

江苏省高等学校立项精品教材

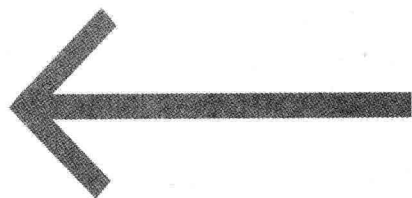
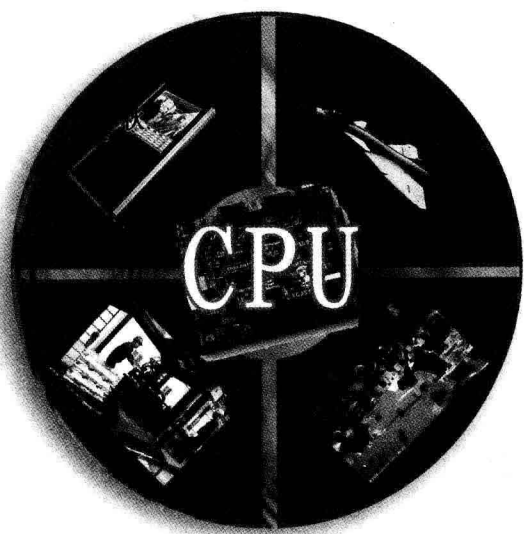
# 单片机技术与项目实施

主 编 戴 娟  
副主编 吴国中 周昱英  
主 审 赵 秋



 南京大学出版社

# 单片机技术与项目实施



主 编 戴 娟  
副主编 吴国中 周昱英  
参 编 (按姓氏笔画排序)  
孙立香 李从宏 陈 杰  
吴 清 曹建平 程俊静  
主 审 赵 秋

### 图书在版编目(CIP)数据

单片机技术与项目实施 / 戴娟主编. —南京: 南京大学出版社, 2010. 12

ISBN 978-7-305-07880-4

I. ①单… II. ①戴… III. ①单片微型计算机  
IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2010)第 235321 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮 编 210093

网 址 <http://www.NjupCo.com>

出 版 人 左 健

书 名 单片机技术与项目实施

主 编 戴 娟

责任编辑 陈 勇 编辑热线 025-83686531

照 排 南京紫藤制版印务中心

印 刷 盐城市华光印刷厂

开 本 787×1092 1/16 印张 13.5 字数 327 千

版 次 2010 年 12 月第 1 版 2010 年 12 月第 1 次印刷

ISBN 978-7-305-07880-4

定 价 25.00 元

发行热线 025-83594756

电子邮箱 [Press@NjupCo.com](mailto:Press@NjupCo.com)

[Sales@NjupCo.com](mailto:Sales@NjupCo.com)(市场部)

---

\* 版权所有, 侵权必究

\* 凡购买南大版图书, 如有印装质量问题, 请与所购  
图书销售部门联系调换

# 前 言

本教材是针对 MCS—8051 单片机和目前最流行的程序设计语言——C 语言,以 Keil C51 编译器为工具,讲解单片机接口技术的教材。C 语言具有开发速度快,代码可重复使用,程序结构清晰、易懂、易维护,在不同的单片机编程中移植性强,特别适宜开发一些比较大型的项目。

本教材由多年从事单片机教学与实际工程项目开发的教学团队共同编写。全书通过 4 个真实的项目,巧妙而系统性地融入嵌入式单片机系统的工作原理、硬件结构、内部资源、各种硬件接口设计、接口驱动程序、通信技术,在这 4 个项目中按照知识点、难点分别用多个案例紧紧围绕项目展开学习,这些案例既是相互独立,又呈递进关系,同时这些案例程序既可以给读者以开拓思路、参考的用途,又是实际的开发程序,可以直接作为程序应用在相同的开发系统上。项目的学习过程还涉及 C 语言编程技术、单片机的仿真开发技术、系统开发技术及各种市场上流行的外围接口器件技术。全书结构合理,逻辑性强,文字精炼,通俗易懂。通过本教材的学习,读者可以了解和掌握 C51 编程的思路和方法。

本教材不是以知识体系为主线结构展开,而是以应用项目引领知识、技能体系结构展开教学,强调做中学、好教好学。为了便于教师教学使用,同时也便于学生的学习,我们将 MCS51 单片机的知识点在附录中系统地给出。

本书可作为高职高专院校电类专业中单片机课程的教材以及毕业设计的参考书,也可供从事自动控制、智能仪器仪表、电力电子、机电一体化以及各类单片机应用的工程技术人员参考,同时也是广大单片机爱好者的参考用书。

