

职业教育课程改革创新规划教材



基于C语言与Proteus联合仿真的单片机技术

主编 / 丘利丽 何 波



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

职业教育课程改革创新规划教材

基于 C 语言与 Proteus 联合 仿真的单片机技术

主 编 丘利丽 何 波

参 编 李旭伟 陆志强

主 审 陈璐韶

贵州师范学院内部使用

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书适用于一体化教学。全书包括五个学习情境，分别是海珠桥灯饰工程的设计与调试、数字钟的设计与调试、轻工LED电子显示屏的设计与调试、家居报警系统的设计与调试和超声波汽车倒车雷达的设计与调试。这五个学习情境的整体结构采用由易到难、循序渐进的方式，内容包含了单片机最小系统、传感器、按键输入、定时中断、流水灯、数码管、点阵、LCD液晶显示器、继电器、蜂鸣器、步进电机和超声波知识点。每个学习情境分为几个学习任务，学习任务之间互有关联，都是为了实现学习情境中的最终产品而服务。每个学习任务中的程序层层递进，后面的程序在前面程序的基础上，稍作改动，即可实现新任务，让读者轻轻松松学习单片机。

本书可作为职业院校的单片机教材，也可以作为广大单片机爱好者及相关电子工程技术人员的参考书。

为方便教学，本书配有电子教学资料包，有需要的教师请登录华信教育资源网(www.hxedu.com.cn)，免费注册后下载。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

基于C语言与Proteus联合仿真的单片机技术 / 丘丽丽, 何波主编. —北京: 电子工业出版社, 2019.4

ISBN 978-7-121-35409-0

I. ①基… II. ①丘… ②何… III. ①单片微型计算机—职业教育—教材 IV. ①TP368.1

中国版本图书馆CIP数据核字(2018)第247164号

策划编辑: 白楠

责任编辑: 白楠 特约编辑: 王纲

印刷: 北京七彩京通数码快印有限公司

装订: 北京七彩京通数码快印有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开本: 787×1092 1/16 印张: 12.75 字数: 326.4千字

版次: 2019年4月第1版

印次: 2019年4月第1次印刷

定价: 30.50元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254592, bain@phei.com.cn。

前 言

随着嵌入式技术的飞速发展,嵌入式系统产品正不断渗透到各行各业,如智能家居、车载电子设备等。因此,单片机技术作为嵌入式计算机控制系统的重要技术,已经越来越受到各个应用领域的重视,尤其对于直接面向企业的职业院校,掌握单片机技术已经成为机电技术应用、电气控制、数控技术、电子信息、计算机应用等专业学生的基本技能。世界技能大赛之电子技术大赛中,单片机技术竞赛内容占有半壁江山。因此,全国的高职院校越来越重视单片机技术的教学。

单片机技术是一门理论与实践结合较强的技术。目前有关单片机的教材大多偏重理论,应用性方面的介绍比较少,并且章节之间没有太多的联系,不适应于现在高职教育提倡的“工学结合”的一体化教学模式。基于上述背景,编者结合自身十余年的单片机教学和指导学生参加世界技能大赛之电子技术大赛的经验,花费了两年多的时间编写本书,该书基于企业的“P(任务分析)-D(硬件及软件设计)-C(任务检查)-A(任务评估)”模式,体现了高职“学中教、教中学”的一体化教学特色。本书的特点包括以下几个方面。

1. 一体化教学,按企业的电子技术产品开发过程实施教学。

本书遵循企业的电子技术产品开发过程原则,让学生经历“任务分析→硬件设计→软件设计→硬件安装→整机调试→整机产品评估”工作过程,让学生从任务中来,到任务中去,提早让学生体验就业岗位,提高学生的职业认同感。

2. 选取典型、完整、难度适中的产品贯彻学习情境,结合理论和实践教学。

本书创设了五个学习情境,分别是海珠桥灯饰工程的设计与调试、数字钟的设计与调试、轻工LED电子显示屏的设计与调试、家居报警系统的设计与调试和超声波汽车倒车雷达的设计与调试。这五个学习情境的整体结构采用由易到难、循序渐进的方式,内容包含了单片机最小系统、传感器、按键输入、定时中断、流水灯、数码管、点阵、LCD液晶显示器、继电器、蜂鸣器、步进电机和超声波知识点。每个学习情境分为几个学习任务,学习任务之间互相关联,都是为了实现学习情境中的最终产品而服务。每个学习任务中的程序层层递进,后面的程序在前面程序的基础上,稍作改动,即可实现任务,让读者轻轻松松学单片机。每个实例演练完后,进一步提出“思考题”,让读者能即学即用,所学知识更加扎实。

