

# 单片机与微机应用 实验实训案例

主 编 朱 斌 朱清友 王玉平

副主编 杨增宝

西南交通大学出版社  
· 成 都 ·



# 前言

## PREFACE

《单片机与微机应用实验实训案例》主要针对“单片机原理及应用”“微机原理及应用”课程的实验教学，为教师和学生提供相应的实验实训案例。

本书主要由单片机实验案例、单片机实训案例和微机应用实验案例三部分内容组成。单片机实验、实训案例部分主要针对 AT89C52 单片机，并利用 C51 语言进行编程实践；微机应用实验案例部分主要针对 8086CPU，并利用汇编语言进行编程实践。

单片机实验案例部分安排了 17 个案例，主要包括 Keil 的应用，Proteus 的应用，单片机开发板的应用，I/O 接口程序设计，定时器应用、中断程序设计，键盘程序设计，LED、LCD 显示驱动程序设计，常用传感器驱动程序设计等，旨在培养学生对单片机基础知识的掌握和应用能力。每一个实验案例都提供了参考电路设计和参考程序设计，给出了程序的仿真运行结果。

单片机实训案例部分安排了 8 个案例，主要包括计算器设计，秒表设计，日历系统设计，数控电源设计，直流电机控制，步进电机控制，智能温湿控制和电子密码锁等，着重培养学生针对单片机的综合应用能力，包括系统硬件设计能力和程序设计能力。每一个实训案例都提供了参考电路设计和参考程序设计以及仿真运行结果。

微机应用实验案例部分安排了 17 个案例，其中 8 个实验案例为软件实验，主要包括算术运算程序设计，数值处理程序设计，字符串处理程序设计，图形程序设计，文件/文件夹操作程序设计等，每个实验案例都提供了程序设计流程图，参考程序以及程序运行结果。另外 9 个为与硬件有关的实验案例，主要是针对并行接口芯片 8255，定时/计数器芯片 8253，D/A 转换器 DAC0832，A/D 转换器 ADC0809 等的编程应用，并结合相应的硬件实验平台完成实验。每个实验都提供了参考电路、程序设计流程图和参考程序。

本书提供的所有参考程序，都经过调试、运行和仿真，能够达到预期的实验结果，但有些程序的算法可能不是最优，读者可以进一步优化。

由于编者水平有限，书中疏漏与不足之处在所难免，恳请读者批评指正。

编者  
2021 年 1 月



# 目录 CONTENTS

## 第一部分 单片机实验案例

案例 1	Keil 软件开发环境的应用 .....	1
案例 2	Proteus 仿真软件的使用 .....	17
案例 3	单片机开发系统的硬件结构及使用方法 .....	35
案例 4	单片机查表程序设计 .....	43
案例 5	七段数码管驱动控制 .....	49
案例 6	字符型 LCD 显示控制实验 .....	52
案例 7	点阵型 LCD 显示驱动控制 .....	59
案例 8	单片机键盘设计与控制 .....	69
案例 9	单片机中断应用 .....	76
案例 10	单片机定时器应用 .....	80
案例 11	单片机串口通信应用 .....	83
案例 12	DS18B20 温度传感器应用 .....	86
案例 13	DS1302 日期时间应用 .....	92
案例 14	ADC0832 A/D 转换器应用 .....	101
案例 15	SRF04 超声波传感器应用 .....	106
案例 16	24C02EEPROM 芯片的应用 .....	111
案例 17	DHT11 温度/湿度传感器应用 .....	117

## 第二部分 单片机实训案例

案例 18	简易计算器设计 .....	121
案例 19	多功能秒表设计 .....	131
案例 20	智能温湿控制系统设计 .....	137

案例 21	支持农历的日历系统设计	146
案例 22	直流电机转速控制/测量系统设计	169
案例 23	步进电机转速控制/测量系统设计	175
案例 24	电子密码锁设计	180
案例 25	数控直流电源设计	193

### 第三部分 微机应用实验案例

案例 26	DEBUG 调试程序的应用	199
案例 27	算术运算程序设计	205
案例 28	未组合 BCD 加减运算程序设计	211
案例 29	数值处理程序设计	217
案例 30	字符串处理程序设计	221
案例 31	图形程序设计	225
案例 32	文件/文件夹操作程序设计	231
案例 33	模拟电子钟设计	237
案例 34	8255 并行接口应用	242
案例 35	8253 定时器/计数器应用	246
案例 36	DAC0832D/A 转换器应用	250
案例 37	ADC0809A/D 转换器应用	255
案例 38	直流电机控制	258
案例 39	步进电机控制	263
案例 40	七段数码管驱动控制	267
案例 41	交通灯控制实验	273
案例 42	两台微机串行通信	279
参考文献		282

# 第一 部分

## 单片机实验案例

### 案例 1 Keil 软件开发环境的应用

#### 一、目的要求

- (1) 熟悉 Keil 软件的安装和应用方法。
- (2) 掌握 Keil 集成环境中程序的编辑、编译、运行、调试和仿真方法。

#### 二、实验内容

在 Keil 集成环境中编写一个简单的 I/O 口控制程序，并对其进行调试、运行和仿真，从而熟悉 Keil 软件的应用。

#### 三、具体要求

参考电路如图 1.1 所示。

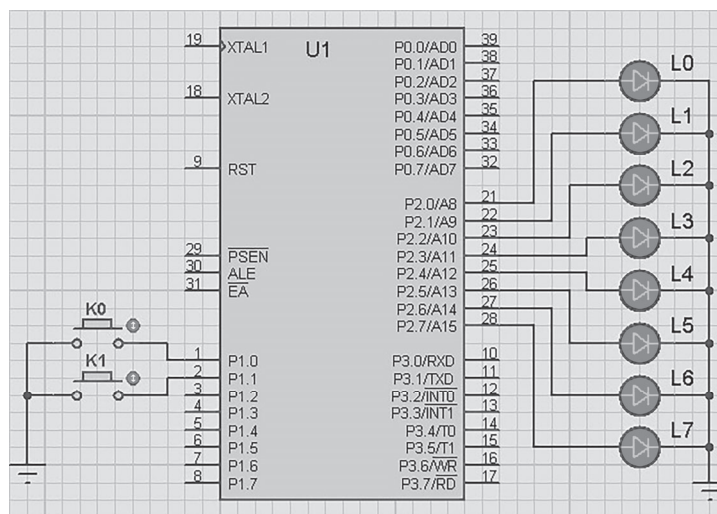


图 1.1 参考电路

(1) 当 K0、K1 都断开时，L0~L7 都灭。

(2) 当 K0 闭合、K1 断开时，L0~L7 按流水方式点亮 (L0→L1→L2→……→L7→L0→……)，大约每秒钟变化 1 次。

(3) 当 K0 断开、K1 闭合时，上面 4 只 LED 和下面 4 只 LED 交替点亮，大约每秒钟变化 1 次。

(4) 当 K0、K1 都闭合时，偶数 LED 和奇数 LED 交替点亮，大约每秒钟变化 1 次。

## 四、Keil 介绍

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统。与汇编语言相比，C 语言在功能、结构性、可读性、可维护性上有明显的优势，易学易用。Keil 提供了包括 C 编译器、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境 (μVision) 将这些部分组合在一起。运行 Keil 软件需要 Windows 7、Windows XP 等操作系统。如果使用 C 语言进行单片机编程，那么 Keil 几乎就是不二之选，即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会使用户事半功倍。

