

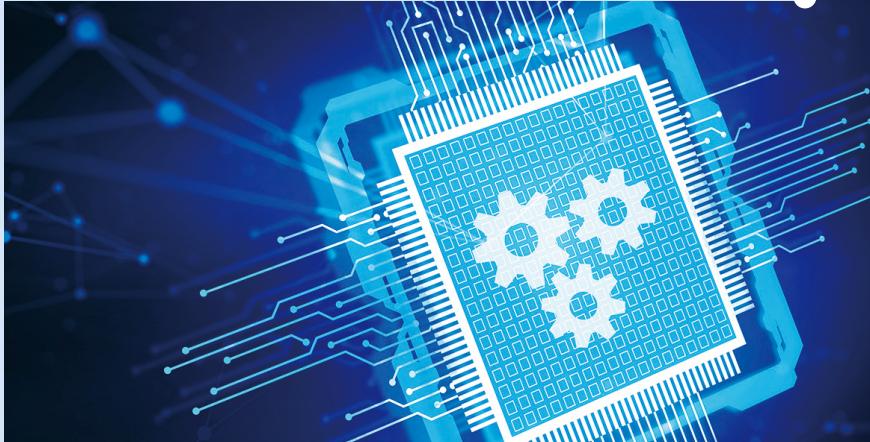


“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定
电子信息类专业“互联网+”创新型

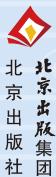
单片机 技术及应用

主编 冯 佳 尹静涛

主编 冯佳 尹静涛



DANPIANJI JISHU JI YINGYONG



北京出版社集团
北京出版社

图书在版编目 (CIP) 数据

单片机技术及应用 / 冯佳, 尹静涛主编 .-- 2 版 .

北京: 北京出版社, 2025.7. -- ISBN 978-7-200

-19575-0

I. TP368.1

中国国家版本馆 CIP 数据核字第 20259SY912 号

单片机技术及应用 (第二版)

DANPIANJI JISHU JI YINGYONG (DI-ER BAN)

主 编: 冯 佳 尹静涛

出 版: 北京出版集团

北京出版社

地 址: 北京北三环中路 6 号

邮 编: 100120

网 址: www.bph.com.cn

总 发 行: 北京出版集团

经 销: 新华书店

印 刷: 定州启航印刷有限公司

版 印 次: 2025 年 7 月第 2 版 2025 年 7 月第 1 次印刷

成品尺寸: 185 毫米 × 260 毫米

印 张: 12.5

字 数: 281 千字

书 号: ISBN 978-7-200-19575-0

定 价: 45.00 元

教材意见建议接收方式: 010-58572341 邮箱: jiaocai@bphg.com.cn

如有印装质量问题, 由本社负责调换

质量监督电话: 010-82685218 010-58572341 010-58572393

目 录

单元一 基本输入与输出控制	1
任务一 流星灯	2
任务二 汽车转向灯	27
任务三 智能感应迎宾灯	40
<hr/>	
单元二 字符显示控制	55
任务一 公交车头电子路牌	56
任务二 点阵广告屏	75
<hr/>	
单元三 定时与计数控制	105
任务一 汽车倒车雷达	106
任务二 电子密码锁	122
<hr/>	
单元四 中断编程控制	144
任务一 歌曲点唱机	145
任务二 定额感应计数器	166
<hr/>	
参考答案	189
参考文献	192

单元一

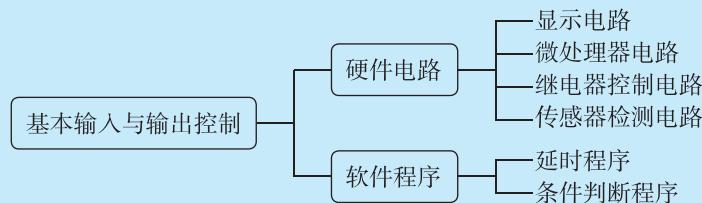
基本输入与输出控制

本单元教学内容在学生未来工作岗位中的作用是帮助学生掌握单片机方面的基本专业理论知识，熟悉单片机产品开发的整个流程，能够使用常用的软件开发工具、测试仪器进行软硬件联调，达到项目所要求的各项功能和性能指标。本单元由任务一流星灯、任务二汽车转向灯、任务三智能感应迎宾灯组成，借助三个任务的学习，了解单片机输入信号和输出信号的特点，掌握单片机基本输入与输出信号的控制方法，熟练掌握单片机C语言的编写规范和流程，为后续程序的编写奠定基础。

学习目标

1. 理解单片机的组成、特点、发展及应用。
2. 能够认识单片机的引脚及功能。
3. 掌握单片机应用系统的开发过程。
4. 掌握单片机应用系统搭建及程序编写、固化与调试的方法。
5. 在电路搭建中，逐步养成规范操作意识和精益求精的工匠精神。
6. 在程序调试中，养成细致耐心的学习态度和严谨认真的工作态度。
7. 在实操训练中，树立科技强国，学习报国的理想信念。

知识导图





任务一 ··· 流星灯



任务描述

随着科技的快速发展和城市建设的不断推进,流星灯作为一种创新的建筑装饰元素,成为城市景观美化的新亮点(图1-1-1)。本设计灵感来源于流星划过夜空的自然奇观,通过单片机精准控制LED灯珠,实现了模拟流星动态轨迹的灯光效果。这种创新的照明系统能够为商业建筑、城市景观和公共空间带来极具视觉冲击力的动态光影体验,不仅提升了环境美感,更赋予建筑立面流动的生命力。

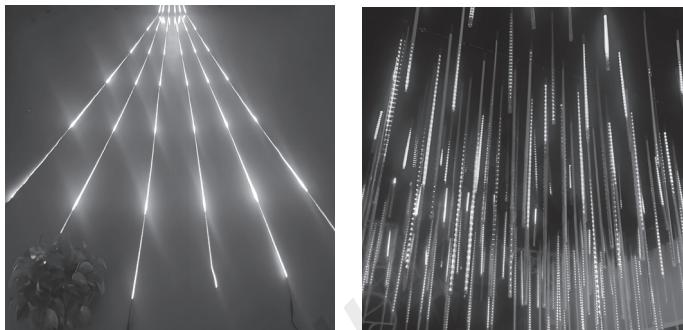


图1-1-1 流星灯



任务要求

1. 电源工作正常:接上+5V电源,电源指示灯红灯亮。
2. 显示电路工作正常:正常通电后,单片机P1口每次点亮一只发光二极管,从左至右、从右至左循环点亮。
3. 微处理器电路工作正常:能按照任务要求正确输出控制信号,实现控制功能。



任务分析

本任务功能的实现是从硬件和软件两个方面进行分析。

1. 硬件电路

本任务的硬件电路较为简单,主要由显示电路和微控制器电路两部分构成。显示电路由发光二极管和限流电阻构成,应注意限流电阻的选取。

2. 软件程序

本任务主要使用单片机Keil C51编写控制程序,驱动发光二极管连续点亮,实现流星灯效果。

任务实施

1. 识读电路原理图

流星灯电路原理图如图 1-1-2 所示。

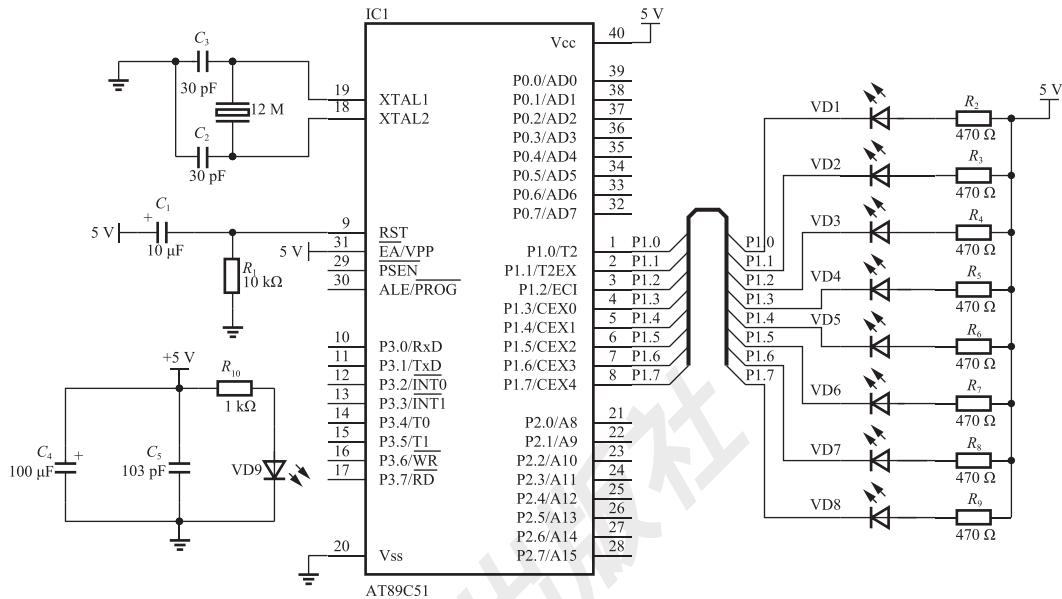


图 1-1-2 流星灯电路原理图

2. 配置电路模块

根据电路原理图，组建该电路可配置 EDM001-MCS51 主机模块。

3. 连接实物模块

流星灯电路连接实物图如图 1-1-3 所示。

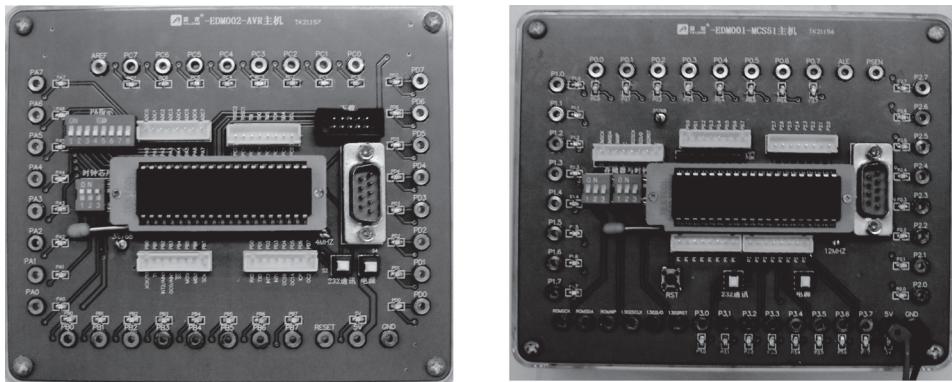


图 1-1-3 流星灯电路连接实物图



实物连接说明：该项目只用到 EDM001-MCS51 主机模块，将该模块连接 +5 V 电源和地。

4. 绘制电路框图

根据图 1-1-2 电路原理图，流星灯电路框图如图 1-1-4 所示。

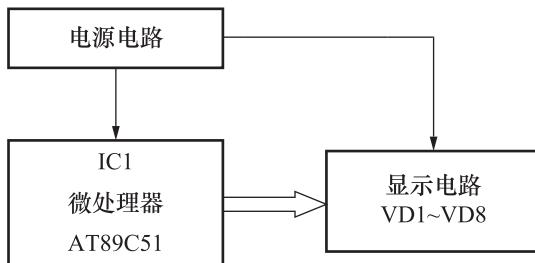


图 1-1-4 流星灯电路框图

5. 编制程序流程图

流星灯程序流程图如图 1-1-5 所示。

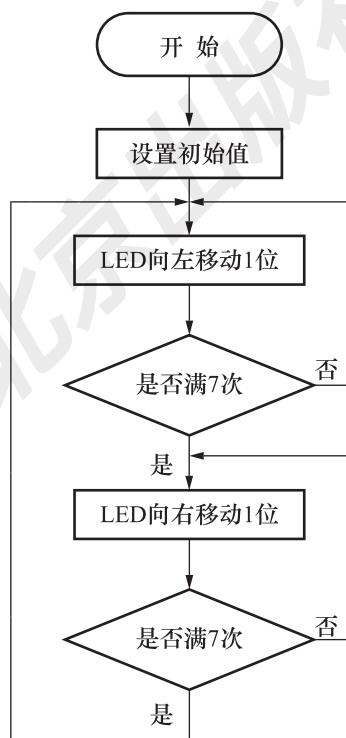


图 1-1-5 流星灯程序流程图

6. 程序设计

```

#include <reg51.h> // 定义8051存储器头文件
void delay( unsigned int x ) // 定义延时子程序
{
    int i,j; // 声明整数变量i,j
    for (i=0;i<x;i++) // 计数x次, 延时时间为x*5 ms
        for (j=1;j<=600;j++); // 计数600次, 延迟5 ms
}
void main( ) // 主程序开始
{
    unsigned char i; // 声明无符号数字变量i
    P1=0xfe; // 为P1口送初值
    while(1) // 嵌入死循环
    {
        for(i=0;i<7;i++) // 计数判断左移7次
        {
            P1= (P1<<1) | 0x01; // 左移1位, 并设定最低位为1
            delay(100); // 延时500 ms
        }
        for(i=0;i<7;i++) // 计数判断右移7次
        {
            P1= (P1>>1) | 0x80; // 右移1位, 并设定最高位为1
            delay(100); // 延时500 ms
        }
    }
}

```

7. 任务检测

序号	检测内容	检测结果
1	电源电路工作是否正常	<input type="checkbox"/> 是 <input type="checkbox"/> 否
2	显示电路工作是否正常	<input type="checkbox"/> 是 <input type="checkbox"/> 否
3	微处理器电路工作是否正常	<input type="checkbox"/> 是 <input type="checkbox"/> 否



任务评价

序号	评价项目	分值	评价要素
1	正确性	80	(1) 电气模块接线图绘制 10 分：接线图正确、规范计 10 分，有电气性错误每错 1 处扣 2 分（同类错误计算 1 处），接线图绘制不规范酌情扣 1 ~ 3 分，本项累计最多扣 10 分。 (2) 程序流程图绘制 10 分：流程图表示规范、逻辑清晰计 10 分，有表达或逻辑错误每处扣 2 分（同类错误计算 1 处），本项累计最多扣 10 分。



续表

序号	评价项目	分值	评价要素
1	正确性	80	(3) 电路连接 10 分: 电路连线正确计 10 分, 每错 1 处扣 3 分 (同类错误计算 1 处), 本项累计最多扣 10 分。 (4) 功能实现 40 分: 程序编写后通过硬件验证实现工作任务书中的功能要求, 每正确实现一个功能计 5~10 分, 每部分实现一个功能计 2~5 分。 (5) 性能指标 10 分: 性能指标达到工作任务书的要求计 10 分, 未达到要求每处扣 3 分, 本项累计最多扣 10 分
2	工艺性	10	(1) 操作工艺 3 分: 操作正确、规范计 3 分, 每出现 1 次操作工艺不良情况扣 1 分, 本项累计最多扣 3 分。 (2) 工具、仪表的使用 2 分: 工具、仪表使用正确计 2 分, 每出现 1 次使用错误扣 1 分, 本项累计最多扣 2 分。 (3) 电路装接、布线工艺 5 分: 电路装接、布线工艺合理计 5 分, 每出现 1 次工艺不良情况扣 1 分, 本项累计最多扣 5 分
3	职业与安全意识	10	(1) “7S” 规范 5 分: 符合 “7S” 管理规范计 5 分, 不符合计 0 分。 (2) 文明安全意识 5 分: 保持赛场安静, 尊重赛场工作人员, 爱惜赛场的设备和器材计 5 分, 有违反情况计 0 分

相关知识

一、认识单片机的引脚

8051 单片机采用 40 引脚双列直插封装形式, 部分引脚具有第二功能, 引脚图如图 1-1-6 所示。

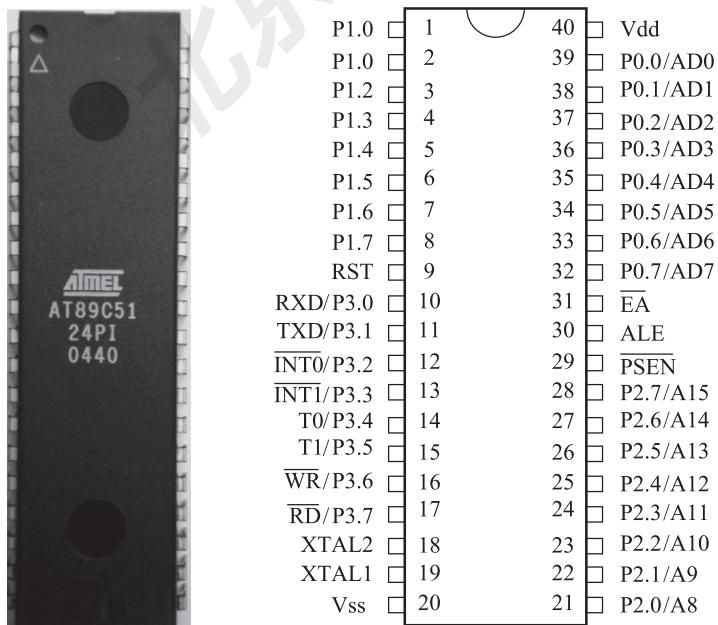


图 1-1-6 8051 单片机引脚图

【写一写】根据 8051 单片机芯片 40 条引脚的定义及简单功能说明, 请对照图 1-1-6 找到相应的引脚名称填写在下表中, 并认真阅读单片机引脚功能介绍。

序号	引脚序号	引脚名称	引脚功能
1	1 ~ 8	P1.0 ~ P1.7 输入 / 输出口	P1 口是一个带有内部上拉电阻的 8 位准双向 I/O 口
2	9		当输入的复位信号延续两个机器周期以上的高电平时即为有效, 用以完成单片机的复位初始化操作
3	10 ~ 17		P3 口是一个带有内部上拉电阻的 8 位准双向 I/O 口, 它具有第二功能
4	18、19		当使用芯片内部时钟时, 此引线端用于外接石英晶体和微调电容; 当使用外部时钟时, 用于接收外部时钟脉冲信号
5	20		接地端
6	21 ~ 28		P2 口是一个带有内部上拉电阻的 8 位准双向 I/O 口
7	29		在读外部 ROM 时, 有效 (低电平), 以实现外部 ROM 单元的读操作
8	30		在系统扩展时, ALE 用于控制把 P0 口输出的低 8 位地址锁存起来, 以实现低位地址和数据的隔离。此外, 由于 ALE 是以晶振 1/6 的固定频率输出的正脉冲, 因此, 可作为外部时钟或外部定时脉冲使用
9	31		当信号为低电平时, 对 ROM 的读操作限定在外部程序存储器; 当信号为高电平时, 对 ROM 的读操作从内部程序存储器开始, 并可延至外部程序存储器
10	32 ~ 39		P0 口是一个 8 位漏极开路的双向 I/O 口
11	40		电源端