本书在编写过程中还得到众多老师和领导的建议与帮助,在此,编者衷心感谢各位。

编 者  
2010 年 8 月

# 目 录

|                                 |    |
|---------------------------------|----|
| ◆ 项目 1 LED 电子显示屏 .....          | 1  |
| 1.1 项目描述 .....                  | 1  |
| 1.2 学习目标 .....                  | 1  |
| 1.3 项目案例 .....                  | 1  |
| 1.3.1 案例 1 一只 LED 闪烁 .....      | 2  |
| 1.3.2 案例 2 流水显示八只 LED .....     | 6  |
| 1.3.3 案例 3 32 只彩灯多模式 .....      | 8  |
| 1.3.4 案例 4 LED 点阵显示屏 .....      | 13 |
| 1.4 项目相关知识 .....                | 21 |
| 1.4.1 C51 程序的开发过程 .....         | 21 |
| 1.4.2 C51 位操作的用法 .....          | 21 |
| 1.4.3 指针与地址运算符 .....            | 25 |
| 1.4.4 MCS-51 单片机的外部存储器的扩展 ..... | 26 |
| 1.4.5 常用芯片介绍 .....              | 29 |
| 1.5 硬件电路分析 .....                | 30 |
| 1.6 软件编程分析 .....                | 31 |
| 1.7 项目实施 .....                  | 34 |
| 1.8 练习 .....                    | 35 |
| ◆ 项目 2 定时开关控制器 .....            | 38 |
| 2.1 项目描述 .....                  | 38 |
| 2.2 学习目标 .....                  | 38 |
| 2.3 案例分析 .....                  | 38 |
| 2.3.1 案例 1 数码静态显示 .....         | 39 |
| 2.3.2 案例 2 数码动态显示 .....         | 41 |
| 2.3.3 案例 3 独立键盘 .....           | 44 |
| 2.3.4 案例 4 0—9 键控秒表 .....       | 46 |
| 2.3.5 案例 5 行列键盘 .....           | 50 |
| 2.3.6 案例 6 99—00 键控倒计时秒表 .....  | 54 |
| 2.3.7 案例 7 定时开关控制器(中断) .....    | 60 |
| 2.4 相关知识 .....                  | 68 |

|        |                                             |     |
|--------|---------------------------------------------|-----|
| 2.4.1  | LED 数码管的结构与原理 .....                         | 68  |
| 2.4.2  | 键盘及其接口 .....                                | 70  |
| 2.4.3  | 定时器/计数器(T/C) .....                          | 72  |
| 2.4.4  | MCS-51 单片机的中断系统 .....                       | 74  |
| 2.5    | 硬件分析 .....                                  | 79  |
| 2.6    | 软件分析 .....                                  | 82  |
| 2.7    | 项目实施 .....                                  | 88  |
| 2.8    | 练习 .....                                    | 88  |
| ◆ 项目 3 | 调光控制系统 .....                                | 92  |
| 3.1    | 项目描述 .....                                  | 92  |
| 3.2    | 学习目标 .....                                  | 92  |
| 3.3    | 案例分析 .....                                  | 92  |
| 3.3.1  | 案例 1 串口通信 .....                             | 94  |
| 3.3.2  | 案例 2 数字电压计设计 .....                          | 99  |
| 3.3.3  | 案例 3 信号发生器设计 .....                          | 102 |
| 3.3.4  | 案例 4 基于 SPI 总线的电子时钟系统设计 .....               | 104 |
| 3.3.5  | 案例 5 调光控制系统设计 .....                         | 108 |
| 3.4    | 相关知识 .....                                  | 116 |
| 3.4.1  | 串行通信 .....                                  | 116 |
| 3.4.2  | AD/DA 概念介绍典型电路与程序设计 .....                   | 125 |
| 3.4.3  | I <sup>2</sup> C、SPI 总线概念及应用、无线信号通讯概念 ..... | 132 |
| 3.4.4  | 无线信号通讯(红外遥控) .....                          | 137 |
| 3.5    | 硬件电路分析 .....                                | 139 |
| 3.6    | 软件编程 .....                                  | 140 |
| 3.7    | 项目实施 .....                                  | 140 |
| 3.8    | 练习 .....                                    | 140 |
| ◆ 项目 4 | 多路温湿度巡检仪 .....                              | 142 |
| 4.1    | 项目描述 .....                                  | 142 |
| 4.2    | 学习目标 .....                                  | 142 |
| 4.3    | 项目案例 .....                                  | 142 |
| 4.4    | 项目相关知识 .....                                | 145 |
| 4.4.1  | STC 微处理器的性能特点 .....                         | 145 |
| 4.4.2  | AHT11 温湿度传感器电气参数 .....                      | 145 |
| 4.4.3  | A/D 转换芯片 TLC1543 .....                      | 146 |
| 4.4.4  | LM358 集成运放的使用方法 .....                       | 150 |