### 1. 软件安装与启动

Keil C51 集成开发环境的安装比较简单，下面进行简单介绍。先下载软件，双击安装包，进入安装向导界面，如图 1.2 所示，点击“Next”按钮。

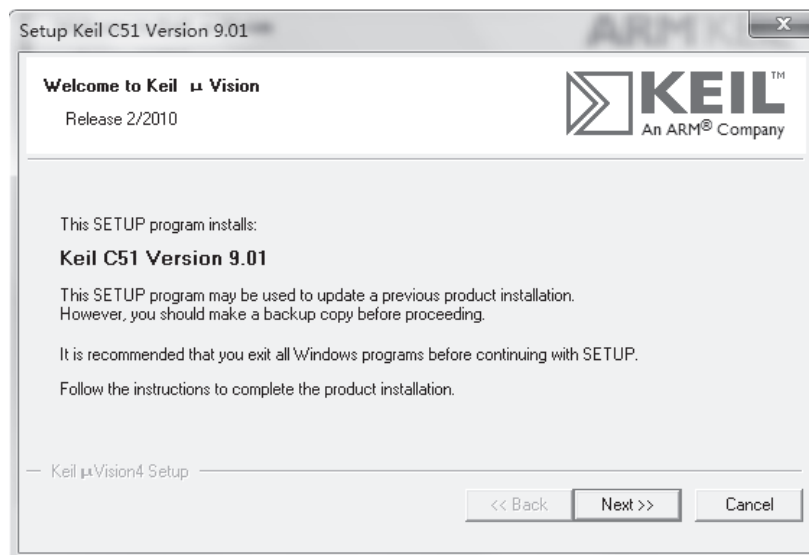


图 1.2 安装向导界面

如图 1.3 所示，勾选“I agree to all the terms of the preceding License Agreement”，点击“Next”按钮。

如图 1.4 所示，选择路径（可以默认），点击“Next”按钮。

如图 1.5 所示，填写用户信息，点击“Next”按钮。



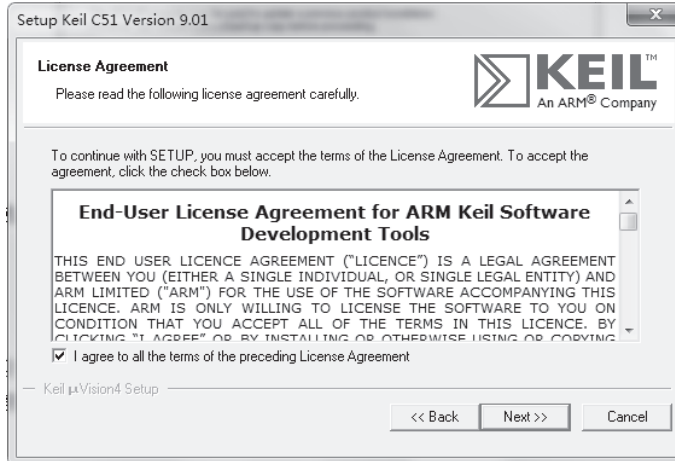


图 1.3 同意安装许可协议界面

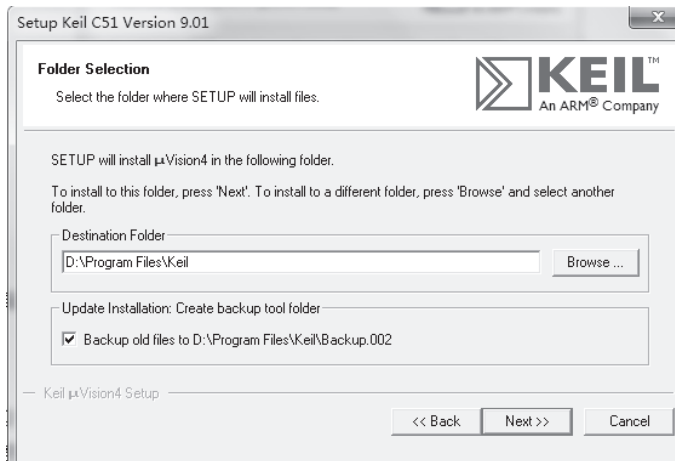


图 1.4 安装路径设置

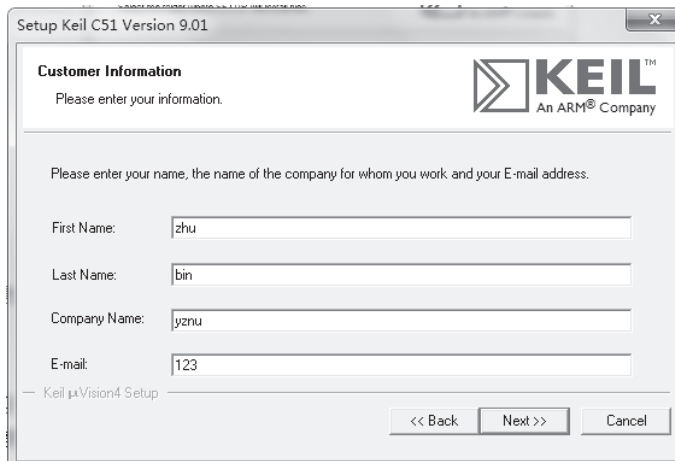


图 1.5 填写用户信息

如图 1.6 所示，安装过程需要等待两分钟，最后点击“Finish”，如图 1.7 所示，完成安装。

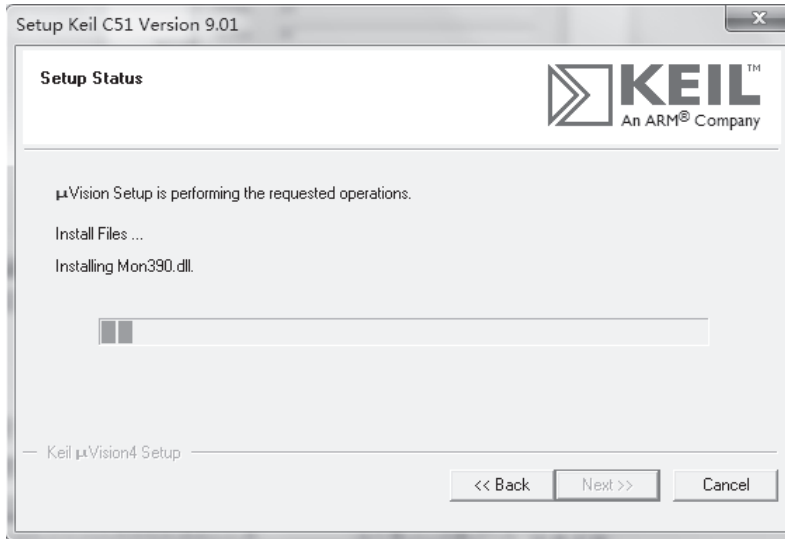


图 1.6 安装过程

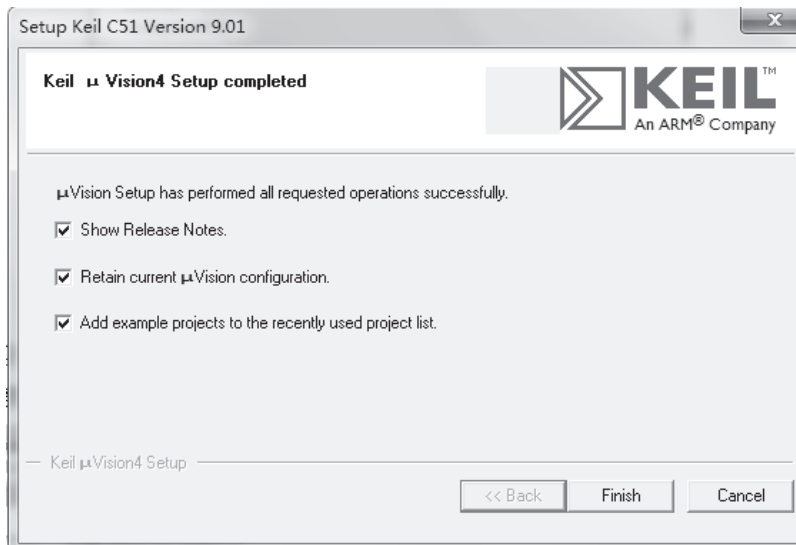


图 1.7 安装完成确认

Keil C51 软件安装完毕后，在桌面上出现 Keil C51 软件的快捷图标。单击该快捷图标，则启动该软件，出现如图 1.8 所示的 Keil C51 界面，图中标出了 Keil C51 界面各窗口的名称。