8051 单片机 P3 口除了可以作为一般 I/O 口使用外, 每一位都有各自的第二功能, 如表 1-1-1 所示。

表 1-1-1 P3 口各引脚与第二功能

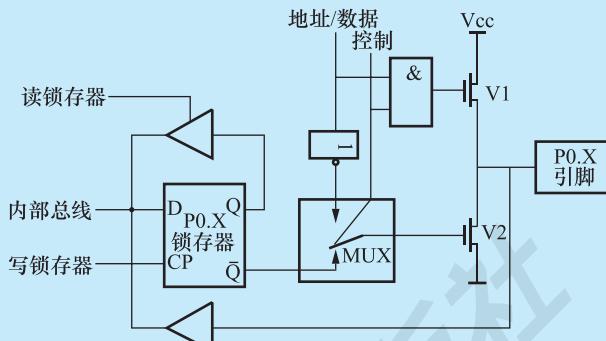
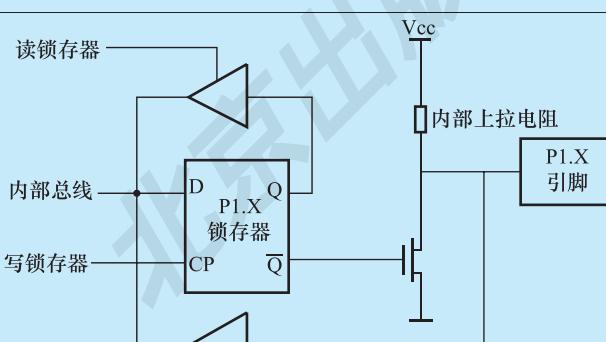
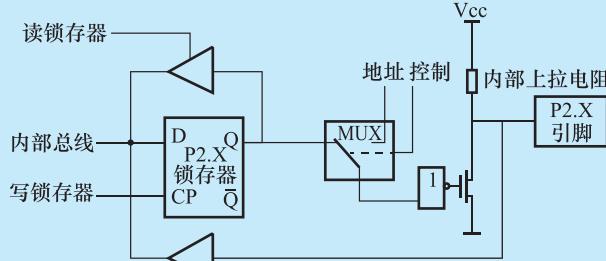
引脚	第二功能	信号名称	引脚	第二功能	信号名称
P3.0	RXD	串行数据接收	P3.4	T0	计数器 0 的外部输入
P3.1	TXD	串行数据发送	P3.5	T1	计数器 1 的外部输入
P3.2	<u>INT0</u>	外部中断 0 输入	P3.6	<u>WR</u>	外部数据 RAM 写选通
P3.3	<u>INT1</u>	外部中断 1 输入	P3.7	<u>RD</u>	外部数据 RAM 读选通



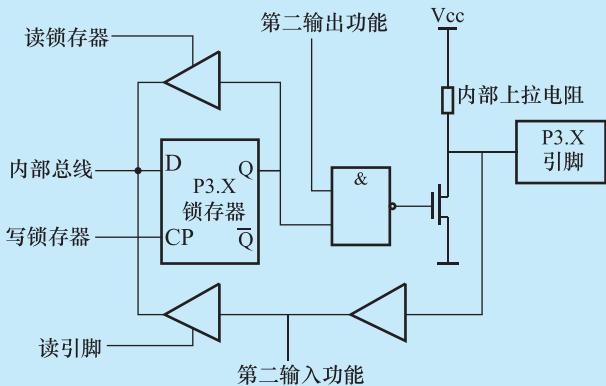
二、认识 51 系列单片机输入 / 输出端口

单片机芯片内还有一项主要内容就是并行 I/O 口。8051 单片机共有 4 个 8 位的并行 I/O 口，分别记作 P0、P1、P2、P3。每个口都包含一个锁存器、一个输出驱动器和输入缓冲器。8051 单片机的 4 个 I/O 口都是 8 位准双向口，这些口在结构和特性上基本相同，但又各具特点，如表 1-1-2 所示。

表 1-1-2 单片机输入 / 输出端口结构

序号	端口名称	端口线逻辑电路	端口使用注意事项
1	P0 口	 <p>读锁存器 内部总线 写锁存器 读引脚</p> <p>地址/数据控制 & - MUX V1 V2 P0.X 引脚</p>	当 P0 口进行一般的 I/O 输出时，由于输出电路是漏极开路电路，因此，必须外接上拉电阻才能有高电平输出；当 P0 口进行一般的 I/O 输入时，必须先向电路中的锁存器写入“1”，使场效应管 FET 截止，以避免锁存器为“0”状态时对引脚读入的干扰
2	P1 口	 <p>读锁存器 内部总线 写锁存器 读引脚</p> <p>Vcc 内部上拉电阻 P1.X 引脚</p>	当 P1 口作为输出口使用时，已经能向外提供推拉电流负载，无须再外接上拉电阻。当 P1 口作为输入口使用时，同样也需先向其锁存器写“1”，使输出驱动电路的 FET 截止
3	P2 口	 <p>读锁存器 内部总线 写锁存器 读引脚</p> <p>地址控制 MUX Vcc 内部上拉电阻 P2.X 引脚</p>	P2 口可以作为通用 I/O 口使用，这时多路转接电路开关倒向锁存器 Q 端。通常情况下，P2 口作为高位地址线使用，此时多路转接电路开关应倒向相反方向

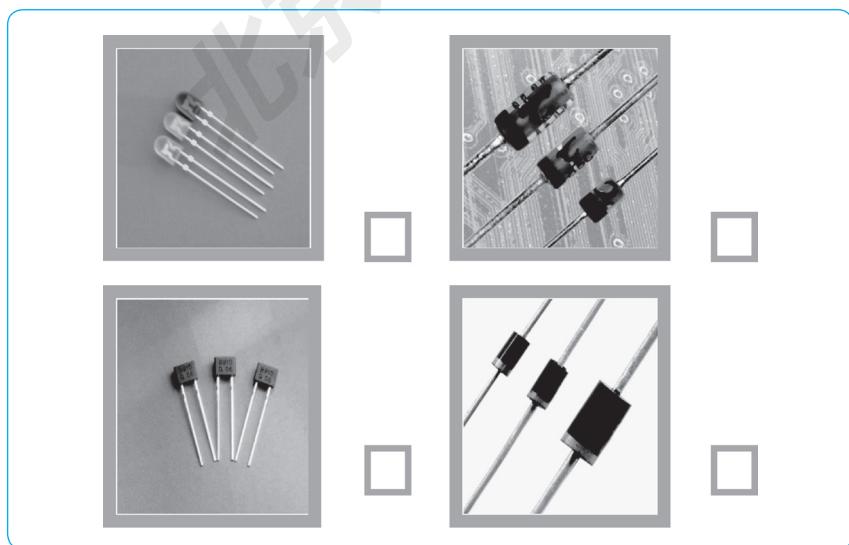
续表

序号	端口名称	端口线逻辑电路	端口使用注意事项
4	P3 口		<p>对于第二功能为输出的信号引脚，当作为 I/O 使用时，第二功能信号引线应保持高电平，与非门开通，以维持从锁存器到输出端数据输出通路的畅通。当输出第二功能信号时，该位的锁存器应置“1”，使与非门对第二功能信号的输出是畅通的，从而实现第二功能信号的输出</p>

三、驱动发光二极管

发光二极管简称 LED，具有体积小、功耗低等特点，常被用以显示信号状态。近年来 LED 技术发展很快，除了能显示红色、绿色、黄色外，还出现了蓝色与白色。高亮度、低功耗的 LED 已经逐渐取代了传统的灯泡，成为交通信号灯的发光器件，就连汽车尾灯也开始流行使用 LED。

【写一写】请利用已有知识，从下列图片中选择出发光二极管的图片。

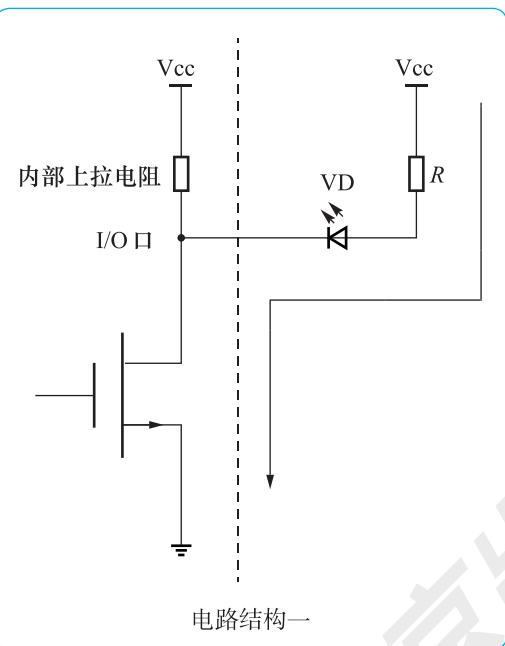


LED 具有二极管的特点。正向导通时会发光；反向偏压时，LED 将截止不发光。以红色 LED 为例，正向导通时 LED 两端电压约为 1.7 V，比普通硅材料二极管的导通电压高很多。

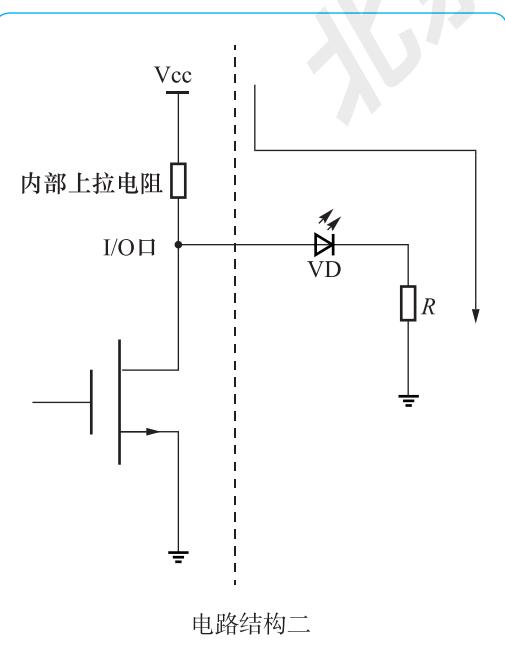


LED 属于电流型元器件, 随着流过电流的增加, LED 的亮度也将增加, 但电流不能过大, 否则会引起 LED 的损毁, 一般导通电流应控制在 $10 \sim 20 \text{ mA}$ 。由于 8051 单片机的输入 / 输出端口属于漏极开路结构, 其中的 P1、P2 与 P3 内部有 $30 \text{ k}\Omega$ 上拉电阻, 因此, 想从 P1、P2 或 P3 输出 $10 \sim 20 \text{ mA}$ 的电流存在困难, 即单片机输出电流的能力较差, 但其 I/O 口却具有较强的输入电流能力。

【写一写】 请利用已有知识, 判断下图中哪一种电路的结构更为合理, 并简述理由。



简述理由



简述理由

如上图所示，当输出低电平时，输出端的场效应管将导通，输出端电压接近 0 V，若发光二极管 LED 正向导通时，两端电压 U_D 为 1.7 V，则限流电阻 R 两端将存在 3.3 V 电压。

【写一写】若想要 LED 更亮一些，可使 I_D 提高到 15 mA，则限流电阻 R 可替换为多少？

如果此时把流过 LED 的电流 I_D 限制在 10 mA，则限流电阻 R 为

$$R = \frac{V_{cc} - U_D}{I_D} = \frac{(5-1.7) \text{ V}}{10 \text{ mA}} = 330 \Omega$$

如果此时把流过 LED 的电流 I_D 限制在 15 mA，则限流电阻 R 为

$$R = \frac{V_{cc} - U_D}{I_D} =$$

对于限制 LED 的限流电阻一般取值为 330 Ω 或 470 Ω ，电阻越小，LED 越亮。

四、创建工程文件

1. Keil C51 简介

Keil C51 是德国知名软件公司 Keil 开发的基于 8051 内核的微控制器软件开发平台，是目前开发 8051 内核单片机的主流工具。Keil C51 支持汇编语言、C 语言等各种开发语言。其中，μVision5 集成开发环境包含项目管理、源代码编辑和强大的程序调试环境。μVision5 调试器是一个强大的全特性调试器，允许用户在 PC 机上完全模拟目标程序、指令集和片内外围功能。



创建工程文件

2. 建立一个 Keil C51 程序

Keil C51 是一个功能强大的软件，但是使用起来并不复杂。接下来通过建立一个简单的 LED（发光二极管）闪烁程序发光的实例来初步掌握 Keil C51 的基本用法。硬件电路如图 1-1-7 所示，单片机 I/O 输出低电平可点亮 LED。

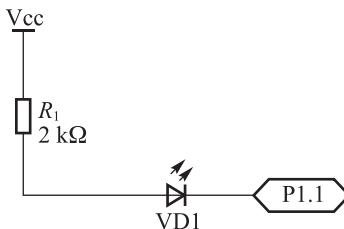


图 1-1-7 LED 闪烁发光电路

(1) 第一步：打开 Keil C51 软件。双击桌面上的“Keil μVision5”图标，启动 Keil C51 程序，启动界面如图 1-1-8 所示。

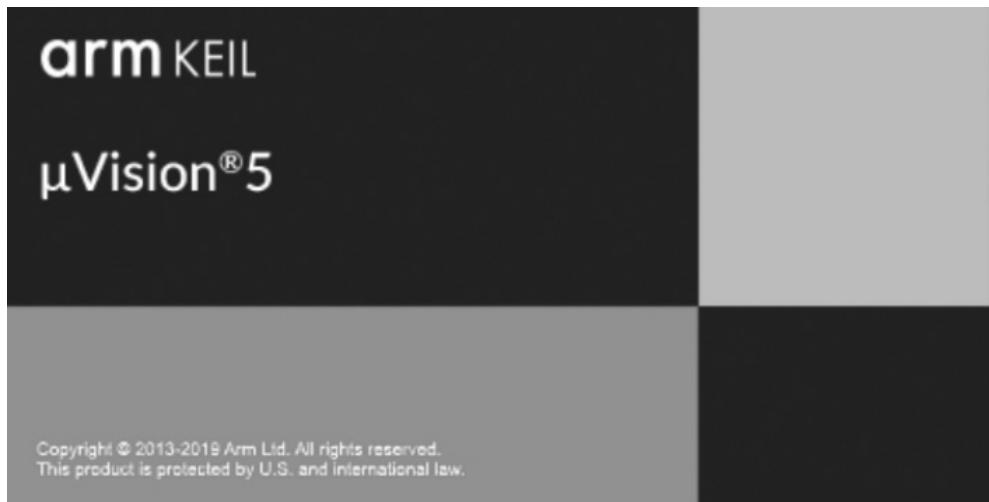


图 1-1-8 Keil C51 的启动界面

打开 Keil μVision5 的主界面如图 1-1-9 所示。

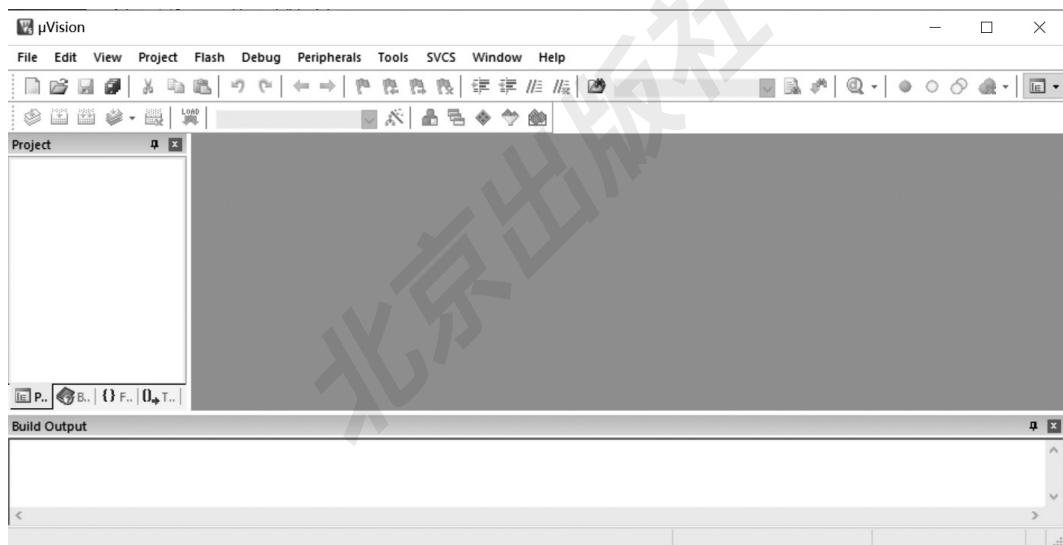


图 1-1-9 Keil μVision5 主界面

(2) 第二步：新建工程。执行 Keil C51 软件的菜单“Project | New μVision Project...”，弹出一个名为“Create New Project”的对话框。先选择一个合适的文件夹来存放工程文件，如“D:\项目一流星灯\led”，其中“led”是新建的文件夹。最后，为工程取名为“led”，并保存。如图 1-1-10 所示。