|                                  |     |
|----------------------------------|-----|
| 4.5 硬件电路分析 .....                 | 150 |
| 4.5.1 微控器模块 .....                | 150 |
| 4.5.2 ATH11 传感器模块的引脚以及接线实例 ..... | 150 |
| 4.5.3 信号放大模块 .....               | 152 |
| 4.5.4 AD 转换模块 .....              | 152 |
| 4.5.5 显示模块 .....                 | 153 |
| 4.5.6 通信模块(下载烧录口)模块 .....        | 153 |
| 4.6 软件分析 .....                   | 154 |
| 4.6.1 程序设计分析 .....               | 154 |
| 4.6.2 TLC1543 采样函数 .....         | 154 |
| 4.6.3 温度湿度转换函数 .....             | 156 |
| 4.7 项目实施 .....                   | 157 |
| 4.7.1 拟定总体设计方案 .....             | 158 |
| 4.7.2 硬件设计 .....                 | 158 |
| 4.7.3 软件开发 .....                 | 158 |
| 4.8 练习 .....                     | 159 |
| ◆ 总练习 .....                      | 161 |
| ◆ 附录 .....                       | 170 |
| 附录 A Keil C51 开发系统基本知识 .....     | 170 |
| 附录 B MCS-51 系列单片机组成及工作原理 .....   | 176 |
| 附录 C C51 语言 .....                | 192 |
| 附录 D C51 重要库函数 .....             | 199 |
| 附录 E PROTEUS 常用元器件 .....         | 201 |
| 附录 F 美国标准信息交换标准码 .....           | 204 |
| ◆ 参考文献 .....                     | 208 |

# 项目 1 LED 电子显示屏

## 1.1 项目描述

LED 显示屏(LED panel);LED 就是 light emitting diode 发光二极管的英文缩写。它是一种通过控制半导体发光二极管的显示方式,用来显示文字、图形、图像、动画、行情、视频、录像信号等各种信息的显示屏幕。本项目是以发光二极管为例,用单片机控制发光二极管实现 LED 显示屏,为了便于理解,按照难易程度,由浅入深地将从简单的单只发光二极管的点亮和熄灭控制开始,流水显示八只发光二极管到实现 32 只发光二极管组成的各种样式的 LED 电子显示屏显示。

## 1.2 学习目标

通过本项目的学习,掌握编译软件 KEIL、仿真软件 proteus 的操作与联调,掌握单片机管脚及功能,学会对单片机端口应用,掌握 C51 语言基本知识,了解常用锁存器、译码器、总线驱动器等芯片。

## 1.3 项目案例

为了让单片机控制发光二极管,需要将发光二极管连接到单片机的端口上,如图 1-1 所示,就是说单片机和 LED 电子屏控制的电路要连接上。

要实现 LED 的闪烁发光,就是要求 LED 所对应的单片机引脚不断交替输出高电平和低电平。为了让人眼能够看清楚 LED 的闪烁变化,LED 的闪烁的频率应在零点几到十几赫兹之间,一般取 1 Hz 左右。

因此,要实现发光二极管以不同的快慢闪亮,只需编写不同的程序,使单片机与发光二极管相连的端口输出不同频率的脉冲信号。若要控制发光二极管发出不同的闪亮样式,则需单片机对端口的各位在不同时间输出不同的驱动电平,就能驱动发光二极管以不同闪亮样式发光。步骤为:

- (1) 确定 LED 闪亮的方式;
- (2) 在计算机上编写原程序,程序功能就是控制单片机在与 LED 连接的端口上输出需



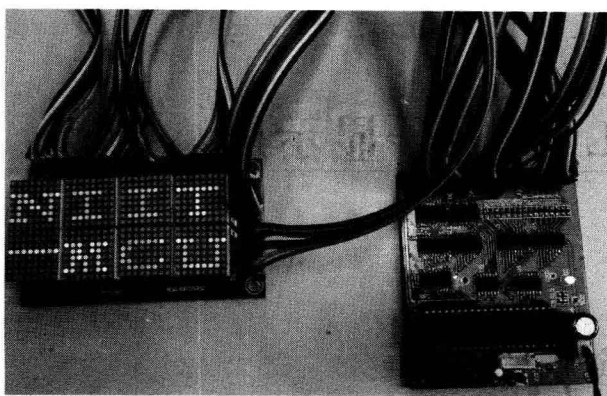


图 1-1 LED 电子显示屏

要的高低电平信号；

(3) 用软件调试和编译原程序；

(4) 将编译后的程序代码写入单片机；

(5) 在 LED 上观察程序运行效果。若要修改程序，可在线修改程序，再编译和下载，再观察修改后的效果，直至完全准确为止。

### 1.3.1 案例 1 一只 LED 闪烁

#### 1. 任务描述

本任务是要用单片机实现一只 LED 彩灯按 1 Hz 的频率闪烁，见图 1-2。就是要求在 1 s 内，让彩灯亮 0.5 s，熄灭 0.5 s。也就是让单片机端口的某一位每隔 0.5 s 改变一次输出电平。

