

51测试网从0开始学单片机入门教程

深圳 51 测试网 www.51c51.com 石学军
2004/6/30 整理编撰



前言 单片机是一门实践性非常强的学科,为此我们突破传统思路,全面围绕单片机试验,从简单的流水灯开始,逐步的带领大家从这些简单的几行或者 10 几行的程序,来熟悉和理解单片机的指令。

[返回 51 测试网 网站首页](#) [返回 51 测试网 自学单片机](#)

[返回 51 测试网 系列单片机例子程序](#) [我要参加讨论](#) [如何购买本站产品](#)

目录

第 1 篇 倚天版编程试验套件的硬件说明

第 2 篇 关于 dais 开发环境的操作方法,初学者可用他来编译文件,软件仿真

第 3 篇 倚天版实验部分原理图硬件链接

第 4 篇 入门程序

1 把所有端口的同时置高置低,不断闪烁

2 p1 口 3 路流水灯理解 2 进制数与端口的关系

3 单片机的加法:把 52h+0fch 结果送 p1 口

4 单片机的乘法:把 ff*03h 结果送 p1

5 单片机的二进制加法

6 单片机的两位计数器

7 学习单片机的逻辑运算

8 进一步学习单片机的逻辑运算

9 循环移位指令的流水灯

10 理解熟悉散转结构的程序

11 位操作指令的学习

12 比较指令的学习与 cy 位

13 该程序的功能是小喇叭 1khz 信号

14 按 p3.5 10 次 p1 口 led 按照 2 进制加 1

15 使用定时器实现长时间的延时.

16 中断的响应,p3.3 的小喇叭 1khz 输出

17 p3.2 的键盘数码管显示 0

18 中断的响应,两级中断嵌套

19 顺序程序的结构

20 p1 口的 led 闪烁 10 次后停止子程序的嵌套

倚天版编程试验一体化自学 8051 单片机开发套件

倚天版编程试验一体化开发套件,适合于零基础,或者初中级单片机爱好者使用,并适合于单片机程序员的前期试验.他由一个多功能的 51 烧写器和一个具有扩展功能的多功能试验板组成.同时标配 2 片试验用 AT89S51,串行试验电缆,遥控器,1602 液晶屏.您不需要另外购买其他任何材料即可完成 51 单片机的流水

灯, 小键盘, 数码管动态显示, 数码管静态显示, 计数器, 小喇叭报警器, 唱歌. 串口通信试验, 24c04 扩展, 外中断, 红外遥控, 液晶显示等试验.

烧写器部分采用并口编程, 烧写速度飞快, 实际烧写一个 4K 的程序仅需要 4 秒的时间. 经过我们上千套的测试, 证明其具有烧写的可靠性高, 品质稳定的特点. 该套件的可以直接烧写的芯片包括

AT89C51, AT89C52, AT89C55, AT89S51, AT89S52, AT89S53, AT89C51RC, AT89C55WD, AT89S8252, AT89C1051U, AT89C2051, AT89C4051, D87C51, D87C52 .



套件的试验部分包括了以下的硬件资源

8个Led指示灯 P1口 . 4个小键盘 P3. 2, P. 3. 3, P3. 4, P3. 5 . 1个小喇叭P3. 3 . 1个硬件中断 P3. 2 . 1个硬件复位.

3 个双八位的动态扫描数码管 位选

P2. 1, P2. 2, P2. 3, P2. 4, P2. 5, P2. 6 段位

P0. 0 P0. 7 可作 6 位显示器, 6 位时钟, 计数器

等 , 1 个标准 232 通信接口.

1 个标准的 1602 液晶显示器. 1 个红外线遥控接收头, 1 个扩展储存器 24C02. 同时为了便于用户实验的扩展, 我们还特别设计了 32 个可独立扩展的 IO 口. 可以通过板上的跳线选择将芯片资源外接, 这样即可通过扩展实现不同的功能.

芯片试验也采用零插拔力卡座, 轻松插拔, 不易损坏芯片.

由于液晶显示技术在工业和民用产品中有重要的作用, 也是最重要的人机接口设备. 为此我们特别为这个开发套件设计了一片标准的 1602 (2 行 16 位字符型, HD44780 兼容芯片) 液晶显示屏, 一屏可以显示 2 行 16 个字的英文字母, 与主板采用排针连接, 我们可以利用这个来学习单片机控制液晶显示的原理与方法, 掌握了这个以后, 再学习其他的汉字型或者更大屏幕的液晶显示也不是难事.

与套件配套的标准 ht6221 超薄型红外线遥控器, 一共 28 个有效键位, 用户可根据光盘中的 6221.asm 文件 按遥控器 123456789, 倚天版显示 123456789, 来分析遥控器解码 , 当然您可以根据这个程序和遥控器来开发更多有意义的具有遥控功能的单片机程序, 这样也就避免了在开发过程中走弯路的情况.

安装使用

软件全面兼容 WINDOWS98, WINDOWSME, WINDOWS2000/XP, 全中文操作界面, 易学易用. 配送的软件光盘包括了大量的自学单片机教程, 程序实例, KEIL 正式版, UVW51E 中文版, 单片机的 C 语言入门教程, 单片机以及数字电路的超星格式图书等.

直接点击光盘的驱动程序目录\倚天版驱动\setup.exe, 按照提示点下一步直到完成安装. 此时在桌面生成成为倚天版驱动.exe 和倚天版 win2000/xp.bat 的快捷方式, 如果是 98 系统直接双击倚天版驱动.exe 即可, 如果是 win2000/winxp 系统双击 倚天版 win2000/xp.bat 即可. 也可直接将整个倚天版驱动的目录复制到硬盘, 并去掉所有文件的只读属性, 否则会出现关闭不了软件的情况, 然后, 如是 win98/me, 直接点击倚天版驱动.exe 如果是 win2000/winxp 系统双击 倚天版 win2000/xp.bat

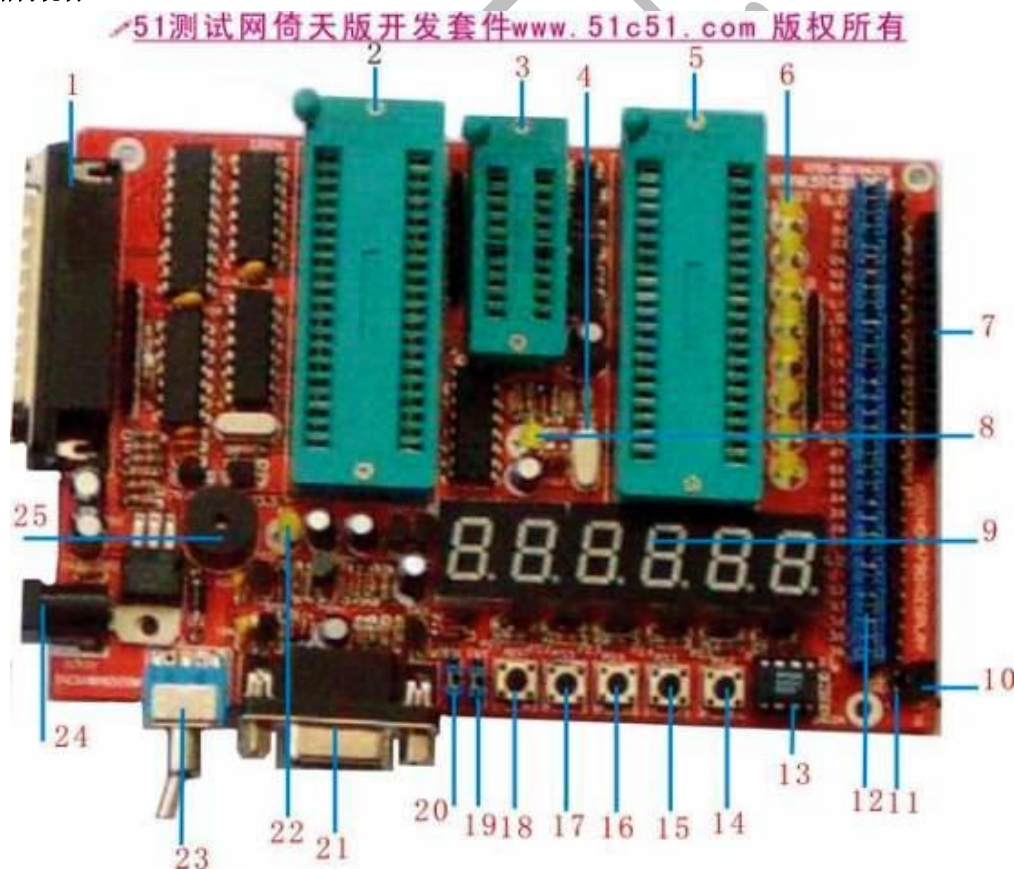
编译软件, 建议安装开发工具中的 dais 开发环境即可. 易学易用, 同时光盘中开发工具配有 keil, uvw51e, madwin 软件, 大家可根据需要安装. 在开发工具目录有这些软件的使用说明的 pdf 文档. 可以用 pdf 阅读器打开.

将随机的并口线插入电脑的打印口. 然后电源插入开发版的电源插孔, 打开电源开关, 运行软件. 程序的主界面如下



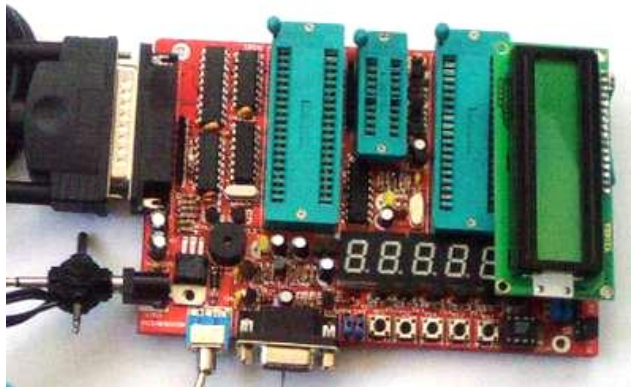
我们以烧写配用的 AT89S51 为例说明使用方法. 假定要写入的文件是 test.hex . 首先, 连接编程器到计算机并口, 然后连接电源. 编程器指示灯亮, 将 89S51 插入 ZIF 插座, 锁紧手柄. 然后启动软件, 鼠标单击在右边的 芯片 下拉框, 选择好芯片类型, 这里选 89S51 . 在 文件 菜单选择 打开 Hex 文件 , 找到 test.hex 文件, 单击 打开 . 窗口中显示文件内容, 右下角显示校验和. 单击 写芯片 按钮, 编程器烧写指示灯亮, 烧写完毕报告结果, 写入之前会自动擦除芯片内容, 写入完毕会自动校验. 如果是需要加密, 可以选择加密级别. 如烧写 2051, 4051 等芯片, 则芯片插入中间较小的烧写卡座.