3. 本书配套了“单片机实训开发板”和Proteus仿真图,方便“虚实相结合”的教学。

编者根据多年的教学经验,自行研发了与教材对应的一套“单片机实训开发板”。该套开发板全部是PCB板,分为多个模块,如单片机最小系统模块、流水灯模块、点阵模块、LCD模块等。该开发板使用简单,模块与模块之间采用跳线的形式连接,并且只要1条USB线把开发板与计算机连接,就可以实现程序下载。

编者建议:为提高学生的学习兴趣,有条件的学校可以为每位上课的学生配一套“单片机实训开发板”。这样学生可以利用开发板在实验室、图书馆、宿舍等地方随时随地学习

单片机技术。若读者需要本书配套的开发板,可以与编者联系,邮箱为“38729128@qq.com”。

4. 本书基于 C 语言与 Proteus 联合仿真,采用多文件、多任务的编程思路与方法。

C 语言具有易阅读、易移植的特点,现已成为嵌入式产品开发的主流语言。本书结合学习情境,采用 C 语言与 Proteus 联合仿真,在任务中理解和掌握 C 语言的理论知识,并且应用到实际任务中,达到举一反三的目的。在实际工作中,项目是一个大的工程,需要按功能进行分解。一般一个功能对应一个任务,一个任务对应一个程序文件,所以编者在本书中引入多文件、多任务的编程思路和方法。例如,学习情境五就是一个较大的工程,包含多个任务。通过该学习情境的学习,读者可以掌握多任务、多实时调度、多文件程序结构的综合系统调试方法。

广州市轻工高级技工学校的丘利丽、何波对本书的编写思路与大纲进行了总体策划,指导了全书的编写。丘利丽编写了学习情境一和学习情境二,何波编写了学习情境四和学习情境五,广州市轻工高级技工学校的陆志强参与编写了学习情境二,广州市轻工高级技工学校的李旭伟编写了学习情境三。广州市轻工高级技工学校的陈琨韶担任了本书主审,并提供了宝贵的编写建议。在此,一一表示衷心的感谢!

由于时间仓促和编者水平有限,书中难免有错误和不妥之处,恳请读者对本书提出批评与建议。

编者

目 录

情境 1 海珠桥灯饰工程的设计与调试	1
学习任务一: 8 位流水灯的设计与调试	1
学习任务二: 海珠桥灯饰工程的设计与调试	35
情境 2 数字钟的设计与调试	45
学习任务一: 1 位数码管的静态显示电路的设计与调试	45
学习任务二: 日期显示电路的设计与调试	56
学习任务三: 中断的设计与调试	63
学习任务四: 数字钟的设计与调试	74
情境 3 轻工 LED 电子显示屏的设计与调试	93
学习任务一: 8×8 点阵静态显示系统的设计与调试	93
学习任务二: 轻工 LED 点阵电子显示系统的设计与调试	104
情境 4 家居报警系统的设计与调试	121
学习任务一: 液晶显示屏的设计与调试	121
学习任务二: 家居报警系统的设计与调试	143
情境 5 超声波汽车倒车雷达的设计与调试	156
学习任务一: 步进电机正反转电路的设计与调试	156
学习任务二: 超声波测距仪的设计与调试	168
学习任务三: 超声波汽车倒车雷达的设计与调试	181
参考文献	195

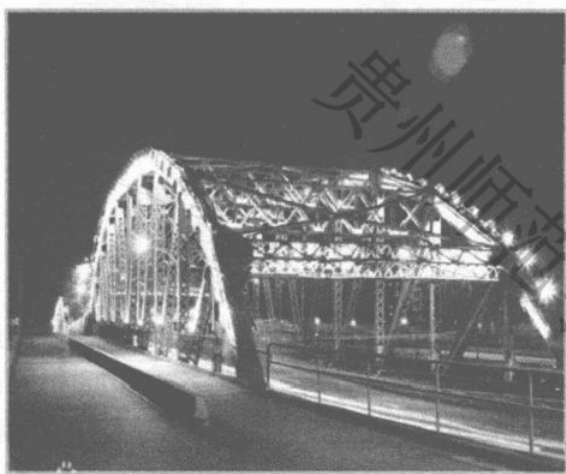
情境 1 海珠桥灯饰工程的设计与调试



情境介绍

随着人们生活环境的不断改善和美化，在许多场合可以看到彩色霓虹灯不断变化。闪烁的 LED，以造价低廉、控制简单等特点得到了广泛的应用，用彩灯来装饰街道和城市建筑物已经成为一种时尚。

