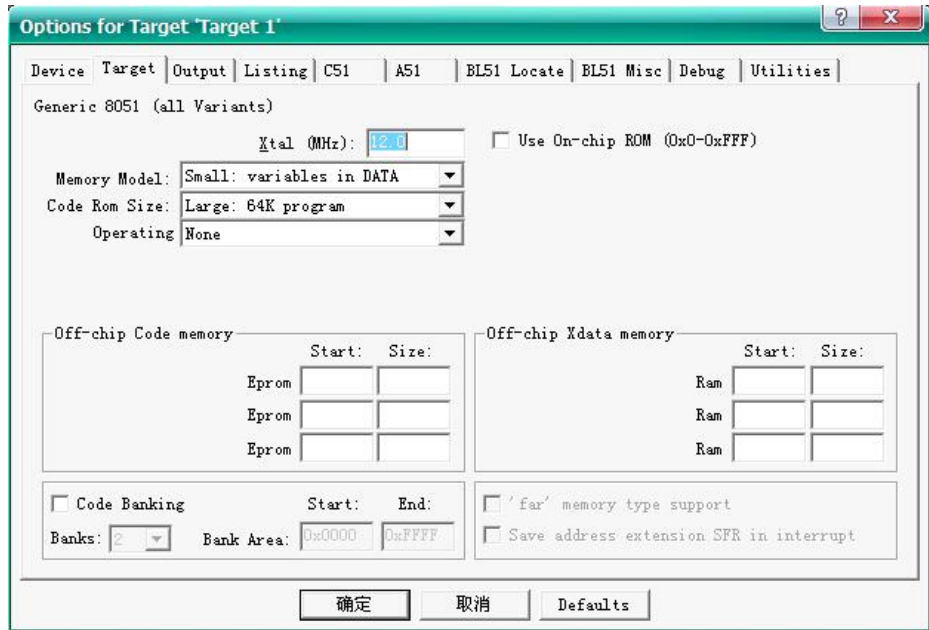
2. 打开工程文件：在主菜单上选“Project”项，下拉列表中选择“Open project”打开已有的工程文件。如实验一的第 1 个实验，在汇编语言文件夹中找到文件夹 T1，打开其中的 T1\_1.uv2 文件。

3、环境设置：打开 T1\_1.uv2 工程文件后，在工具栏中选择如下图选项设置调试参数及运行环境 ，或从主菜单“Project”项中选择“Options for Target ‘Target1’”，打开如下图设置窗口。

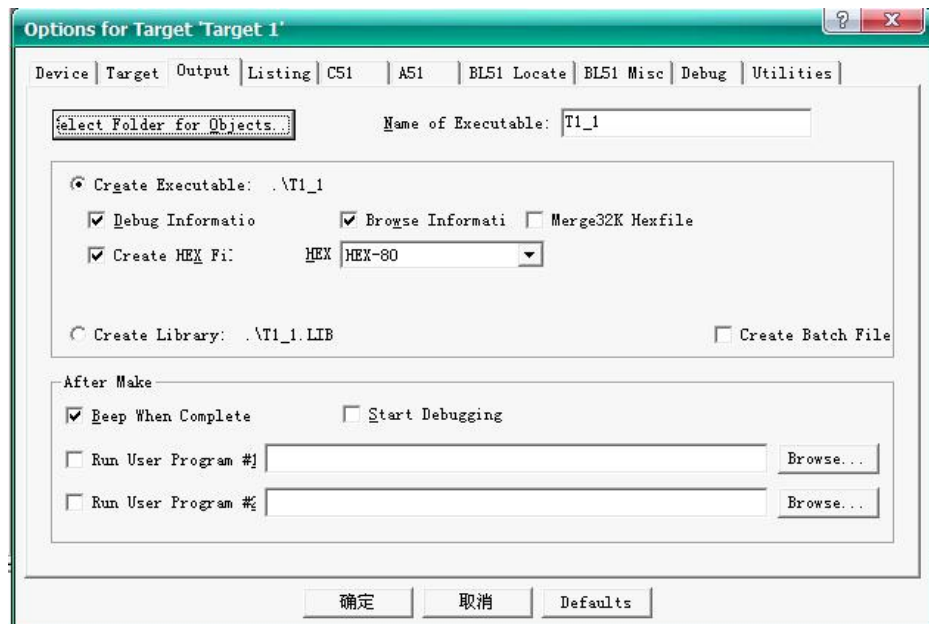


在“Device”项下选择要仿真的芯片类型，如：Generic -> 8051。见上图。

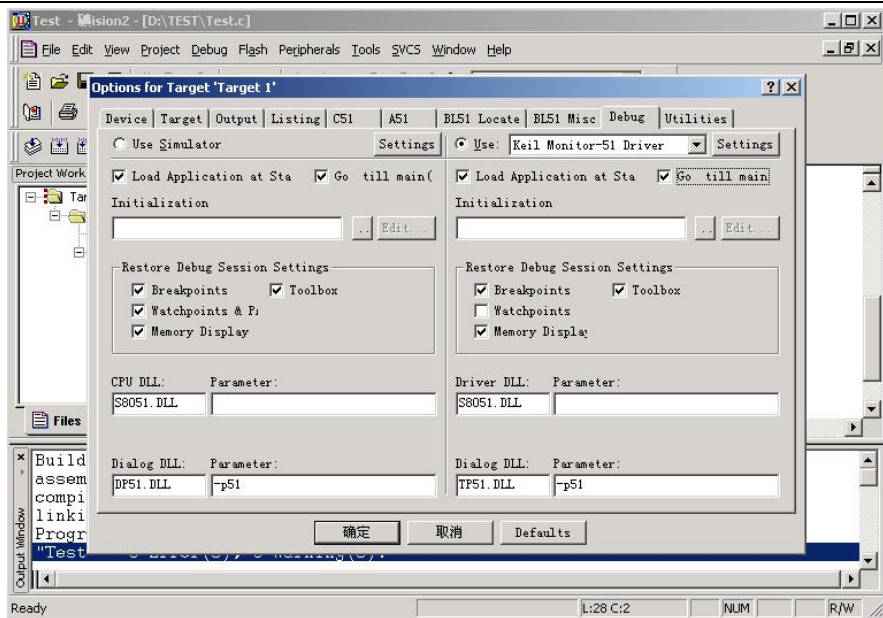
在“Target”项下的晶振设置中修改为硬件电路所用晶振频率，为12MHz。见下图。



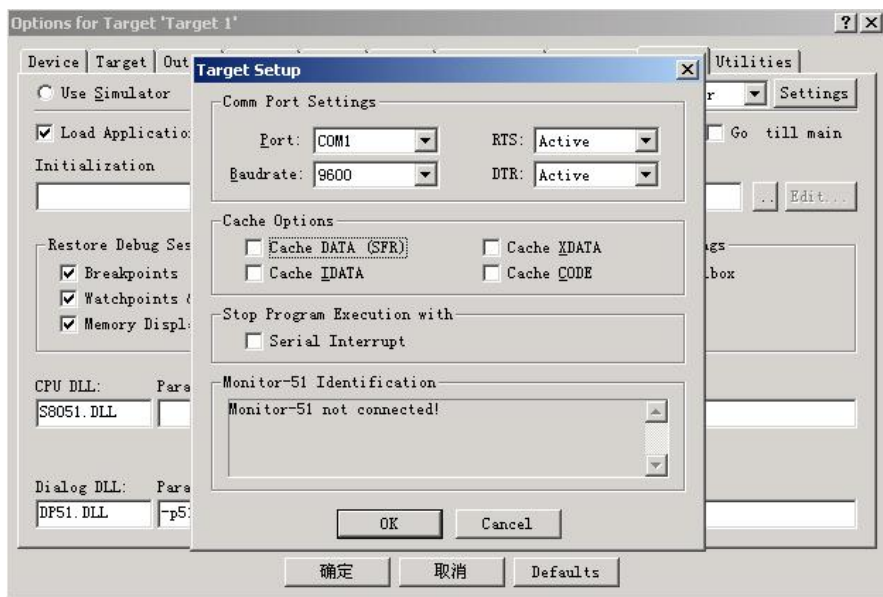
在“Output”项下在 Creat HEX File 选项前打勾，则在编译的同时生成可下载执行的 HEX 文件。见下图。



在“Debug”项下在 Use: 选项前打勾，并选择 Keil Monitor-Driver。



点击“Settings”按钮，进入串口选择及波特率设置窗口（如下图所示）。



选择合适的波特率及串口号，方法如下：

- 波特率选择 9600。
- 串口号选择前需要查看，串口号查看：光标选中桌面上的“我的电脑”图标，点击右键——单击设备管理器——打开“端口 (COM 和 LPT)”查看是 COM 几 (如 COM1)，就在 Port 选项中选择 COM 几 (COM1)。