图 1.8 Keil C51 软件开发环境界面

## 2. 创建工程

编写一个新的应用程序前，首先要建立工程（Project）。

Keil C51 用工程管理的方法把一个程序设计中所需要用到的、互相关联的程序链接在同一工程中。这样，打开一个工程时，所需要的关联程序也都跟着进入了调试窗口，方便用户对工程中各个程序进行编写、调试和存储。项目管理便于区分不同工程中所用到的程序文件和库文件，非常容易管理。因此，编写程序前，需要首先创建一个新的工程，操作如下：

(1) 在编辑界面下，单击菜单栏中的“工程”（Project），出现下拉菜单，再点击选择“新建 μVision 工程”（New Project），如图 1.9 所示。



图 1.9 新建工程菜单

(2) 单击“新建 μVision 工程”（New Project）选项后，就会弹出“Create New Project”窗口，如图 1.10 所示。在“文件名 (N)”中输入一个工程的名称，保存后的文件扩展名为“.uvproj”，即工程文件的扩展名，以后直接单击此文件就可打开先前建立的工程。

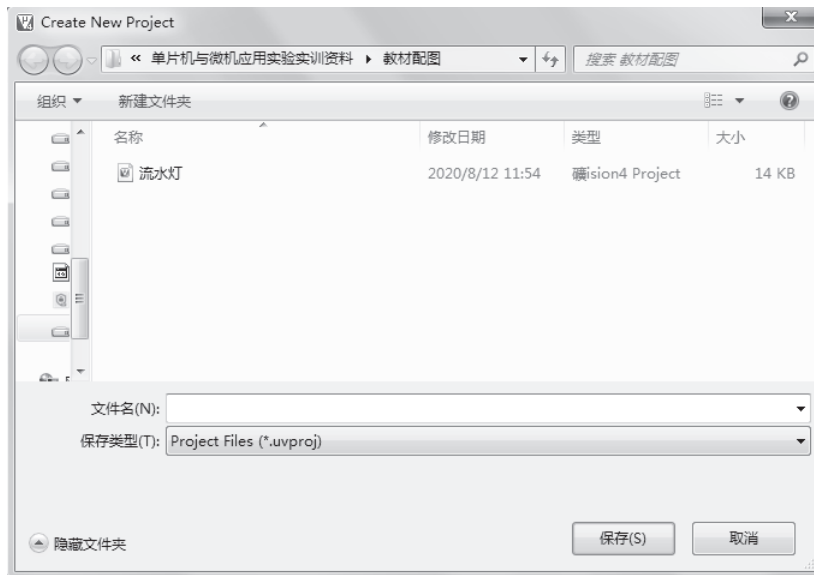


图 1.10 “Create New Project” 窗口

在“文件名 (N)” 窗口中输入新建工程文件的的名字后，选择工程的保存目录，单击“保存 (S)” 即可，如图 1.11 所示。

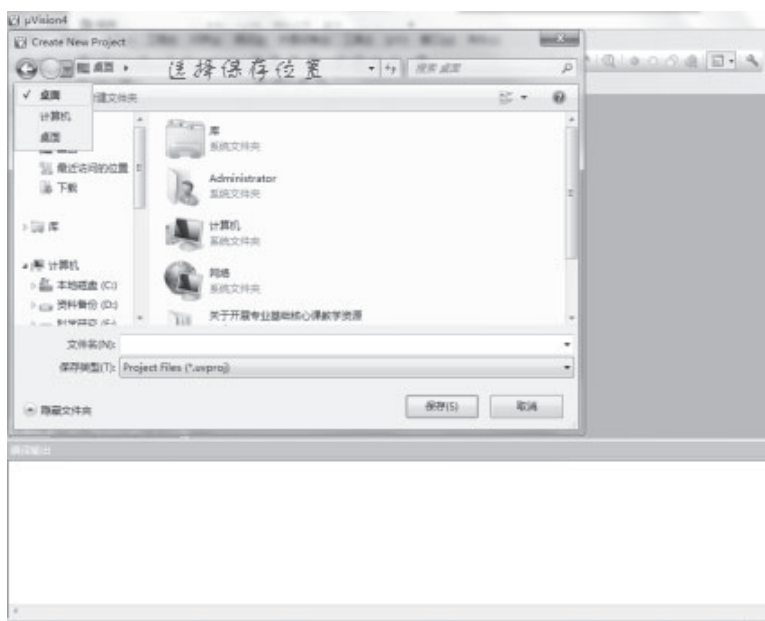


图 1.11 新建工程保存目录选择窗口

(3) 选择单片机，单击“保存 (S)” 后，会弹出如图 1.12 所示的选择单片机窗口，按照提示选择相应的单片机。这里选择“Atmel”目录下的“AT89S52”。

(4) 点击“确定”按钮后，会出现如图 1.13 所示的对话框。如果需要复制启动代码到新建的工程，选择“是”出现图 1.14 的窗口；如选择单击“否”，图 1.14 中的启动代码项“SARTUP.A51”，不会出现在对话框中，这时新的工程已经创建完毕。