结构说明



1	并行通信口, 用于连接电缆到电脑打印口下载烧写数据
2	40 脚单片机的烧写卡座, 用于 AT89C51/S51 等单片机的烧写. 注意所有卡座插芯片均为芯片缺口向上 否则会烧毁芯片. 下同

3	20 脚单片机的烧写卡座, 用于 AT89C2051/4051 等单片机的烧写
4	试验部分晶体, 12mhz
5	实验卡座 用于 C51/S51 等兼容 40 脚单片机试验
6	8 个指示灯接在 P1.0-P1.7, 低电平点亮
7	1602 液晶插座.
8	烧写指示灯, 在烧写芯片时他会闪烁, 表示烧写中. 否则通信异常, 请检查并口设置.
9	六位数码管可作计数器, 时钟等
10	红外线接收头, 端口位置为 P3.7, 用于接收红外线遥控信号
11	34pin 排针, 从上至下为 P0.0---P3.7, +5V, GND 当取下相邻跳线时端口资源由排针外接, 可由此作外接扩展试验
12	32 位跳线, 分别为 P0.0---P3.7 的选择跳线, 插上为板上试验, 取下为端口外接
13	24C02 储存器, 插座安装, 用户可自行升级
14	小键盘 P3.2 一端接端口一端接地, 按下端口位低电平
15	小键盘 P3.3 一端接端口一端接地, 按下端口位低电平
16	小键盘 P3.4 一端接端口一端接地, 按下端口位低电平
17	小键盘 P3.5 一端接端口一端接地, 按下端口位低电平
18	复位键 作实验时按此键, 则试验部分复位运行, 用于重启程序.
19	INT 跳线, 默认向下, 需要时跳于向上可将 INT 指示灯的信号引入 P3.5
20	PEN 跳线, 默认向下, 需要时跳于向上可将单片机 29 脚接地 请参阅 89C51 芯片资料
21	标准 232 通信口, 可以做上位机和单片机通信等试验. 参照试验程序
22	Int 指示灯, 开机即会不停闪烁, 兼电源指示. 配合 int 跳线可作中断试验
23	电源开关. 注意打开后指示灯会更亮一些. 如异常可短路电源开关.
24	电源插座. 插入随机专用 13.8V150MA 电源
25	小喇叭, 作唱歌或者报警试验用, 端口为 P3.3



注意事项

- 1 芯片不需要擦除即可直接写入, 此时会自动首先擦除原来的内容. 如果需要擦除芯片, 可以用 清空缓冲区 此时窗口显示全部为 ff, 再执行写入操作即可.
- 2 开发版对于电源要求较高, 如出现错误, 请检查当地电源电压是否过低.
- 3 开发版对电脑并口没有特殊要求, 一般默认值即可, 对于某些笔记本电脑, 可能需要把并口设置为并口 1, 地址为 0378-037f, 如异常请尝试更改并口模式. 对于 WINDOWS XP 系统可能需要更改控制面板\硬件\端口\lpt 并口属性为 使用任何中断 , 请注意 在点击写入时, 中间小卡座下方的 write 指示灯应当闪烁. 否则就是电脑与开发板没有正确的联机, 此时应当尝试更改并口地址设置与中断.
- 5 作液晶显示器实验时须将 P3.0 的跳线线取下. 因为液晶用到了 P3.0 口 按上图方向插入液晶. 液晶功

较大烧写程序时须取下液晶。

6 烧写完成程序请断开电源开关, 将烧好的芯片放入右边试验卡座, 卡好芯片, 打开电源, 按下 REST 键, 程序即可运行。

配套的 HT6221 遥控器的遥控键值如下

键值	丝印	键值	丝印	键值	丝印	键值	丝印
01011111	POWER	01010000	MODE	01011101	STOP	01000100	>
01000001	MIC	00001010	VOL+	01000010	LANG	01000110	BAND
01000111	<<	01001001	SET	00000101	>>	00000110	L/R
00000111	<<	00001001	VOL-	01000101	>>	00000011	OSD
01011100	1	00011110	2	01011110	3	00000010	PBC
01011000	4	00011010	5	01011011	6	00000001	PGM
00011011	7	01011001	8	01011010	9	00011000	GOTO
悬空	0	悬空	10+				

配套硬件包括 倚天版开发套件主板一块, 精美红色PCB 已经调试好, 板载24C02存储器, LM386功放, 声音洪亮, 并行电缆一根, 试验用串行电缆一根, 1602液晶一个, 超薄HT6221红外线遥控器一个, 配套 13.8V150MA专用电源一个, 软件光盘一张 含大量的自学单片机教程, 程序实例, KEIL正式版, UVW51E 中文版, 单片机的C语言入门教程, 单片机以及数字电路的超星格式图书等。

使用说明书一份, 试验用芯片 AT89S51 两片. 关于该套件的更多信息和最新源程序请

点击 www.8951.com www.51c51.com

另外我们准备了部分试验例程, 这些程序都是在该套件调试通过的, 并配有详细的注释说明, 用户可根据这些例程来学习试验其编程方法和思路, 从而快速的掌握单片机编程. 在以后我们也将陆续登出一些有代表性的例子程序的讲解. 由于篇幅所限, 更多的示例程序程序在配套的光盘中。

8led.asm 八个左右跑马灯	1602-A.asm 为驱动 1602 液晶显示一个字母 A
keyshuzi.asm 四个小键盘控制的数字显示	1600logo.asm 1602 液晶的 51 测试网 logo
keyled.asm 四个小键盘控制的花样彩灯	16020-9.asm 为 1602 液晶的全屏显示 0-9
2-8led.asm p1 口八个灯作二进制加法	key1602.asm 小键盘控制 1602 液晶显示器.
3-8LED.asm P1 口八个灯做 3 路跑动显示	keyled.asm 小键盘控制的数字显示与彩灯
dida.asm 小喇叭的警报器.	6221.asm 遥控显示数字的程序.
music.asm 小喇叭唱歌祝你平安	999999.asm 为 0-999999 的计数器
clock.asm 六位数码管时钟程序.	1ed895152.asm 显示 895152 同时 8 个小灯跑动

单片机学习的免费午餐——dais 集成开发环境与软件仿真 石学军

学习单片机之初最先遇到的问题就是不知道怎样把程序生成 hex 文件. 常用的 keil demo 版使用起来设置比较复杂, 并且有 2k 代码的限制, 为此向大家介绍一款 dais 开发环境软件. 该软件全中文界面, 可以实现编译, 硬件仿真, 软件仿真一体化. 并且程序支持汇编/C 混合调试, 中文注释, 全兼容于 WINDOWS98/2000/XP 操作体统。

经过本人一段时间的使用, 深感使用简易, 值得向广大初学单片机的朋友推荐. DAIS 的软件仿真功能强大, 初学者利用他来进行编译软件生成 hex 文件学习和软件仿真, 修改一些小程序很实用。

以下以 DAIS-8052M 仿真器的配套软件为例, 介绍该软件的编译方法和软件仿真使用方法. 点击安装目录下的 SETUP, 一路回车, 即可安装成功, 在桌面上生成图标



双击运行, 出现连接对话框, 此时如已连接仿真器, 点确定即可. 该仿真器默认的波特率为 9600, 串口号则根据计算机的实际情况选择. 一般情况下是串口 1.

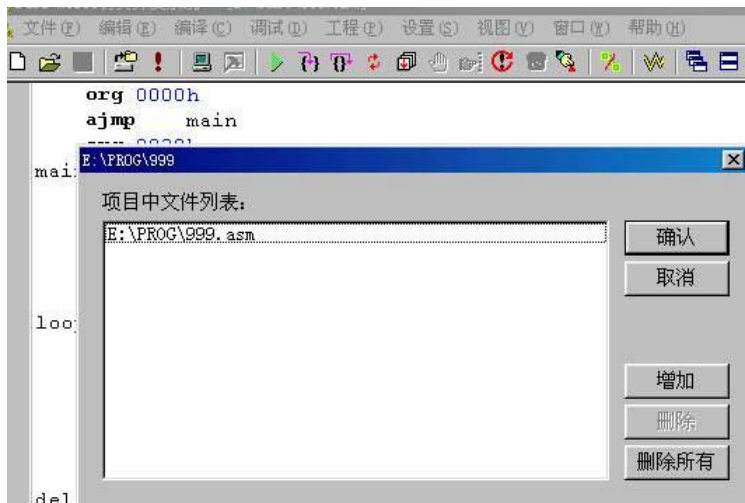


在这里需要注意的是没有仿真器的情况下, 点取消, 我们可以利用该软编译生成 HEX 文件, 并进行软件仿真. 在本篇中, 主要介绍它的软件仿真和编译功能, 如果您采用的 DAIS-8052M 仿真器, 则进入设置仿真模式对其进行如下设置, 否则可能造成仿真器工作异常.

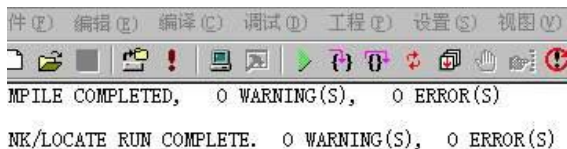


进入后, 点新建文件, 在窗口中输入源程序, 然后把他另存为 ASM, 假定我们把他存为 E:\PROG\999.ASM 具体方法为点文件\另存为即可, 如果在 E 盘没有 PROG 目录, 应当首先建立一个, 当然并不是一定要取这个名字, 只是举例说明而已, 你也可以把他存在 C 盘或者 D 盘. 汇编是底层操作, 需要注意的是不要使用中文的文件夹, 更不能把文件名取为中文名字. 例如你把文件存在 D:\程序\流水灯.ASM 就不行. 同时文件的名称不要超过 8 个英文字符. 没办法, 谁叫咱比老外落后呢 另外也不要系统的盘符设置为 系统盘, 软件盘, 游戏盘 之类的中文盘符.

点 保存 后, 再点菜单的工程 新建 在这里, 我们仍然把工程的名称取为 999 这时将弹出工程的对话框, 此时它是一个新的工程, 我们需要为它增加一个项目文件, 即刚刚保存的 999.ASM 点 增加 然后点 确定



好了, 准备工作完成了, 点击菜单/编译/文件编译, 连接, 装载 或者下图图标的红色感叹号



如没有异常将出现成功的对话框. 这里需要补充一点, 汇编语言对语法要求比较严格, 如果编译不成功请仔细的检查程序本身有没有语法错误. 当然这种情况下, dais 会提示您的第几行出现错误.



此时在我们的 E:\PROG 目录下即生成了以上的八个文件, 其中的 999.ASM 是汇编的源程序, 999.HEX 就是我们所需要的目标代码. 我们可以用编程器将这个文件写进单片机中, 程序就可以运行了. 如果以上操作中已经连接了 DAIS-8052M 仿真器, 把仿真头插入目标板, 直接点图标栏目的绿色小箭头, 即可连续运行. 其他单步运行, 复位, 寄存器窗口, 储存器窗口等大家自己体会一下.

现在我们介绍利用他的软件仿真功能. 即便没有仿真器, 我们也可以利用这个软件完成 51 单片机的大部分仿真工作. 对于初学者来说, 用他来熟悉 8051 单片机的指令系统非常的直观, 如我们用他来学习单片机的加减法.

以下面的一个简单的小例子来说明 25h+36h 等于多少

我们可以首先输入下面的程序

MOV R0, #25H; 将立即数 25 送寄存器 R0

MOV A, #36H; 将立即数 36 送累加器 A

ADD A, R0; a 与 r0 相加

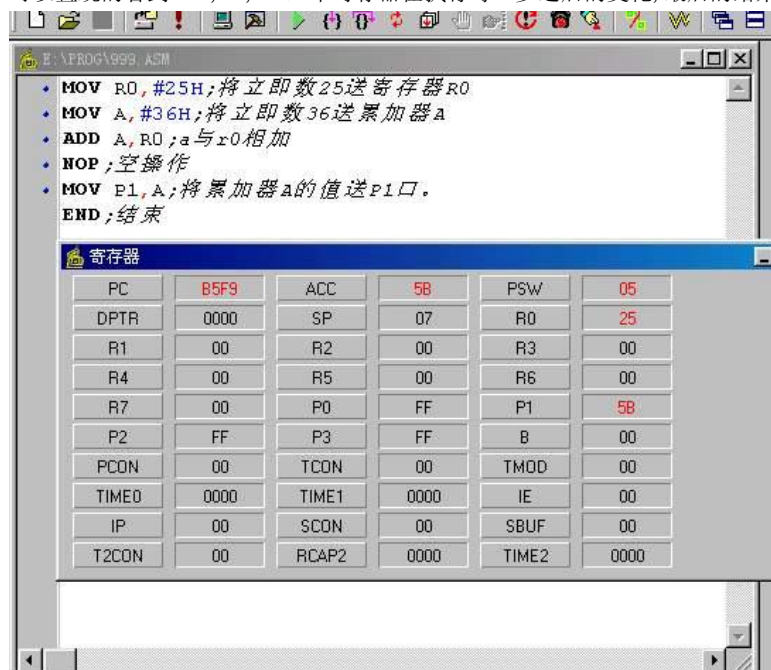
NOP; 空操作

MOV P1, A; 将累加器 A 的值送 P1 口

END; 结束

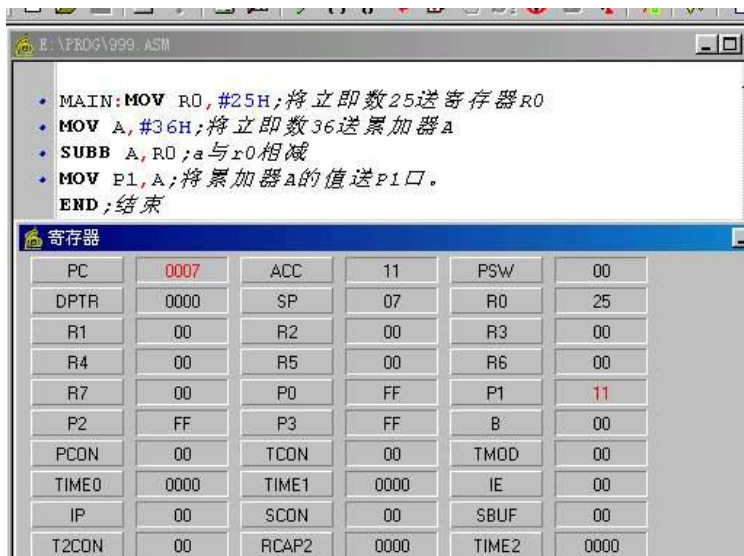
然后把程序按照上面的方法编译, 打开视图\寄存器窗口并单步运行,

可以直观的看到 ACC, P1, R0 三个寄存器在执行每一步之后的变化, 最后的结果如下图



我们可以看到 ACC 的值为 5BH, 这个就是 25H+36H 的结果了. P1 口的值也是 5BH, 这是 MOV P1, A 把 A 送往 P1 口的结果. 同时我们也可以观察到 R0 的值是 25H, 这个是由于程序中我们把 R0 送入 25H 的结果.