要点：端口位处理，程序结构。

#### 2. 电路原理

其电路原理图见图 1-2，该电路在单片机最小系统的基础上加了一个 LED 彩灯和限流电阻 R2。电路的核心部分是 AT89C51 单片机，这里只用到 P1 口的一个引脚。图中 C3、R1 组成典型的上电复位（即在加电时单片机复位）电路，X1、C1、C2 与 AT89C51 片内振荡电路组成时钟振荡器。值得注意的是，C2、C3 的容量不能与图中数值偏差太大，否则可能引起不起振或振荡不稳定。XTAL 的频率可以在 4 MHz~20 MHz 之间；不过，频率的变化会导致程序运行速度的变化，这样就需要调整延时子函数的参数。事实上，不调整参数亦可，只是此时延迟时间不再是 1 秒。

用单片机点亮一只发光二极管 LED，显然，这个 LED 必须要和单片机的某个引脚相连，我们将这个 LED 和 P1.0 脚相连。（见图 1-2，其中 R2 是限流电阻）

按照图 1-2 的接法，当 P1.0 脚是高电平时，LED 不亮，只有 P1.0 脚是低电平时，LED 才发亮。因此我们要让 P1.0 引脚按要求变为高电平或低电平。P1.0 脚是 Intel 公司规定的名称，不可以更改。

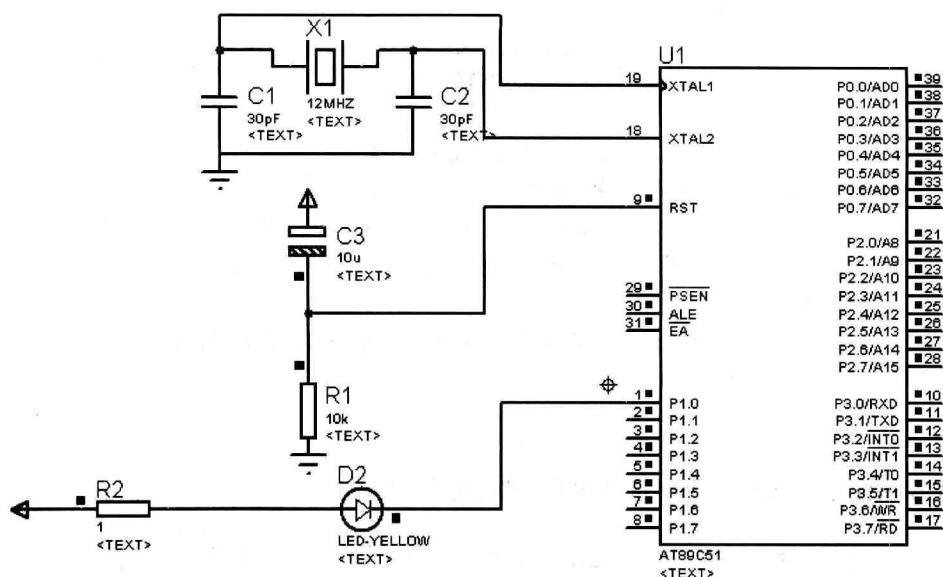


图 1-2 案例 1 电路参考原理图

### 3. 软件编制

怎样让它变“高”或变“低”呢？当给某一个引脚送“1”数据，即输出高电平；当给某一个引脚送“0”数据，即输出低电平，就可以使 LED 灯不停地一亮一灭。单片机重复地实现输出高低电平这个过程可用流程图表示出来，如图 1-3 所示。

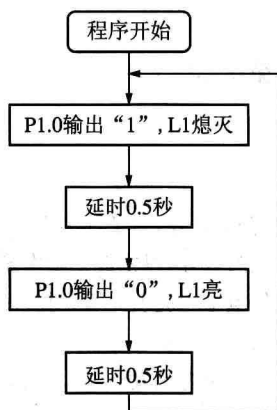


图 1-3 1 Hz 的闪烁程序设计框图

软件采用 C51 编程。共有两段程序：主程序 main()、延时程序 delay05()。使用了 for 语句，当然也可以使用别的控制语句来实现相同的功能。

其中延时程序的设计方法如下：

单片机的指令执行的时间很短，数量为微秒级。因此，我们要求的闪烁时间间隔为 0.5 秒，相对于微秒来说，相差太大，所以我们在执行某一指令时，插入延时程序，来达到我们的要求。这样的延时程序该如何设计呢？