图 1.12 选择单片机窗口

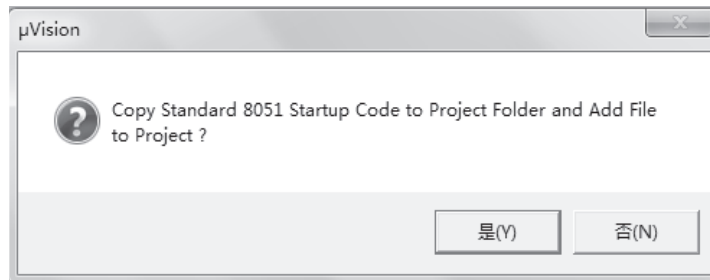


图 1.13 是否复制启动代码到工程对话框

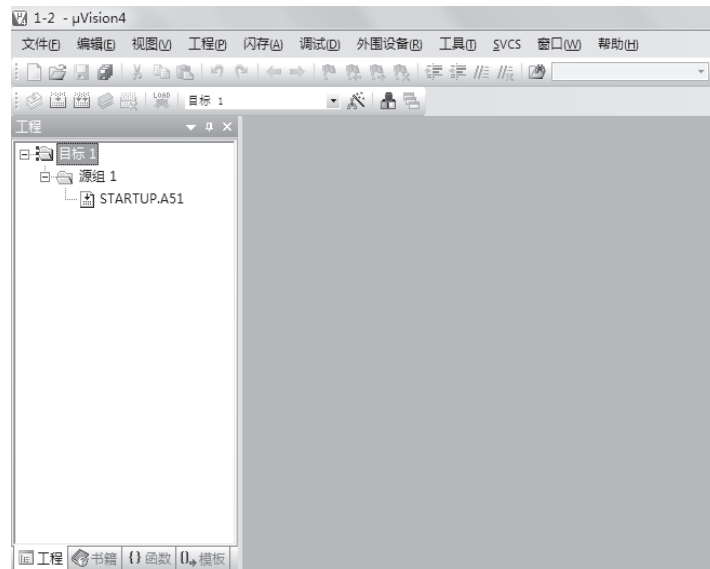


图 1.14 完成工程的创建

### 3. 添加用户源程序文件


新的工程文件创建完成后，就需要将用户源程序文件添加到这个工程中。添加用户程序文件通常有两种方式：一种是新建文件；另一种是添加已创建的文件。

#### 1) 新建文件

(1) 单击图 1.8 中“新建”快捷按钮，这时会出现如图 1.15 所示窗口。窗口中会出现一个空白的文件编辑区域，用户可在这里输入程序源代码。



图 1.15 建立新文件

(2) 单击图 1.8 中“保存”快捷按钮, 保存用户程序文件，这时会弹出如图 1.16 所示窗口。

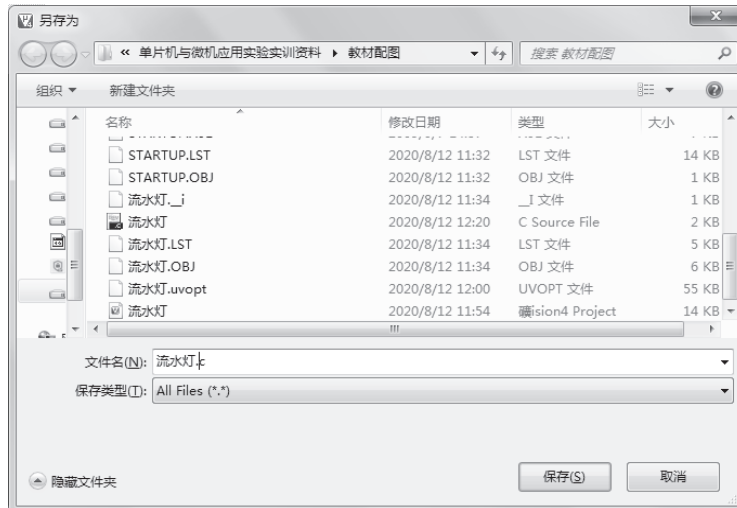


图 1.16 文件保存对话框

(3) 在图 1.16 “另存为”对话框中，在下拉框中选择新文件的保存目录，将这个新文件与刚才建立的工程保存在同一个文件夹下，然后在“文件名(N)”窗口中输入新建文件的名称。

字“流水灯”，如果使用 C51 语言编程，则文件名的扩展名应为“.c”。

如果用汇编语言编程，文件扩展名应为“.asm”。完成上述步骤后单击“保存”，此时新文件已创建完成。这个新文件还需添加到刚才创建的工程中，操作步骤如下所示。

## 2) 添加文件

(1) 在工程窗口（见图 1.8）中，右键单击“源组 1”（Source Group 1），选择“添加文件到组‘源组 1’”（Add File to ‘Source Group 1’）选项，如图 1.17 所示。

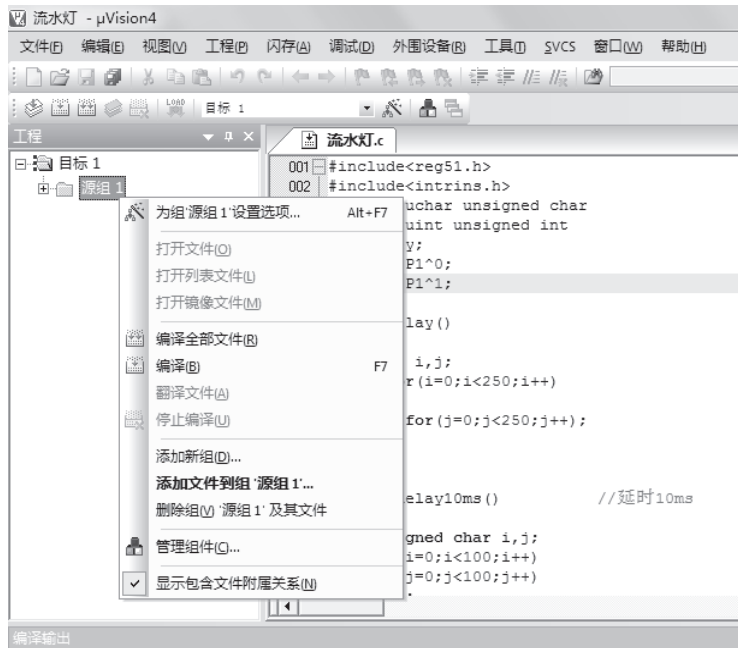


图 1.17 添加文件

(2) 完成上述操作后，会出现“添加文件到组‘源组 1’”对话框，如图 1.18 所示。在该窗口中选择要添加的文件，这里只有刚刚建立的文件“流水灯.c”，点击这个文件后，单击“添加”（Add）按钮，再单击“关闭”（Close）按钮，文件添加就完成了，这时的项目窗口如图 1.19 所示，用户程序文件“流水灯.c”已经出现在“源组 1”（Source Group 1）目录下了。

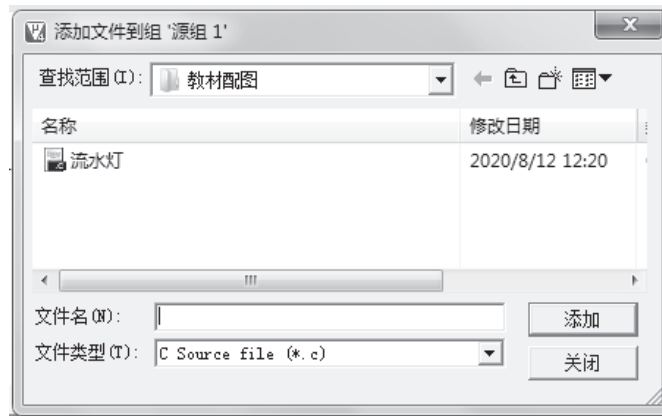


图 1.18 “添加文件到组‘源组 1’”对话框

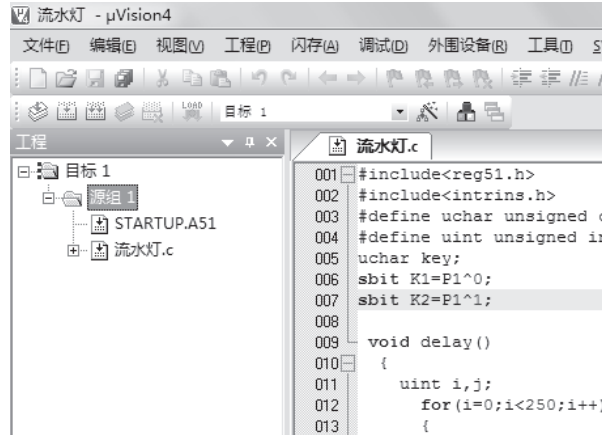



图 1.19 文件已添加到项目中

#### 4. 程序的编译

通过前面的学习，在文件编辑窗口建立了文件“流水灯.c”（或“流水灯.asm”），并且将文件添加到项目中。下面需要将文件进行编译和调试，最终生成可执行的.hex 文件，具体步骤如下：