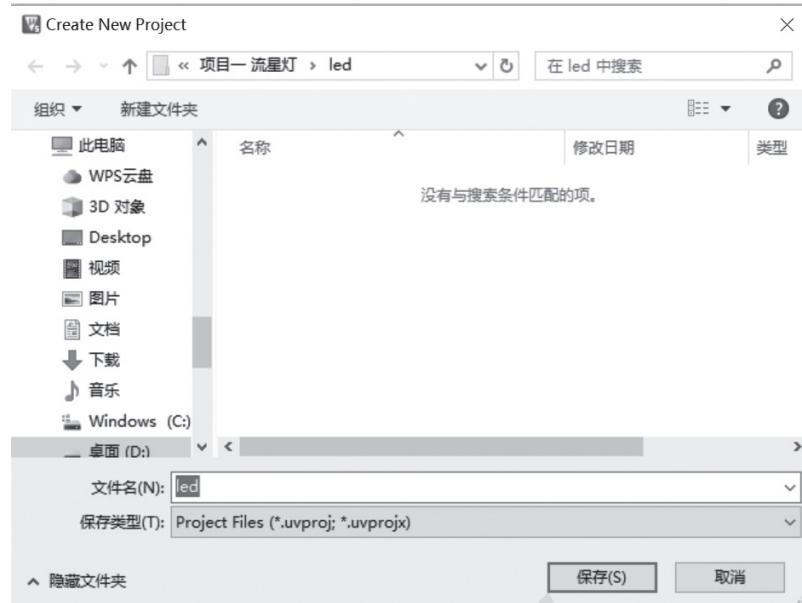


图 1-1-10 新建 Keil C51 工程

(3) 第三步：选择 CPU。紧接着，Keil C51 提示选择 CPU 器件。8051 内核单片机最早是由 Intel 公司开发的，后来其他厂商如 Philips、Atmel、Winbond 等先后推出其兼容产品，并在 8051 的基础上扩展了许多增强功能。在这里可以选择 Atmel 公司的一个器件“AT89C51”，该器件与 Intel 的 8051 单片机完全兼容。如图 1-1-11、图 1-1-12 所示。

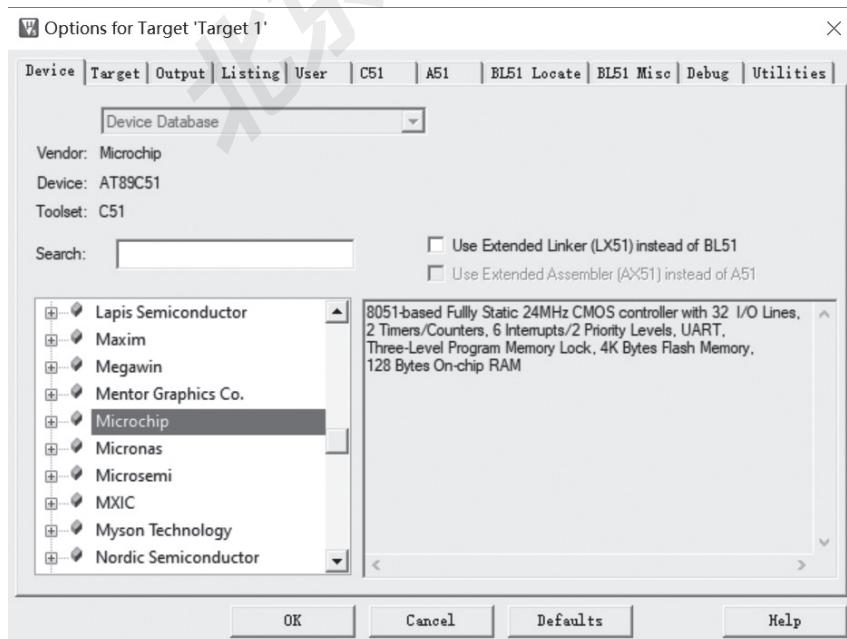


图 1-1-11 选择 CPU

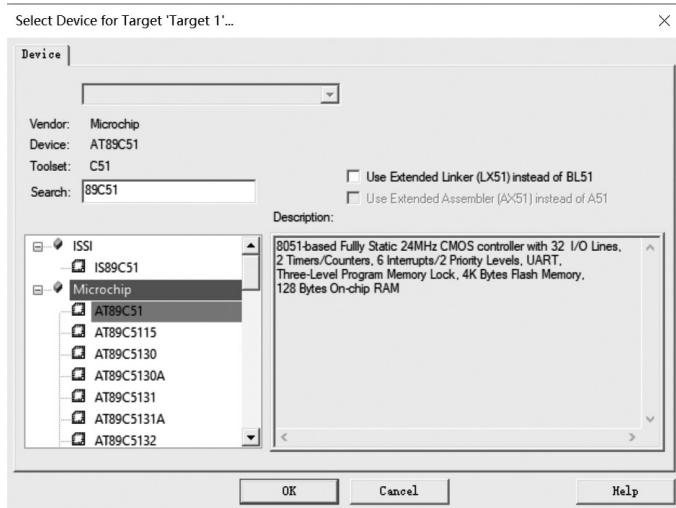


图 1-1-12 选择 AT89C51

接下来，弹出一个如图 1-1-13 所示的对话框。该对话框提示是否要把标准 8051 的启动代码添加到工程中去，可以增强某些功能的初始化配置，一般选择“是(Y)”，即添加启动代码。



图 1-1-13 选择是否要添加启动代码

至此，一个空的 Keil C51 工程建立完成。如图 1-1-14 所示。

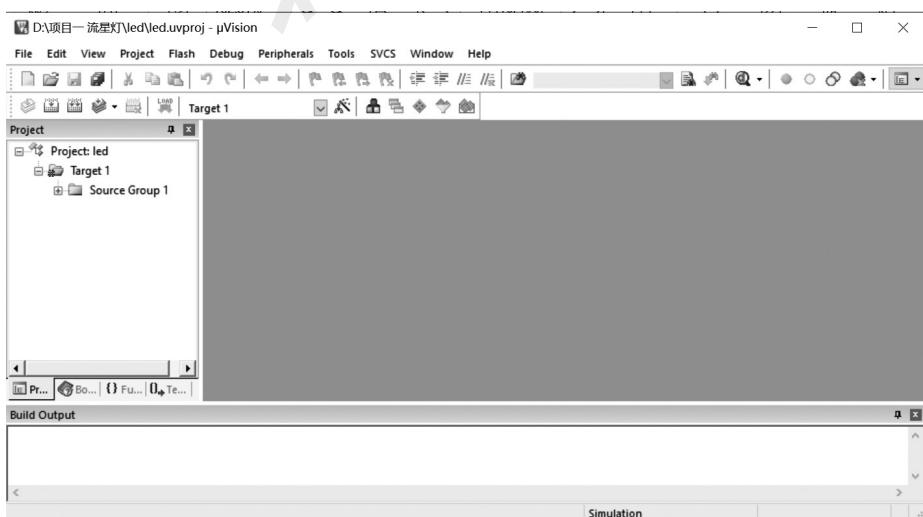


图 1-1-14 空的 Keil C51 工程建立完成

(4) 第四步：设置软件参数。单击“”按钮，设置此芯片的选项，屏幕上出现如图 1-1-15 所示的对话框。在这个对话框里要设置此芯片的工作频率与所需要输出的文件。首先在 Target 选项卡的“Xtal (MHz)”字段输入“12”，表示设置工作频率为 12 MHz。

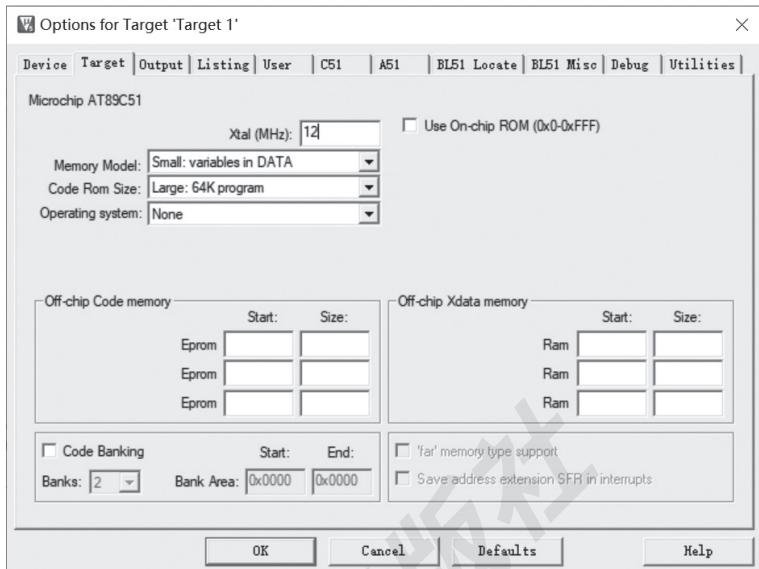


图 1-1-15 Options for Target 'Target 1' 对话框

然后切换到 Output 选项卡，勾选“Create HEX File”复选框，只有选择此项才能产生 16 进制文件 (*.hex)，单击“确定”按钮，关闭此对话框即可完成设置。如图 1-1-16 所示。

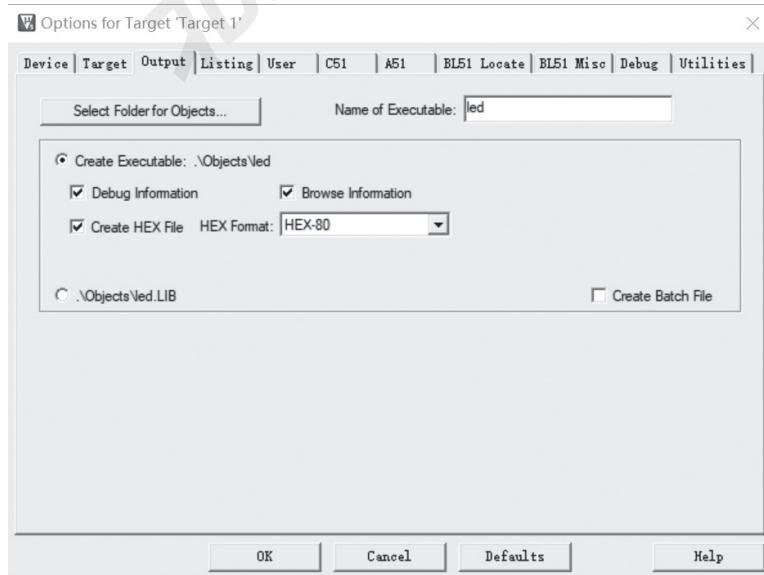


图 1-1-16 Output 选项卡



(5) 第五步: 新建文档文件。执行菜单“File | New...”，出现一个名为“Text n”(其中 n 表示序号)的文档。如图 1-1-17 所示。

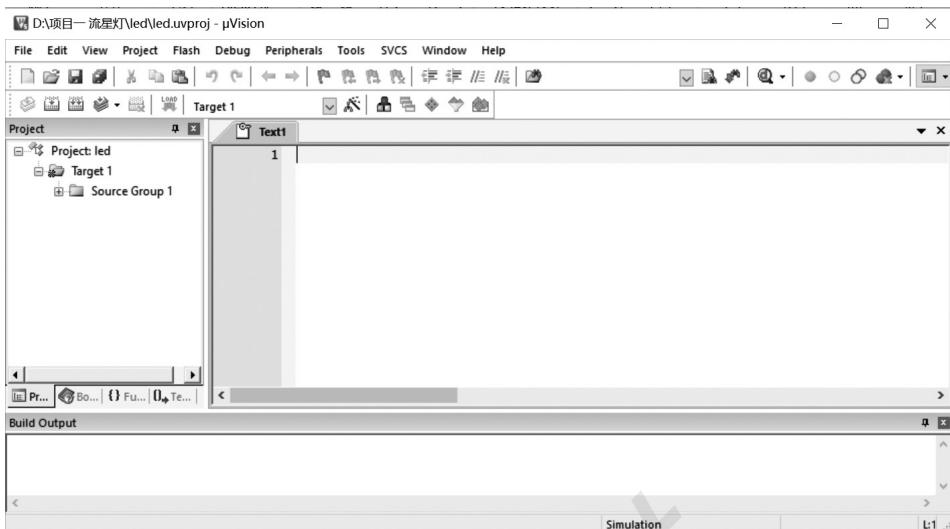


图 1-1-17 新建文档文件

执行菜单“File | Save”，弹出一个名为“Save As”的对话框。将文件名改为“led. c”，然后保存。注意：扩展名“. c”不可省略。如图 1-1-18 所示。

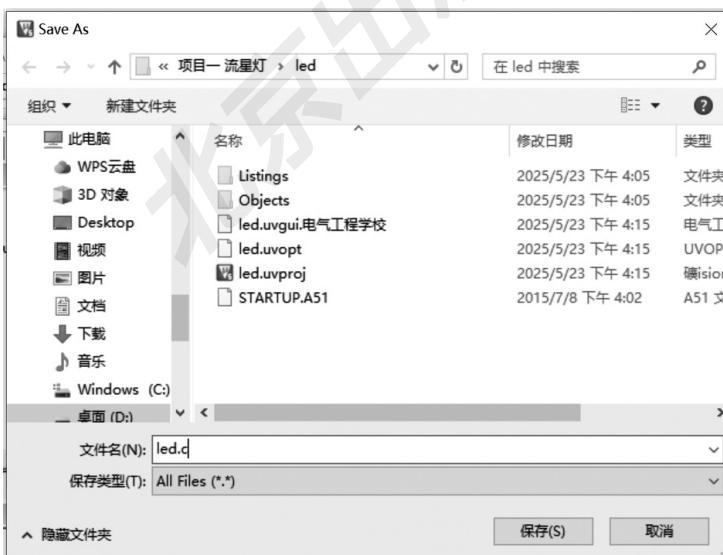


图 1-1-18 保存新建的文件

(6) 第六步: 添加源程序文件到工程中。现在,一个空的源程序文件“led. c”已经建立,但是这个文件与刚才新建的工程之间并没有什么内在联系,我们需要把它添加到工程中去。单击 Keil C51 软件左边项目工作窗口“Target 1”上的“+”，将其展开。然后右键单击“Source Group 1”文件夹,弹出如图 1-1-19 所示的选择菜单。单击其中的“Add

Existing Files to Group ‘Source Group 1’” ... 项。

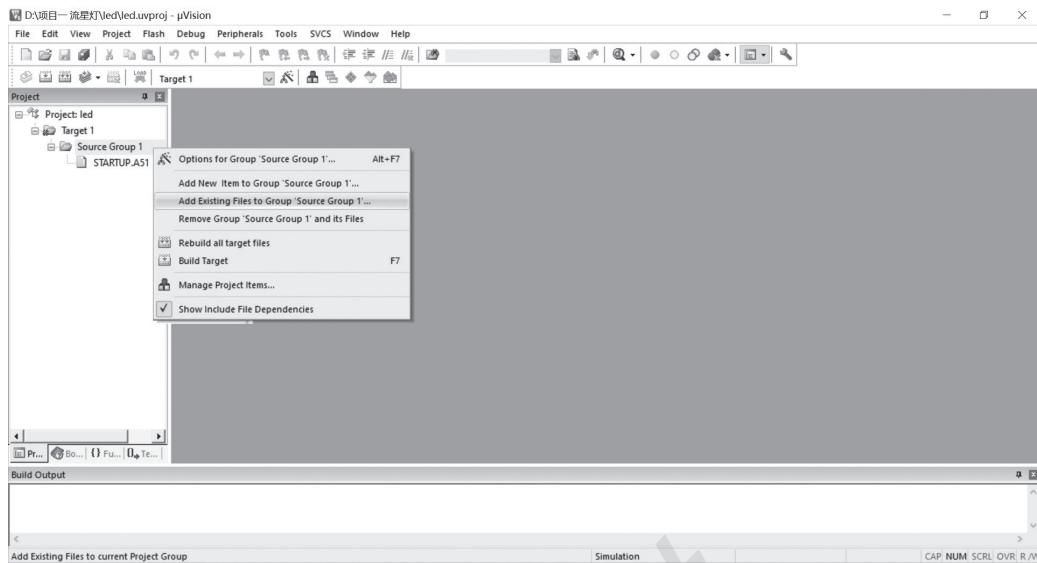


图 1-1-19 准备添加源程序文件到工程中

首先选择文件类型为“C Source file (*.c)”，对话框内将出现刚才保存过的“led.c”文件。选中文件“led.c”，单击“Add”按钮（请不要多次单击“Add”按钮），最后单击“Close”按钮。这时，源程序文件“led.c”已经出现在项目工作窗口的“Source Group 1”文件夹内，可以单击左边的“+”展开后查看。如图 1-1-20、图 1-1-21 所示。