同样的我们来实验单片机的减法指令, 例子如下 计算 36H-25H 的值, 把上面的程序第 3 行改为 SUBB A, R0 其余的部分相同, 编译后打开寄存器窗口并单步运行, 我们可以观察到 P1 口和 ACC 的值都变成了 11H, 这就是 36-25h 的值了.



参照以上方法我们可以利用这个软件来熟悉 8051 单片机的其他指令，体会各个寄存器之间的变化，对于初学单片机的朋友很有帮助。

当然，dais 仿真器本身还有很多非常实用的功能，限于篇幅，我们仅仅介绍这一个部分。可以看到它使用很简单，并且具有全中文的操作界面，真正是一款易学易用的开发软件。

对 DAIS 软件感兴趣的朋友可以到本刊的配套光盘中找到该软件或者到作者主页 www.8951.com 下载。

倚天版开发套件试验部分原理图

p1.0-1.7 为 8 个发光管 p3.3 小喇叭 p3.2, p3.3 p3.4 p3.5 小键盘 p2.1p2.2p2.3p2.4p2.5p2.6 数码管位选, p0 数码管段选

28h, 7eh, 0a2h, 62h, 74h, 61h, 21h, 7ah, 20h, 60h

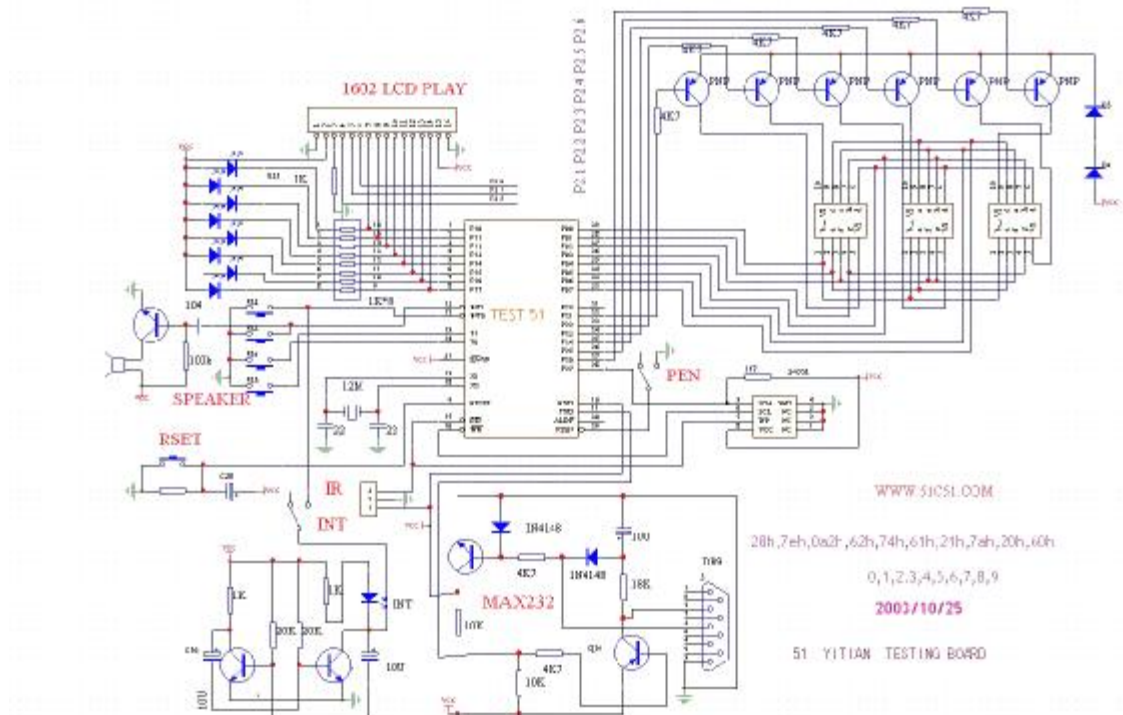
; 0 1 2 3 4 5 6 7 8 9 数码管的代码表

增加了红外线端口, p3.7

增加了 1602 液晶接口 1gnd 2vcc 3vol (对比度调整) 4p3.0 (RS 数据命令选择)

5p3.1 R/W 读写选择 6p3.5 E, 使能信号

7---14 p1.0-p1.7 数据总线



1 该程序的功能是把所有端口的同时置高置低, 不断闪烁.

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程. www.51c51.com www.8951.com

; 知识要点 理解单片机的运行过程, 理解延时程序,

ORG 0000H;程序开始

LJMP MAIN;

ORG 030H;

MAIN:MOV P0, #00H; 把 p0, p1, p2, p3 全部置低电平

MOV P1, #00H;

MOV P2, #00H;

MOV P3, #00H;

ACALL DEL; 延时

MOV P0, #0FFH; 把 p0, p1, p2, p3 全部置高电平

MOV P1, #0FFH;

MOV P2, #0FFH;

MOV P3, #0FFH;

ACALL DEL; 延时

AJMP MAIN; 重新开始

ORG 0200H; 从 200h 开始存放

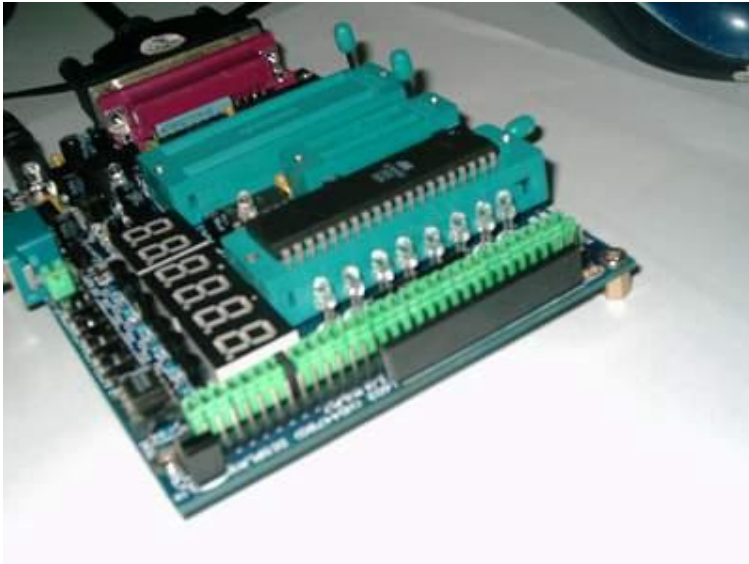
DEL:MOV R5, #04H; 延时子程序

F3:MOV R6, #0FFH;

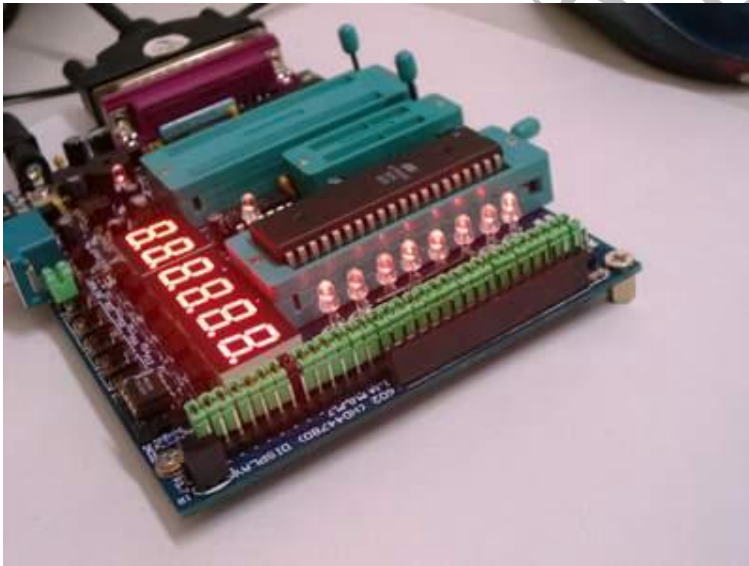
F2:MOV R7, #0FFH;

F1: DJNZ R7, F1;

```
DJNZ R6, F2;  
DJNZ R5, F3;  
RET; 子程序结束  
END
```



程序在倚天版运行的状态 1, 全部不亮



程序在倚天版运行的状态 2, 全部亮

2 该程序的功能是 p1 口 3 路流水灯

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

; 知识要点 理解 2 进制数与端口的关系

ORG 0000H; 程序开始

LJMP MAIN;

ORG 030H;

```
MAIN:MOV P1,#0DBH;11011011--零为亮
ACALL DEL;延时
MOV P1,#06DH;01101101
ACALL DEL;延时
MOV P1,#0B6H;10110110
ACALL DEL;延时
AJMP MAIN;
DEL: MOV R7,#0FFH;延时子程序
DEL1:MOV R6,#0FFH;
DEL2: DJNZ R6,DEL2;
DJNZ R7,DEL1;
RET;
END
```



程序在倚天版运行的状态 1, 跑马灯的瞬间 11011011 靠近数码管的一边为高位, 1 为灭 0 为灭



程序在倚天版运行的状态 2, 跑马灯的瞬间 10110110 靠近数码管的一边为高位, 1 为亮 0 为灭

;3 该程序的功能是单片机的加法:把 52h+0fch 结果送 p1 口

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解 2 进制数的加法

;运行结果为 p1.7 到 p1.0 为亮灭亮亮灭灭灭亮就是二进制 01001110

;注意是从高向低排列, 同时该数已经超过 256, 所以进位 cy=1

ORG 0000H;程序开始

LJMP MAIN;

ORG 030H;

MAIN:ACALL DEL;

MOV A, #052H;寄存器 a 送 52h

MOV R0, #0FCH;寄存器 r0 送 0fch

ADD A, R0;相加

NOP;空操作

MOV P1, A;结果送 p1 口

DEL:MOV R7, #0FFH;;延时子程序

DEL1:MOV R6, #0FFH;

DEL2:MOV R5, #01FH;

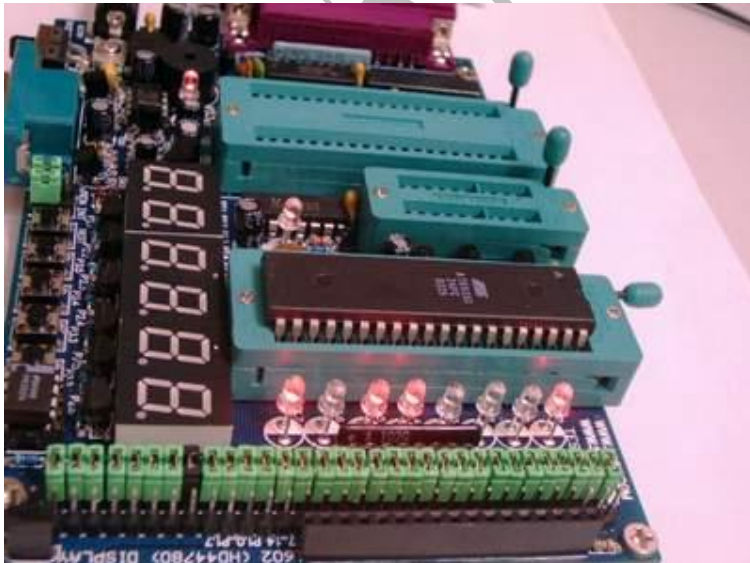
DEL3: DJNZ R5, DEL3;

DJNZ R6, DEL2;

DJNZ R7, DEL1;