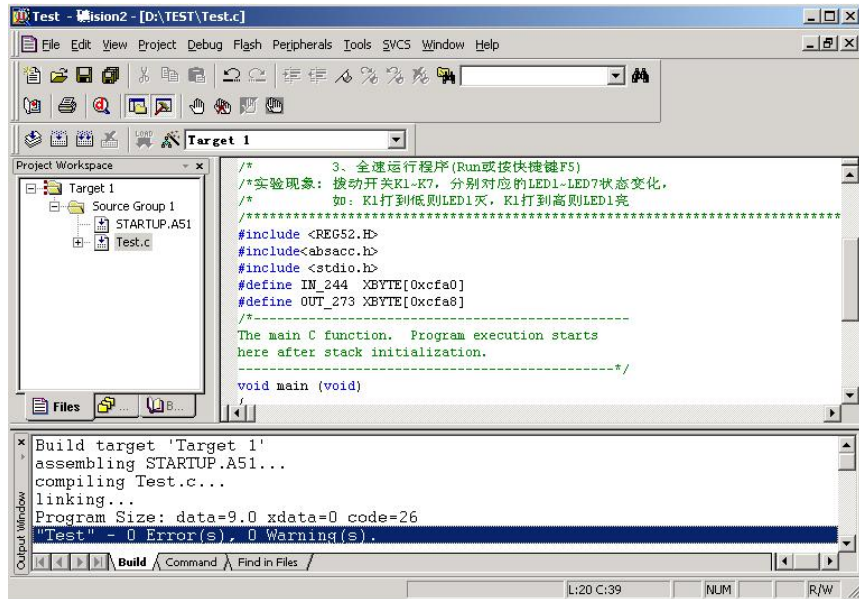
“Serial Interrupt”项不可选，把前面的勾打掉。点 OK 保存设置。

在“Option for Target ‘Target1’”窗口中，点确定，退出环境设置。

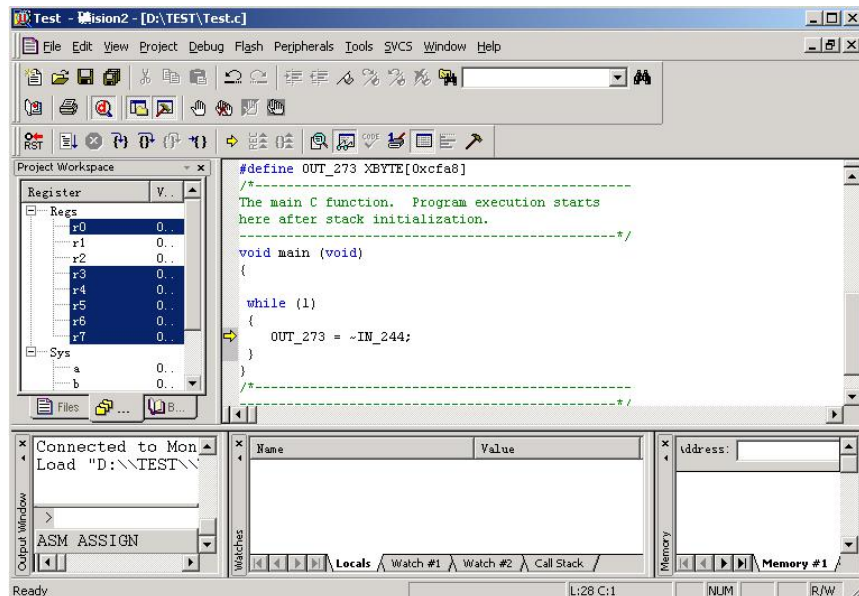
浏览添加编辑好的 C 或 ASM 文件。添加完毕点“Close”，关闭窗口。


4. 编译链接：在“Build Bar”  工具条中，选第二项编译当前文件，


第三项为编译全部。编译完成，在下方“Output”窗口中出现编译结果。



5. 仿真调试：点击调试按钮          ，或从主菜单选取调试“Start/Stop Debug Session CTRL+F5”（快捷键 CTRL+F5），程序下载到仿真芯片中。窗口下方显示下载进度条。100%下载完成后出现如下窗口。




下载完毕后，运用“Debug Bar”调试工具条  上不同图标可分别进行单步、停止、进入、跳出、运行到光标、全速运行等。

全速运行后，程序不受控。如需再次单步调试，需点击调试工具  中的红色停止按钮，停止调试。



---

若要重新修改程序、编译程序，则点击工具条中红色的 Start/Stop Debug Session 按钮，退出仿真运行，返回进行程序修改、编译。编译通过后，需要重新将程序下载到实验箱上的单片机芯片后，再运行程序，查看效果。

## 四、实验程序说明

实验可用汇编语言或 C 语言；

汇编语言程序：

在汇编语言程序实验文件夹中，找到每个实验对应的文件夹，打开相应的扩展名为 “.uv2” 的工程文件。如：做实验二时，找到文件夹 “T2”，打开工程文件 “T2.uv2”；做实验三时，找到文件夹 “T3”，打开工程文件 “T3.uv2”。（实验代码本书略）

C 语言程序：

在 C 语言程序实验文件夹中，找到每个实验对应的文件夹，打开扩展名为 “tmp.uv2” 的工程文件。如：做实验二时，找到文件夹 “P2”，打开工程文件 “tmp.uv2”；做实验三时，找到文件夹 “P3”，打开工程文件 “tmp.uv2”。（实验代码本书略）

---

## 第二部分 基本实验

### 实验一 P1 口实验一

#### 一、实验目的:

1. 学习 P1 口的使用方法。
2. 学习延时子程序的编写和使用。

#### 二、实验内容:

1. P1 口做输出口，接八只发光二极管，编写程序，使发光二极管循环点亮。
2. P1 口做输入口，接八个按钮开关，以实验箱上 74LS273 做输出口，编写程序读取开关状态，在发光二极管上显示出来。

#### 三、实验环境: 单片机实验箱

#### 四、实验原理:

P1 口为准双向口，P1 口的每一位都能独立地定义为输入位或输出位。作为输入位时，必须向锁存器相应位写入“1”，该位才能作为输入。8051 中所有口锁存器在复位时均置为“1”，如果后来在口锁存器写过“0”，在需要时应写入一个“1”，使它成为一个输入。

可以用第二个实验做一下实验。先按要求编好程序并调试成功后，可将 P1 口锁存器中置“0”，此时将 P1 做输入口，会有什么结果。

延时程序的实现。现常用的有两种方法，一是用定时器中断来实现，一是用指令循环来实现。在系统时间允许的情况下可以采用后一种方法。

若系统晶振为 6.144MHZ，则一个机器周期为  $12 \div 6.144\mu\text{s}$  即  $1 \div 0.512\mu\text{s}$ 。现要写一个延时 0.1s 的程序，可写出如下代码：

```
MOV R7, #X          (1)
DEL1: MOV R6, #200  (2)
DEL2: DJNZ R6, DEL2 (3)
      DJNZ R7, DEL1  (4)
```

上面 MOV、DJNZ 指令均需两个机器周期，所以每执行一条指令需要  $1 \div 0.256\mu\text{s}$ ，现求出 X 值：

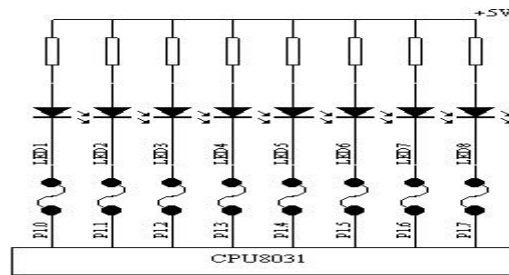
$$1 \div 0.256 + X (1 \div 0.256 + 200 \times 1 \div 0.256 + 1 \div 0.256) = 0.1 \times 10^6$$

指令 (1)	指令 (2)	指令 (3)	指令 (4)
所需时间	所需时间	所需时间	所需时间

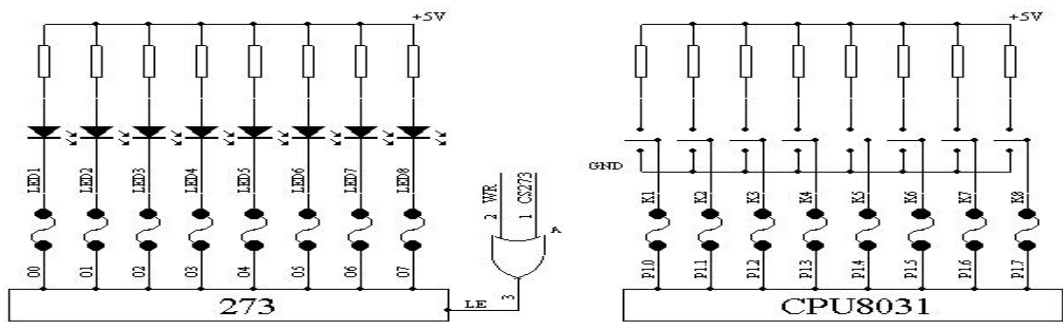
$$X = (0.1 \times 10^6 - 1 \div 0.256) / (1 \div 0.256 + 200 \times 1 \div 0.256 + 1 \div 0.256) = 127D = 7FH$$