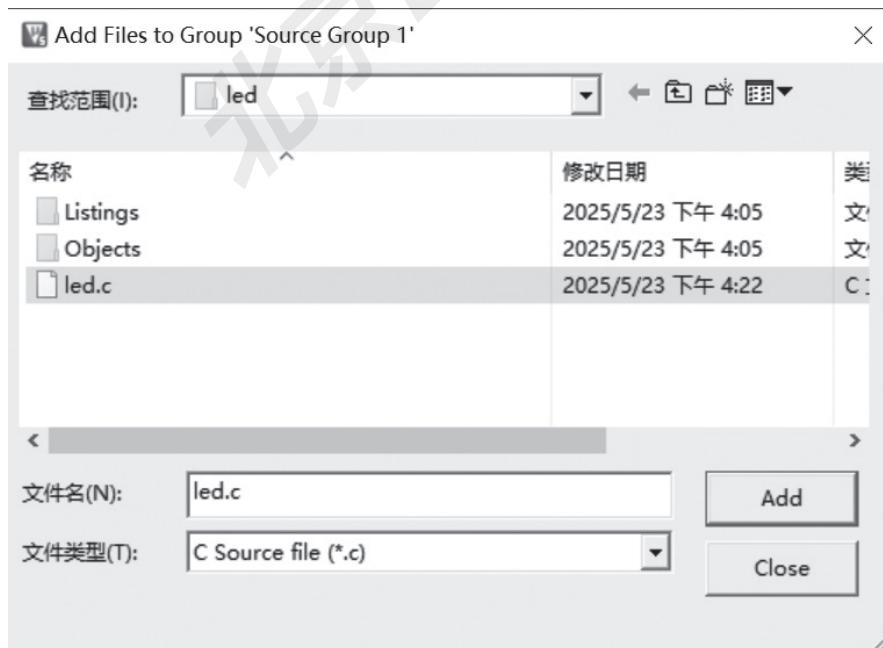


图 1-1-20 向工程中添加源程序文件

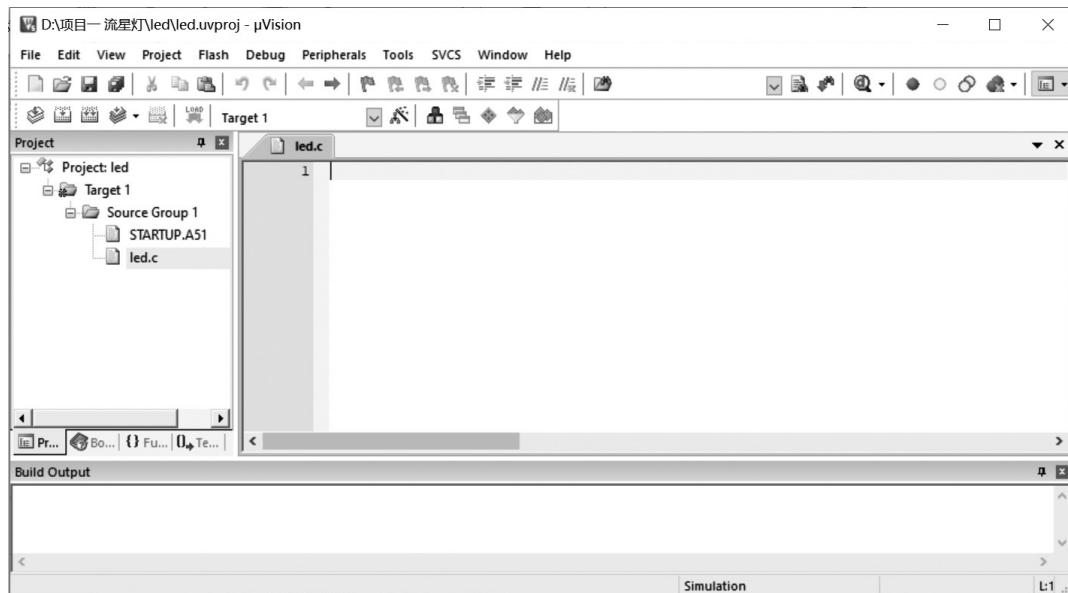


图 1-1-21 展开“Source Group 1”文件夹

(7) 第七步：开始输入源程序。先最大化 led.c 源程序窗口，然后请按以下程序清单输入程序代码。如图 1-1-22 所示。

```
/*****************************************************************************  
程序清单：LED闪烁发光程序led.c  
程序名称：LED闪烁发光  
硬件接法：P1.0控制LED，低电平点亮  
运行效果：LED亮200 ms，灭200 ms，反复循环  
*****  
  
#include <reg51.h> // 包含8051的SFR寄存器定义头文件  
sbit LED = P1^0; // 定义I/O接口  
void Delay(unsigned int x) // 延时子程序  
{  
    int i,j;  
    for (i=0;i<x;i++)  
        for (j=1;j<=600;j++);  
}  
void main() // 主函数  
{  
    while (1)  
    {  
        LED = 0; // 点亮LED  
        Delay(40); // 延时200 ms  
        LED = 1; // 熄灭LED  
        Delay(40); // 延时200 ms
```

```

    }
}

```

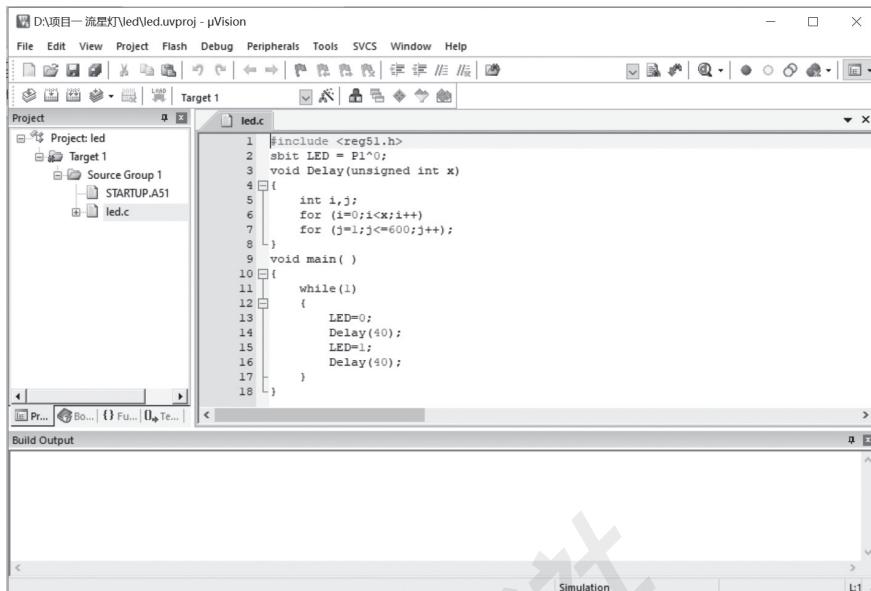


图 1-1-22 在 led.c 文本中输入程序

(8) 第八步：进行编译与链接。单击左上方的“”按钮即可进行编译与链接，而其过程将记录在下方的输出窗口，“0 Error (s)，0 Warning (s)”表示“没有错误，没有警告”，否则就要对程序进行修改直至将错误全部改正为止。如图 1-1-23 所示。

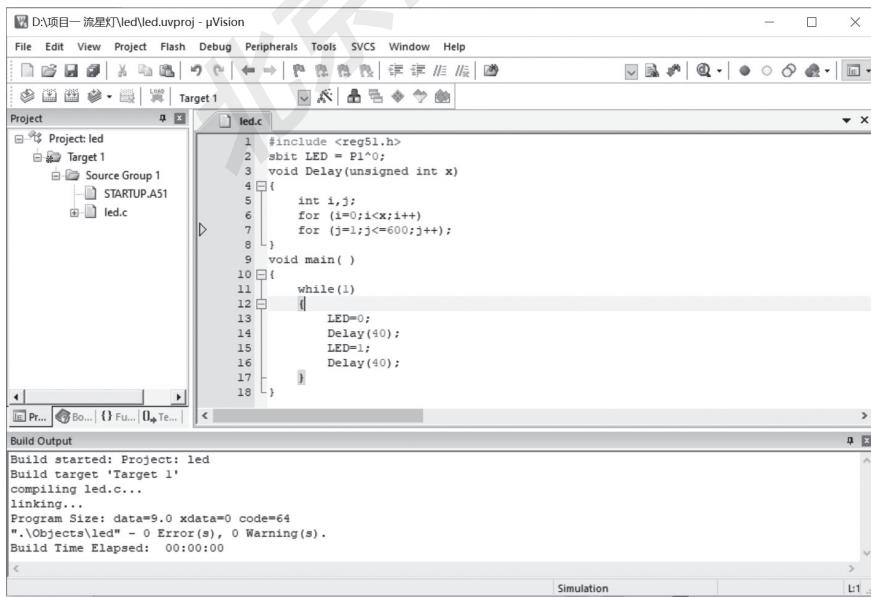


图 1-1-23 编译与链接

接下来进行调试 / 仿真，单击“Debug”菜单中的“Start/Stop Debug Session”命令（或



按“Ctrl+F5”组合键)开始调试工具栏, 页面也将相应发生变化。如图 1-1-24、图 1-1-25 所示。

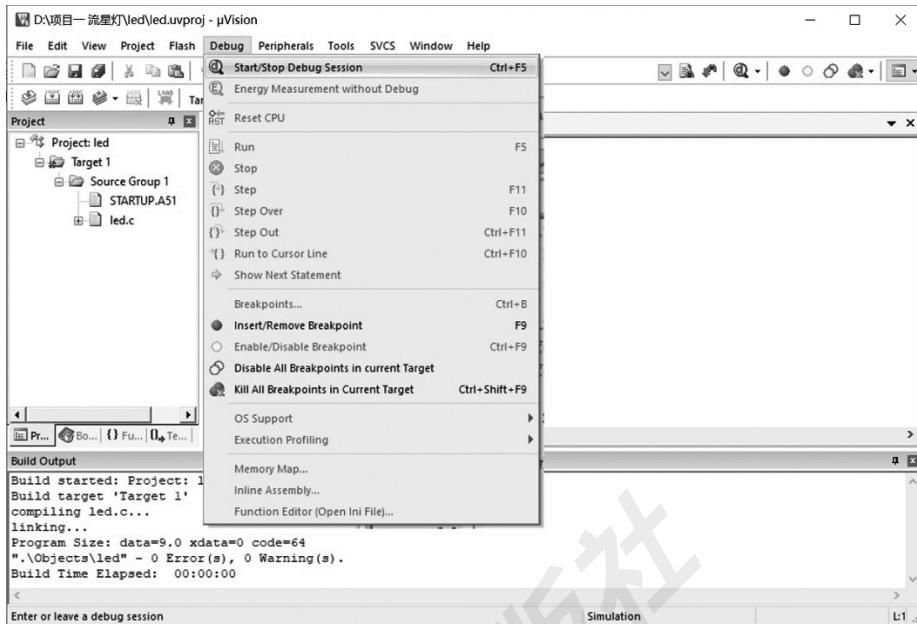


图 1-1-24 单击“Start/Stop Debug Session”命令

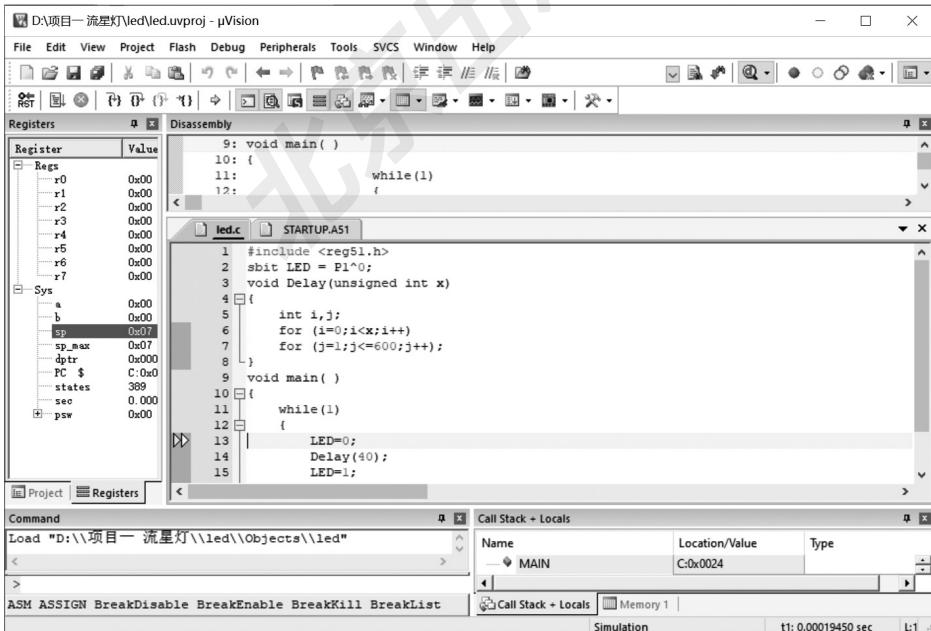


图 1-1-25 调试窗口

单击“Peripherals”菜单中的“I/O-Ports”命令, 再选中“Port 1”选项, 即可打开“Port 1”窗口, 如图 1-1-26、图 1-1-27 所示。

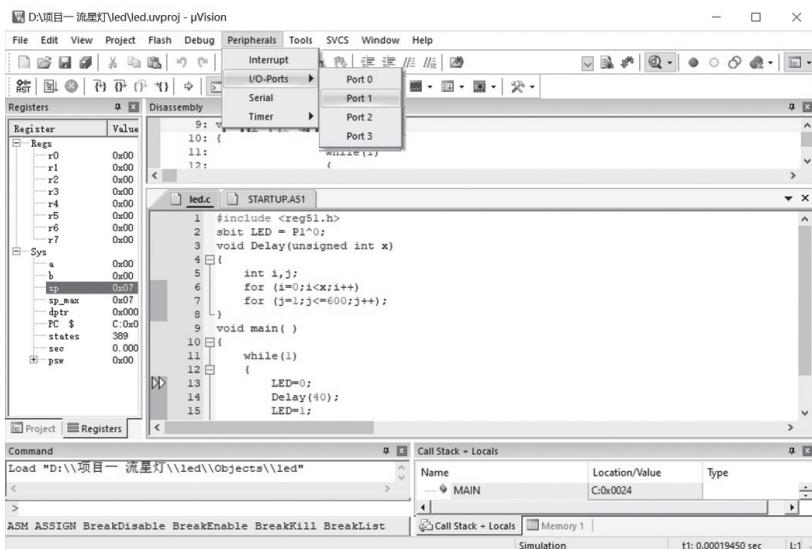


图 1-1-26 Peripherals 菜单

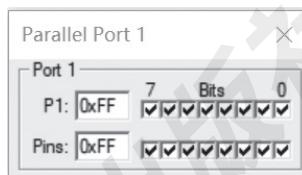


图 1-1-27 Port 1 窗口

单击“”按钮，则监视窗口与 Port 1 窗口的内容将随程序的进行而变化。若想从头开始，则可单击“”按钮停止运行程序，单击“”按钮复位 CPU，再单击“”按钮即可。如图 1-1-28 所示。

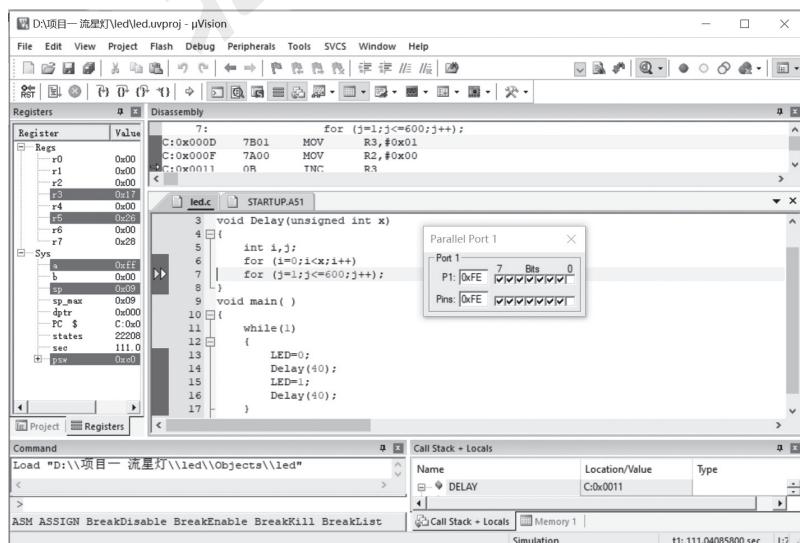


图 1-1-28 程序运行界面



在本项目所保存的文件夹里可以找到“led.hex”文件，这个文件就是可执行文件。在 Proteus ISIS 界面调入所设计的硬件图，双击 CPU，填入相应的 HEX 运行文件的名称，如图 1-1-29 所示，单击运行按钮，即可实现与硬、软件的联合调试。

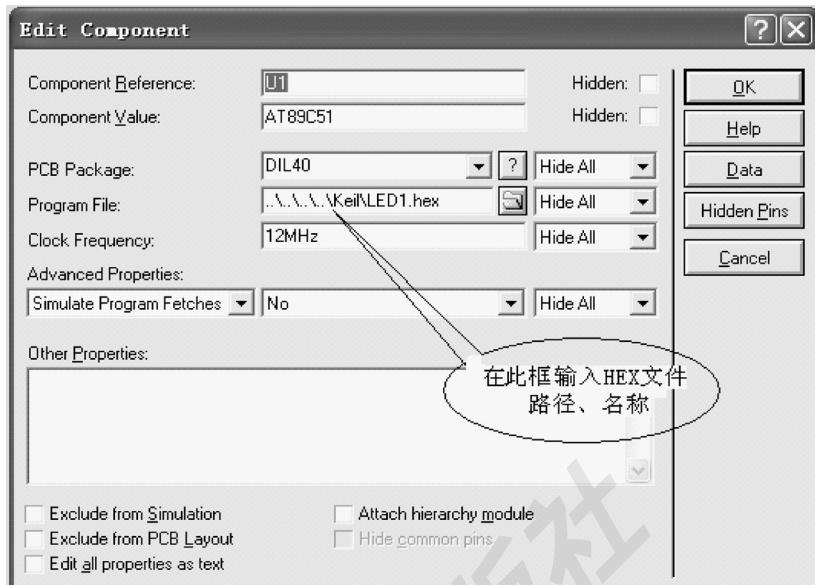


图 1-1-29 单片机程序可执行文件的路径、名称输入

五、编程知识

在单片机程序的编制过程中，无论是简单程序还是复杂程序，编写时都应遵循一定规律，应先按照控制要求根据实际电路和设计思路编写程序流程图。因为程序流程图是程序设计的主线，往往初学者极容易忽略此环节。在此我们选用移植性较高的 C 语言来向大家介绍单片机控制程序的编写方法。