下面具体介绍其原理:

```
void delay05( )           //软件定时延时 0.5 s 函数
{
    uchar i,j,k;          //声明三个无符号字符型变量 i,j,k
    for(i=5;i>0;i--)      //外循环 5 次,每次约 0.1s,共延时为 5×0.1 s=0.5 s
    for(j=200;j>0;j--)    //循环 200 次,每次约 0.5ms,所以延时约为 200×0.5 ms= 100 ms=0.1 s
    {                      //内循环共 250 次,延时约为 250×2 μs = 500 μs=0.5 ms
        for(k=250;k>0;k--)
        { ; }             //最里面循环的循环体,什么也不做,但每次循环延时约为 2 μs
    }
}
```

参考源程序/\* 驱动 LED 每秒闪烁一次的演示程序:

```
#include <reg51.h>        //包含头文件,声明各个特殊功能寄存器
#define uchar unsigned char //为了书写简单,定义 uchar 表示无符号字符型
sbit p1=P1.0;             //定义 P1.0 为开关输入
void delay05( )           //软件延时函数

{
    uchar i,j,k;          //声明三个无符号字符型变量 i,j,k
    for(i=5;i>0;i--)      //外循环 5 次,每次约 0.1 s,共延时为 5×0.1 s=0.5 s
    for(j=200;j>0;j--)    //循环 200 次,每次约 0.5 ms,所以延时约为 200×0.5 ms= 100
                           ms=0.1 s

    for(k=250;k>0;k--);   //循环延时,延时约为 250×2 us=500 us=0.5 ms
}

void main( )
{
    while (1)             //while 循环,当条件为真时,执行下面的循环体,由于条件
                           //恒为真,所以构成无限次循环

    { p1=0;               //低电平点亮发光管,对应 P1.0 为低,即 L0 点亮
      delay05( );         //延时 0.5 秒
      p1=1;               //高电平熄灭发光管,对应 P1.0 为高,即 L0 熄灭
      delay05( );         //延时 0.5 秒
    }
}
```

#### 4. 实施步骤

第一步:硬件设计,打开 proteus ,按照图 1-2 添加元器件。

表 1-1 案例 1 元件清单

| 关键字          | 标 值    | 数 量 | 名 称   |
|--------------|--------|-----|-------|
| AT89C51      |        | 1   | 单片机   |
| CAP          | 30 pF  | 2   | 电容    |
| CAP - ELEC   | 10 uF  | 1   | 电解电容  |
| CRYSTAL      | 12 MHz | 1   | 晶振    |
| RES          | 1,10k  | 2 个 | 电阻    |
| LED - YELLOW | yellow | 1   | 发光二极管 |

元器件添加后,在 Terminals Mode 选中电源,地,进行电路连接,如图 1-4 所示。

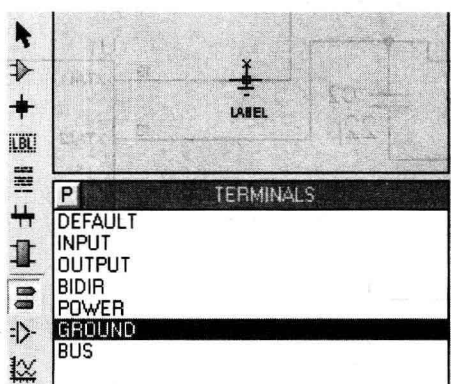


图 1-4 Proteus 选择电源、地

第二步:程序设计。

首先使 P1.0 输出一个低电平,使得灯亮,然后延时,使得灯灭,无限循环。

第三步:调试与仿真。

- (1) 打开 keil uvision,新建 keil 项目。
- (2) 选 CPU,选择 AT89C51。
- (3) 新建.c 文件,编写程序,并保存。
- (4) 在“project workspace”窗口中,将新建.c 文件添加到“source group 1”中。
- (5) 在“project workspace”窗口中的“target 1”文件夹单击右键,在弹出的右键菜单中选择“option for target 1”选项,弹出“option for target”对话框,选择“output”选项,选择“create hex file”选项。
- (6) 在“option for target 1”对话框中选择“debug”选项卡,并选中“use: proteus VSM simulator”选项,将 proteus VSM simulator 作为 keil 的调试工具。
- (7) 在 keil 的菜单中选择“project”下“build project”,编译程序,如有错误,自行修改。
- (8) 用 proteus 生成的电路中,在 AT89C51 芯片上单击左键,打开“edit component”对话框,设置晶振频率 12 MHz,并在“program file”中,选择用 keil 生成的 hex 文件。在“file”选项下,选“save design”,保存设计。
- (9) 在 proteus 菜单中,打开“debug”,选中“use remote debug monitor”选项,以支持与