RET;延时子程序返回

END



程序在倚天版运行的状态, 运行结果为 p1.7 到 p1.0 为亮灭亮亮灭灭灭亮就是二进制 01001110

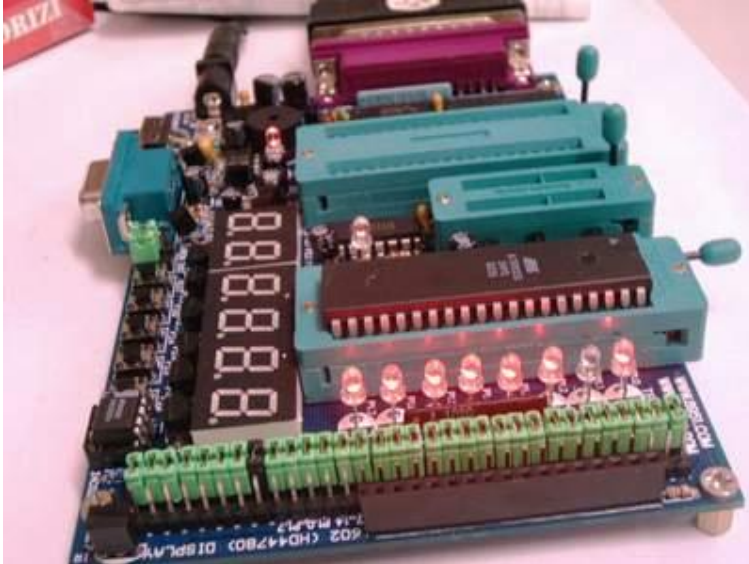
靠近数码管的一边为高位, 1 为亮 0 为灭

;4 该程序的功能是单片机的乘法:把 ff*03h 结果送 p1 口先送高 8 位再送低 8 位

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com
;知识要点 理解 2 进制数的乘法
;运行结果先输出 00000010 ,然后输出 11111101,就是 1011111101 换算后等于 10 进制 765,就是两数相乘的结果
;注意是从高向低排列,同时该数已经超过 256, 所以进位 cy=1
ORG 0000H;程序开始
LJMP MAIN;
ORG 030H;
MAIN:ACALL DEL;
MOV A, #0FFH;寄存器 a 送 0ff
MOV B, #03H;寄存器 b 送 03h
MUL AB; 相乘
MOV P1, b ;高 8 位输出结果是 00000010
acall del
MOV P1, a;低 8 位输出结果是 11111101
DEL:MOV R7, #0FFH;延时子程序
DEL1:MOV R6, #0FFH;
DEL2:MOV R5, #01FH;
DEL3: DJNZ R5, DEL3;
DJNZ R6, DEL2;
DJNZ R7, DEL1;
RET;
END



程序在倚天版运行的状态 11111101



程序在倚天版运行的状态 00000010

5 该程序的功能是单片机的二进制加法

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

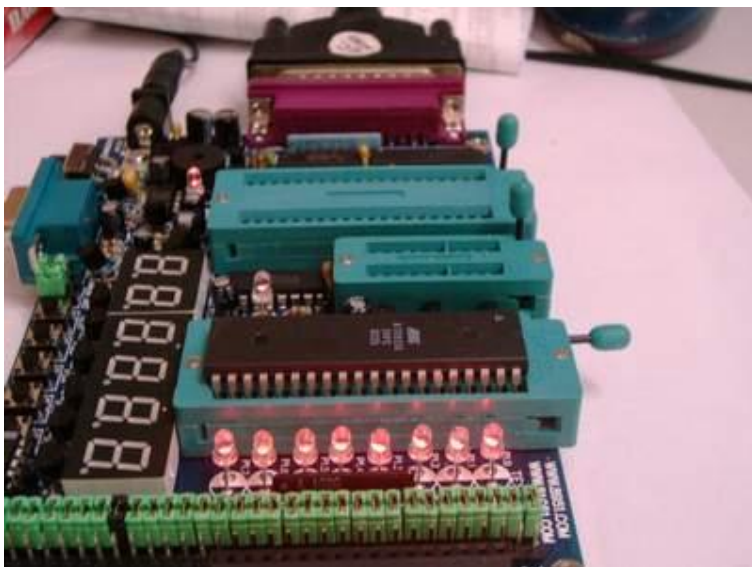
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解 2 进制数的加法

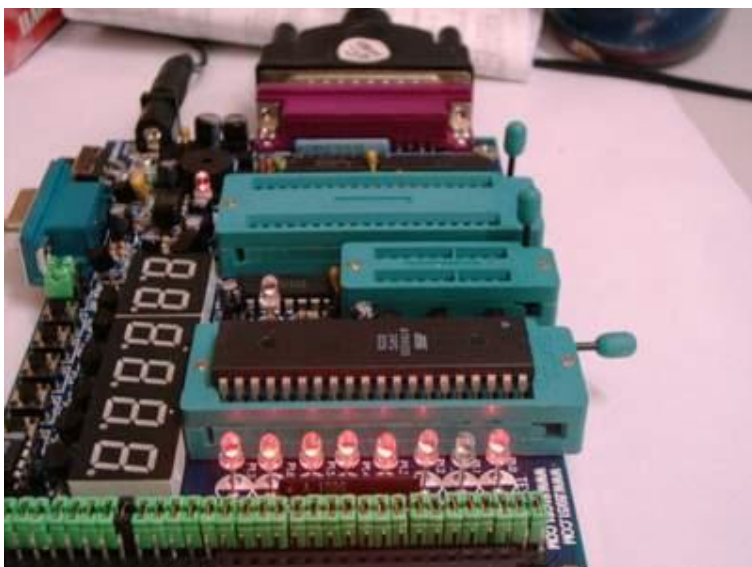
;运行结果先输出 00000000 (全亮), 然后输出 00000001, 然后 00000010, 直到 11111111

;从 P1 口的 8 个灯看二进制的 0, 1, 2, 3,

```
ORG 0000H;程序开始
LJMP MAIN;
ORG 030H;
MAIN:MOV A, #00H;先送 0
PLAY:MOV P1, A;输出
ACALL DEL;延时
INC A;加 1
AJMP PLAY;再输出
DEL:MOV R7, #0FFH;延时子程序
DEL1:MOV R6, #0FFH;
DEL2:MOV R5, #01FH;
DEL3:  DJNZ R5, DEL3;
DJNZ R6, DEL2;
DJNZ R7, DEL1;
RET;
END
```



程序在倚天版运行的状态 00000000



程序在倚天版运行的状态 00000010



程序在倚天版运行的状态 00000100

6 该程序的功能是单片机的两位计数器

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解数的概念

;运行结果 数码管从 00 一直加到 99

;ANL 0FH 就是把数据与 00001111 相与那么遇 1 得原来的数遇 0 则为 0, 结果是前面的 4 位变 0, 后面的 4 位保留

;P2.1 , P2.2 分别是数码管控制位. CLR 则点亮

ORG 0000H;程序开始

LJMP MAIN;

ORG 030H;

MAIN:MOV 20H, #00H;20H 单元送 00

MOV A, 20H;

GOON:CLR C;进位 CY 置 0 这里的作用是当数字超过 9 要求进位时屏蔽进位, 意思永不进位

ANL A, #0FH;屏蔽 A 中的高 4 位

MOV DPTR, #TAB;查表

MOVC A, @A+DPTR;

PP:CLR P2.2;打开个位数码管

MOV P0, A;输出到 P0

ACALL DEL;延时以便观察结果

SETB P2.2;关断显示

MOV A, 20H;20H 再送 A

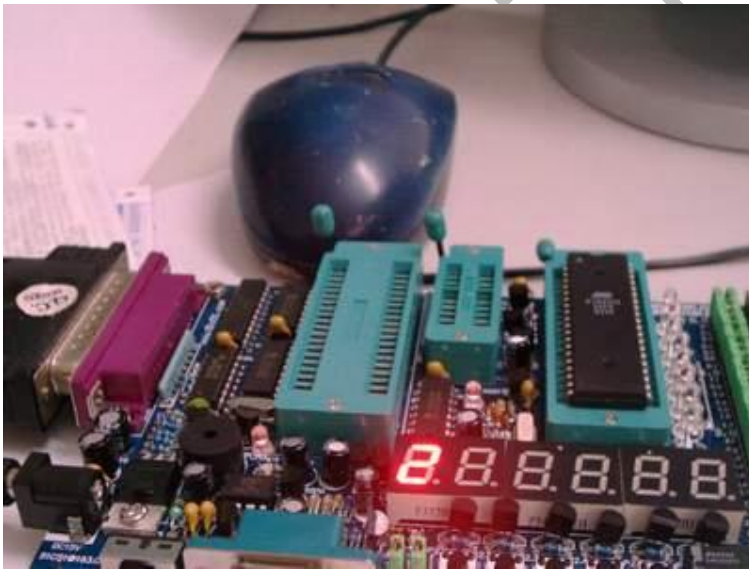
SWAP A;交换 A 的高低 4 位

ANL A, #0FH;屏蔽 A 中的高 4 位

MOVC A, @A+DPTR;查表

CLR P2.1;打开十位数码管

```
MOV P0, A; 输出到 P0
ACALL DEL ; 延时
SETB P2.1 ; 关断显示
MOV A, 20H; 20H 送 A
INC A; 加 1
DA A; 2-10 进制调整
MOV 20H, A; 调整后送 20H
AJMP GOON; 重新开始
DEL: MOV R7, #0FFH; 延时子程序
DEL1: MOV R6, #01FH;
DEL2: MOV R5, #01FH;
DEL3: DJNZ R5, DEL3;
DJNZ R6, DEL2;
DJNZ R7, DEL1;
RET;
ORG 0100H; 从 100H 开始存放数据表 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
TAB: db 28h, 7eh, 0a2h, 62h, 74h, 61h, 21h, 7ah, 20h, 60h
; 0 1 2 3 4 5 6 7 8 9
END
```



程序在倚天版运行的状态瞬间第一个数码管显示 2



程序在倚天版运行的状态瞬间第数码管显示 34

7 该程序的功能是学习单片机的逻辑运算

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解熟悉单片机的逻辑运算

;运行结果 P1 口输出 4 次计算的数值, 注意 1 为灭 0 为亮, 从 1.7 到 1.0 排列

;与的概念: 有 0 为 0 全 1 出 1. 或的概念: 有 1 为 1 全 0 出 0. 取反的概念遇 1 为 0 遇 0 为 1

;异或的概念: 相同为 0 不同为 1

ORG 0000H;

LJMP MAIN; 程序开始

ORG 030H;

MAIN: MOV A, #03CH; A 送 03C, 就是 00111100

MOV R0, #0AAH; R0 送 0AA 就是 10101010

CPL A; A 的内容取反

MOV P1, A; 输出到 P1 结果为 11000011

ACALL DEL; 调用延时便于观察

MOV P1, #0FFH; 关断显示

MOV A, #0C3H; A 的值重新载入

ANL A, R0; A 与 R0 相与

MOV P1, A; 输出到 P1 结果为 10000010

ACALL DEL; 调用延时便于观察

MOV P1, #0FFH

MOV A, #0C3H; A 的值重新载入

ORL A, R0; A 与 R0 相或结果为 11101011

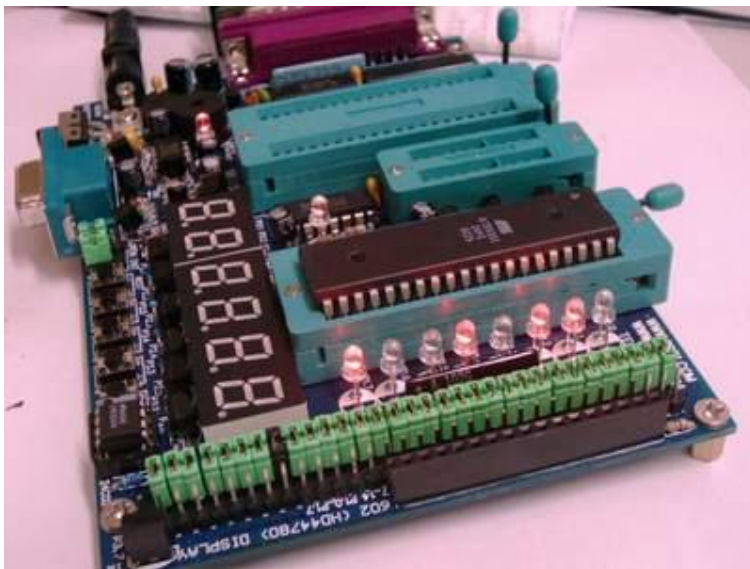
MOV P1, A; 输出到 P1

ACALL DEL; 调用延时便于观察

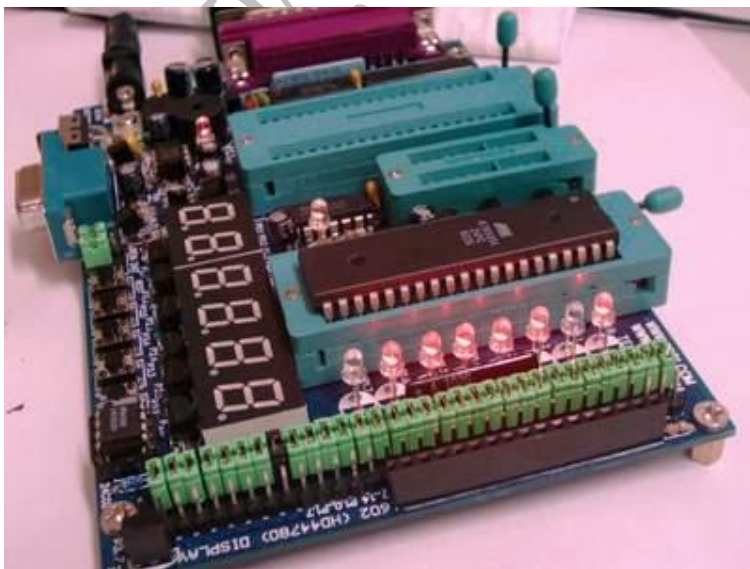
MOV P1, #0FFH; 关断显示

MOV A, #0C3H; A 的值重新载入

```
XRL A,R0;;A 与 R0 相异或结果为 01101001
MOV P1,A;;输出到 P1
ACALL DEL;
MOV P1,#0FFH;;关断显示
AJMP MAIN;重新开始
DEL: MOV R7,#0FFH;延时子程序
DEL1:MOV R6,#0FFH;
DEL2:MOV R5,#01FH;
DEL3: DJNZ R5,DEL3;
      DJNZ R6,DEL2;
      DJNZ R7,DEL1;
      RET;
      END
```