经计算得  $X=127$ 。代入上式可知实际延时时间约为  $0.100215s$ ，已经很精确了。

## 五、实验原理图：



P1 口输出实验



P1 口输入实验

## 六、线路连接和实验步骤：

执行程序 1 (T1\_1.ASM) 时：

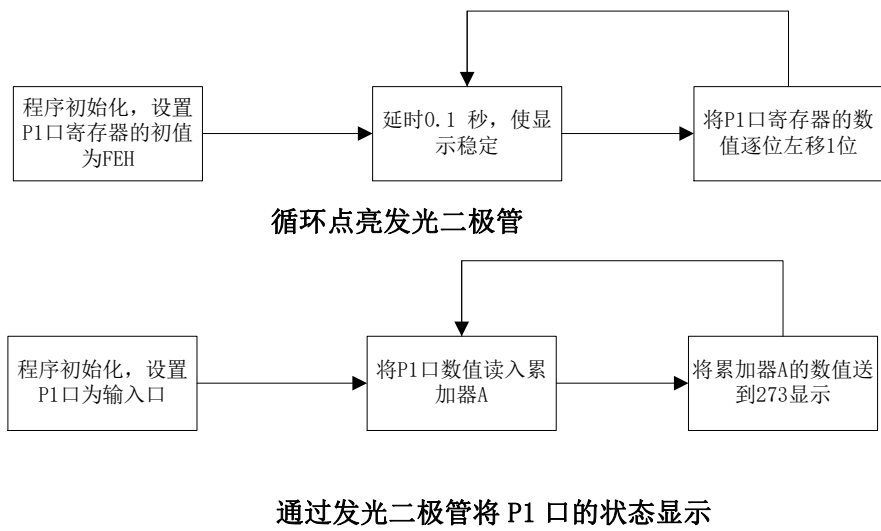
1. CPU 板上的 P1.0~P1.7 接试验箱的发光二极管 L1~L8；
2. 试验箱上电，打开软件，设置好端口和波特率，然后加载实验程序，依次点击汇编 (F3)，调试 (F5)，运行 (F9)，观察实验结果，八个 LED 等循环点亮。

执行程序 2 (T1\_1.ASM) 时：

1. CPU 板上的 P1.0~P1.7 接试验箱的平推开关 K1~K8；
2. 试验箱 74LS273 的 00~07 接试验箱发光二极管 L1~L8；
3. 试验箱上 74LS273 的片选端 CS273 接试验箱上 CPLD ENCODER UNIT 的 CS0（由程序所选择的入口地址而定，与 CS0~CS7 相应的片选地址请查看第一部分系统资源，以后不赘述）；

4. 试验箱上电，加载实验程序，依次点击汇编，调试，运行，观察实验结果，可用开关来控制 LED 灯的亮灭。

## 七、程序框图：



## 实验二 P1 口实验二

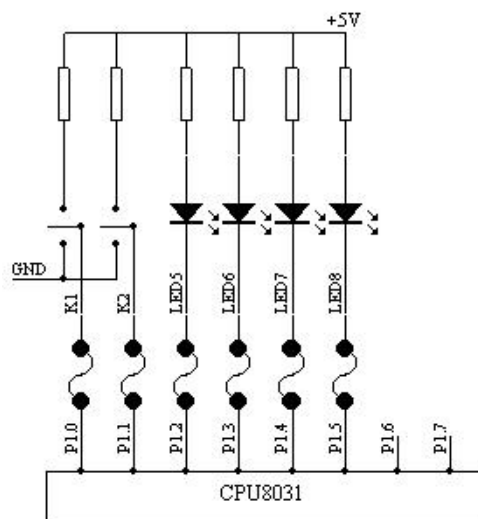
### 一、实验目的:

1. 学习 P1 口既做输入又做为输出的使用方法。
2. 学习数据输入、输出程序的设计方法。

### 二、实验环境: 单片机实验箱

### 三、实验原理:

### 四、实验原理图:



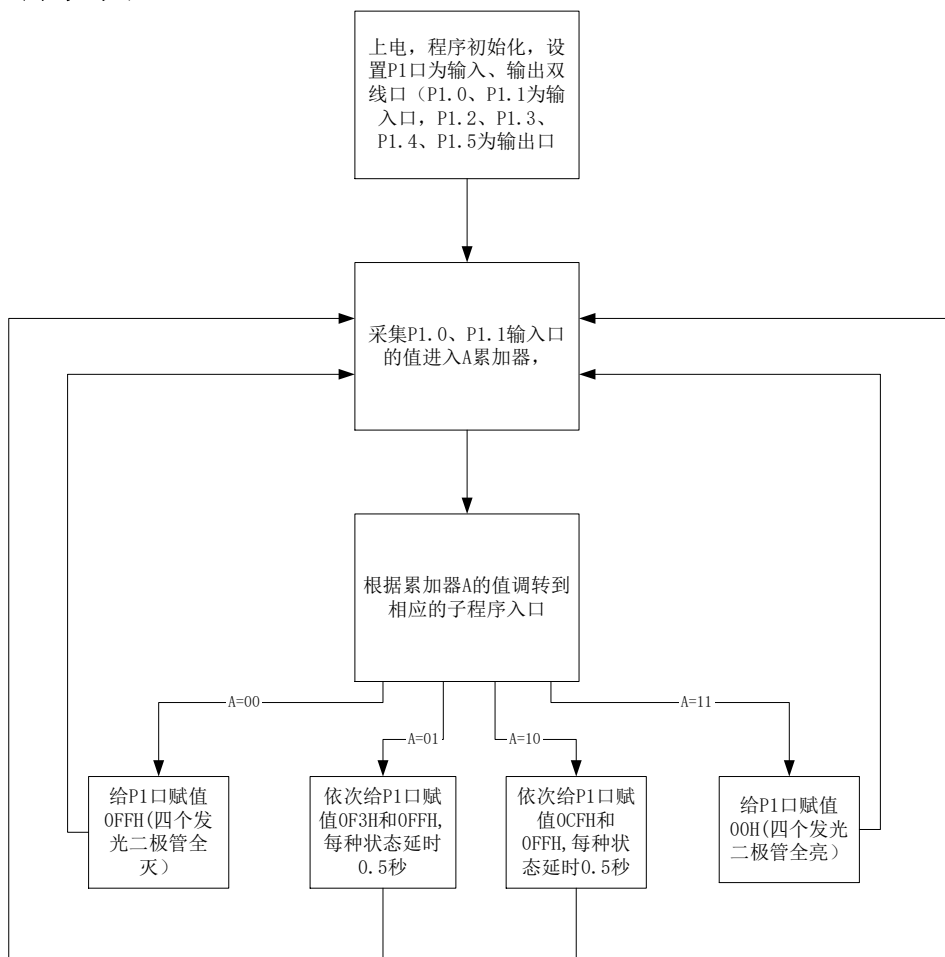
## 五、线路连接和实验步骤:

1. 实验箱的平推开关的输出 K1、K2 分别接 CPU 板的 P1.0、P1.1;
2. 实验箱上发光二极管的输入 L1、L2、L5、L6 分别接 CPU 板上的 P1.2、P1.3、P1.4、P1.5;
3. 运行实验程序, K1 做为左转弯开关, K2 做为右转弯开关。L5、L6 做为右转弯灯, L1、L2 做为左转弯灯;

### 结果显示:

- 1: K1 接高电平 K2 接低电平时, 右转弯灯 (L5、L6) 灭, 左转弯灯 (L1、L2) 以一定频率闪烁;
- 2: K2 接高电平 K1 接低电平时, 左转弯灯 (L1、L2) 灭, 右转弯灯 (L5、L6) 以一定频率闪烁;
- 3: K1、K2 同时接低电平时, 发光二极管全灭;
- 4: K1、K2 同时接高电平时, 发光二极管全亮。