单击快捷按钮中的对当前文件进行编译，这里是以“流水灯.c”文件为例，在图 1.20 中的输出窗口会出现编译结果的提示信息。

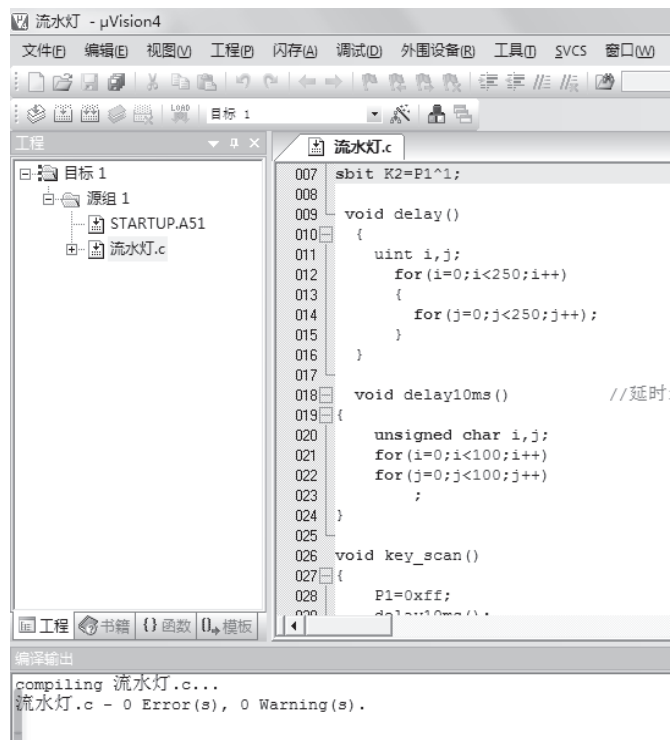



图 1.20 文件编译信息

从输出窗口中的提示信息可以看出，程序没有错误。如果程序有误，则认真检查程序，



找到错误并改正，改正后再次单击进行编译，直至提示信息显示没有错误为止。

## 5. 工程的设置

工程创建后，还需对其进行进一步设置。右键单击工程窗口的“目标 1”（Target 1），选择“为‘目标 1’设置选项”（Options for Target ‘Target 1’）（见图 1.21），出现工程设置对话框，如图 1.22 所示。该对话框下有多个页面，通常需要设置的有两个，一个是“项目”（Target）页面，另一个是“输出”（Output）页面，其余设置取默认值就可。

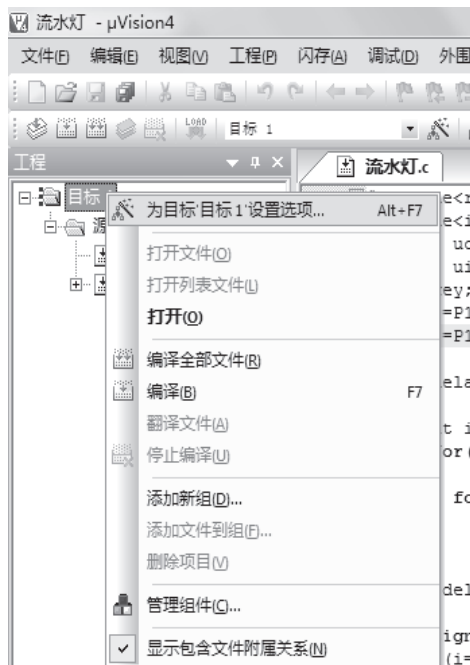


图 1.21 项目参数设置



图 1.22 工程设置对话框

## 1) “项目” 页面

(1) 时钟 (Xtal) (MHz): 设置晶振频率值, 默认值是所选目标 CPU 的最高可用频率值, 可根据需要重新设置。该设置与最终产生的目标代码无关, 仅用于软件模拟调试时显示程序执行时间。正确设置该数值可使显示时间与实际所用时间一致, 一般将其设置成与硬件目标样机所用的频率相同, 如果不需要了解程序执行的时间, 也可以不设置。

(2) 存储模式 (Memory Model): 设置 RAM 的存储器模式, 有 3 个选项。

Small: 所有变量都在单片机的内部 RAM 中。

Compact: 可以使用 1 页外部 RAM。

Large: 可以使用全部外部的扩展 RAM。

(3) 代码 Rom 大小 (Code Rom Size): 设置 ROM 空间的使用, 即程序的代码存储器模式, 有 3 个选项。

Small: 只使用低于 2 KB 的程序空间。

Compact: 单个函数的代码量不超过 2 KB, 整个程序可以使用 64 KB 程序空间。

Large: 可以使用全部 64 KB 程序空间。

(4) 使用片内 ROM (Use on-chip ROM): 是否仅使用片内 ROM 选项。

(5) 操作系统 (Operation System): 操作系统选项。Keil 提供了两种操作系统: Rtx tiny 和 Rtx full。通常不选操作系统, 所以选用默认项 None。

(6) 片代码存储 (off-chip Code Memory): 用以确定系统扩展的程序存储器的地址范围。

(7) 片外 Xdata 存储 (off-chip Xdata Memory): 用以确定系统扩展的数据存储器的地址范围。

(2) 和 (3) 中的 3 个选项必须根据所用硬件来决定, 如果是最小应用系统, 不进行任何扩展, 则按默认值设置。

## 2) “输出” 页面

点击窗口的“输出”选项, 会出现输出页面, 如图 1.23 所示。



图 1.23 Output 页面

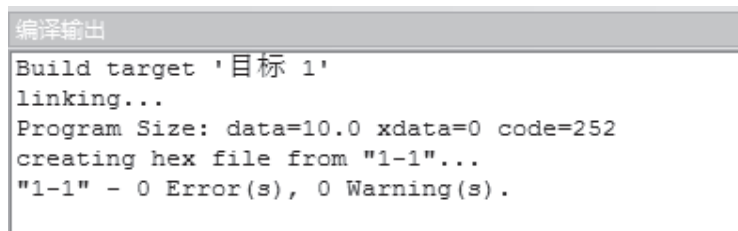
(1) 创建可执行文件 (Create HEX File): 生成可执行文件代码文件。选择此项后即可生成单片机可以运行的二进制文件 (.hex 格式文件), 文件的扩展名为.hex。

(2) 为目标文件选择目录 (Select Folder for objects): 选择最终的目标文件所在的文件夹, 默认与工程文件在同一文件夹中, 通常选默认。

(3) 执行的名字 (Name of Executable): 用于指定最终生成的目标文件的名字, 默认与工程文件相同, 通常选默认。

(4) 调试信息 (Debug information): 将会产生调试信息。这些信息用于调试, 如果需要对程序进行调试, 应选中该项, 其他选项选默认即可。


完成设置后, 就可在程序编译时单击快捷按钮, 此时会产生如图 1.24 的提示信息。该信息说明程序占用片内 RAM 共 11 字节, 片外 RAM 共 0 字节, 程序存储器共 252 字节。最后生成的.hex 文件名为“1-1.hex”, 至此, 整个程序编译过程就结束了, 生成的.hex 文件就可在后面介绍 Proteus 环境下进行虚拟仿真时, 装入单片机运行。





```
编译输出
Build target '目标 1'
linking...
Program Size: data=10.0 xdata=0 code=252
creating hex file from "1-1"...
"1-1" - 0 Error(s), 0 Warning(s).
```

图 1.24 hex 文件生成的提示信息

下面对用于编译、连接时的快捷按钮、与作简要说明:

(1) : 建立工程按钮, 用来编译、链接当前工程, 并产生相应目标文件, 如.hex 文件。

(2) : 编译按钮, 用于在工程文件有改动时, 来重建整个工程, 并产生相应的目标文件, 如.hex 文件。

(3) 按钮: 重新编译按钮, 用于重新编译当前项目中的所有文件, 并生成相应的目标程序 (.hex 文件), 供单片机直接下载。主要用在当项目文件有改动时, 来全部重建整个项目。因为一个项目不止一个文件, 当有多个文件时, 可用本按钮进行编译。

用 C51 编写的源代码程序不能直接使用, 需对该源代码程序进行编译, 生成可执行的目标代码.hex 文件, 并加载到 Proteus 环境下的虚拟单片机中, 才能进行虚拟仿真。