keil 联合调试。

(10) 在 keil 菜单的“debug”项下,选“start debug”,进入调试,按 F5 键运行。调出 proteus 界面,就会看到运行结果。

### 1.3.2 案例 2 流水显示八只 LED

#### 1. 任务描述

用单片机流水形式点亮与它相连的八只发光二极管 LED。见图 1-5。

要点:单片机并口处理,单片机内部结构与管脚功能。

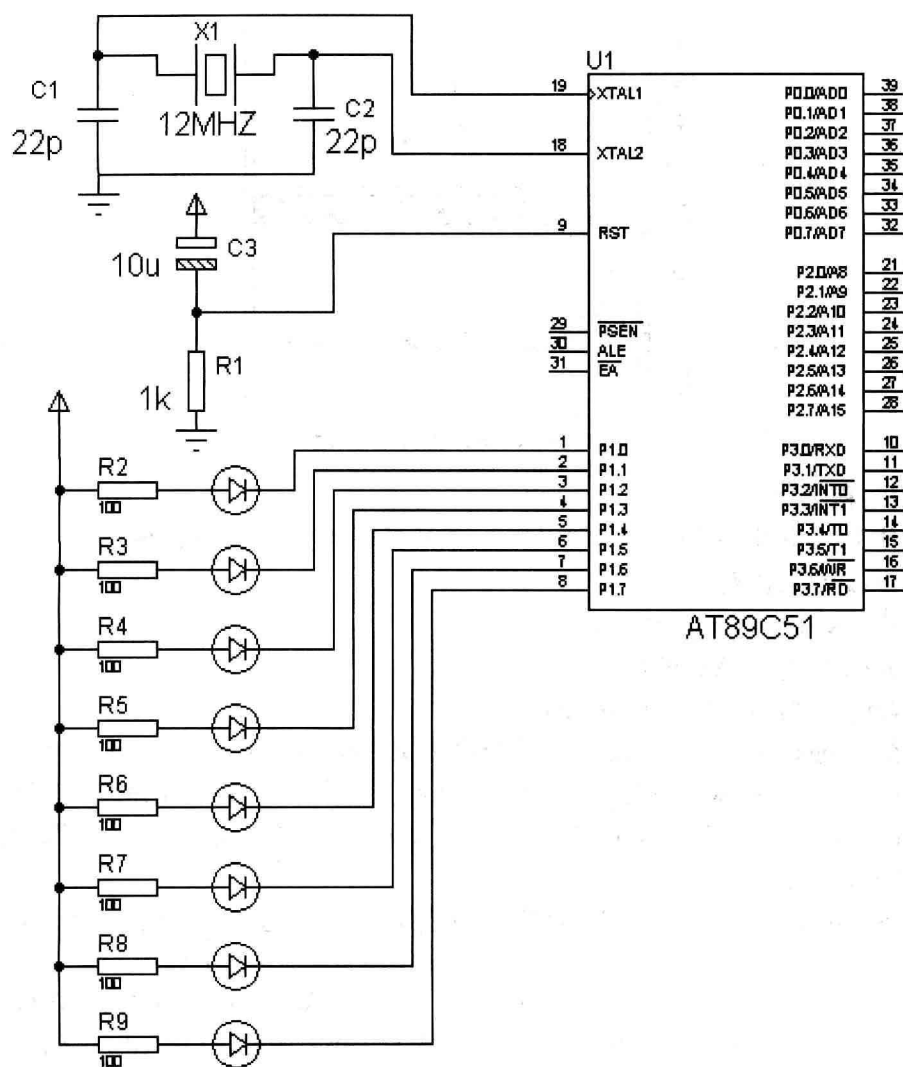


图 1-5 案例 2 电路参考原理图

## 2. 电路原理

八个发光二极管分别接在单片机的 P1.0—P1.7 上,只要那个脚上输出“0”,与它相连的发光二极管就被点亮。开始后,按 P1.0→P1.1→P1.2→P1.3→……→P1.7→P1.6→……→P1.0 次序逐个点亮与它相连的 LED,重复循环这个过程。

## 3. 软件编制

通过案例 1 我们已掌握使 LED 发光,但如何流水点亮? 原理是使用延时。每次送出的数据是不同,具体的数据如下表 1-2 所示即可达到要求的输出控制动作。

表 1-2 输出控制与指示灯对应表

| P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 | 说明   |
|------|------|------|------|------|------|------|------|------|
| L8   | L7   | L6   | L5   | L4   | L3   | L2   | L1   |      |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | L1 亮 |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 1    | L2 亮 |
| 1    | 1    | 1    | 1    | 1    | 0    | 1    | 1    | L3 亮 |
| 1    | 1    | 1    | 1    | 0    | 1    | 1    | 1    | L4 亮 |
| 1    | 1    | 1    | 0    | 1    | 1    | 1    | 1    | L5 亮 |
| 1    | 1    | 0    | 1    | 1    | 1    | 1    | 1    | L6 亮 |
| 1    | 0    | 1    | 1    | 1    | 1    | 1    | 1    | L7 亮 |
| 0    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | L8 亮 |

根据前面的分析,实现任务的思路是:程序开始时,给某一变量赋初始值 0x01,并从 LED 驱动端口输出变量的反码,等待一段时间后,让变量的值左移,再次输出反码并延时,直到所有数据输出完毕,再次重复整个过程。