## 六、程序框图:



# 实验三 简单 I/O 口扩展实验一

——交通灯控制实验

## 一、实验目的：

1. 学习在单片机系统中扩展简单 I/O 接口的方法。
2. 学习数据输出程序的设计方法。
3. 学习模拟交通灯控制的实现方法。

## 二、实验环境： 单片机实验箱

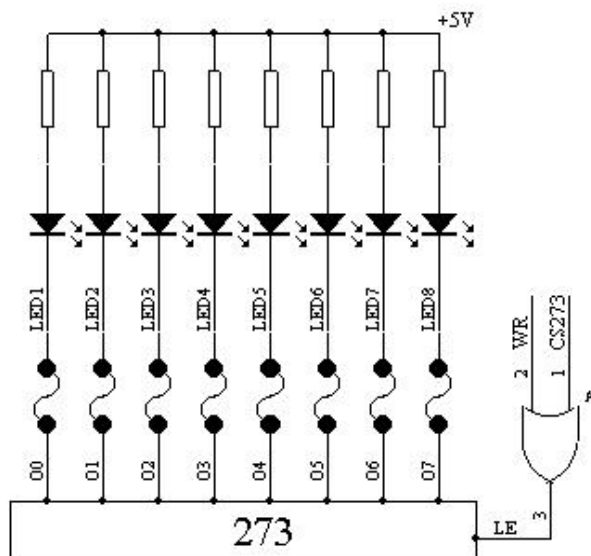
## 三、实验内容：

扩展实验箱上的 74LS273 做为输出口，控制八个发光二极管亮灭，模拟交通灯管理。

## 四、实验原理：

要完成本实验，首先必须了解交通路灯的亮灭规律。本实验需要用到实验箱上八个发光二极管中的六个，即红、黄、绿各两个。不妨将 L1(红)、L2(绿)、L3(黄)做为东西方向的指示灯，将 L5(红)、L6(绿)、L7(黄)做为南北方向的指示灯。而交通灯的亮灭规律为：初始态是两个路口的红灯全亮，之后，东西路口的绿灯亮，南北路口的红灯亮，东西方向通车，延时一段时间后，东西路口绿灯灭，黄灯开始闪烁。闪烁若干次后，东西路口红灯亮，而同时南北路口的绿灯亮，南北方向开始通车，延时一段时间后，南北路口的绿灯灭，黄灯开始闪烁。闪烁若干次后，再切换到东西路口方向，重复上述过程。各发光二极管的阳极通过保护电阻接到+5V 的电源上，阴极接到输入端上，因此使其点亮应使相应输入端为低电平。

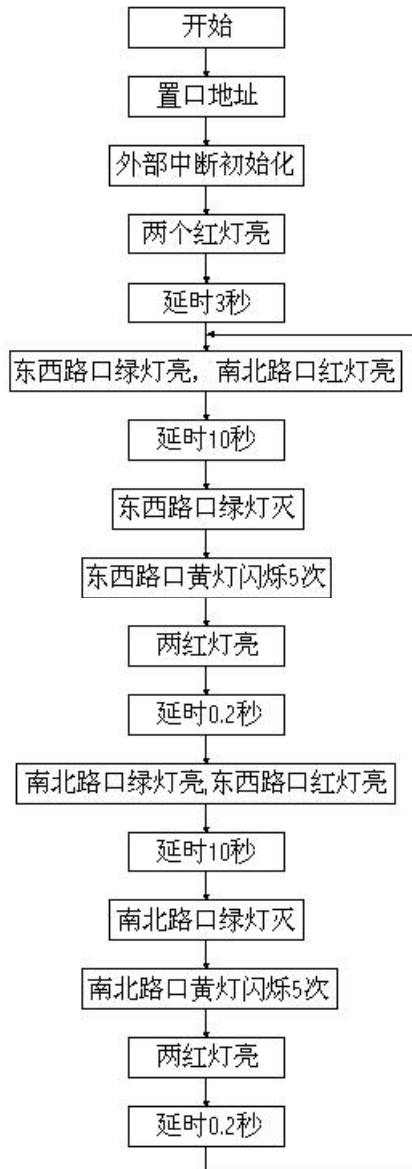
## 五、实验原理图



## 六、线路连接和实验步骤:

1. 实验箱 244/273 PORT 单元的 00~07 接实验箱上发光二极管 L1~L8;
2. 74LS273 的片选 CS273 接片选信号 CS0 (CPLD ENCODER UNIT), 此时 74LS273 的片选地址为 CFA0H~CFA7H 之间任选;
3. 运行实验程序, 观察 LED 显示情况是否与实验内容相符;

## 七、程序框图:



## 实验四 简单 I/O 口扩展实验二

### 一、实验目的：

1. 学习在单片机系统中扩展简单 I/O 口的方法。
2. 学习数据输入，输出程序的编制方法。

### 二、实验环境： 单片机实验箱

### 三、实验原理：

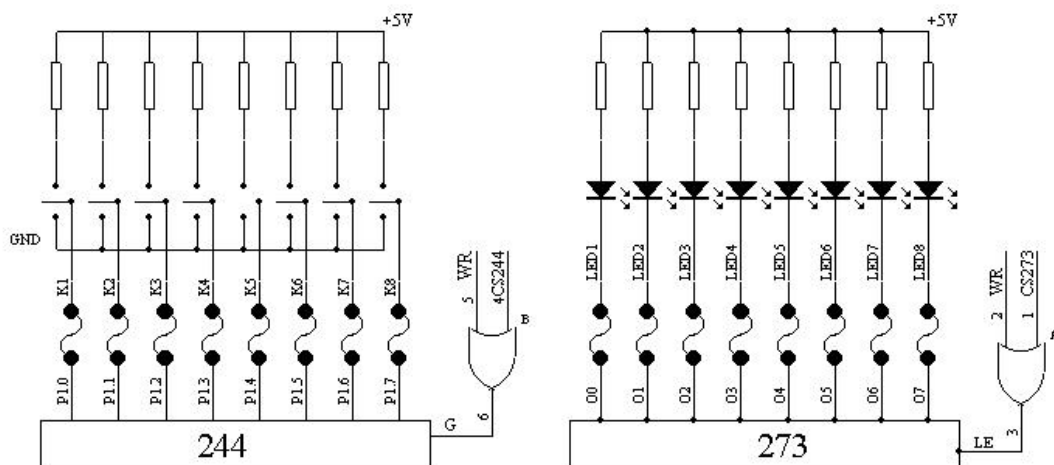
MCS-51 外部扩展空间很大，但数据总线口和控制信号线的负载能力是有限的。若需要扩展的芯片较多，则 MCS-51 总线口的负载过重，74LS244 是一个扩展输入口，同时也是一个单向驱动器，以减轻总线口的负担。

程序中加了一段延时程序，以减少总线口读写的频繁程度。延时时间约为 0.01 秒，不会影响显示的稳定。

### 四、实验内容：

利用 74LS244 做为输入口，读取开关状态，并将此状态通过发光二极管显示出来。

### 五、实验原理图：



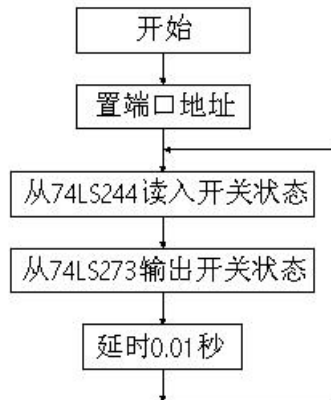
简单 I/O 实验 2

### 六、线路连接和实验步骤：

1. 244/273PORT 单元的 IN0~IN7 接平推开关的 K1~K8，片选信号 CS244 接 CS1。
2. 244/273PORT 单元的 00~07 接发光二极管的 L1~L8，片选信号 CS273 接 CS2。
3. 编程、全速执行。
4. 拨动开关 K1~K8，观察发光二极管状态的变化。



## 七、程序框图：



## 实验五 中断实验

——有急救车的交通灯控制实验

### 一、实验目的：

1. 学习外部中断技术的基本使用方法。
2. 学习中断处理程序的编程方法。

**二、实验环境：** 单片机实验箱

### 三、实验内容：

在实验三的内容的基础上增加允许急救车优先通过的要求。当有急救车到达时，两个方向上的红灯亮，以便让急救车通过，假定急救车通过路口的时间为 10 秒，急救车通过后，交通灯恢复中断前的状态。本实验以单脉冲为中断申请，表示有急救车通过。

### 四、实验原理：

交通灯的亮灭规律见实验三。

本实验中断处理程序的应用，最主要的地方是如何保护进入中断前的状态，使得中断程序执行完毕后能回到交通灯中断前的状态。要保护的地方，除了累加器 ACC、标志寄存器 PSW 外，还要注意：一是主程序中的延时程序和中断处理程序中的延时程序不能混用，本实验给出的程序中，主程序延时用的是 R5、R6、R7，中断延时用的是 R3、R4 和新的 R5。第二，主程序中每执行一步经 74LS273 的端口输出数据的操作时，应先将所输出的数据保存到一个单元中。因为进入中断程序后也要执行往 74LS273 端口输出数据的操作，中断返回时如果没有恢复中断前 74LS273 端口锁存器的数据，则显示往往出错，回不到中断前的状态。还要注意一点，主程序中往端口输出数据操作要先保存再输出，例如有如下操作：

```
MOV A, #0F0H      (0)
```

```
MOVX @R1, A      (1)
```