## 五、参考程序

```
#include<reg52.h>//加载头文件
#include<intrins.h>//为了使用循环移位指令
#define uchar unsigned char//定义宏
uchar x=1;//定义变量 x, 用于流水灯控制
//定义按钮开关 k0 和 k1
sbit k0=P1^0;
sbit k1=P1^1;
```

```

//延时, 大约 1 s
void delay()
{unsigned int i=1000;
  uchar j;
  while(i--)for(j=0;j<125;j++);
}
//主函数
void main()
{while(1)
  {while(k0&k1)P2=0;//当 k0、k1 都断开时全灭
    while(~k0&k1)//当 k0 按下、k1 断开时, LED 流水灯点亮
    {P2=x;
      delay();
      x=_crol_(x,1);//x 循环左移 1 位
    }
    while(k0&~k1)//k0 断开、k1 按下时, 上下 4 个 LED 交替点亮
    {P2=0x0f;//点亮 L0 ~ L3
      delay();
      P2=0xf0;//点亮 L4 ~ L7
      delay();
    }
    while(~k1&~k0)//k0、k1 都按下时, 奇数灯和偶数灯交替点亮
    {P2=0x55;
      delay();
      P2=0xaa;
      delay();
    }
  }
}

```

## 六、实验结果

(1) 当 K0、K1 都断开时, L0 ~ L7 都灭, 实验结果如图 1.25 所示。

(2) 当 K0 闭合、K1 断开时, L0 ~ L7 按流水方式点亮, 大约每秒钟变化 1 次, 如图 1.26 所示。

(3) 当 K0 断开、K1 闭合时, 上面 4 只 LED 和下面 4 只 LED 交替点亮, 大约每秒钟变化 1 次, 如图 1.27 所示。

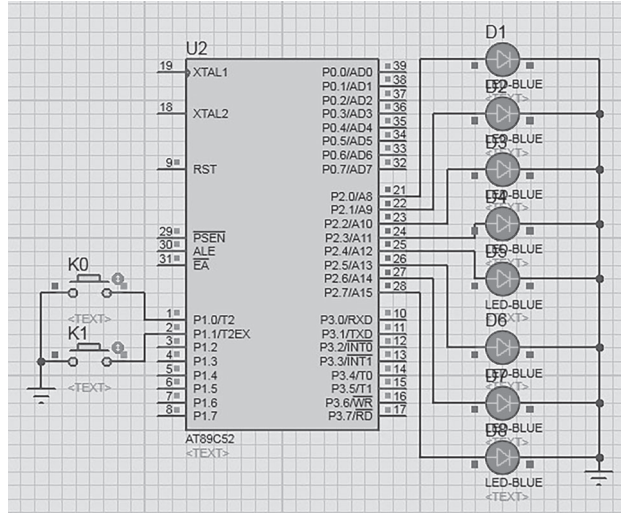


图 1.25 LED 全灭

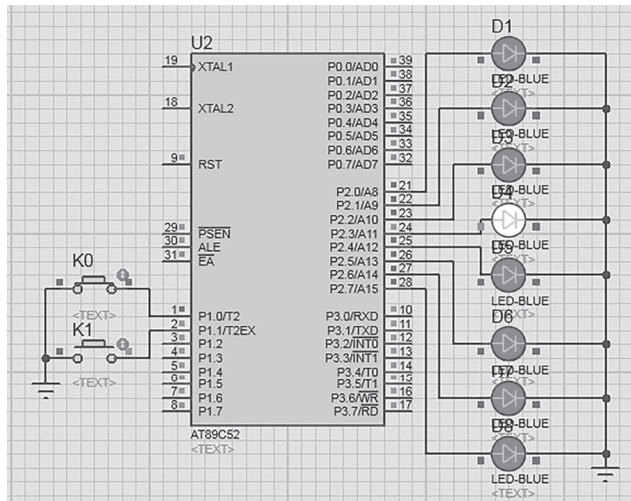


图 1.26 LED 流水灯点亮

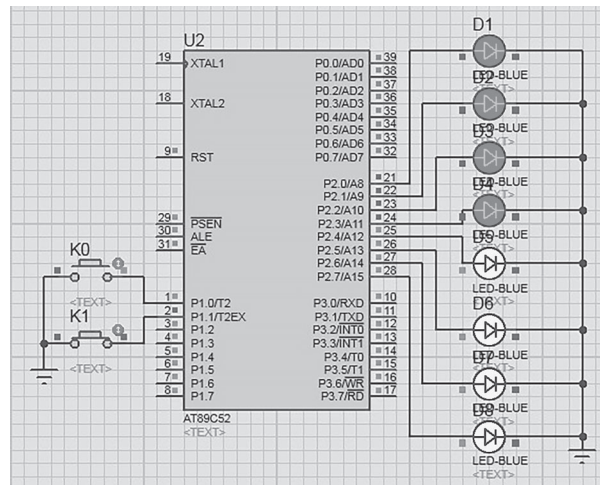


图 1.27 上下 4 个 LED 交替点亮

(4) 当 K0、K1 都闭合时，偶数 LED 和奇数 LED 交替点亮，大约每秒钟变化 1 次，如图 1.28 所示。

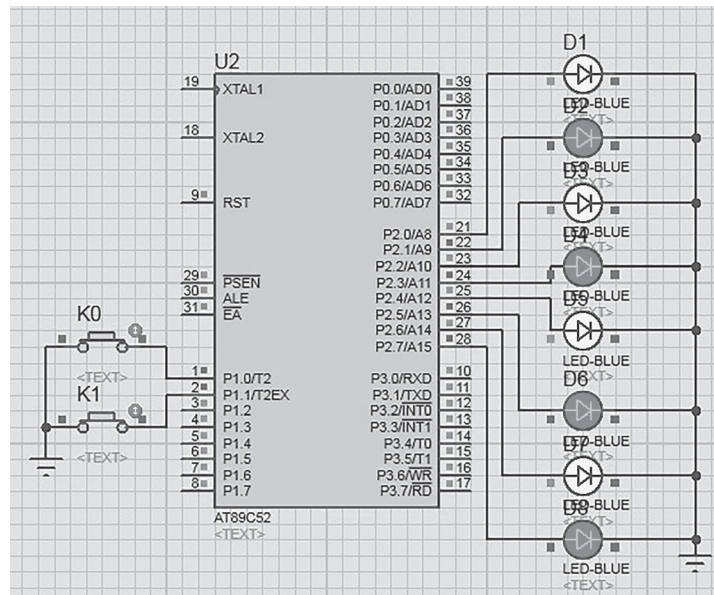


图 1.28 奇数 LED 和偶数 LED 交替点亮

## 案例 2 Proteus 仿真软件的使用

### 一、目的要求

- (1) 熟悉 Proteus 仿真软件环境，掌握常用菜单项的操作方法。
- (2) 掌握 Proteus 软件环境中绘制电路图的方法及仿真单片机程序的步骤。

### 二、实验内容

- (1) 在 Proteus 软件环境中绘制单片机及各类常用电子器件，连接元件引脚。
- (2) 编写 I/O 口控制程序，并对其进行调试，运行和仿真，从而熟悉 Proteus 软件的应用。

### 三、具体要求

按图 2.1 所示选择元件，并绘制仿真电路，编程实现：当 K1 按下时，L1 亮；当 K2 按下时，L2 亮；当 K3 按下时，电机转动。

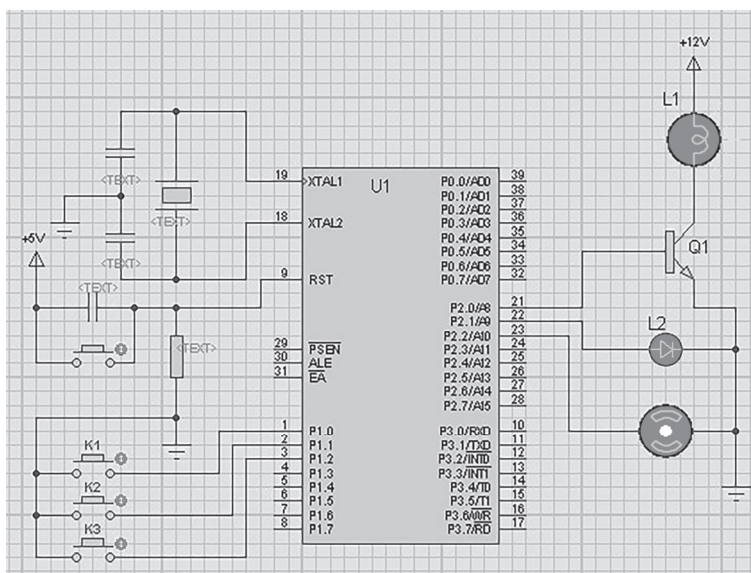


图 2.1 仿真电路

### 四、Proteus 虚拟仿真平台介绍

Proteus 是英国 Lab Center Electronics 公司于 1989 年推出的完全使用软件手段来对单片机应用系统进行虚拟仿真的软件开发平台。



## 1. Proteus 功能简介

Proteus 是目前支持嵌入式处理器的虚拟仿真平台之一，除可仿真模拟电路、数字电路外，还可仿真 8051、PIC12/16/18 系列、AVR 系列、MSP430 等各主流系列单片机，以及各种外围可编程接口芯片。此外，它还支持 ARM7、ARM9 等型号的嵌入式微处理器的仿真。