根据思路得出的程序流程图如图 1-6 所示。

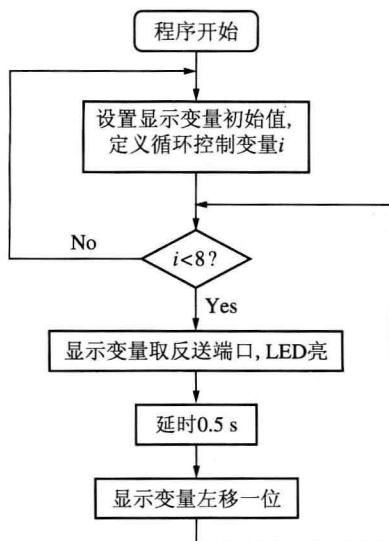


图 1-6 流水显示八只 LED 程序流程图

参考源程序如下:

```

/* 依次点亮发光二极管的完整的程序: */
#include <reg51.h>           //包含头文件
#define uchar unsigned char
void delay05( )              //延时 0.5s 子程序
{
    uchar i,j,k;
    for(i=5;i>0;i--)
        for(j=200;j>0;j--)
            for(k=250;k>0;k--);
}
void main( )                 //主函数
{
    uchar m,n;

    while (1)                //无限循环
    {
        n=0x01;              //n 初始化为 0x01,即 00000001
        for(m=0;m<=7;m++)    //for 循环,完成 8 次循环,重复执行 8 次循环体
        {
            P1=~n;            /* n 表示将变量 j 中的二进制位取反。n 初始值为 0x01,即
                                00000001 将 n 各位取反后为 11111110,输出到 P1 口相对应的
                                端口信号为 0,LED 就亮;为 1,LED 熄灭 */

            delay05( );        //延时 0.5 s
            n=n<<1;           /* 可采用“n=n+n;”或“n=n*2;”。n<<1 表示变量 n
                                中的二进制位左移 1 位,并最低位补 0。 */

        }
    }
}

```

#### 4. 实施步骤

同案例 1。

### 1.3.3 案例 3 32 只彩灯多模式

#### 1. 任务描述

本案例是带扩展功能的 32 只彩灯多模式显示,见图 1-7。要求流水轮流显示 32 只彩灯。

要点:利用单片机的读写信号实现并口扩展电路结构与编程。

#### 2. 电路原理

该电路用 AT89S52 作为主控芯片,用 74LS373 作为锁存驱动,用 4 个 373 分别连接 8 个 LED,32 个 LED 构成显示模块。方案如图 1-7 所示。由于是要求扩展能力,最方便的

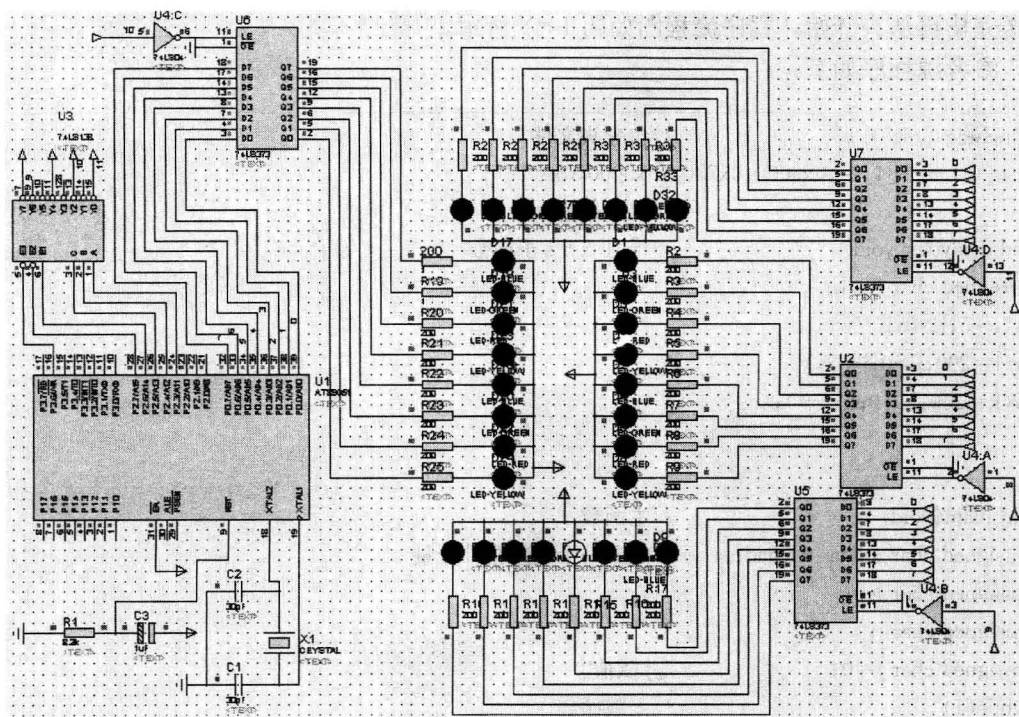


图 1-7 32 只彩灯多模式电路原理图

是使用单片机的 /WR 引脚即 16 脚。/WR(外部数据写控制)与 /RD(外部数据读控制),这两个引脚在访问外部芯片时能根据时序要求自动发出控制信号。我们可以使用 74LS373、74LS138 与单片机组完成带扩展功能的彩灯显示。