单片机 C 语言程序流程图如图 1-1-30 所示。

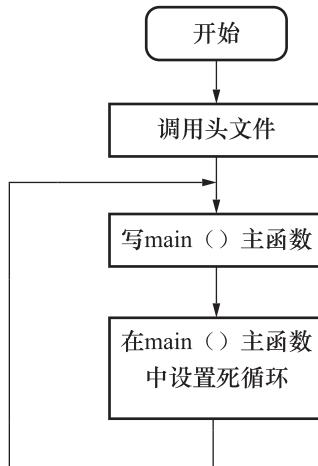


图 1-1-30 单片机 C 语言程序流程图

【记一记】单片机 C 语言程序基本结构框架。

```
#include < reg51.h > // 调用51单片机头文件reg51.h
main( ) // 写main主函数
{
    while(1) // 死循环, 用于保证主程序的正常运行
    {
        // 在此处书写要重复执行的程序语句
    }
}
```



C 语言程序
基本结构框架

在上述程序中使用到了前条件循环“while ()”指令，该指令属于循环指令。这条指令常被用来作为循环条件的判断，与计数循环“for ()”和后条件循环“do…while ()”都是日常编程中最常见的循环指令。这里“while (1) { 程序内容 }”的含义是让在“while (1)”大括号中所包含的内容重复执行。

while () 和 do while () 的使用方法

1. while () 指令将判断条件放在循环体语句之前，称为前条件循环，其格式如下。

```
while(条件表达式)
{
    循环体语句;
}
```

在执行过程中先进行判断，再运行执行语句。循环体语句运行完毕后，自动返回继续判断 while 指令中的条件是否成立，若成立，则继续运行循环体语句；若不成立，则退出循环。

2. do while () 语句提供先执行再判断的功能，称为后条件循环，其格式如下。

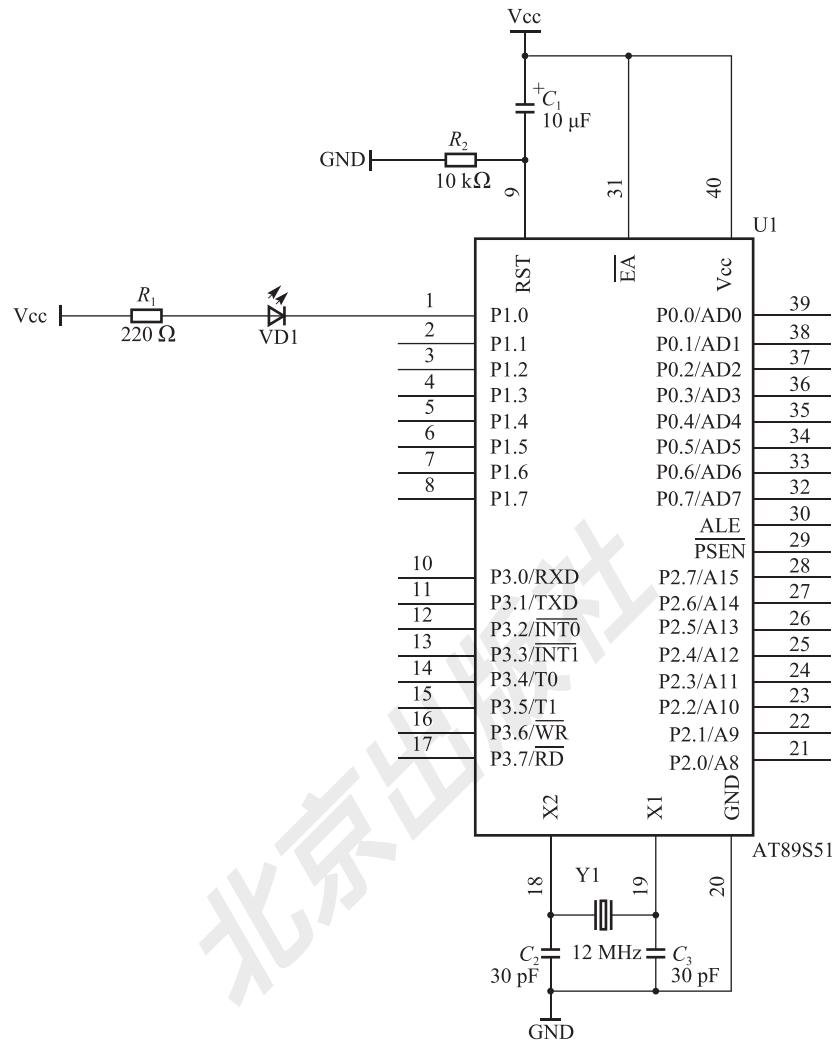
```
do
{
    循环体语句;
}
while(条件表达式);
```

在这个语句中，将先执行一次循环后，再判断表达式是否成立。若不成立，则停止执行该循环；若成立，则继续执行该循环。

在认真阅读完上述内容后，在单片机 P1.0 端口上接一只发光二极管 VD1 使之长亮。



请认真分析电路图,思考应如何实现LED长亮的控制功能。



【写一写】请根据程序的控制要求,自行编制程序流程图。

程序流程图

在对 while 指令有一定程度的了解后，我们编写了下面的实验程序实现上述功能。

【试一试】请对以下程序进行测试，观察并记录发光二极管的状态。

```
#include <reg51.h> // 调用51单片机头文件reg51.h
sbit LED=P1^0; // 定义用LED替换P1.0
main( ) // main主函数
{
    while(1) // 死循环，在主程序中设置死循环
    {
        LED=0; // 主要用于保证主程序的正常运行
        LED=1; // 点亮LED
    }
}
```

现象：

通过阅读程序我们不难发现，要想点亮 LED，只需在单片机和 LED 对应的引脚上输出低电平即可，目前程序实现的功能为控制 LED 长亮，但如果需要使 LED 一亮一灭，应当如何实现呢？请执行下面的程序，观察 LED 现象并加以记录，思考如何修改实现 LED 一亮一灭的控制效果。

【试一试】一只发光二极管点亮、熄灭程序。

```
#include <reg51.h> // 调用51单片机头文件reg51.h
sbit LED=P1^0; // 用LED替代P1.0
main( ) // main主函数
{
    while(1) // 死循环
    {
        LED=0; // 点亮二极管
        LED=1; // 熄灭二极管
    }
}
```

现象：

在本程序中，虽然为 LED 输入低电平使其点亮，但由于 LED 具有辉光效应，人眼无法在较短的时间间隔内正确分辨出 LED 的亮灭，故在编程中通常可以采用延时的方法加以解决，即在程序中加入延时子程序，原因是人眼要想看清 LED 的亮灭变化，需要给人眼一定的反应时间，如果速度过快人眼将无法正常识别。通常当物体移去时，视神经对物体的印象不会立即消失，而要延续 0.1 ~ 0.4 s 的时间，人眼的这种性质被称为眼睛的“视觉暂留”。



【试一试】延时时间为 5 ms 的延时子程序。

```
void Delay(unsigned int x)
{
    unsigned int i, j;
    for (i=0; i<x; i++)
        for (j=1; j<=600; j++);
}
```



延时程序解读

本程序是一个延时时间为 5 ms 的延时子程序，每执行一次本程序就可以达到延时 5 ms 的效果，如果想增加延时时间可以在为变量赋值时增大其数值，但变量的最大数值不能超过 65 535。

【试一试】请讨论如何在 P1.0 端口上接一个发光二极管 VD1，使 VD1 不停地一亮一灭，实现一亮一灭的时间间隔为 0.2 s 的亮灭闪烁效果。



任务拓展

一、单片机内部存储器

表 1-1-3 列出了 MCS-51 单片机系列的芯片型号以及它们的技术性能指标，使我们对单片机内部存储器结构的基本情况有了一个概括的了解。MCS-51 系列又分为 51 和 52 两个子系列，并以芯片型号的最末位数字作为标志。其中，51 子系列是基本型，而 52 子系列则属增强型。

表 1-1-3 单片机内部存储器结构

子系统	片内 ROM 形式			片内 ROM 容量	片内 RAM 容量	寻址范围	I/O 特性			中断源
	无	ROM	EPROM				计数器	并行口	串行口	
51 子系列	8031	8051	8751	4 KB	8 B	2*64 KB	2*16	4*8	1	5
	80C31	80C51	87C51	4 KB	8 B	2*64 KB	2*16	4*8	1	5
52 子系列	8032	8052	8752	8 KB	256 B	2*64 KB	3*16	4*8	1	6
	80C32	80C52	87C52	8 KB	256 B	2*64 KB	3*16	4*8	1	6

二、单片机时钟电路

在 MCS-51 芯片内部有一个高增益反相放大器，其输入端为芯片引脚 XTAL1，其输出端为引脚 XTAL2。而在芯片的外部，XTAL1 和 XTAL2 之间跨接晶体振荡器和微调电容，从而构成一个稳定的自激振荡器，这就是单片机的时钟电路。

时钟电路产生的振荡脉冲经过触发器进行二分频之后，才成为单片机的时钟脉冲信号。一般电容 C_1 和 C_2 取 30pF 左右，晶体的振荡频率范围是 2 ~ 12 MHz，如图 1-1-31 所示。晶体振荡频率高，则系统的时钟频率高，单片机运行速度快。MCS-51 在通常应

用情况下，使用振荡频率为 6 MHz 或 12 MHz。

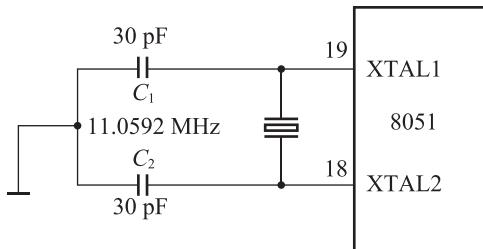


图 1-1-31 单片机时钟电路

三、单片机复位电路

单片机复位是使 CPU 和系统中的其他功能部件都处在一个确定的初始状态，并从这个状态开始工作，无论是在单片机刚开始接上电源时，还是断电后或者发生故障后都需要复位，所以必须弄清楚 MCS-51 单片机复位条件、复位的状态和复位电路。

单片机复位的条件是：必须使 RST/VPD 引脚 9 加上持续两个以上机器周期的高电平。例如，若时钟频率为 12 MHz，则每个机器周期为 1 μ s，那么只需 2 μ s 以上的高电平，在 RST 引脚出现高电平后的第二个机器周期单片机执行复位。单片机复位的状态是：CPU 将进行初始化操作，CPU 与其他功能部件处于一个确定的初始状态，并从这个状态开始工作。复位后 PC 的地址为 0000H，使单片机从第一个单元读取指令。单片机常见的复位电路如图 1-1-32 所示。

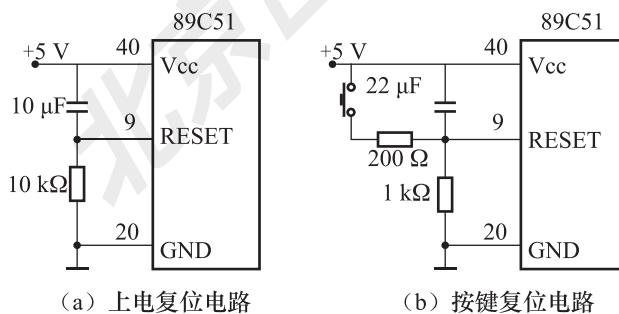


图 1-1-32 复位电路

任务二 ··· 汽车转向灯



任务描述

近年来，北京市汽车保有量逐年攀升，中、高档汽车消费比重也在逐年提高。无论是低档汽车，还是中、高档汽车在电控部分都离不开单片机。从雨刷器的控制到车窗的自动升降，汽车档次越高用到单片机的数量也就越多，控制程序也越复杂。从某种角度讲，



单片机控制电路的复杂程度决定了汽车的整体价格。本任务选取汽车电控部分最为基础的汽车转向灯进行介绍。(图 1-2-1)



图 1-2-1 汽车转向灯

任务要求

1. 电源工作正常: 接上 +5 V 电源, 电源指示灯红灯亮。
2. 继电器控制电路工作正常: 正常通电后, 继电器线圈受单片机 P2 口控制实现断开或吸合效果。
3. 按键设定电路工作正常: 正常通电后, 按键 S1 按下时, 左灯 L1 闪烁; 按键 S2 按下时, 右灯 L2 闪烁; 按键 S3 按下时, L1、L2 停止闪烁。
4. 微处理器及显示电路工作正常: 以上 3 部分正常工作。

任务分析

本任务功能的实现应该从硬件和软件两个方面进行分析。

1. 硬件电路

本任务中硬件电路为单片机控制三极管驱动继电器工作, 显示电路由灯泡作为发光元件, 利用单片机 P2 口控制继电器断开或吸合实现转向闪烁。在硬件电路中应注意继电器工作时的续流保护以及可控外接电路中负载的性质。

2. 软件程序

本任务主要使用单片机 Keil C51 编写控制程序, 利用延时子程序实现单片机驱动控制继电器对灯泡进行控制, 实现转向闪烁的控制效果。

任务实施

1. 识读电路原理图

汽车转向灯电路原理图如图 1-2-2 所示。

2. 配置电路模块

根据电路原理图, 组建该电路可配置 EDM001-MCS51 主机模块、EDM403 按键模块、EDM402 继电器驱动电路模块和 EDM604 直流灯泡模块。

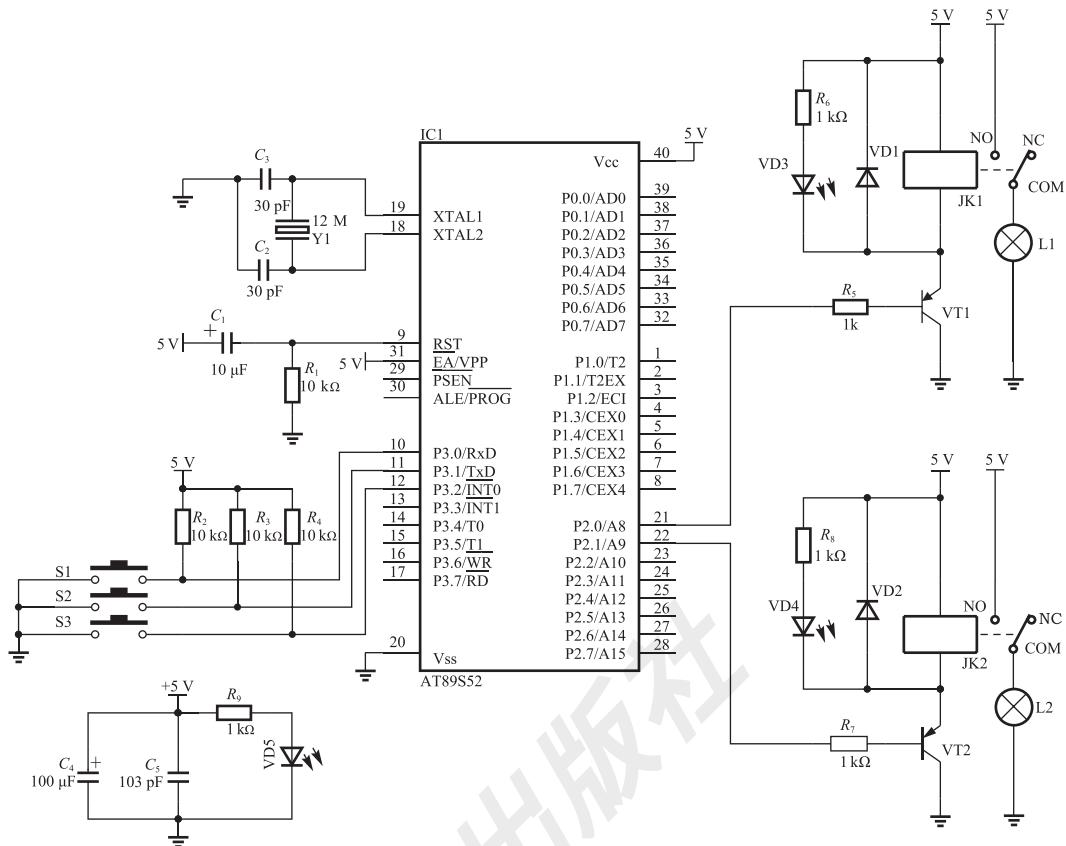


图 1-2-2 汽车转向灯电路原理图

3. 连接实物模块

汽车转向灯电路连接实物图如图 1-2-3 所示。

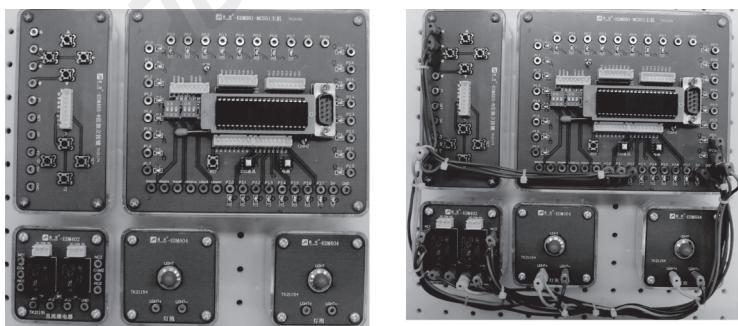


图 1-2-3 汽车转向灯电路连接实物图

实物连接说明如下。

- (1) 该电路用到的 5 个模块都需要连接 +5 V 电源和地。
- (2) EDM403 模块的 \blacktriangleleft \blacktriangleright 插孔分别连接 EDM001-MCS51 主机模块的 P3.0、P3.1、P3.2 插孔。
- (3) EDM402 模块的 JK1、JK2 插孔分别连接 EDM001-MCS51 主机模块的 P2.0、P2.1。



插孔, 实现继电器的控制。两块 EDM604 模块的 LIGHT+ 插孔接电源正极, LIGHT- 接 EDM402 模块的 COM 插孔, EDM402 模块的 NC 插孔接地。