程序在倚天版运行的状态 01101001



程序在倚天版运行的状态 10000010

8 该程序的功能是进一步学习单片机的逻辑运算

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解熟悉单片机的逻辑运算

;运行结果 P1 口输出结果为 01100000, 注意 1 为灭 0 为亮, 从 1.7 到 1.0 排列

;ANL 0FH 就是把数据与 00001111 相与那么遇 1 得原来的数遇 0 则为 0, 结果是前面的 4 位变 0, 后面的 4 位保留

ORG 0000H; 程序开始

LJMP MAIN;

ORG 030H;

MAIN:MOV A, #59H; A 中送 59H 这个数 01011001

ANL A, #0FH; 59H 和 00001111 相与就是保留后 4 位结果为 1001

SWAP A; A 中的高低字节交换结果为 10010000

ANL P1, #0FH; P1 口的内容与 A 相与就是清除高 4 位注意 P1 本来为 11111111 即 FF 此时变为 00001111

ORL P1, A; P1 的内容与相或 10010000 和 00001111 相或结果为 01100000

ACALL DEL; 调用延时以便观察

AJMP MAIN; 重新开始

DEL: MOV R7, #0FFH; 延时子程序

DEL1: MOV R6, #0FFH;

DEL2: MOV R5, #01FH;

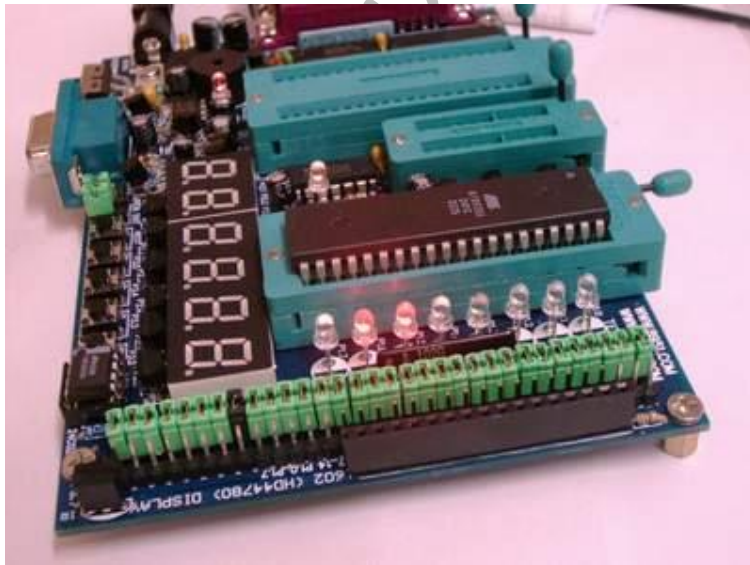
DEL3: DJNZ R5, DEL3;

DJNZ R6, DEL2;

DJNZ R7, DEL1;

RET;

END



运行结果 P1 口输出结果为 01100000, 注意 1 为灭 0 为亮, 从 1.7 到 1.0 排列

9 该程序的功能是用循环移位指令的流水灯

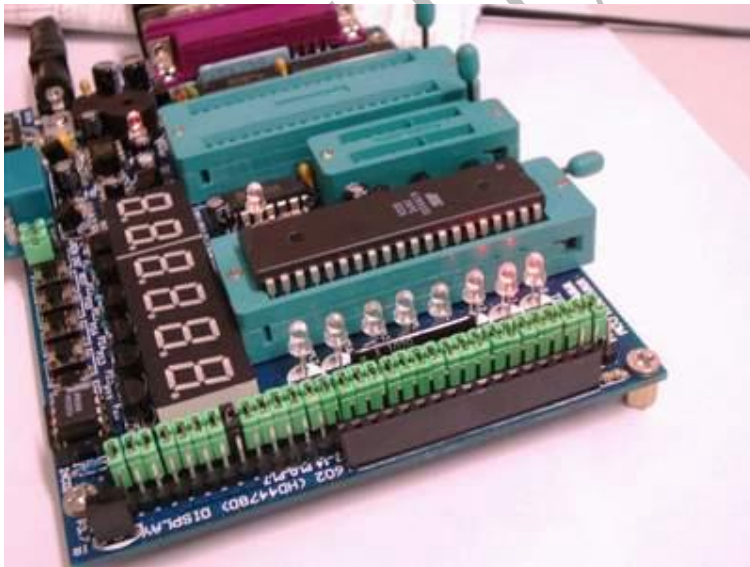
;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

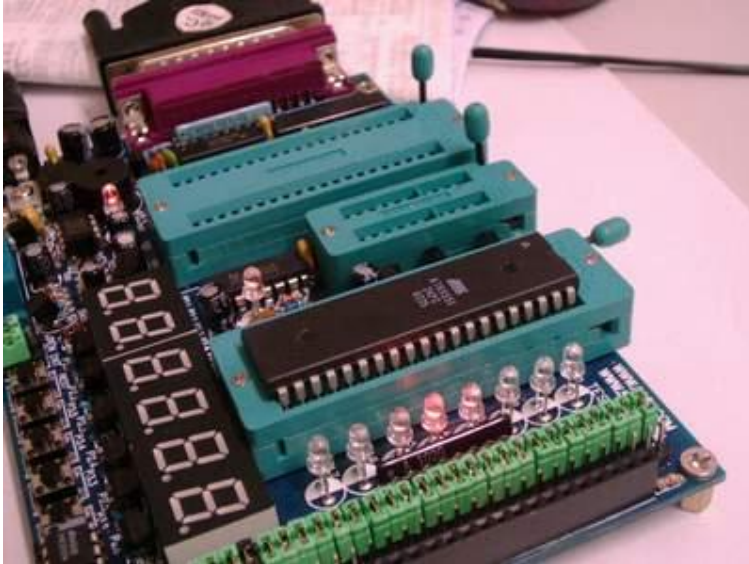
;知识要点 理解熟悉移位指令

;运行结果 P1 口 8 个灯循环点亮

```
ORG 0000H;程序开始
    LJMP MAIN;
    ORG 030H;
MAIN:MOV R0, #08H;把 8 送入 R0 中,
    MOV A, #0FEH;把 FE 送入 A 中, 即 11111110
PLAY:RR A;右移动 1 位
    MOV P1, A;输出显示
    ACALL DEL;延时
    DJNZ R0, PLAY;一共移动 8 次
    AJMP MAIN;重新开始
DEL: MOV R7, #0FH;延时子程序
DEL1:MOV R6, #0FFH;
DEL2:MOV R5, #01FH;
DEL3: DJNZ R5, DEL3;
    DJNZ R6, DEL2;
    DJNZ R7, DEL1;
    RET;
END
```



运行结果流水灯的瞬间 注意由于照相机的延时, 看到的两个灯亮实际为一个灯亮



运行结果流水灯的瞬间 注意由于照相机的延时,看到的两个灯亮实际为一个灯亮

10 该程序的功能是用按键计数器

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 理解熟悉散转结构的程序

;运行结果 按 P3.2 或 3.3 (低 4 位的任意一个键盘) 第一位数码管加 1 直到 9 循环

;开始显示 0 把 P3 送 0FH, 判断有没有按键, 没有则反复循环, 有按 R0 加 1, 如到 10

;那么则清除, 最后根据 R0 的值散转.

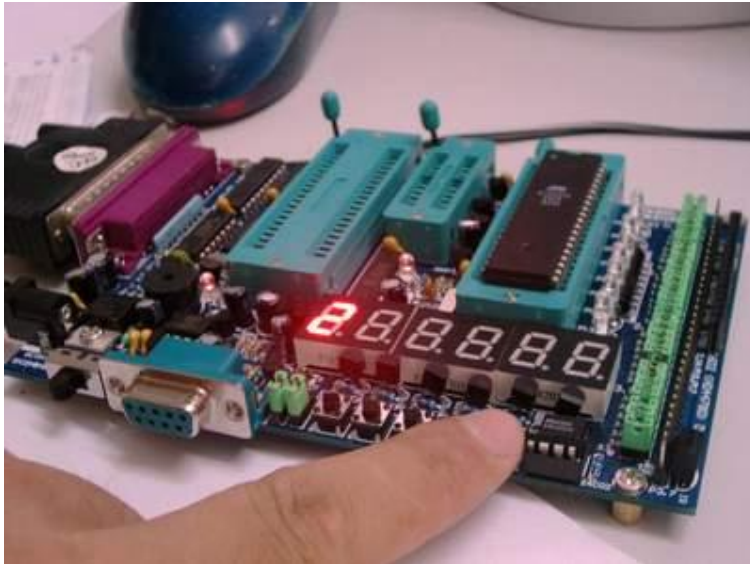
```

ORG 0000H;
    LJMP MAIN; 程序开始
    ORG 030H;
MAIN:  MOV P0, #028H; 数码管显示 0
      clr p2.1    ; 打开第一个数码管
      MOV R0, #00H; r0 送 0fh
ST:    MOV P3, #0FH; p3 送 0fh
      MOV A, P3;
      CJNE A, #0FH, F1; 如果 p3 口的值等于 0f, 那么按键没有按,
      ACALL DEL;
      AJMP ST; 继续检测
F1:    ACALL DEL;
      CJNE A, #0FH, F2; 再次判断有没有按
      AJMP ST; 继续检测
F2:    INC R0; r0 加 1
      CJNE R0, #0AH, F3; 如果 r0 等于 10 转道 f3; 0AH 就是 10, 到这里转移重新
      MOV R0, #00H;
F3:    MOV DPTR, #JPTAB; 取地址
      MOV A, R0;
      CLR C; 清除进位

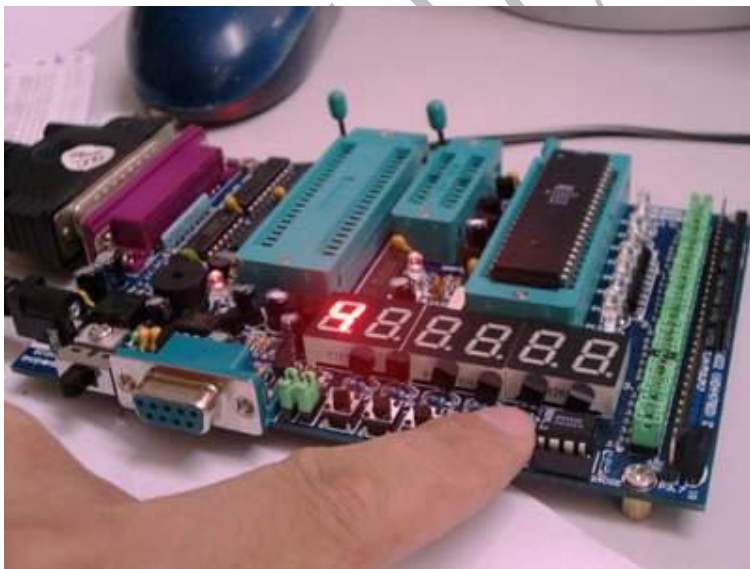
```

```
RLC A;a 的内容右移
JNC NADD;判断有没有进位, 有向下无到 NDDA
INC DPH;有进位 DPH 加 1
NADD:  JMP @A+DPTR;根据 A 的内跳转到 PR1, PR2...
JPTAB:  NOP;
        NOP;
        AJMP PR1;
        AJMP PR2;
        AJMP PR3;
        AJMP PR4;
        AJMP PR5;
        AJMP PR6;
        AJMP PR7;
        AJMP PR8;
        AJMP PR9;
DEL:    MOV R7, #04H;延时子程序
DEL1:   MOV R6, #0FFH;
DEL2:   MOV R5, #01FH;
DEL3:   DJNZ R5, DEL3;
        DJNZ R6, DEL2;
        DJNZ R7, DEL1;
        RET;
PR1:    MOV P0, #7eH;显示 1
        ACALL DEL;
        AJMP ST;
PR2:    MOV P0, #0a2H;显示 2
        ACALL DEL;
        AJMP ST;
PR3:    MOV P0, #62H;显示 3
        ACALL DEL;
        AJMP ST;
PR4:    MOV P0, #74H;显示 4
        ACALL DEL;
        AJMP ST;
PR5:    MOV P0, #61H;显示 5
        ACALL DEL;
        AJMP ST;
PR6:    MOV P0, #21H;
        ACALL DEL;
        AJMP ST;
PR7:    MOV P0, #7aH;
        ACALL DEL;
        AJMP ST;
PR8:    MOV P0, #20H;
```

```
ACALL DEL;  
AJMP ST;  
PR9:  MOV P0, #60H;  
ACALL DEL;  
AJMP ST;  
END
```