MOV SAVE, A (2)

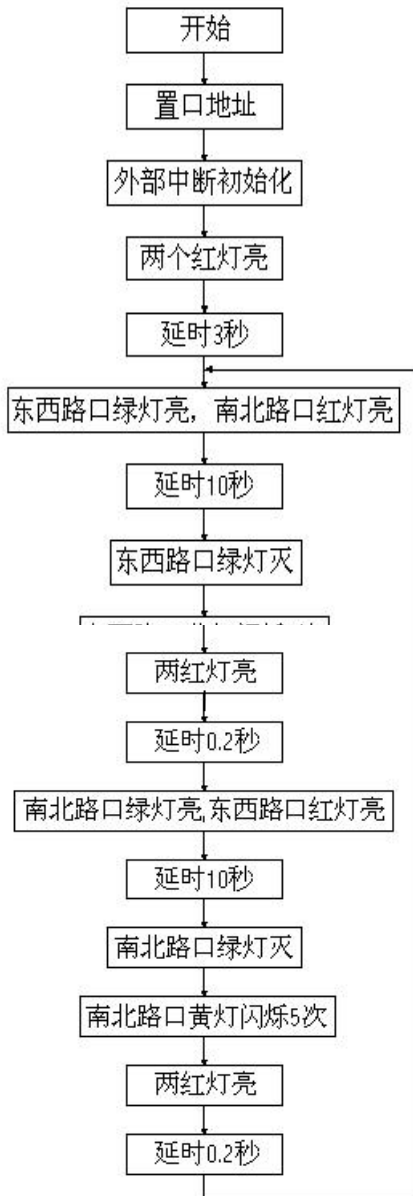
程序如果正好执行到(1)时发生中断,则转入中断程序,假设中断程序返回主程序前需要执行一句MOV A,SAVE指令,由于主程序中没有执行(2),故SAVE中的内容实际上是前一次放入的而不是(0)语句中给出的0F0H,显示出错,将(1)、(2)两句顺序颠倒一下则没有问题。发生中断时两方向的红灯一起亮10秒,然后返回中断前的状态。

### 五、实验原理图:(同实验三)

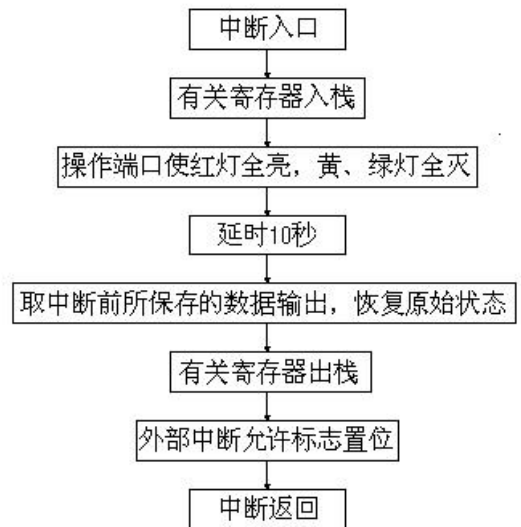
### 六、线路连接和实验步骤:

1. 244/273PORT 单元的的 00~07 接发光二极管 L1~L8;
2. 74LS273 的片选 CS273 接片选信号 CS2,此时 74LS273 的片选地址为 CFB0H~CFB7H 之间任选;
3. 单脉冲输出端 P-接 CPU 板上的 INTO。

### 七、程序框图: 主程序框图



### 中断程序框图



---

# 实验六 定时器实验

——循环彩灯实验

## 一、实验目的：

1. 学习 8031 内部计数器的使用和编程方法。
2. 进一步掌握中断处理程序的编写方法。

## 二、实验环境： 单片机实验箱

## 三、实验原理：

### 1. 定时常数的确定

定时器/计数器的输入脉冲周期与机器周期一样，为振荡频率的 1/12。本实验中时钟频率为 6.0 MHz，现要采用中断方法来实现 0.5 秒延时，要在定时器 1 中设置一个时间常数，使其每隔 0.1 秒产生一次中断，CPU 响应中断后将 R0 中计数值减一，令 R0=05H，即可实现 0.5 秒延时。

时间常数可按下述方法确定：

机器周期=12÷晶振频率=12/(6×10<sup>6</sup>)=2us

设计数初值为 X，则  $(2e+16-X) \times 2 \times 10^6 = 0.1$ ，可求得 X=15535

化为十六进制则 X=3CAFH，故初始值为 TH1=3CH，TL1=AFH

### 2. 初始化程序

包括定时器初始化和中断系统初始化，主要是对 IP、IE、TCON、TMOD 的相应位进行正确的设置，并将时间常数送入定时器中。由于只有定时器中断，IP 便不必设置。

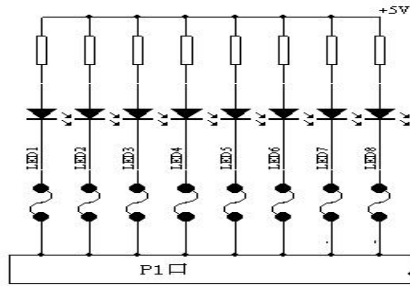
### 3. 设计中断服务程序和主程序

中断服务程序除了要完成计数减一工作外，还要将时间常数重新送入定时器中，为下一次中断做准备。主程序则用来控制发光二极管按要求顺序燃灭。

## 四、实验内容

由 8051 内部定时器 1 按方式 1 工作，即作为 16 位定时器使用，每 0.1 秒钟 T1 溢出中断一次。P1 口的 P1.0~P1.7 分别接发光二极管的 L1~L8。要求编写程序模拟一循环彩灯。彩灯变化花样可自行设计。例程给出的变化花样为：①L1、L2、…L8 依次点亮；②L1、L2、…L8 依次熄灭；③L1、L2、…L8 全亮、全灭。各时序间隔为 0.5 秒。让发光二极管按以上规律循环显示下去。