4. 绘制电路框图

根据图 1-2-2 电路原理图, 汽车转向灯电路框图如图 1-2-4 所示。

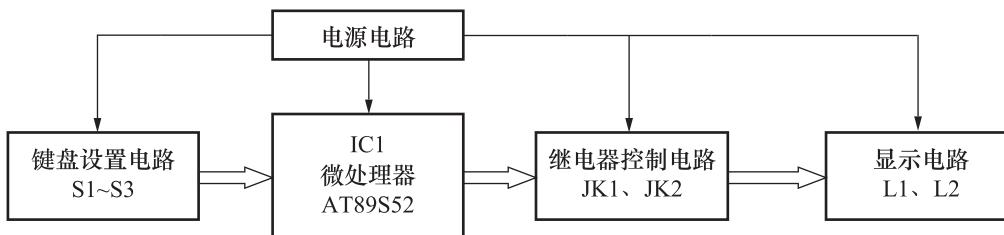


图 1-2-4 汽车转向灯电路框图

5. 编制程序流程图

汽车转向灯程序流程图如图 1-2-5 所示。

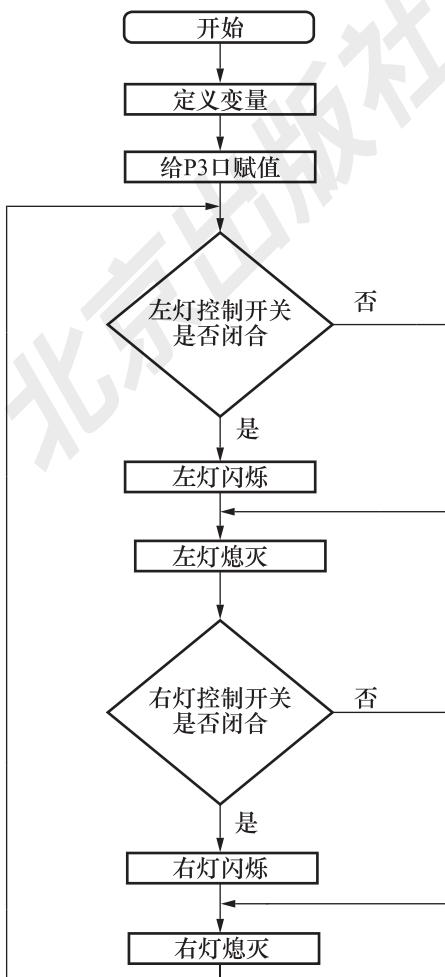


图 1-2-5 汽车转向灯程序流程图

6. 程序设计

```
#include <reg51.h> // 定义8051存储器头文件
#define uchar unsigned char
#define uint unsigned int
sbit relay1=P2^0; // 变量定义
sbit relay2=P2^1;
sbit P30=P3^0; // 定义按键端口
sbit P31=P3^1;
sbit P32=P3^2;
void delay (uint ms) // 定义延时函数
{
    uchar t;
    while(ms--)
    {
        for(t=0;t<120;t++);
    }
}
void left() // 定义左转向灯闪烁程序
{
    if(P30==0&&P31==1) // 判断左转向开关是否到位
    {
        relay1=1; // 控制继电器开合实现LED灯闪烁
        delay(400);
        relay1=0;
        delay(400);
    }
    else // 继电器打开向左LED灯熄灭
    {
        relay1=1;
    }
}
void right() // 定义右转向灯闪烁程序
{
    if(P32==0&&P31==1) // 判断右转向开关是否到位
    {
        relay2=1; // 控制继电器开合实现LED灯闪烁
        delay(400);
        relay2=0;
        delay(400);
    }
    else // 继电器打开向右LED灯熄灭
    {
        relay2=1;
    }
}
void main ( ) // 定义主函数
{
    P3=0xff;
    while(1) // 实现死循环
    {
    }
}
```



```

    lift(    );
    right(   );
}

}

```

// 调用左转向灯闪烁程序
// 调用右转向灯闪烁程序

7. 任务检测

序号	检测内容	检测结果	
1	电源电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否
2	显示电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否
3	微处理器电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否
4	继电器控制电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否



任务评价

序号	评价项目	分值	评价要素
1	正确性	80	(1) 电气模块接线图绘制 10 分：接线图正确、规范计 10 分，有电气性错误每错 1 处扣 2 分（同类错误计算 1 处），接线图绘制不规范酌情扣 1~3 分，本项累计最多扣 10 分。 (2) 程序流程图绘制 10 分：流程图表示规范、逻辑清晰计 10 分，有表达或逻辑错误每处扣 2 分（同类错误计算 1 处），本项累计最多扣 10 分。 (3) 电路连接 10 分：电路连线正确计 10 分，每错 1 处扣 3 分（同类错误计算 1 处），本项累计最多扣 10 分。 (4) 功能实现 40 分：程序编写后通过硬件验证实现工作任务书中的功能要求，每正确实现一个功能计 5~10 分，每部分实现一个功能计 2~5 分。 (5) 性能指标 10 分：性能指标达到工作任务书的要求计 10 分，未达到要求每处扣 3 分，本项累计最多扣 10 分
2	工艺性	10	(1) 操作工艺 3 分：操作正确、规范计 3 分，每出现 1 次操作工艺不良情况扣 1 分，本项累计最多扣 3 分。 (2) 工具、仪表的使用 2 分：工具、仪表使用正确计 2 分，每出现 1 次使用错误扣 1 分，本项累计最多扣 2 分。 (3) 电路装接、布线工艺 5 分：电路装接、布线工艺合理计 5 分，每出现 1 次工艺不良情况扣 1 分，本项累计最多扣 5 分
3	职业与安全意识	10	(1) “7S” 规范 5 分：符合 “7S” 管理规范计 5 分，不符合计 0 分。 (2) 文明安全意识 5 分：保持赛场安静，尊重赛场工作人员，爱惜赛场的设备和器材计 5 分，有违反情况计 0 分



相关知识

一、驱动继电器

继电器是一种电子控制器件，它实际上是用较小电流去控制较大电流的一种自动开关，故在电路中起着自动调节、安全保护、转换电路等作用。继电器所使用的工作电压

有 DC 12 V、DC 9 V、DC 5 V 等, 如图 1-2-6 所示。

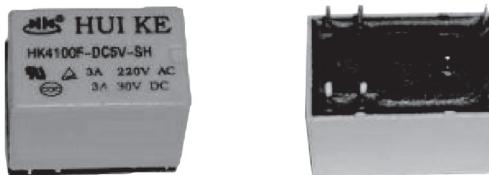


图 1-2-6 继电器实物图

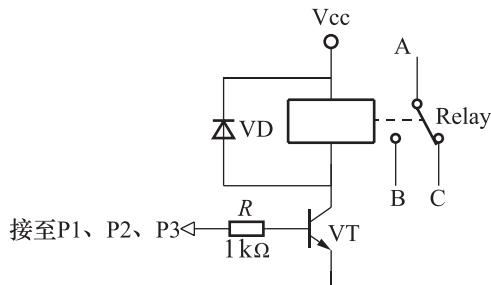
【写一写】请画出图 1-2-6 中继电器的内部电路结构。

二、单片机对继电器的控制

当控制信号为高电平时, 继电器常开触点吸合 (AB 导通); 当控制信号为低电平时, 继电器常开触点断开, 常闭触点吸合 (AC 导通), 驱动多个输出端口, 可使用 7405 驱动 5 V 继电器、7406 驱动最高电压为 30 V 的继电器。

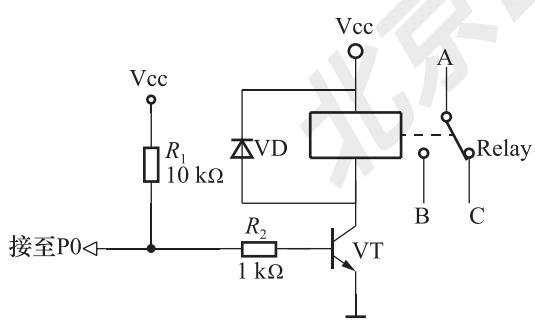


【写一写】请简要分析以下两种电路的结构特点、元器件作用及控制方式。



电路结构 (1)

简述理由

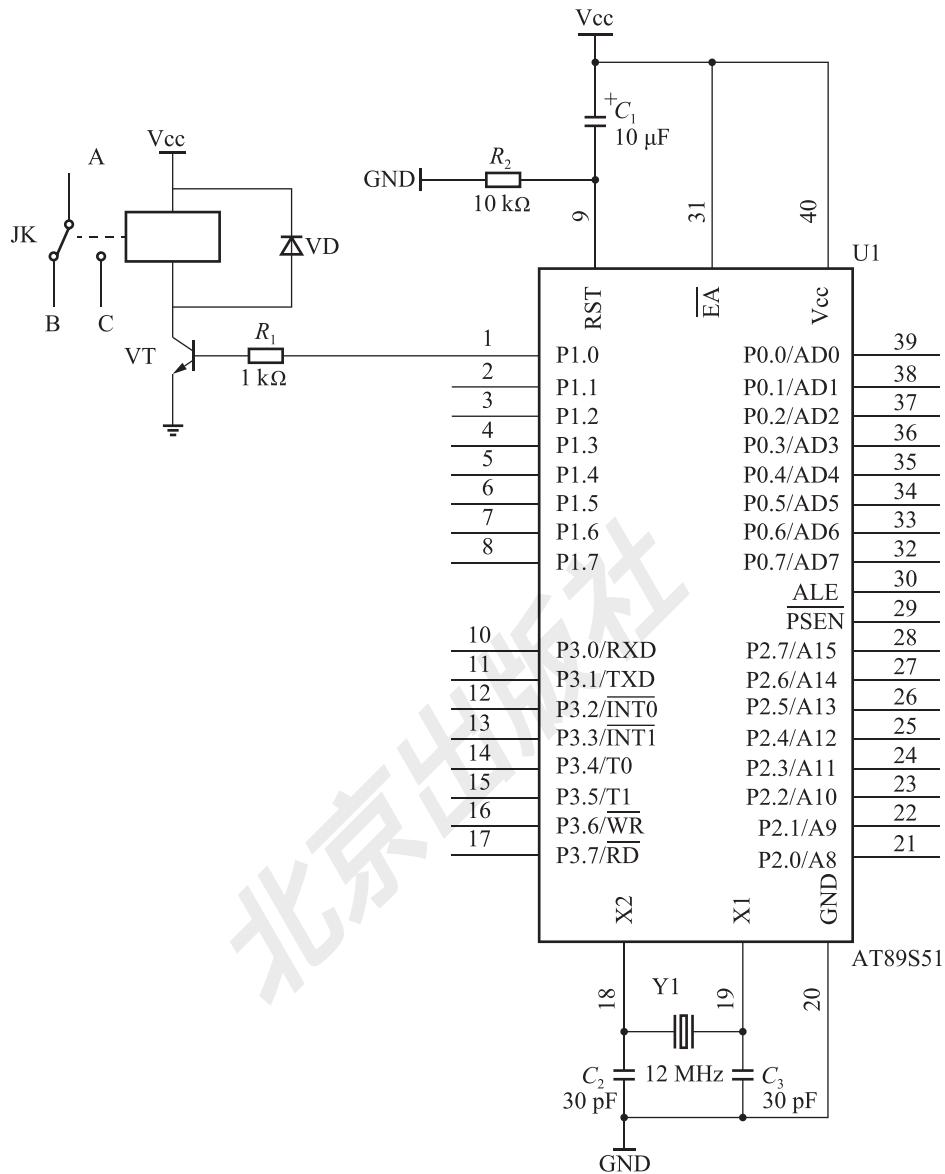


电路结构 (2)

简述理由

当单片机接口电路输入高电平信号，三极管 VT 将导通使继电器线圈得电。若接口电路输入低电平信号，三极管 VT 将截止，继电器线圈不得电。所以只要通过单片机 I/O 端口输出高、低电平信号到三极管基极就能够控制继电器的工作情况。当单片机输出高电平信号时，继电器线圈得电，触头吸合；当单片机输出低电平信号时，继电器线圈不得电，触头断开。

请认真分析下面电路原理图，思考如何用单片机实现对继电器的开关控制。



请认真阅读下面继电器开关程序。

【记一记】继电器开关程序（三极管采用 NPN 型）。

```
#include <reg51.h> // 调用51单片机头文件reg51.h
sbit KA=P1^0; // 定义用LED替换P1.0
main( ) // main主函数
{
    while(1) // 死循环
```



```
{  
    KA=0; // 继电器断开  
    KA=1; // 继电器吸合  
}
```

从电路中可以看出单片机 I/O 口控制不同类型三极管与继电器连接时，单片机对继电器输出控制信号截然不同，请认真思考并完成下面的内容。

当继电器线圈的驱动三极管 VT 由 NPN 型转换成 PNP 型，请自行绘制电路原理图，并编写与电路对应的继电器开关控制程序。

【想一想】【写一写】请画出 P1 口与继电器连接的电路图 (PNP 型三极管驱动)，并编写开关控制程序。

电路图

源程序

在上述程序中使用到了计数循环 for () 指令，该指令属于循环指令，这条指令常被用来作为循环条件的判断与计数。

for () 语句的使用方法

```
for(表达式1;表达式2;表达式3)  
{  
    循环体语句;  
}
```

for 循环的语句执行过程如下。

1. 先对表达式 1 赋初值，进行初始化。
2. 判断表达式 2 是否满足给定的循环条件，若满足循环条件，则执行循环体内语句，然后执行第三步；若不满足循环条件，则结束循环，转到第五步。
3. 若表达式 2 为真，则在执行指定的循环语句后，求表达式 3。

4. 回到第二步继续执行。
5. 退出 for 循环，执行下一条语句。

最后对 for 循环语句的三种特例进行说明：

- (1) for 语句中的小括号内的 3 个表达式全部为空。

```
for( ; ; )  
{ 循环体语句 }
```

在小括号内只有两个分号，无表达式。

这意味着没有设初值，无判断条件，循环变量为增值，这将导致一个死循环，其效果与 while(1) 相同。

- (2) for 语句 3 个表达式中，有一个或两个缺省。

① 表达式 1 缺省

```
for ( ; i<=50 ; i++) sum=sum+i ;
```

即不对 i 设初值。

② 表达式 2 缺省

```
for (i=0;;i++) sum=sum+i;
```

即不判断循环条件，认为表达式始终为真，循环将无休止地进行下去。

③ 表达式 1, 3 缺省

```
for ( ;i<=100; )  
{ sum= sum+i;i++ ;  
}
```

- (3) 没有循环体的 for 语句。

```
int sum=1000;  
for (i=0 ; i<sum ; i++) ;
```

没有循环体的 for 语句一般在程序中起到延时的作用。

下面是使用 for 语句和 while 语句编制的一个延时时间为 1 ms 的延时子程序。当给传递变量 x 送入不同数值时，可以产生不同的延时时间，本程序最大的延时时间为 65.535 s。



【试一试】延时 1 ms 为单位的延时程序。

```
void delay(unsigned int x)
{
    unsigned char y;
    while(x--)
    {
        for(y=0 ; y<125 ; y++)
        { ; }
    }
}
```

心得体会：



任务拓展

单片机头文件的指定

51 单片机经常使用的头文件有 reg51.h、reg52.h 或 AT89x51.h 等，对于指定头文件的方式通常有以下两种。

1. 在 #include 之后，以 <> 包含头文件的文件名，如下。

格式：#include <头文件文件名>

例如：#include <reg51.h>

若采用这种方式，编译程序将从 Keil μVision5 的头文件文件夹中查找所指定的头文件。如果 Keil μVision5 安装在 C 盘根目录下，则编译程序将从 C:\Keil\C51\INC 路径中查找它。

2. 在 #include 之后，以 “” 包含头文件的文件名，如下。

格式：#include “头文件文件名”

例如：#include “reg51.h”

若采用这种方式，编译程序将从源程序所在文件夹中查找所使用的头文件。

对于用户自定义头文件或由其他公司提供的单片机头文件，我们在使用前将其拷贝到 INC 文件夹里就可以正常使用了。

```
-----  
REG51.H  
Header file for generic 80C51 and 80C31 microcontroller.  
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.  
All rights reserved.  
-----  
#ifndef __REG51_H__  
#define __REG51_H__  
/* BYTE Register */
```

```
sfr P0 = 0x80;
sfr P1 = 0x90;
sfr P2 = 0xA0;
sfr P3 = 0xB0;
sfr PSW = 0xD0;
sfr ACC = 0xE0;
sfr B = 0xF0;
sfr SP = 0x81;
sfr DPL = 0x82;
sfr DPH = 0x83;
sfr PCON = 0x87;
sfr TCON = 0x88;
sfr TMOD = 0x89;
sfr TL0 = 0x8A;
sfr TL1 = 0x8B;
sfr TH0 = 0x8C;
sfr TH1 = 0x8D;
sfr IE = 0xA8;
sfr IP = 0xB8;
sfr SCON = 0x98;
sfr SBUF = 0x99;
/* BIT Register */
/* PSW */
sbit CY = 0xD7;
sbit AC = 0xD6;
sbit F0 = 0xD5;
sbit RS1 = 0xD4;
sbit RS0 = 0xD3;
sbit OV = 0xD2;
sbit P = 0xD0;
/* TCON */
sbit TF1 = 0x8F;
sbit TR1 = 0x8E;
sbit TF0 = 0x8D;
sbit TR0 = 0x8C;
sbit IE1 = 0x8B;
sbit IT1 = 0x8A;
sbit IE0 = 0x89;
sbit IT0 = 0x88;
/* IE */
sbit EA = 0xAF;
sbit ES = 0xAC;
sbit ET1 = 0xAB;
sbit EX1 = 0xAA;
sbit ET0 = 0xA9;
sbit EX0 = 0xA8;
/* IP */
```



```
sbit PS = 0xBC;  
sbit PT1 = 0xBB;  
sbit PX1 = 0xBA;  
sbit PT0 = 0xB9;  
sbit PX0 = 0xB8;  
/* P3 */  
sbit RD = 0xB7;  
sbit WR = 0xB6;  
sbit T1 = 0xB5;  
sbit T0 = 0xB4;  
sbit INT1 = 0xB3;  
sbit INT0 = 0xB2;  
sbit TXD = 0xB1;  
sbit RXD = 0xB0;  
/* SCON */  
sbit SM0 = 0x9F;  
sbit SM1 = 0x9E;  
sbit SM2 = 0x9D;  
sbit REN = 0x9C;  
sbit TB8 = 0x9B;  
sbit RB8 = 0x9A;  
sbit TI = 0x99;  
sbit RI = 0x98;  
#endif
```