程序在倚天版运行的状态按键显示数字 2



程序在倚天版运行的状态按键显示数字 4

11 该程序的功能是用按键控制灯

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

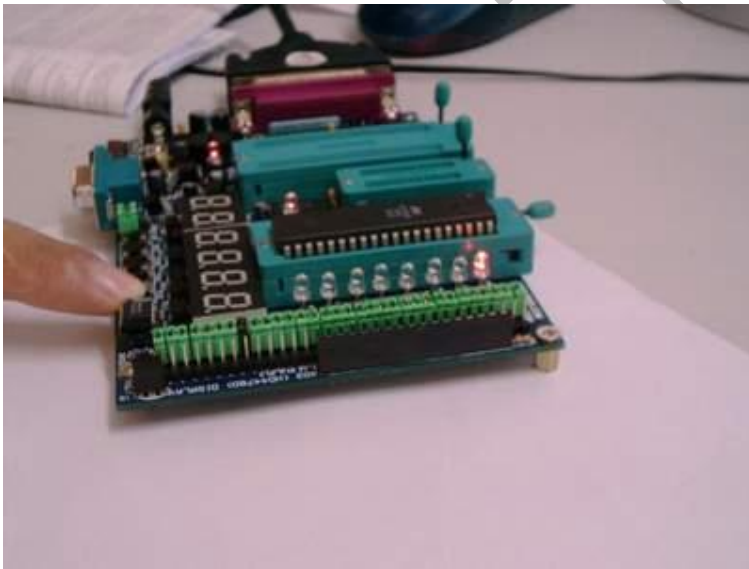
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 位操作指令的学习

;运行结果 按 P3.2 那么 p1.0 的灯亮
 ; p3.2 接一个按键到地, p1.0 led. 程序把 p3.2 的状态送给 p1.0
 ;位地址有多种表达方式, 如直接地址 p2.2 字节地址 21h.5 寄存器 acc.7 位定义如 rs0 伪指令定义如 fi
 bit psw.1

```

ORG 0000H;程序开始
    LJMP MAIN;
    ORG 030H;
MAIN:
    MOV C, P3.2; 把 p3.2 的状态送入 c
    MOV P1.0, C; 把 c 的状态送 p1.0
    ACALL DEL; 调用延时
    AJMP MAIN; 反复循环
DEL:  MOV R7, #0FFH; 延时程序
DEL1: MOV R6, #0FFH;
DEL2: DJNZ R6, DEL2;
      DJNZ R7, DEL1;
      RET;
      END
  
```



程序在倚天版运行的状态按键数码管 p10 亮

12 该程序的功能是比较两个数的大小

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492
 ;51 测试网系列自学单片机例程, www.51c51.com www.8951.com
 ;知识要点 比较指令的学习与 cy 位
 ;开始初始化为 ff. 大于 fe, 那么显示 1. 如果用导线把 p1.1, \...p1.7
 ;短路到地, p1 将小于 fe, 显示 0, 如果把 p1.0 短路到地 p1=fe 则继续检测无显示.
 ;该程序示例了比较指令. 同时有助于理解数的概念.

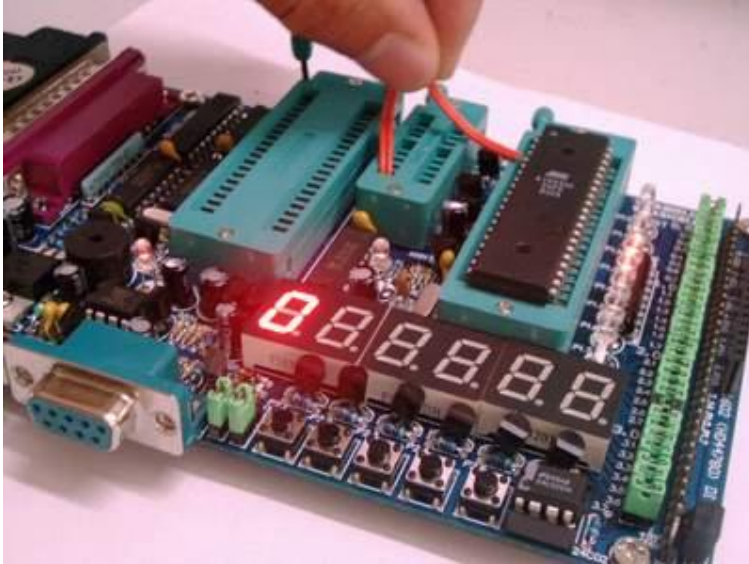
```

ORG 0000H;程序开始
  
```

```
LJMP MAIN;
ORG 030H;
MAIN:  MOV P0, #0FFH; 初始化 p0
      MOV r0, #0FeH; r0 送立即数
      clr p2.1; 打开第一个数码管
      MOV A, r0; r0 送到 a
      CJNE A, P1, L1; pi 与 r0 比较如相等则顺序如果不等则跳到 11. 如大 cy=1 否则为 0
      ACALL DEL; 调用延时以便观察
      AJMP MAIN; 重新开始
L1:    JC L2; cy 如果为 1 则跳到 12 否则顺序
      mov p0, #28h; 如果为 0 显示 0
      ACALL DEL;
      AJMP MAIN; 继续检测
L2:    mov p0, #7eh; 如果为 1 显示 1
      ACALL DEL;
      AJMP MAIN; 继续检测
DEL:   MOV R7, #0FFH; 延时程序
DEL1:  MOV R6, #0FFH;
DEL2:  DJNZ R6, DEL2;
      DJNZ R7, DEL1;
      RET;
      END
```



程序在倚天版运行的状态 开始初始化为 ff. 大于 fe, 显示 1.



程序在倚天版运行的状态 用导线把 p10-p17 任意一个接地数码管显示 0

;13 该程序的功能是小喇叭 1khz 信号

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com
;知识要点 定时器的应用学习定时器的应用
;1 如果输出 1khz 那么 1 个周期是 500us 交替出现. 晶体 12m, 每周期 1us
;t1 为方式 0 则 m1m0=00h, 使用定时器 c/t=0 gate=0
;方式 0 为 13 位定时器, 总时间为 2 的 13 次方微秒. =8192 微秒, 减去 500 微秒=7692
;1111000001100 换算为 16 进制高 8 位等于 0fh 低 8 位 0ch
;注意:这两个字段不是连续的 低 8 位 01100 前面加 0 补足后再算
;初学请多多练习掌握计算方法, 有条件用示波器看频率

```
ORG 0000H;
LJMP MAIN;
ORG 030H;
MAIN: MOV TMOD, #00H;置定时器为方式 0
      MOV TH1, #0f0H;载入定时的初值
      MOV TL1, #0CH;
      MOV IE, #00H;禁止中断
      SETB TR1;起动定时器
LOOP: JBC TF1, LOOP1;查询定时器时间到了到 loop1
      AJMP LOOP;否则继续查询
LOOP1: MOV TH1, #0f0H;重新载入定时的初值
      MOV TL1, #0CH;
      CLR TF1;清除溢出标志
      CPL P3. 3;小喇叭响
      AJMP LOOP;继续查询
END;
```

14 该程序的功能是按 p3.5 10 次 p1 口 led 按照 2 进制加 1 计数器的应用

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 计数器的应用

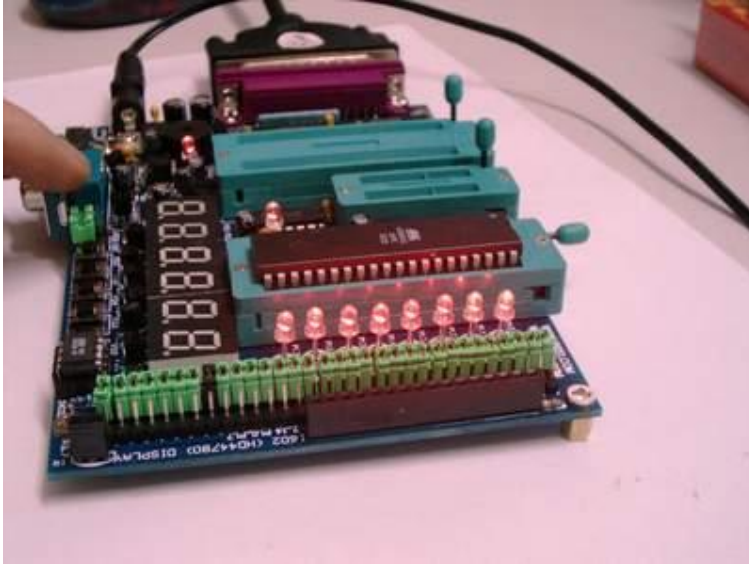
;计数值 10 次

;t1 为方式 2 则 m1m0=10h, 使用定时器 c/t=1 gate=0 TMOD=60H

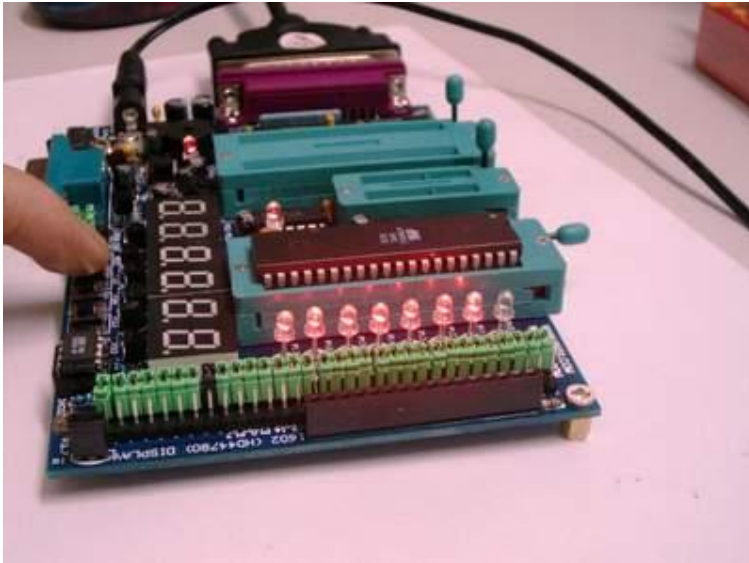
;方式 2 为 8 位计数器, 一共 256, 减去 10 等于 246 则计数器初值为 F6H

;注意定时器 T1 的脚就是 P3.5 该脚位为一个按键到地

```
ORG 0000H
    LJMP MAIN
    ORG 030H
MAIN:  MOV TMOD, #60H; 设置为方式 2
        MOV TH1, #0F6H; 载入初值 F6 即 246 (共 256 计 10 次)
        MOV TL1, #0F6H
        MOV IE, #00H; 禁止中断
        SETB TR1; 起动定时器
        MOV P1, #00H; 点亮 P1 口的灯
        ACALL DEL; 延时
LOOP:  JBC TF1, LOOP1; 查询定时器, 如到了到 LOOP1
        AJMP LOOP; 否则继续查询
LOOP1: INC A; A 加 1
        MOV P1, a; 输出到 P1
        AJMP LOOP; 继续查询
DEL:   MOV R7, #014H; 延时子程序
DEL1:  MOV R6, #0FFH
DEL2:  MOV R5, #01FH
DEL3:  DJNZ R5, DEL3
        DJNZ R6, DEL2
        DJNZ R7, DEL1
        RET
        END
```