Proteus 不需要用户硬件样机就可直接在 PC（个人计算机）上对单片机系统进行虚拟仿真，将系统的功能及运行过程形象化，并可以像焊接好的电路板一样看到单片机系统的执行效果。

Proteus 元件库中有几万种元件模型，可直接对单片机的各种外围元件及电路进行仿真，如 RAM、ROM、总线驱动器、各种可编程外围接口芯片、LED 数码管显示器、LCD 显示模块、矩阵式键盘以及多种 D/A 和 A/D 转换器等。此外，它还可对 RS232 总线、I<sup>2</sup>C 总线、SPI 总线进行动态仿真。

Proteus 提供了各种信号源、虚拟仿真仪器，并能对电路原理图的关键点进行虚拟测试。

Proteus 提供了丰富的调试功能。在虚拟仿真中具有全速、单步、设置断点等调试功能，同时可支持观察各变量、寄存器的当前状态。

目前，Proteus 已在包括剑桥大学、斯坦福大学、牛津大学、加州大学在内的全球数千所高校以及世界各大研发公司得到广泛应用。

尽管 Proteus 具有开发效率高、不需要附加的硬件开发装置成本等优点，但是它不能进行用户样机硬件的诊断，所以在单片机系统的设计开发中，一般是先在 Proteus 环境下绘出系统的硬件原理电路图，在 Keil C51 环境下书写并编译程序，接着在 Proteus 环境下仿真并通过，然后依照仿真结果来完成实际的硬件设计，并把仿真通过的程序代码通过编程器或在线烧录到单片机的程序存储器中，最后运行程序并观察用户样机的运行结果，如有问题，再连接硬件仿真器或免费在线修改程序以进行分析、调试。

## 2. Proteus ISIS 的虚拟仿真

Proteus ISIS（智能原理图输入）用来绘制单片机系统的电路原理图，还可直接实现单片机系统的虚拟仿真，可模拟产生声、光及各种动作等逼真的效果。

当电路连接无误后，单击单片机芯片载入经 Keil C51 调试编译后生成的.hex 文件，点击仿真运行按钮，即可检验电路硬件及软件的设计正确与否。

按要求把 Proteus 安装在 PC 机上。安装完后，单击桌面上的 ISIS 运行界面图标，即可出现如图 2.2 所示的 Proteus ISIS 原理电路图绘制界面（以汉化 7.5 版本为例）。

整个屏幕界面分为若干个区域，由原理图编辑窗口、预览窗口、工具箱、主菜单栏、主工具栏等组成。

## 3. ISIS 各窗口简介

ISIS 界面主要有 3 个窗口：原理图编辑窗口、预览窗口和对象选择窗口，如图 2.2 所示。

### 1) 原理图编辑窗口

该窗口用来绘制电路原理图，设计电路以及各种符号模型的区域，蓝色方框内为可编辑区，元件放置、电路设置尽量在此框中完成。



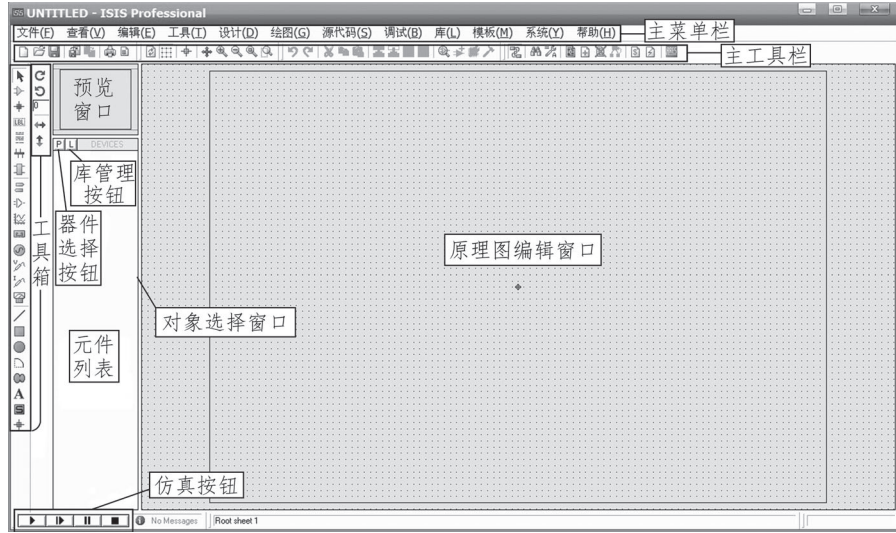


图 2.2 Proteus ISIS 的界面

## 2) 预览窗口

该窗口可对选中的元器件进行预览，也可对原理图编辑窗口进行预览，如图 2.3 所示。可显示两个内容：

(1) 如单击元件列表中的元件时，预览窗口会显示该元件符号。

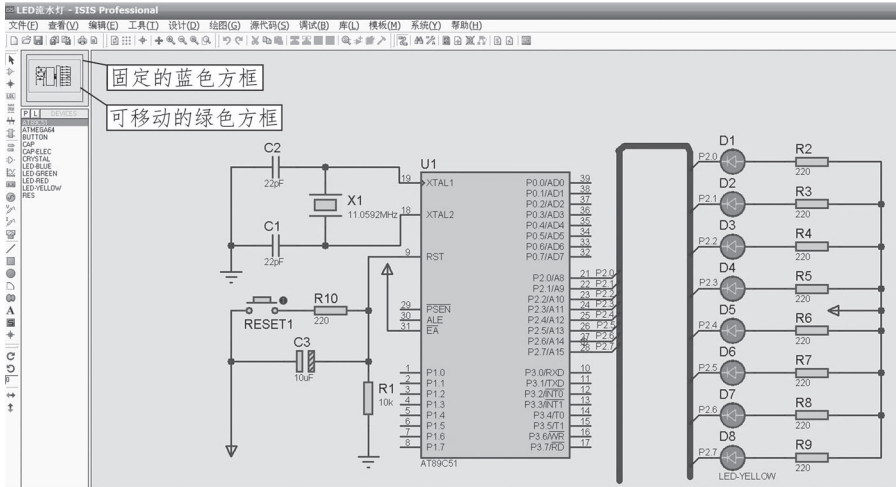


图 2.3 预览窗口调整原理图的可视范围

(2) 当鼠标焦点落在原理图编辑窗口时（即放置元件到原理图编辑窗口后或在原理图编辑窗口中点击鼠标后），它会显示整张原理图的缩略图，并会显示一个绿色方框，框里的内容就是当前原理图窗口中显示的内容。