任务三 智能感应迎宾灯

任务描述

智能感应迎宾灯是应用于中高档轿车的一种电子产品。它的前身是电子防盗报警器，主要作用是当光线较为昏暗时，汽车感应到用户接近车身时，为了便于用户开启车门时看清周围事物而添加的一种辅助电子产品。（图 1-3-1）



图 1-3-1 智能感应迎宾灯

任务要求

1. 电源工作正常：接上 +5 V 电源，电源指示灯红灯亮。
2. 热释电检测电路工作正常：正常通电后，热释电传感器 PIR 能正常检测 50 cm 范围内的红外线辐射信号。
3. 红外反射检测电路工作正常：正常通电后，当被测物体距红外反射传感器小于 15 cm 时，红外反射检测电路能输出检测到物体的报警信号。
4. 继电器电路工作正常：正常通电后，继电器 JK 能接收来自单片机的控制信号完成断开或吸合操作。
5. 微处理器及照明电路工作正常：以上 4 部分正常工作。

任务分析

本任务功能的实现应该从硬件和软件两个方面进行分析。

1. 硬件电路

本任务由信号输入电路、单片机微控制器电路和照明电路三部分组成，在硬件电路中主要学习热释电传感器和红外反射传感器的控制使用方法。

2. 软件程序

本任务主要使用单片机 Keil C51 编写控制程序，实现智能感应迎宾灯的控制效果。

任务实施

1. 识读电路原理图

智能感应迎宾灯电路原理图如图 1-3-2 所示。

2. 配置电路模块

根据电路原理图，组建该电路可配置 EDM001-MCS51 主机模块、EDM402 继电器驱动电路模块、EDM604 直流灯泡模块、EDM107 热释电电路模块、EDM112 红外反射模块。

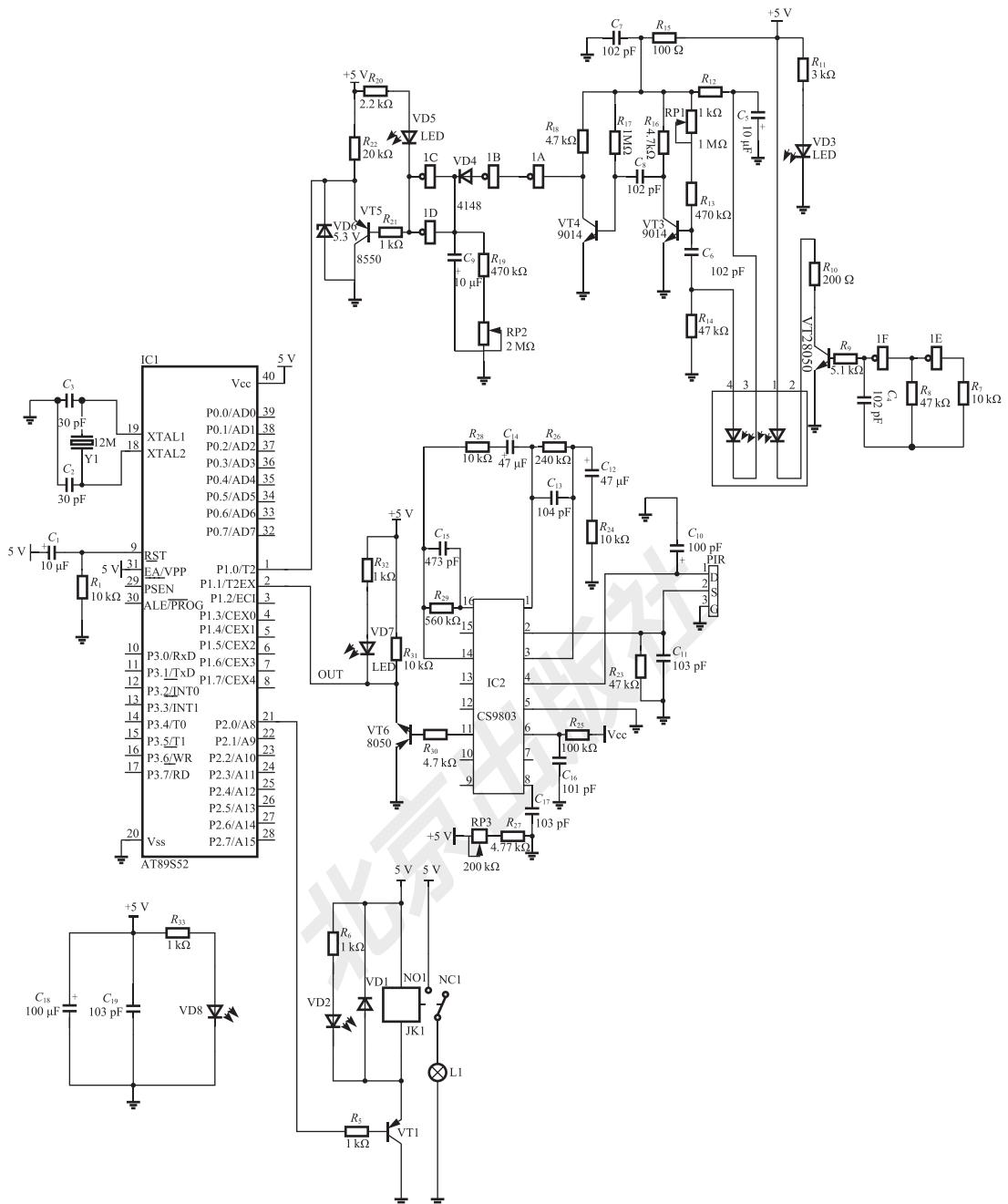


图 1-3-2 智能感应迎宾灯电路原理图

3. 连接实物模块

智能感应迎宾灯电路连接实物图如图 1-3-3 所示。

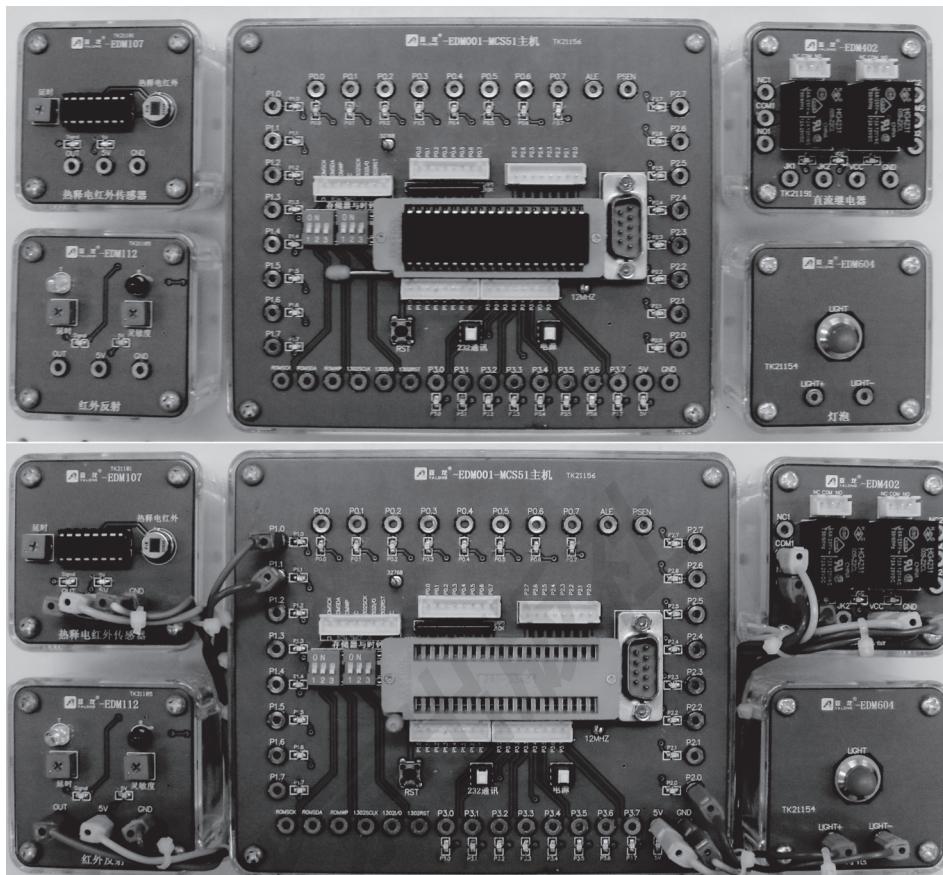


图 1-3-3 智能感应迎宾灯电路连接实物图

实物连接说明如下。

(1) 各模块均接 +5 V 电源和地。

(2) EDM107 模块的 OUT 插孔连接 EDM001-MCS51 主机的 P1.1 插孔，EDM112 模块的 OUT 插孔连接 EDM001-MCS51 主机的 P1.0 插孔，EDM402 模块的 COM1 插孔连接 +5 V 电源插孔，JK1 插孔连接 EDM001-MCS51 主机的 P2.0 插孔，NO1 插孔连接 EDM604 模块的 LIGHT+ 插孔，EDM604 模块的 LIGHT- 插孔接地。

4. 绘制电路框图

根据图 1-3-2 所示电路原理图，智能感应迎宾灯电路框图如图 1-3-4 所示。

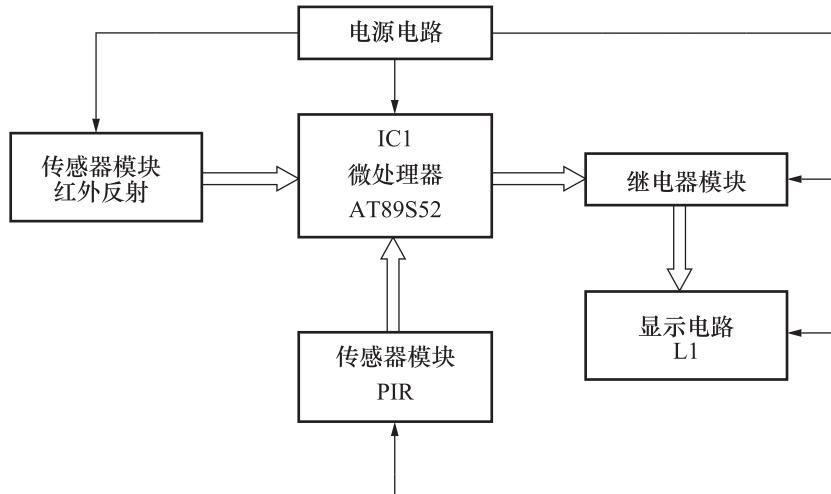


图 1-3-4 智能感应迎宾灯电路框图

5. 编制程序流程图

智能感应迎宾灯程序流程图如图 1-3-5 所示。

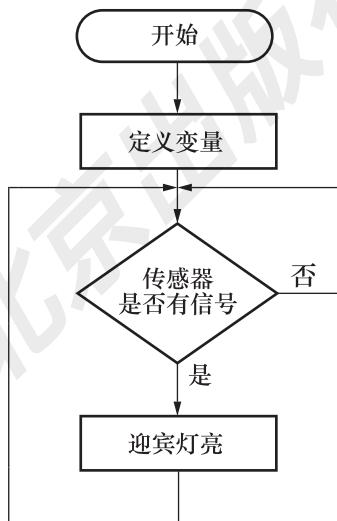


图 1-3-5 智能感应迎宾灯程序流程图

6. 程序设计

```
#include <reg51.h>          // 调用头文件
sbit ELEC=P1^0;             // 光电传感器端口定义
sbit PYRO=P1^1;             // 热释电传感器端口定义
sbit LAMP=P2^0;              // 照明设备端口定义
main ()                     // 主函数
{
    while(1)                // 实现死循环
```

```

{
  if (ELEC == 0 || PYRO == 0)           // 判断光电传感器和热释电传感器是否有信号
    {LAMP = 1; }                      // 如果有信号则照明灯亮
  else {LAMP = 0; }                   // 如果没有信号则照明灯灭
}
}

```

7. 任务检测

序号	检测内容	检测结果	
1	电源电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否
2	显示电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否
3	微处理器电路工作是否正常	<input type="checkbox"/> 是	<input type="checkbox"/> 否



任务评价

序号	评价项目	分值	评价要素
1	正确性	80	(1) 电气模块接线图绘制 10 分：接线图正确、规范计 10 分，有电气性错误每错 1 处扣 2 分（同类错误计算 1 处），接线图绘制不规范酌情扣 1 ~ 3 分，本项累计最多扣 10 分。 (2) 程序流程图绘制 10 分：流程图表示规范、逻辑清晰计 10 分，有表达或逻辑错误每处扣 2 分（同类错误计算 1 处），本项累计最多扣 10 分。 (3) 电路连接 10 分：电路连线正确计 10 分，每错 1 处扣 3 分（同类错误计算 1 处），本项累计最多扣 10 分。 (4) 功能实现 40 分：程序编写后通过硬件验证实现工作任务书中的功能要求，每正确实现一个功能计 5 ~ 10 分，每部分实现一个功能计 2 ~ 5 分。 (5) 性能指标 10 分：性能指标达到工作任务书的要求计 10 分，未达到要求每处扣 3 分，本项累计最多扣 10 分
2	工艺性	10	(1) 操作工艺 3 分：操作正确、规范计 3 分，每出现 1 次操作工艺不良情况扣 1 分，本项累计最多扣 3 分。 (2) 工具、仪表的使用 2 分：工具、仪表使用正确计 2 分，每出现 1 次使用错误扣 1 分，本项累计最多扣 2 分。 (3) 电路装接、布线工艺 5 分：电路装接、布线工艺合理计 5 分，每出现 1 次工艺不良情况扣 1 分，本项累计最多扣 5 分
3	职业与安全意识	10	(1) “7S” 规范 5 分：符合 “7S” 管理规范计 5 分，不符合计 0 分。 (2) 文明安全意识 5 分：保持赛场安静，尊重赛场工作人员，爱惜赛场的设备和器材计 5 分，有违反情况计 0 分



相关知识

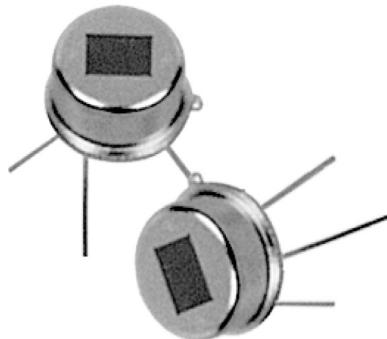
一、热释电红外传感器

热释电红外传感器也可以叫作热释电传感器，是一种被动式调制型温度传感器。它



的工作原理是收集物体表面辐射出来的红外线，然后将红外线能量转换为电气信号作用于电路。在电路图中，热释电传感器通常用字母“PIR”表示。

【写一写】请观察热释电传感器实物图，利用已有知识画出对应的电路符号。

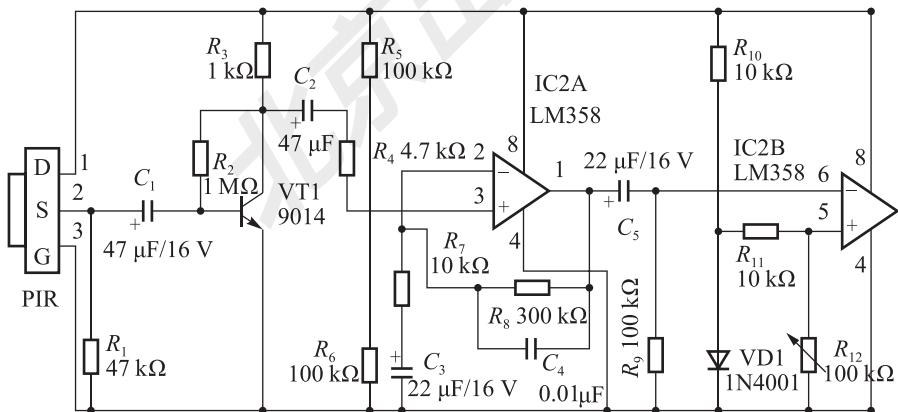


热释电传感器实物图

热释电传感器电路符号

热释电红外传感器是一种非常有应用潜力的传感器。它能检测到人或某些动物发射的红外线并将其转换成电信号输出。

【写一写】下图为热释电报警电路的一部分，请圈出热释电红外传感器。



热释电红外传感器在我们的生活中应用非常广泛，如体温测量仪、入侵报警器、自动开关、火焰报警等，如图 1-3-6 所示。它之所以能够得到广泛的应用，是因为它具有反应速度快、灵敏度和准确度高、测量范围广、使用方便等特点。目前，常用的热释电红外传感器的型号有 RE200B、P228、LHI958 等。



(a) 体温测量仪



(b) 入侵报警器



图 1-3-6 热释电传感器应用

热释电红外传感器通常采用 3 引脚金属封装，3 个引脚分别为电源供电端用“D”表示，信号输出端用“S”表示，接地端用“G”表示。热释电红外传感器侧面有一个凸起标志，这个标志主要是用来识别管脚，将热释电红外传感器管脚朝向自己，距离凸起标志最近的为信号输出端，然后顺时针方向依次为电源供电端、信号输出端和接地端。

二、红外线发光二极管

红外线发光二极管由红外辐射效率高的材料（常用砷化镓）制成 PN 结，外加正向偏压向 PN 结注入电流激发红外光。其最大的优点是可以完全无红暴（采用 940 ~ 950 nm 波长红外管）或仅有微弱红暴（红暴为有可见红光）且寿命长。

【写一写】请观察红外线发光二极管实物图，查阅资料画出对应的电路符号。



红外线发光二极管实物图

电路符号



(1) 红外线发光二极管的极性。

判别红外线发光二极管的正、负电极时，可观察红外线发光二极管两个引脚长短，通常长引脚为正极，短引脚为负极。因为红外线发光二极管呈透明状，所以管壳内的电极清晰可见，内部电极较宽、较大的一个为负极，而较窄且小的一个为正极。