## 五、实验电路：

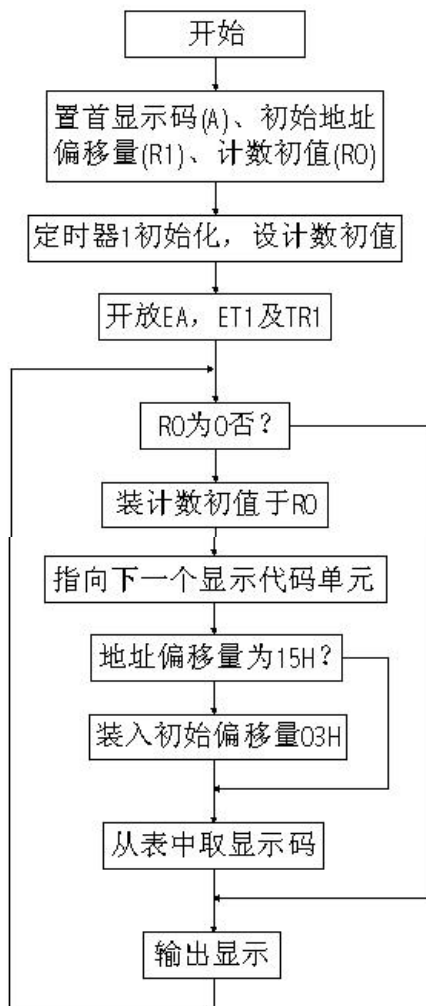


## 六、线路连接和实验步骤：

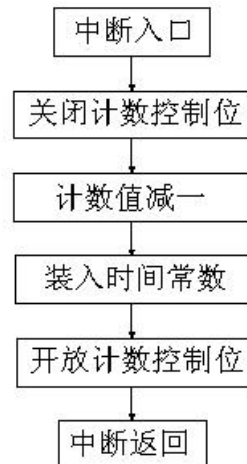
CPU 板的 P1.0~P1.7 分别接发光二极管 L1~L8 即可。

## 七、程序框图：

主程序框图



中断程序框图



## 实验七 8255A 可编程并行接口实验一

### 一、实验目的：

1. 了解 8255A 芯片的结构及编程方法。
2. 掌握通过 8255A 并行口读取开关数据的方法。

### 二、实验环境： 单片机实验箱

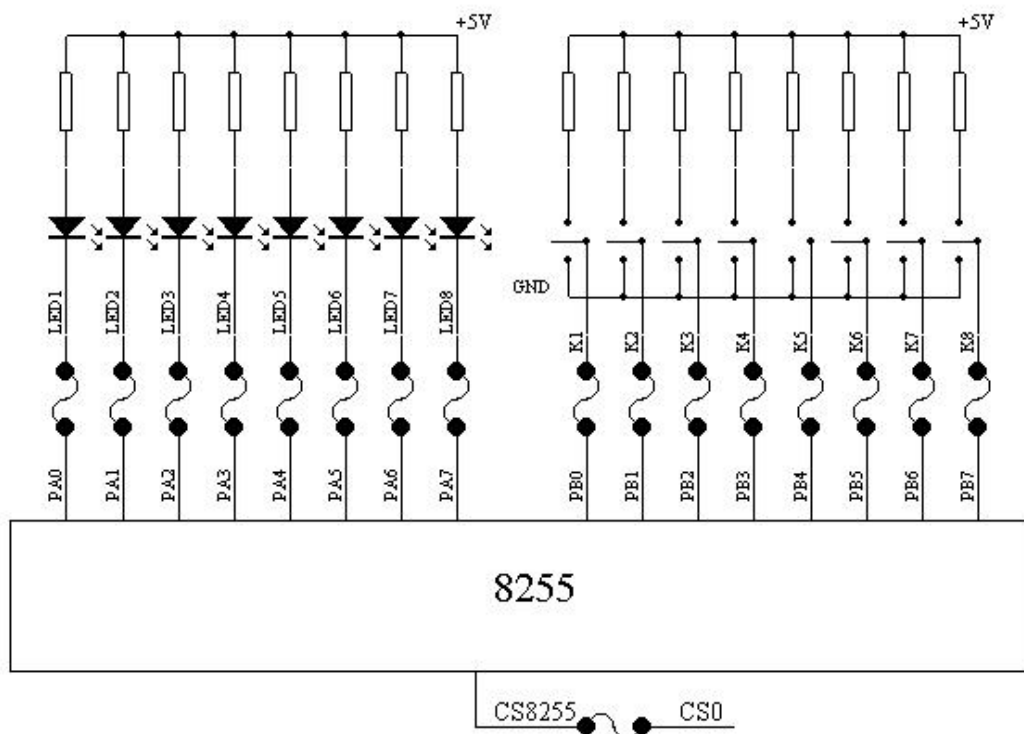
### 三、实验原理：

设置好 8255A 各端口的工作模式。实验中应当使三个端口都工作于方式 0，并使 A 口为输出口，B 口为输入口。

### 四、实验内容：

利用 8255A 可编程并行接口芯片，重复实验四的内容。实验可用 B 通道作为开关量输入，A 通道作为显示输出口。

### 五、实验电路：

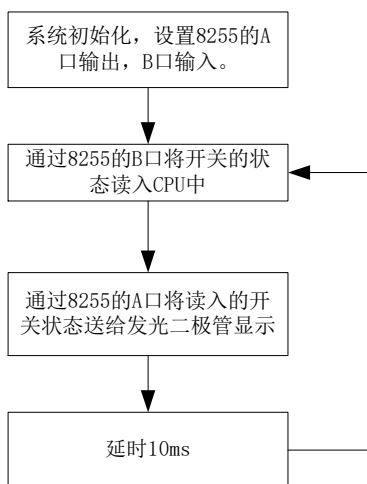


### 六、线路连接和实验步骤：

1. 8255 PORT 单元的 PA0~PA7 接发光二极管 L1~L8；
2. 8255 PORT 单元的 PB0~PB7 接平推开关 K1~K8；

3. 片选信号 8255CS 接 CS0。

## 七、程序框图：



## 实验八 8255A 可编程并行接口实验二

### ——键盘实验

### 一、实验目的：

1. 掌握 8255A 编程原理。
2. 了解键盘电路的工作原理。
3. 掌握键盘接口电路的编程方法。

### 二、实验环境： 单片机实验箱

### 三、实验原理：

1. 识别键的闭合，通常采用行扫描法和行反转法。

行扫描法是使键盘上某一行线为低电平，而其余行接高电平，然后读取列值，如所读列值中某位为低电平，表明有键按下，否则扫描下一行，直到扫完所有行。

本实验例程采用的是行反转法。

行反转法识别键闭合时，要将行线接一并行口，先让它工作于输出方式，将列线也接到一个并行口，先让它工作于输入方式，程序使 CPU 通过输出端口往各行线上全部送低电平，然后读入列线值，如此时有某键被按下，则必定会使某一系列线值为 0。然后，程序对两个并行端口进行方式设置，使行线工作于输入方式，列线工作于输出方式，并将刚才读得的列线值从列线所接的并行端口输出，再读取行线上的输入值，那么，在闭合键所在的行线上的值必定为 0。这样，当一个键被按下时，必定可以读得一对唯一的行线值和列线