单击绿色方框中的某一点，就可拖动鼠标来改变绿色方框的位置，从而改变原理图的可视范围，最后在绿色方框内单击鼠标，绿色方框就不再移动，使得原理图的可视范围固定。

## 3) 对象选择窗口

该窗口用来选择元器件、终端等对象。该窗口中的元件列表区用来表明当前所处模式以

及其中的对象列表，如图 2.4 所示。窗口中两个按钮：“P”为器件选择按钮，“L”为库管理按钮。在图中可以看到元件列表，即已经选择的 AT89C51 单片机、电容电阻、晶振、发光二极管等各种元器件。



图 2.4 元件列表

#### 4. 主菜单栏

图 2.2 最上面一行为主菜单栏，包含如下命令：文件、查看、编辑、工具、设计、绘图、源代码、调试、库、模板、系统和帮助。单击任意菜单命令后，都将弹出其下拉的子菜单命令列表。

##### 1) 文件 (File) 菜单

文件菜单包括工程的新建设计、打开设计、导入位图、导入区域、导出区域和打印等操作，如图 2.5 所示。ISIS 下的文件主要是设计文件 (Design Files)，文件扩展名为 “.DSN”，它包括一个单片机硬件系统的原理电路图及其所有信息，用于虚拟仿真。



图 2.5 文件菜单

下面介绍文件菜单下的新建设计命令。

点击“文件”→“新建设计”，出现一个空的界面，新设计的默认名为“UNTITLED.DSN”。

该命令会把该设计以这个名字存入磁盘文件中，文件的其他选项也会使用它作为默认名。

如想进行新的设计，即需给该设计命名并保存，然后点击“文件”→“保存设计”（也可直接点击主工具栏中的快捷图标），输入新的文件名保存即可。

## 2) 工具 (Tools) 菜单


工具菜单如图 2.6 所示。菜单中的“自动连线 (W)”命令图标，在绘制电路原理图中会用到，在绘制电路原理图时会出现，按下快捷图标即进入自动连线状态。



图 2.6 工具菜单

## 3) 调试 (Debug) 菜单

调试菜单如图 2.7 所示，主要完成单步运行、断点设置等功能。

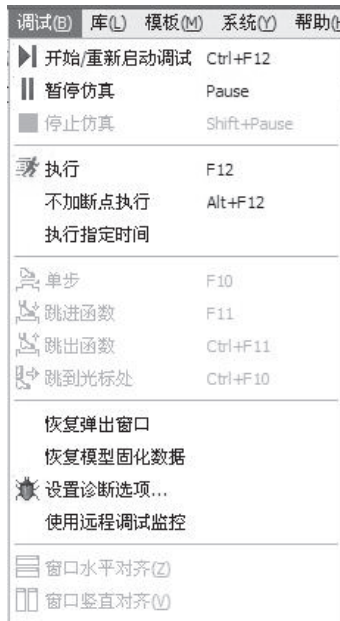


图 2.7 调试菜单








## 5. 主工具栏

主工具栏位于主菜单下面，以图标形式给出，栏中共有 38 个快捷图标按钮：











每一个图标按钮都对应一个具体的菜单命令，主要目的是快捷方便地使用这些命令。下面把 38 个图标分为 4 组，简要介绍快捷图标命令的功能。














图标  的功能如下：

- ：新建一个设计文件；
- ：打开一个已存在的设计文件；
- ：保存当前的电路图设计；
- ：将一个局部文件导入 ISIS 中；
- ：将当前选中的对象导出为一个局部文件；
- ：打印当前设计文件；
- ：选择打印的区域。


图标  的功能如下：









- ：刷新显示；
- ：原理图是否显示网格的开关；
- ：是否显示手动原点；
- ：以鼠标所在点为中心居中；
- ：放大；
- ：缩小；
- ：查看整张图；
- ：查看局部图。

图标  的功能如下：

- ：撤销最后一步操作；
- ：恢复最后一步操作；
- ：剪切选中对象；
- ：复制选中对象至剪切板；
- ：从剪切板粘贴；
- ：复制选中的块对象；
- ：移动选中的块对象；
- ：旋转选中的块对象；
- ：删除选中的块对象；
- ：从库中选取器件；
- ：创建器件；
- ：封装工具；
- ：释放元件。

图标  的功能如下：

- ：自动连线；

-  : 查找并连接;
-  : 属性分配工具;
-  : 设计浏览器;
-  : 新建图纸;
-  : 移动页面/删除页面;
-  : 退出父页面;
-  : 生成元件列表;
-  : 生成电气规则检查报告;
-  : 生成网表并传输到 ARES。

## 6. 工具箱

图 2.2 中最左侧为工具箱，选择相应的工具箱图标按钮，系统将提供不同的操作工具。对象选择器根据不同的工具箱图标决定当前状态显示的内容。显示对象的类型包括：元器件、终端、引脚、图形符号、标注和图表等。

下面介绍工具箱中各图标按钮对应的功能。

### 1) 模型工具栏各图标的功能

 : 选择模式。

 : 元件模式，用来拾取元器件。

设计者可根据需要，从丰富的元件库中拾取元器件并添加元件到列表中。单击此图标可在列表中选择元件。

 : 放置电路的连接点。

此按钮适用于节点的连线，在不用连线工具的条件下，可方便地在节点之间或节点到电路中任意点或线之间连线。

 : 标注线标签或网络标号。


该图标按钮在绘制电路图时，具有重要的意义，它可使连线简单化。例如，从 8051 单片机的 P1.7 脚和二极管的阳极各画出一条短线，并标注网络标号为 1，那么就说明 P1.7 脚和二极管的阳极已经在电路上连接在一起了，而不用真的画一条线把它们连起来。

 : 输入文本。

使用本图标按钮命令，可在绘制的电路上添加说明文本。

 : 绘制总线。

总线在电路图上表现出来的是一条粗线，它代表的是一组总线。当连接到总线上时，要注意标好网络标号。

 : 绘制子电路块。

 : 选择端子。

点击此图标按钮，在对象选择器中列出可供选择的各种常用端子：

DEFAULT：默认的无定义端子；

INPUT：输入端子；


OUTPUT：输出端子；


BIDIR：双向端子；


POWER: 电源端子;  
GROUND: 接地端子;  
BUS: 总线端子。

: 选择元件引脚。

点击此图标, 在对象选择器中列出可供选择的各种引脚(例如, 普通引脚、时钟引脚、反电压引脚和短接引脚)。

: 在对象选择器中列出可供选择的各种仿真分析所需的图表(如模拟图表、数字图表、混合图表和噪声图表等)。

: 当需要对设计电路分割仿真时, 采用此模式。

: 在对象选择器中列出各种信号源(如正弦、脉冲和 FILE 信号源等)模式。

: 在电路原理图中添加电压探针。电路仿真时可显示探针处的电压值。

: 在电路原理图中添加电流探针。电路仿真时可显示探针处的电流值。

: 在对象选择器中列出可供选择的虚拟仪器。

## 2) 2D 图形模式各图标按钮功能

: 画线。点击本图标, 右侧的窗口中提供了各种专用的画线工具, 具体如下:

COMPONENT: 用于元器件的连线;

PIN: 用于引脚的连线;

PORT: 用于端口的连线;

MARKER: 用于标记的连线;

ACTUATOR: 用于激励源的连线;

INDICATOR: 用于指示器的连线;

VPROBE: 用于电压探针的连线;

IProbe: 用于电流探针的连线;

GENERATOR: 用于信号发生器的连线;

TERMINAL: 用于端子的连线。

SUBCIRCUIT: 用于支电路的连线;

4D GRAPHIC: 用于二维图的连线;


WIRE DOT: 用于线连接点的连线;


WIRE: 用于线连接;


BUS WIRE: 用于总线的连线;

BORDER: 用于边界的连线;

TEMPLATE: 用于模板的连线。

: 画一个方框。

: 画一个圆。

: 画一段弧线。

: 图形弧线模式。

: 图形文本模式。

: 图形符号模式。