74LS373 在电路中起到对单片机 P0 口八位数据的锁存功能,74LS138 是 3—8 译码器,负责控制 74LS373 的传递数据。

### 3. 软件编制

通过案例 2 的学习,32 只 LED 彩灯控制器所有亮灯模式均由软件控制完成,本设计的功能是通过改变 LED 亮灭方式使彩灯轮流显示。用一片 74LS138 译码器输出端 Y0,Y2,Y4,Y7 分别负责控制 4 片 74LS373 传递数据。根据前面的分析,实现任务的思路是:程序开始时,定义四个地址指针变量,给第一个地址指针变量赋初始值 0x01,并从端口输出变量的反码。等待一段时间后,让变量的值左移,再次输出反码并延时,直到八个数据输出完毕,再次给第二个地址指针变量赋初始值 0x01,重复输出数据,直至四个地址指针变量的各 8 个数据

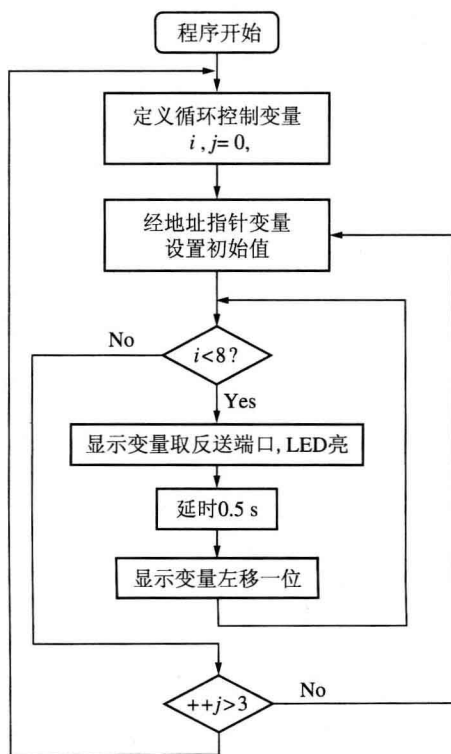


图 1-8 32 只彩灯多模式程序流程图



传送完毕再从头开始。程序流程图如图 1-8 所示。

参考源程序如下：

```
#include<reg51.h>           //包含头文件
unsigned int b[4]={0x6000,0x7800,0x5000,0x4000};
                               //四个地址

void delay05s(void)          //延时 0.5 s 子函数
{
    unsigned char i,j,k;
    for(i=5;i>0;i--)
    for(j=200;j>0;j--)
    for(k=250;k>0;k--);
}

void main(void)              //主函数
{
    unsigned char xdata * ledad; //定义一个地址指针变量
    unsigned char led;
    unsigned char i;
    unsigned char j=0;         //j 赋值 0
    while(1)                   //无限循环
    {
        ledad=b[j];            //将第一个地址赋给地址变量
        led=0x01;              //led 初始化为 0x01,即 00000001
        for(i=0;i<8;i++)       //for 循环,完成 8 次循环,
        { * ledad=~led;         //将变量 led 中的二进制数位取反
            //led 初始值为 0x01,即 00000001 将 led 各位取反后为 11111110
            //输出到 P0 口的那一位信号为 0,LED 就亮;为 1,LED 熄灭
        }
        delay05s( );           //延时 0.5s
        led=led<<1;            //可采用“n=n+n;”或“n=n*2;”。n<<1 表示变量 n 中的二进
            //制位左移 1 位,在最低位补 0。
    }
    if(++j>3) j=0;             //j 加 1,判断 j 是否大于 3,若大于 3,则 j 重新赋予 0
    * ledad=0xff;              //把灯全关闭
}
}
```

#### 4. 实施步骤

(1) 列出图 1-7 电路器件清单。含元件名称、参数及数量。

(2) 用 proteus 软件绘制出如图 1-7 所示的仿真电路图,参数设置同案例 1。

NOT 器件在 proteus 的选取如图 1-9。

(3) 源程序输入、编辑及编译。

运行 keil,建立工程名,输入源程序程序,命名为 \*.c,将文件加入工程中,设置工程,正确编译后生成目标文件“\*.hex”,在 proteus 中加入该文件。