程序在倚天版运行的状态 开机全亮就是 00000000



程序在倚天版运行的状态 按键 p35 10 次后 p10 熄灭, 就是 00000001

;15 该程序的功能使用定时器实现长时间的延时. p1.0 每分钟取反 1 次

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 计数器的应用

;把定时器设置为 50ms, 另外设置两个软件计数器 20, 60 总时间

;为 50ms*20*60=60000ms=1 分钟

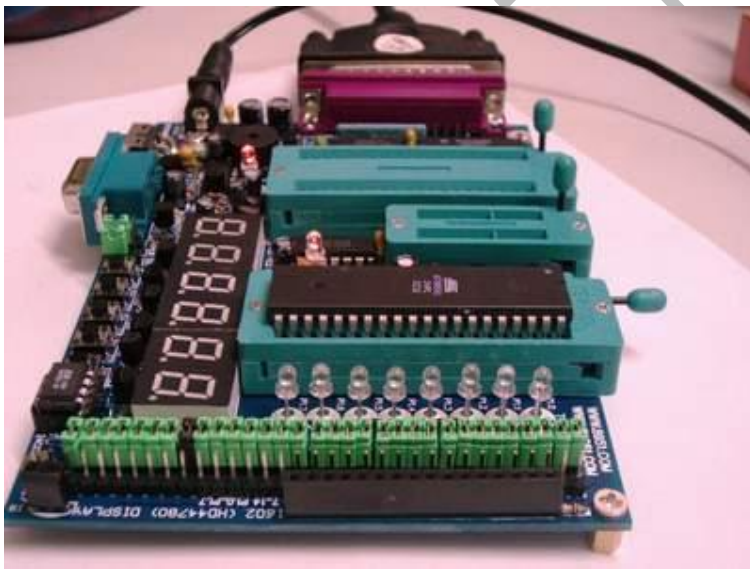
;t1 为方式 1 则 m1m0=01h, 使用定时器 c/t=0 gate=0 TMOD=10H

;方式 1 为 16 位计数器, 一共 65535, 减去 50000 等于 15536 则计数器初值为 3cb0h

ORG 0000H; 程序开始

LJMP MAIN

```
ORG 030H
MAIN:  MOV 30H, #20; 30h, 31h 分别置常数 20, 60
      MOV 31H, #60;
      MOV TMOD, #10H; 定时器方式 1
      MOV TH1, #03CH; 置定时器初值
      MOV TL1, #0B0H
      MOV IE, #00H; 禁止中断
      SETB TR1; 启动定时器
LOOP:  JBC TF1, LOOP1; 查询定时器时间到转 loop1
      AJMP LOOP
LOOP1: MOV TH1, #03CH; 重置定时器
      MOV TL1, #0B0H
      DJNZ 30H, LOOP; 定时是否到了 20 次
      MOV 30H, #20
      DJNZ 31H, LOOP; 定时是否到了 60 次
      MOV 31H, #60
      CPL P1.0; 取反 p1.0
      AJMP LOOP
      END
```



程序在倚天版运行的状态 程序运行后全灭



程序在倚天版运行的状态 1 分钟后 p10 点亮, 实现长时间的延时

;16 该程序的功能中断的响应, p3. 3 的小喇叭 1khz 输出

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492
 ;51 测试网系列自学单片机例程, www.51c51.com www.8951.com
 ;知识要点 中断的应用;1khz 就是每 500us 取反一次
 ;t1 为方式 0 则 m1m0=00h, 使用定时器 c/t=0 gate=0
 ;方式 0 为 13 位定时器, 总时间为 2 的 13 次方微秒.=8192 微秒, 减去 500 微秒=7692
 ;1111000001100 换算为 16 进制高 8 位等于 0fh 低 8 位 0ch
 ;注意:这两个字段不是连续的 低 8 位 01100 前面加 0 补足后再算
 ;初学请多多练习掌握计算方法, 有条件用示波器看频率
 ;中断响应的基本条件是中断有请求, 中断允许寄存器相应位置 1, 总中断开放 ea=1
 ;一旦响应中断, 只保护断点不保护现场, 所以一般要保护现场.

```

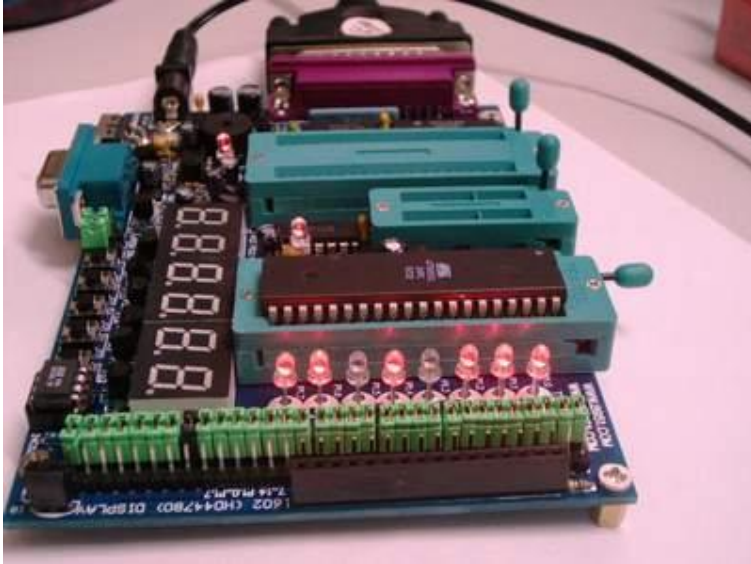
ORG 0000H;
LJMP MAIN;
ORG 001BH;跳转到中断入口
LJMP INSER;到中断服务程序
ORG 030H;
MAIN: MOV TMOD, #00H;ti 工作方式 0
      MOV TH1, #0F0H;载入初值
      MOV TL1, #0CH;
      SETB EA;开中断
      SETB ET1;定时器 t1 允许
LOOP: SETB TR1;起动定时器
HERE: SJMP HERE;等待中断, 虚拟主程序, 实际这里可以插入其他程序
      ORG 0200H;中断服务程序从 200h 开始
INSER: MOV TH1, #0F0H;重新装入定时初值
      MOV TL1, #0CH;
      CPL P3. 3;取反小喇叭
  
```

```
RETI;中断返回  
END
```

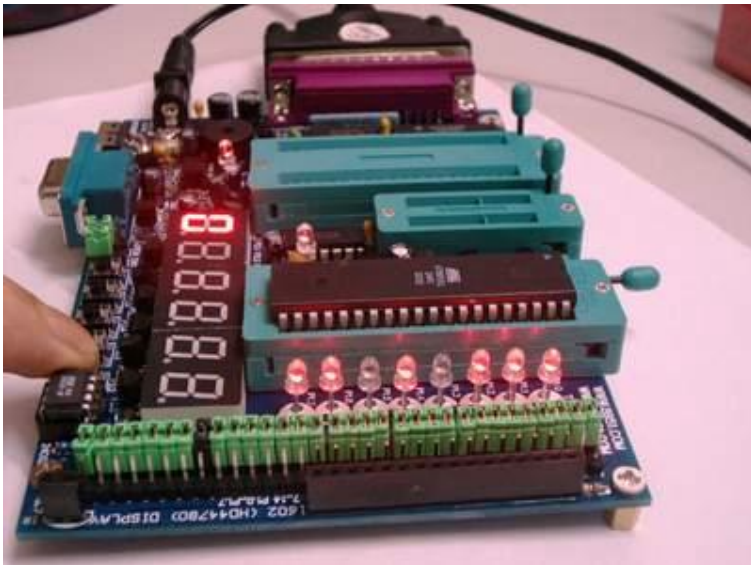
;17 该程序的功能中断的响应, p3. 2 的键盘数码管显示 0

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com
;知识要点 中断的应用;熟悉中断的地址入口, 中断的返回与响应
;主程序是把 p1 口送一个数据 28h
;所以 p1 的状态为 00101000 响应中断(按 p32, 那么到中断服务数码管显示 0)
;中断响应的基本条件是中断有请求, 中断允许寄存器相应位置 1, 总中断开放 ea=1
;一旦响应中断, 只保护断点不保护现场, 所以一般要保护现场。

```
ORG 0000H  
    LJMP MAIN  
    ORG 0003H; 外中断 0 的入口地址(注意是 p32)  
    LJMP 0100H; 跳到 100h 地址处  
    ORG 030H  
MAIN: MOV A, #28H; a 送 28h  
    SETB EA; 开总中断  
    SETB EX0; 开外部中断  
LOOP: MOV P1, A; 送 a 到 LED 显示  
    LCALL DEL1; 调用延时  
    AJMP LOOP; 重新开始  
DEL:  MOV R7, #0FFH; 延时程序  
DEL1:  MOV R6, #0FFH  
DEL2:  DJNZ R6, DEL2  
    DJNZ R7, DEL1  
    RET  
    ORG 0100H; 中断服务程序从 100h 开始  
    MOV P0, #28h; 数码管显示 0  
    CLR P2.1; 打开第一个数码管  
    RETI  
    END
```



程序在倚天版运行的状态 P1 的状态为 00101000



程序在倚天版运行的状态 按 p32 后数码管显示 0, p1 口不变

18 该程序的功能中断的响应, 两级中断嵌套, 中断优先级设置

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 中断的应用;熟悉中断的地址入口, 中断的返回与响应;中断的优先级

;开始主程序数码管显示 0, 按 p32 显示 1, 延时 10 秒返回主程序

;按 p33 显示 2 延时 2 秒返回主程序,

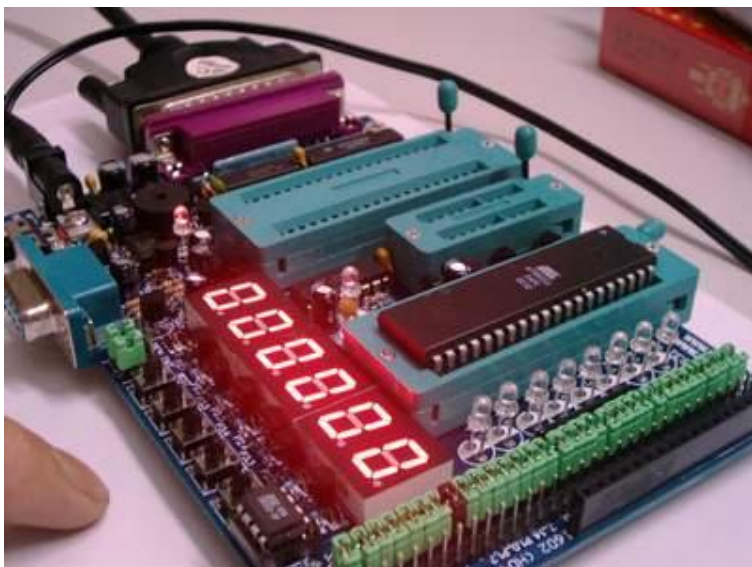
;如果按 p32 显示 1 后再按 p33, 那么显示完成后继续显示 1, 完成后返回主程序显示 0