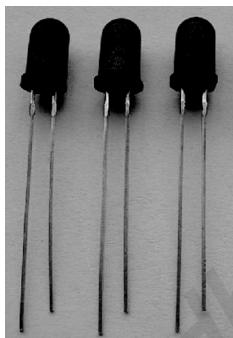
(2) 红外线发光二极管的应用。

红外线发光二极管适用于各类光电检测器的信号光源、光电转换的自动控制仪器、传感器等。

三、红外线接收管

红外接收二极管又叫红外光电二极管，也可称红外光敏二极管。它被广泛应用于各种家用电器的遥控接收器中，如音响、彩色电视机、空调器、VCD 视盘机、DVD 视盘机以及录像机等。红外接收二极管能很好地接收红外发光二极管发射的波长为 940 nm 的红外光信号，而对于其他波长的光线则不能接收。

【写一写】请观察红外线接收管实物图，查阅资料画出对应的电路符号。



红外线接收管实物图

电路符号

红外接收二极管的外形和发射管基本上一样，若从外观上识别，常见的红外接收二极管外观颜色呈黑色。识别引脚时，面对受光窗口，从左至右，分别为正极和负极。另外，在红外接收二极管的管体顶端有一个小斜切平面，通常带有此斜切平面一端的引脚为负极，另一端为正极。亦可用万用表来测量。

红外接收电路通常由红外接收二极管与放大电路组成，放大电路通常又由一个集成块及若干电阻、电容等元件组成，并且需要封装在一个金属屏蔽盒里，虽然电路比较复杂，但体积却很小，还不及一个普通小功率三极管体积大。红外对管检测电路如图 1-3-7 所示。

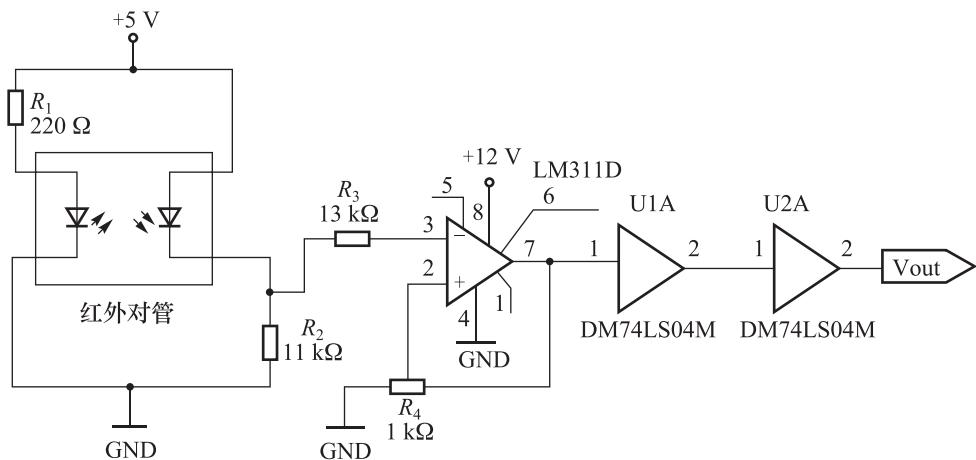


图 1-3-7 红外对管检测电路

任务拓展

单片机下载器软件的使用

AT89S51 单片机下载器是专门用于下载程序到单片机系统中的软件，该软件使用方便。启动软件之后进入如图 1-3-8 所示的界面。

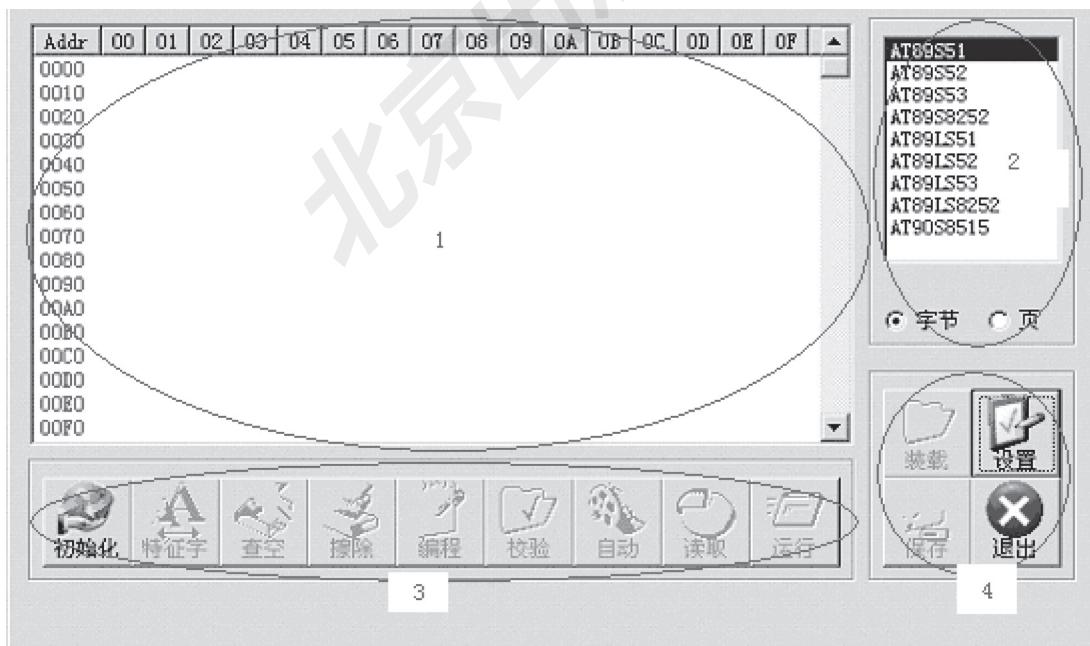


图 1-3-8 下载器界面



在图 1-3-8 中, 区域 1 为程序代码显示区; 区域 2 为下载芯片选择区, 该软件支持多种芯片的程序在线下载, 系统板上的单片机 AT89S51 是其中一种, 软件默认情况下为 AT89S51 单片机。

区域 3 为在线下载的操作区, 它可以提供如下的操作。

(1) 初始化: 启动 AT89S51 单片机进入 ISP 下载状态, 单击“初始化”按钮, 若启动成功, 则操作区的操作按钮就会变成如图 1-3-9 所示的状态, 否则, 没有启动成功会有“初始化失败”的红色字样提示。



图 1-3-9 初始化界面

(2) 特征字: 单击“特征字”按钮, 会读出单片机的芯片的特征字。对于 AT89S51 单片机的特征字为 1E 51 06, 出现的提示框如图 1-3-10 所示。



图 1-3-10 特征字界面

(3) 查空: 检查单片机是否已经被擦除干净。

(4) 擦除: 把单片机的内容擦除干净, 即单片机内部 ROM 的内容全为 FFH。

(5) 编程: 把代码区中的程序代码下载到单片机的内部 ROM 中。注意在编程之前, 要对单片机芯片进行擦除操作。

(6) 校验: 经过编程之后, 对下载到单片机内部 ROM 中的内容与代码区的内容相比较, 若程序下载过程中完全正确, 则提示校验正确, 否则提示出现错误。那就需要重新下载程序到 ROM 中。

(7) 自动: 提供了内部 ROM 从擦除到编程, 最后到校验这三个过程。

(8) 读取: 从单片机内部 ROM 中读取内容到代码显示区中。

区域 4 中, 有“装载”“设置”“保存”和“退出”四个功能。

(1) 装载: 把经过 Keil C 软件转化成 hex 格式的文件装入区域 1 中, 当单击“装载”按钮时, 出现如图 1-3-11 所示的对话框。

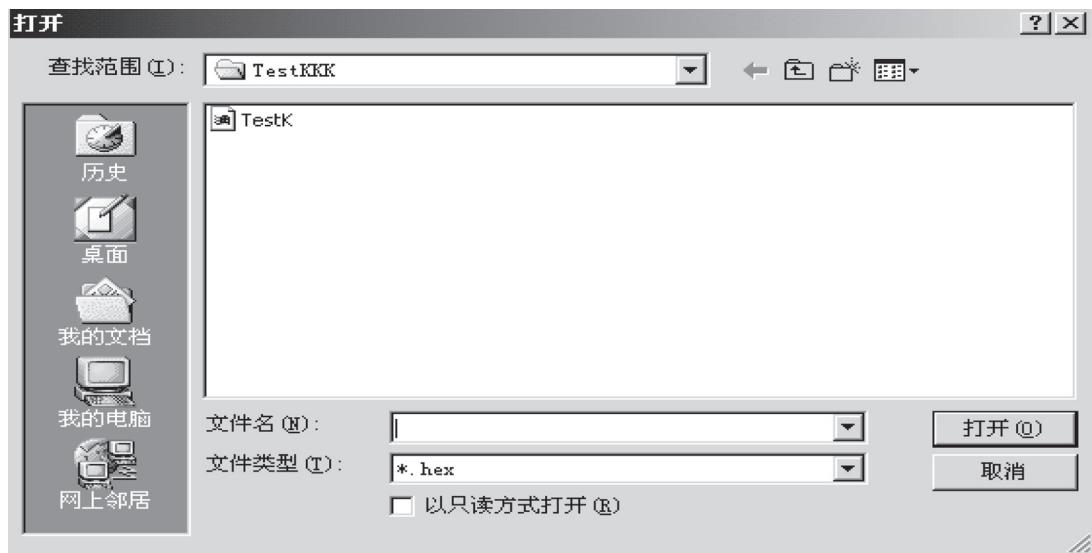


图 1-3-11 打开文件对话框

在这里选择以“.hex”为后缀的文件，选中它并单击“打开”按钮，即把程序代码装入到代码显示区域 1 中。装载之后如图 1-3-12 所示。我们就可以把代码显示区域 1 中的代码通过 ISP 下载线下载到 AT89S51 单片机中。

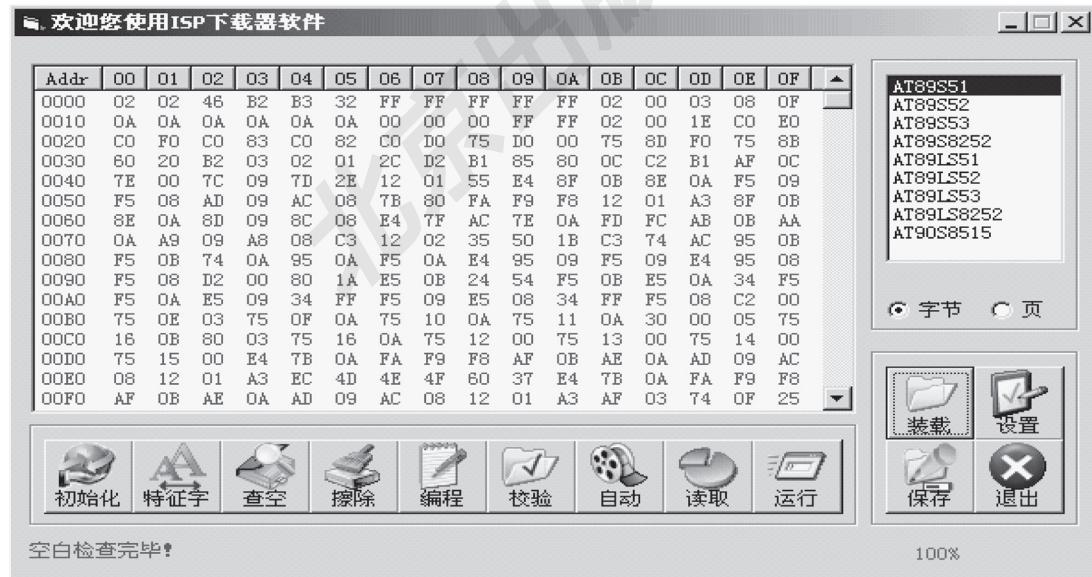


图 1-3-12 hex 文件装载

(2) 设置：对该软件的一些操作方式进行设置，单击按钮之后，出现如图 1-3-13 所示的对话框。

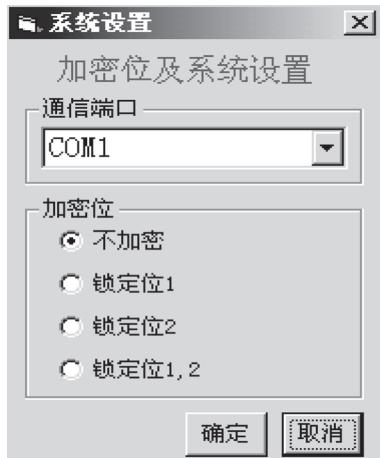


图 1-3-13 系统设置对话框

在这里可以进行通信端口的设置，共设置 COM1、COM2、COM3、COM4 这 4 个串行通信端口，根据计算机的硬件特点来决定，默认情况下为 COM1，即串行通信口 1。

另外，还可以对芯片的加密位进行设置，共有 4 种选择，“不加密”表示对芯片的程序不进行加密保护，“锁定位 1”表示对芯片的程序进行一级加密保护，“锁定位 2”表示对芯片的程序进行二级加密保护，“锁定位 1, 2”表示对芯片的程序进行三级加密保护。

(3) 保存：把从单片机的芯片 ROM 中的程序读取出来的代码显示在区域 1 中，单击“保存”按钮，即可保存为 hex 格式的文件。

(4) 退出：退出该软件系统。

在进行程序调试的时候，我们一般通过 Keil C 软件把编译好的程序转化成 hex 格式的文件，通过上面的方法，装载程序之后，单击“自动”按钮，程序就下载到单片机内部 ROM 芯片中，最后单击“运行”，即可以看到程序的结果。注意，每次重新更新程序的时候，要单击“初始化”按钮，才可以继续相应的操作。



技术应用

国产 C919 大飞机

航空航天领域一直是科技发展的前沿领域之一，单片机作为一种集成电路芯片，具有体积小、功耗低、可靠性高、性能稳定等优点，因此在航空航天领域应用广泛，尤其在 C919 飞机控制与导航方面，其优势得到充分发挥。（图 1-3-14）

C919 飞机控制系统是确保 C919 飞机飞行安全和稳定的关键组成部分，它包括 C919 飞机的导航、稳定控制、姿态控制、动力控制等多个方面。单片机通过其强大的计算和控制能力，成为实现 C919 飞机控制系统的核心。

(1) C919 飞机导航。C919 飞机导航是指通过各种传感器和导航设备获取并处理 C919 飞机在空中的位置、速度、方向等信息，以实现精确的导航和路径规划。单片机通过接收和处理传感器数据，能够实时计算 C919 飞机的位置和方向，并提供相关的导航控制指令。

(2) C919 飞机稳定控制。C919 飞机稳定控制是指通过运算控制方式，使 C919 飞机在飞行中保持稳定的状态，避免意外摇晃或失控。单片机能够利用其高性能处理器和实时反馈控制算法，实现对 C919 飞机的稳定控制，确保 C919 飞机在不同环境下的平稳飞行。

(3) C919 飞机姿态控制。C919 飞机姿态控制是指通过控制 C919 飞机的姿态参数，实现 C919 大飞机的准确控制和姿态调整。单片机通过接收传感器数据和算法处理，能够实时计算和调整 C919 飞机的姿态参数，从而实现对 C919 飞机的精准控制。

(4) C919 飞机动力控制。C919 飞机动力控制是指通过控制 C919 飞机的电机或发动机，调整 C919 飞机的速度和动力输出，以适应不同的飞行任务需求。单片机通过接收来自传感器和控制系统的反馈信号，能够精确控制 C919 飞机的动力输出，保证 C919 飞机的动力系统的正常运行和合理调整。

综上所述，单片机在航空航天应用中对实现 C919 飞机控制与导航具有重要的作用。其高性能处理能力、可编程性和灵活性强、体积小、功耗低等优势，使其成为 C919 飞机控制与导航系统中不可或缺的关键技术。



图 1-3-14 C919 飞机

单元小结

本单元以流星灯、汽车转向灯和智能感应迎宾灯为载体，主要围绕单片机最小系统构成展开讲解，学习 Keil C51 程序语言的基本规则和绘制程序流程图的方法，利用单片机输入、输出端口对发光二极管进行控制，掌握延时程序的不同实现方法。通过正确识读电路原理图，逐步掌握电路框图的绘制方法，并能根据电路原理图进行实物连接。



综合测试

一、填空题

1. 8051 单片机共有_____个引脚。
2. 单片机输入 / 输出端口中具有第二功能的是_____口。
3. 对于 LED 限流电阻一般取值为 330Ω 或 470Ω ，电阻越小，LED 越_____。
4. 单片机正常工作时，电源电压为_____V。
5. 当程序使用 `while (1)` 指令时，程序陷入_____循环。

二、选择题

1. 下列选项中不属于单片机最小系统的是()。
A. 振荡电路 B. 复位电路
C. 延时电路 D. 供电电路
2. 单片机第 9 引脚可以接收()信号。
A. 振荡 B. 复位
C. 延时 D. 中断
3. 单片机输入 / 输出端口具有较强的()能力。
A. 输入电流 B. 输出电流
C. 阻碍电流 D. 疏导电流
4. 语句 `#include <reg51.h>` 中 `reg51.h` 是()文件。
A. 头 B. 尾
C. 合并 D. 独立
5. 语句 `sbit LED = P1^0` 的含义是()。
A. 用 LED 替代 P1.0 B. 用 LED 替代 P1
C. 用 LED 替代 P0 D. 用 LED 替代 P0.1

三、简答题

1. 请编写一个延时时间为 5 ms 的延时子程序。
2. 请对以下程序进行测试，观察并记录发光二极管的状态，为程序填写注释语句。

```
#include < reg51.h > // 调用51单片机头文件reg51.h
sbit LED=P1^0; // 
main( ) // 
{
    while(1) // 
    {
        LED=0; // 
    }
}
```