值。

2. 程序设计时，要学会灵活地对 8255A 的各端口进行方式设置。

3. 程序设计时，可将各键对应的键值（行线值、列线值）放在一个表中，将要显示的 0~F 字符放在另一个表中，通过查表来确定按下的是哪一个键并正确显示出来。

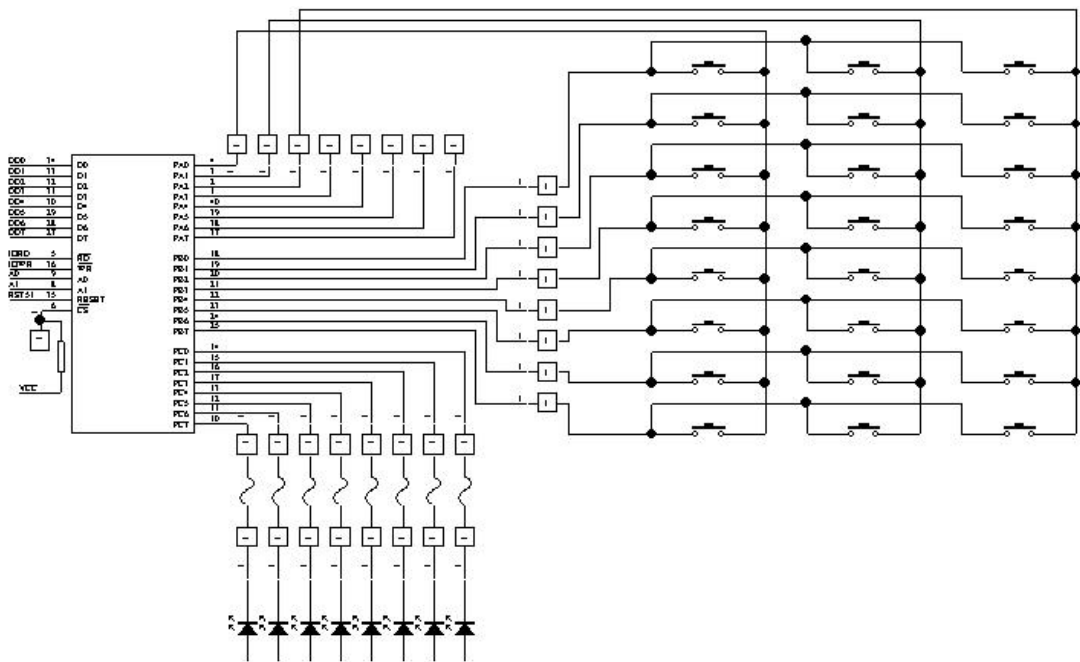
### 实验题目

利用实验箱上的 8255A 可编程并行接口芯片和矩阵键盘，编写程序，做到在键盘上每按一个数字键（0~F），用发光二极管将该代码显示出来。

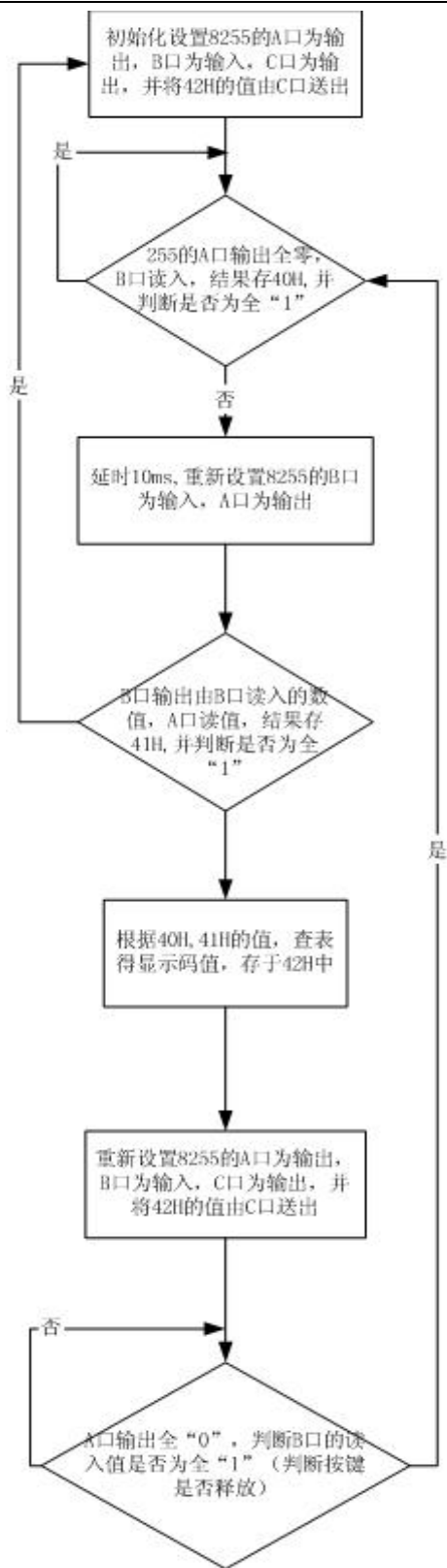
## 四、线路连接和实验步骤：

1. 将实验箱键盘的 RL10~RL17 接 8255 PORT 单元的 PBO~PB7；
2. 实验箱键盘的 KA10~KA12 接 8255 PORT 单元的 PA0~PA2；
3. 8255 PORT 单元的 PC0~PC7 接发光二极管的 L1~L8；
4. 片选信号 8255CS 接 CS0。

## 五、实验电路：



## 六、程序框图





# 实验九 数码显示实验

## 一、实验目的：

1. 进一步掌握定时器的使用和编程方法。
2. 了解七段数码显示数字的原理。
3. 掌握用一个段锁存器，一个位锁存器同时显示多位数字的技术。

## 二、实验环境： 单片机实验箱

## 三、实验原理：

本试验采用动态显示。动态显示就是一位一位地轮流点亮显示器的各个位（扫描）。将 8031CPU 的 P1 口当作一个锁存器使用，74LS273 作为段锁存器。

## 四、实验内容

利用定时器 1 定时中断，控制电子钟走时，利用实验箱上的六个数码管显示分、秒，做成一个电子钟。显示格式为：                      分      秒

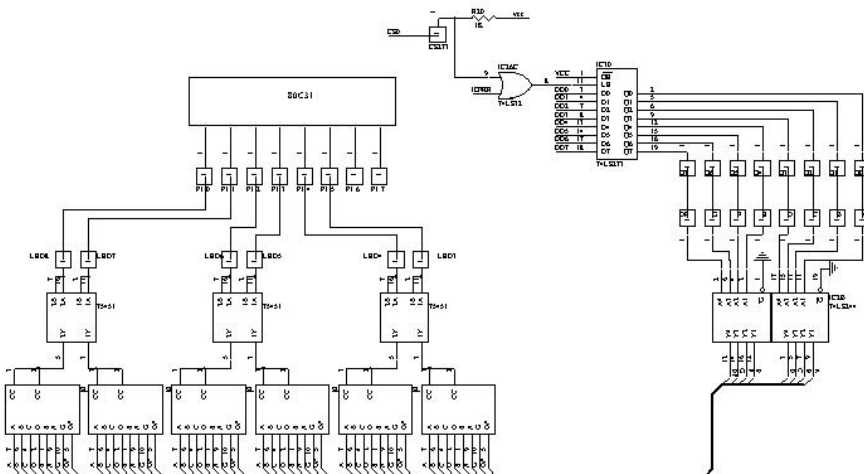
定时时间常数计算方法为：

定时器 1 工作于方式 1，晶振频率为 6MHZ，故预置值 Tx 为：

$$(2e+16-Tx) \times 12 \times 1 / (6 \times 10e+6) = 0.1s$$

$$Tx=15535D=3CAFH, \text{ 故 } TH1=3CH, TL1=AFH$$

## 五、实验电路：

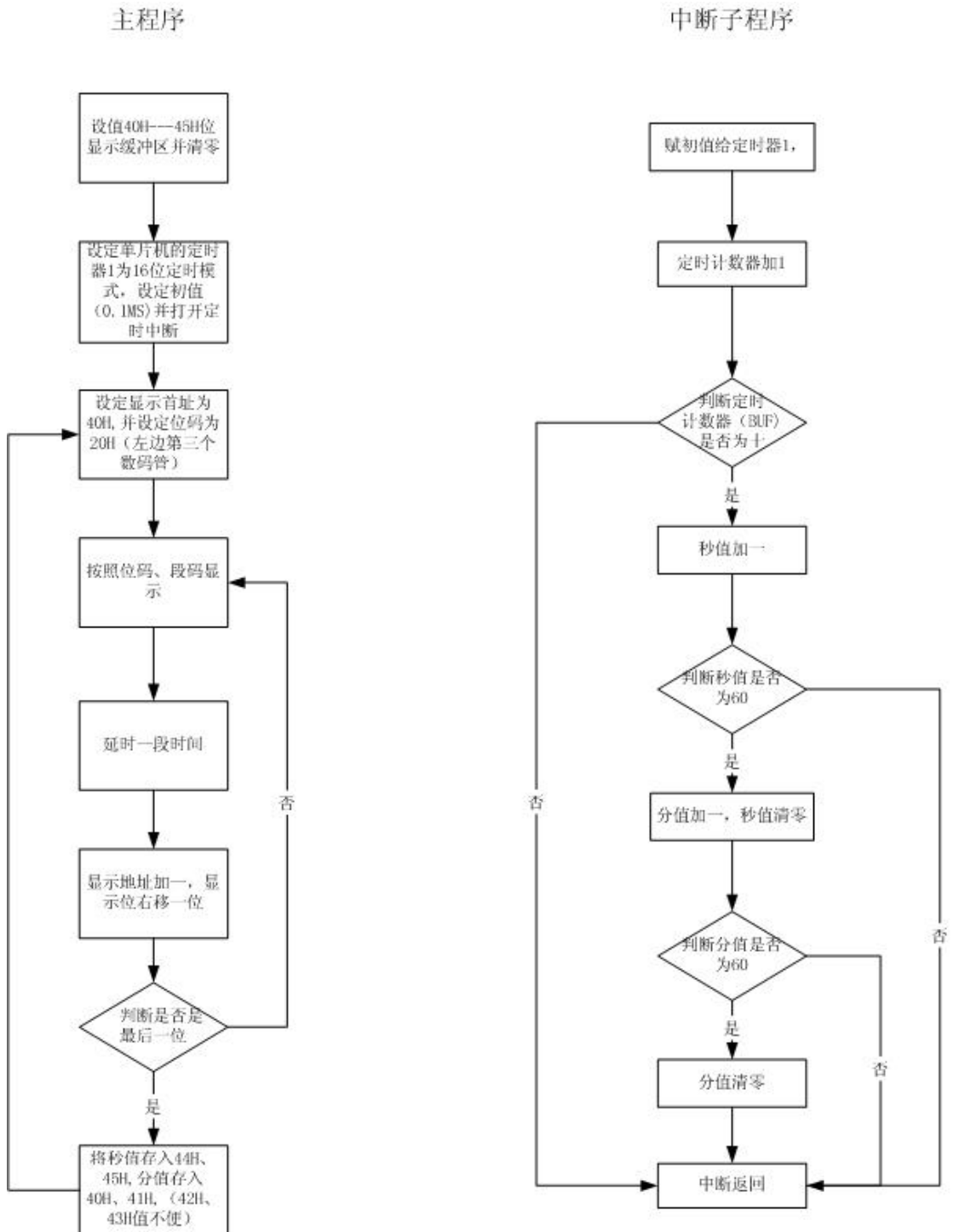


## 六、实验线路连接：

1. 将 CPU 板的 P1.0~P1.5 与实验箱数码管的输入 LED6~LED1 相连；
2. 实验箱 244/273 PORT 单元的 00~07 与数码管单元的 LEDA~LEDDp 相连；
3. 片选信号 CS273 与 CS0 相连；