;由于 p33 的优先级高, 首先响应, 请仔细体会这个程序

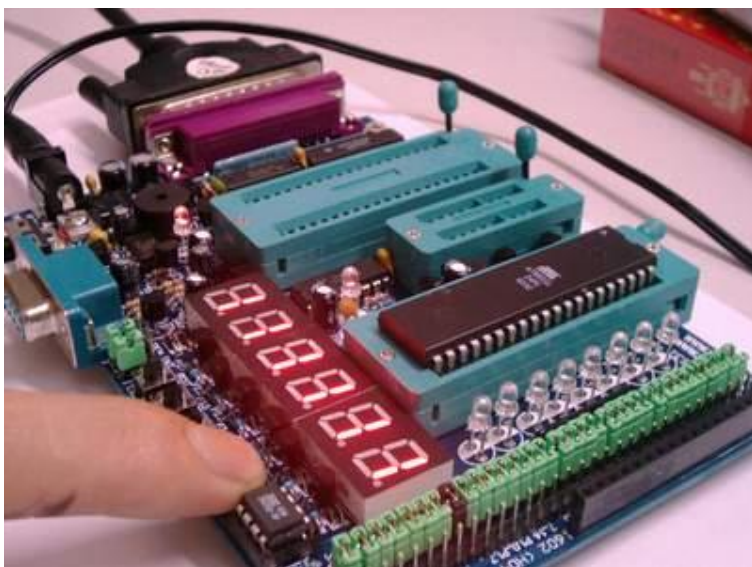
ORG 0000H

LJMP MAIN;跳到主程序

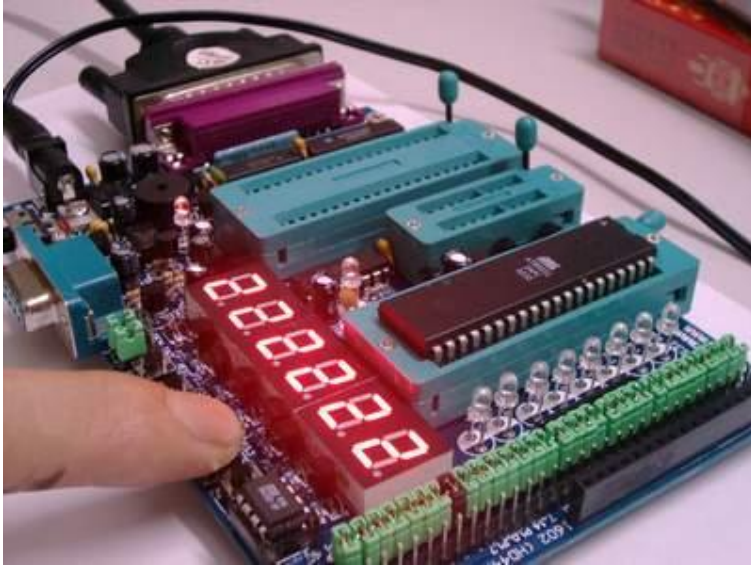
```
ORG 0003H;中断 0 入口
LJMP INSR0;到中断服务 0
ORG 0013H;中断 1 入口地址
LJMP INSR1;到中断服务 0
ORG 030H;主程序从 030h 开始
MAIN:  MOV SP, #70H;设堆栈指针
        MOV IE, #85H;开开放总中断以及中断 0, 1
        SETB PX1;设中断 1 为高优先级别
LOOP:  MOV P0, #28h;数码管显示 0, 主程序
        mov p2, #0h;打开所有数码管
        SJMP LOOP
ORG 0100H;中断 0 服务从 100h 开始
INSR0:  MOV R5, #3FH;置初值
DS0:   MOV P0, #7eh;数码管显示 1, 中断 0 程序低优先级别
        mov p2, #0h
        LCALL DEL
        DJNZ R5, DS0;延时 10 秒
        RETI
ORG 0150H;中断服务 1 从 150h 开始
INSR1:  MOV A, R5
        PUSH ACC;把 r5 内容保护
        MOV R5, #0FH;
DS1:   MOV P0, #0a2h;数码管显示 2, 中断 1 程序高优先级别
        mov p2, #0h
        LCALL DEL;延时 2 秒
        DJNZ R5, DS1
        POP ACC;弹出 acc
        MOV R5, A
        RETI;中断返回
ORG 0200H;延时程序从 200h 开始
DEL:   MOV R4, #0ffh
DEL1:  MOV R3, #0FEH
DEL2:  DJNZ R3, DEL2
        DJNZ R4, DEL1
        RET
END
```



程序在倚天版运行的状态 开始的时候数码管全部显示 0



程序在倚天版运行的状态 按 p32 后数码管全部显示 1



程序在倚天版运行的状态 按 p33 数码管全部显示 2

19 该程序的功能流水灯程序

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

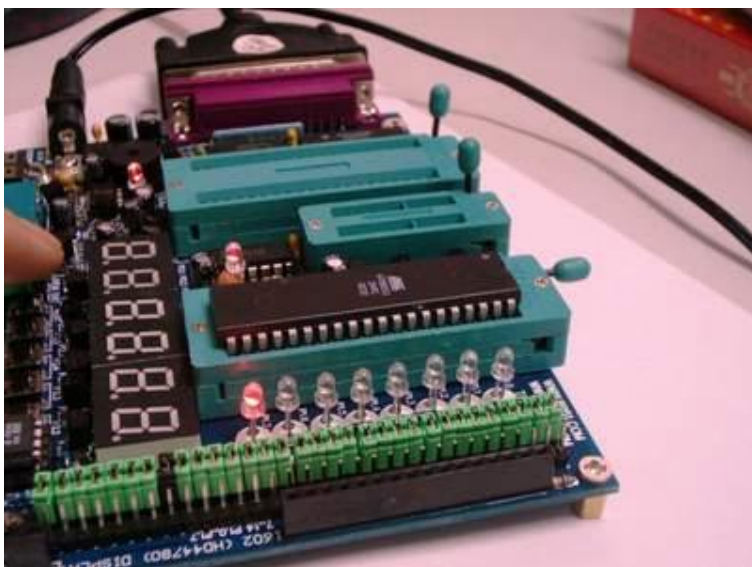
;知识要点 顺序程序的结构, 从头开始逐步运行, 熟悉单片机的端口

;顺序结构是最基本的简单的结构

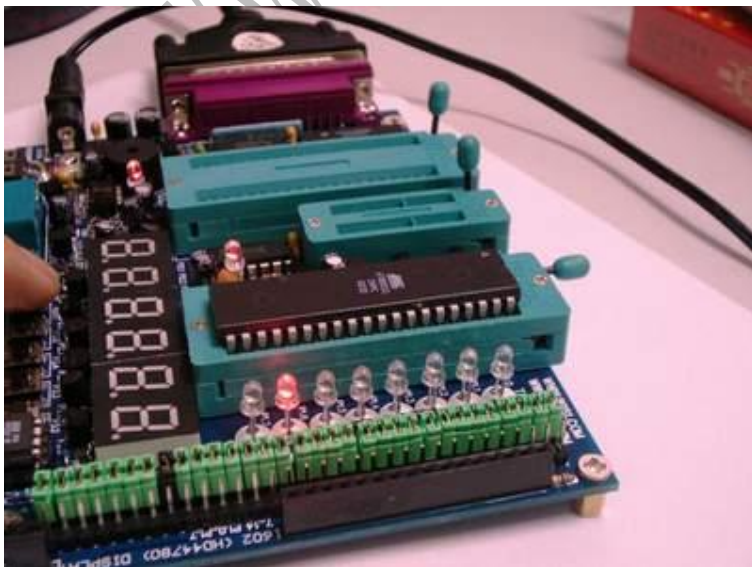
```
ORG 0000H
    LJMP MAIN
ORG 030H
MAIN: CLR P1.7 ;点亮第 1 个灯
      ACALL DEL;延时
      SETB P1.7;关闭低一个灯
      CLR P1.6;点亮第 2 个灯
      ACALL DEL
      SETB P1.6
      CLR P1.5;点亮第 3 个灯
      ACALL DEL
      SETB P1.5
      CLR P1.4;点亮第 4 个灯
      ACALL DEL
      SETB P1.4
      CLR P1.3;点亮第 5 个灯
      ACALL DEL
      SETB P1.3
      CLR P1.2;点亮第 6 个灯
      ACALL DEL
      SETB P1.2
      CLR P1.1
```

```
ACALL DEL
SETB P1.1
CLR P1.0
ACALL DEL
SETB P1.0
AJMP MAIN

DEL:  MOV R7, #0FFH; 延时子程序
DEL1: MOV R6, #0FFH
DEL2: MOV R5, #01FH
DEL3: DJNZ R5, DEL3
      DJNZ R6, DEL2
      DJNZ R7, DEL1
      RET
      END
```



程序在倚天版运行的状态 流水灯的瞬间亮第一个灯



程序在倚天版运行的状态 流水灯的瞬间亮第二个灯

20 该程序的功能 p1 口的 led 闪烁 10 次后停止子程序的嵌套

;该程序在倚天版开发套件运行通过 tel:0755-27529065 28068492

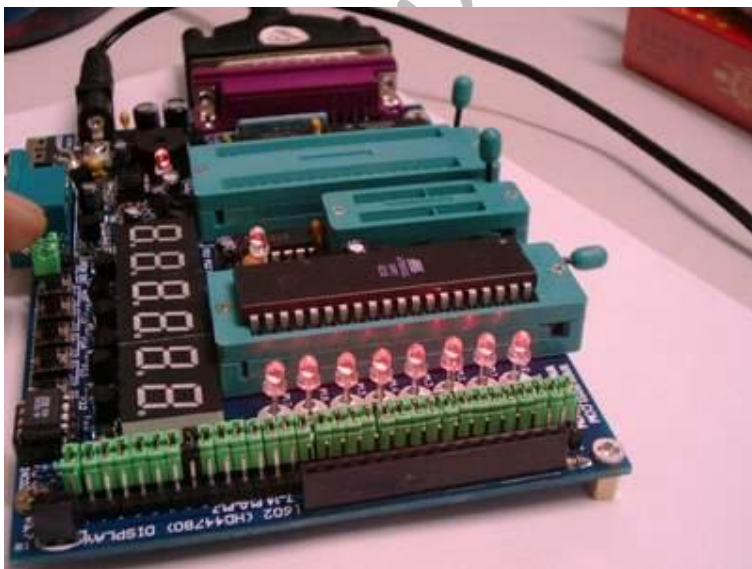
;51 测试网系列自学单片机例程, www.51c51.com www.8951.com

;知识要点 熟悉子程序的嵌套

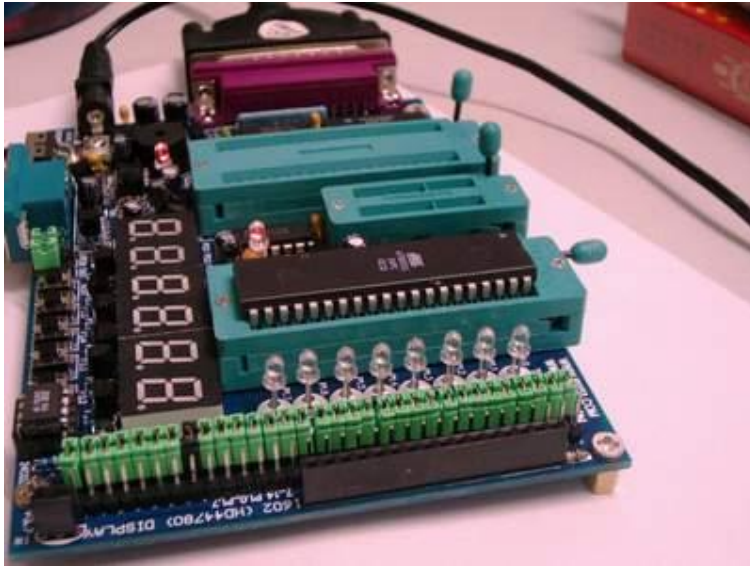
;r0 送 20, 开始为 0, 然后每次加 1, 到 20 后停止:注意闪烁是取反 2 次

;所以取反 20 次为闪烁 10 次

```
ORG 0000H
    LJMP MAIN
ORG 030H
MAIN:  MOV R0, #0; 最开始为 0
        MOV a, #0ffH; a 送 ff, 全部不亮
loop:  cpl a ; 取反 a(变为 00h, 全部亮)
        mov p1, a; 输出到 p1
        acall del; 调用延时以便观察
        cjne r0, #20, loop; 如果海没有到 20 则继续
        sjmp $; 否则原地等待结束
DEL:   MOV R7, #01FH; 延时子程序
DEL1:  MOV R6, #0ffH
DEL2:  MOV R5, #08H
DEL3:  DJNZ R5, DEL3
        DJNZ R6, DEL2
        DJNZ R7, DEL1
        inc r0; 每调用一次子程序 r0 加 1
        RET
END
```



程序在倚天版运行的状态 p1 口灯全亮



程序在倚天版运行的状态闪烁 10 次后停止

后记 本教程为 51 测试网 www.51c51.com www.8951.com 根据电子制作手把手学单片机教程精心整理编撰, 版主耗费大量时间精力, 为尊重作者劳动成果, 如转载请保留网站链接并著明出处.

深圳 51 测试网 跟我来学单片机

电话 0755-27529065

传真 0755-28068492

[返回 51 测试网 网站主页](#)

[返回 51 测试网 自学单片机](#)

[返回 51 测试网 系列单片机例子程序](#)

[我要参加讨论](#)

[如何购买本站产品](#)