4. 去掉实验箱上数码管右侧的短路子连接（跳线帽）；

## 七、程序框图



# 实验十 8279 显示接口实验一

## 一、实验目的：

1. 掌握在 8031 系统中扩展 8279 键盘显示接口的方法。
2. 掌握 8279 的工作原理和编程方法。
3. 进一步掌握中断处理程序的编程方法。

## 二、实验环境： 单片机实验箱

## 三、实验原理：

利用 8279 键盘显示接口电路和实验箱上提供的 2 个数码显示，做成一个电子钟。

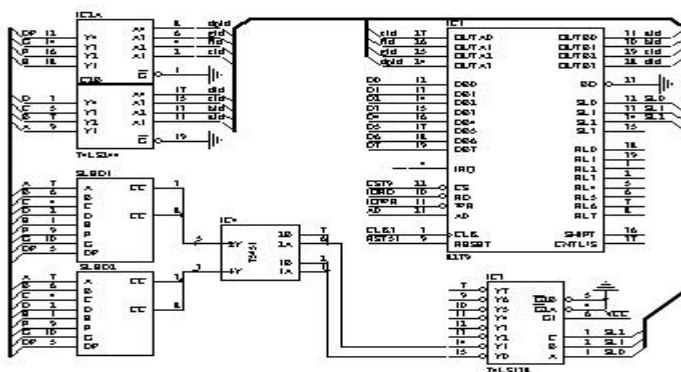
利用 8279 可实现对键盘/显示器的自动扫描，以减轻 CPU 的负担，且具有显示稳定、程序简单、不会出现误动作等特点。本实验利用 8279 实现显示扫描自动化。

8279 操作命令字较多，根据需要来灵活使用，通过本实验可初步熟悉其使用方法。

电子钟做成如下格式：

XX 由左向右分别为十位、个位（秒）。

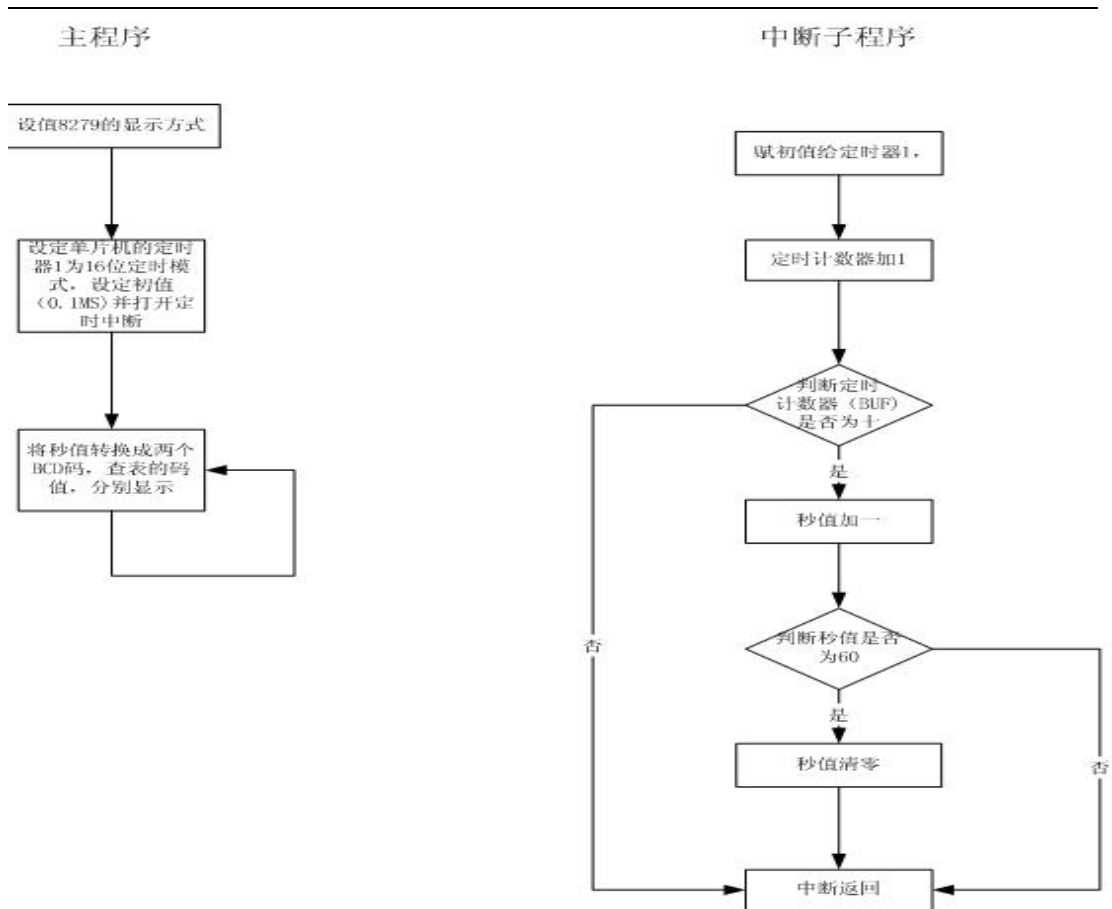
## 四、实验电路：



## 五、线路连接和实验步骤：

本试验不必接线

## 六、程序框图



## 实验十一 8279 键盘显示接口实验二

### 一、实验目的：

1. 进一步了解 8279 键盘、显示电路的编程方法。
2. 进一步了解键盘电路工作原理及编程方法。

### 二、实验环境： 单片机实验箱。

### 三、实验原理：

本实验用到了 8279 的键盘输入部分。键盘部分提供的扫描方式最多可和 64 个按键或传感器阵列相连，能自动消除开关抖动以及对多键同时按下采取保护。

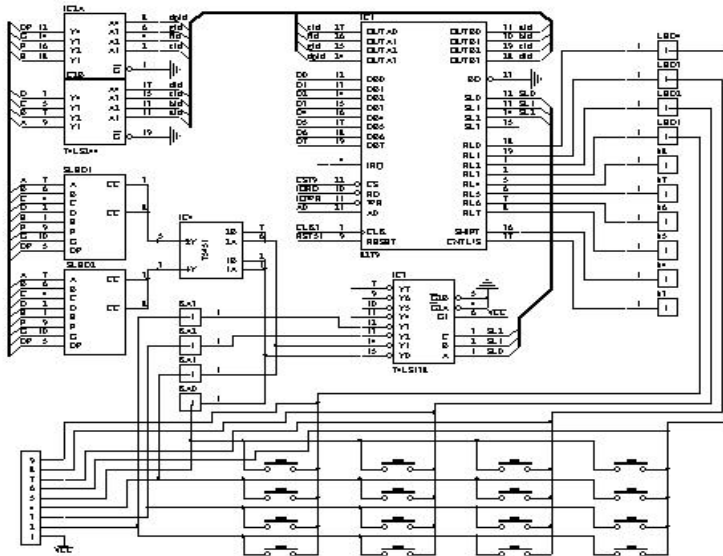
由于键盘扫描由 8279 自动实现，简化了键盘处理程序的设计，因而编程的主要任务是实现对扫描值进行适当处理，以两位十六进制数将扫描码显示在数码管上。

可省略对 8279 进行初始化，因为监控程序对 8279 已经进行了初始化，详见第三章键盘操作说明的 4.5 节。

## 四、实验内容

利用实验箱上提供的 8279，键盘电路，数码显示电路，组成一个键盘分析电路，编写程序，要求在键盘上按动一个键，就将 8279 对此键扫描的扫描码显示在数码管上。

## 五、实验电路：



## 六、实验线路连接和实验步骤：

1. 将键盘的 KA10~KA12 接 LED&KEYBOARD PORT 单元的 KA0~KA2；
2. 键盘的 RL10~RL17 接 LED&KEYBOARD PORT 单元的 RL0~RL7；

## 七、实验提示：

编译全速运行程序后，按某一键，数码管将显示键值，可仔细观察键与键值的对应关系。

## 实验十二 LCD 显示实验

### 一、实验目的：

学习液晶显示的编程方法，了解液晶显示模块的工作原理。

掌握液晶显示模块与单片机的接口方法。

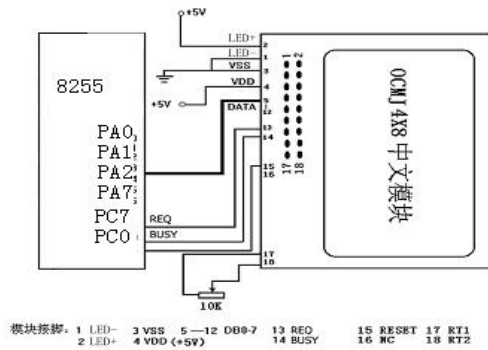
### 二、实验环境及设备

- 1、EL-MUT-III 单片机试验箱
- 4、点阵式 LCD 模块

### 三、实验内容

编程实现在液晶显示屏上显示中文汉字“北京理工达盛科技有限公司”等字符。

### 四、实验原理



### 五、实验步骤

#### 1、实验连线

1. 8255 的 PA0~PA7 接 A/D PORT 单元的 DB0~DB7;
2. 8255 的 PC7 接 A/D PORT 单元的 BUSY;
3. 8255 的 PC0 接 A/D PORT 单元的 REQ;
4. 8255CS 接 CS0;
5. 运行实验程序 T20.asm，观察液晶的显示状态。

### 六、程序框图