海珠桥是广州历史悠久的广府文化传承代表。因此，本情境以单片机控制技术为基础，由学生自主完成海珠桥的硬件和软件设计与调试，设计丰富多彩的海珠桥灯饰效果，使其摇身变为珠水之上的巨型灯屏。



学习任务一：8 位流水灯的设计与调试



任务描述

在 Proteus 仿真软件和单片机开发板上实现 8 位流水灯顺序点亮效果，并能控制它们的点亮速度。



任务目标

- (1) 能正确分析单片机最小系统的电路结构及各部分的功能。
- (2) 学会根据任务要求, 自主设计 8 位流水灯的硬件电路。
- (3) 正确理解 Keil C 语言的基本结构、数据类型、常数、变量、运算符、循环指令及选择指令的知识点。
- (4) 熟练使用 Proteus 和 Keil μ Vision3 软件, 完成程序的设计与调试。
- (5) 能正确使用 STC-ISP-V488 程序下载软件, 完成程序的下载, 并观察开发板上 8 位流水灯的工作过程。

建议课时: 18 课时



任务分析

单片机 P2 口连接 8 个发光二极管, 利用各引脚输出电位的变化, 控制发光二极管的亮灭: 输出电位为高电平, 发光二极管灭; 输出电位为低电平, 发光二极管亮。为了清楚地分辨发光二极管的点亮和熄灭, 编写延时程序, 在 P2 口输出信号由一种状态向另一种状态变化时, 实现一定时间的间隔。



任务实施

一、硬件电路设计

1. 硬件设计思路

设计思路: 利用 STC 单片机芯片, 外加振荡电路、复位电路、控制电路、电源, 组成一个单片机最小系统。在最小系统的基础上, 利用 P2 口的 8 个引脚控制 8 个发光二极管。由于发光二极管具有普通二极管的共性——单向导电性, 因此只要在其两极间加上合适的正向电压, 发光二极管即可点亮; 将电压撤除或加反向电压, 发光二极管即熄灭。根据发光二极管的特性, 结合单片机 P2 口的输出信号, 即可实现流水灯的控制效果。

2. 电路硬件设计

选用 51 单片机芯片, 该芯片共有 40 个引脚, 如图 1-1-1 所示。

(1) 主电源电路

VCC (40 脚): 接+5V 电源, 又称电源引脚。

GND (20 脚): 接地。

(2) 时钟电路

单片机时钟信号的提供有两种方式: 内部方式和外部方式。

内部方式是指使用内部振荡器, 在 XTAL1 (19 脚) 和 XTAL2 (18 脚) 之间外接石英晶体和陶瓷电容 C_1 和 C_2 [图 1-1-2 (a)], 它们和单片机的内部电路构成一个完整的振荡器, 振荡频率和石英晶体的振荡频率相同。电容器 C_1 和 C_2 容量为 30pF, 石英晶体的

振荡频率为 12MHz。

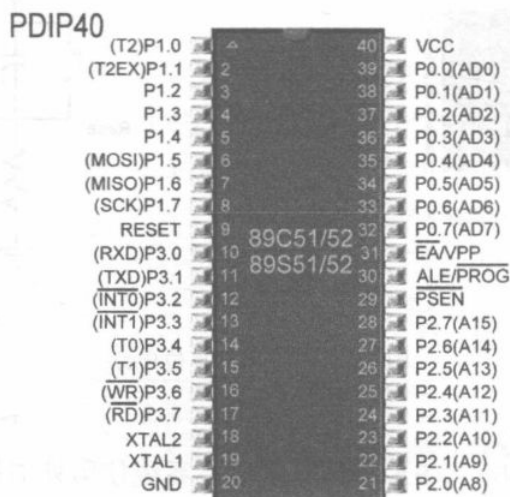


图 1-1-1 单片机芯片引脚

当使用外部信号源为单片机提供时钟信号时，XTAL1 为空引脚，XTAL2 外接时钟信号，如图 1-1-2 (b) 所示。

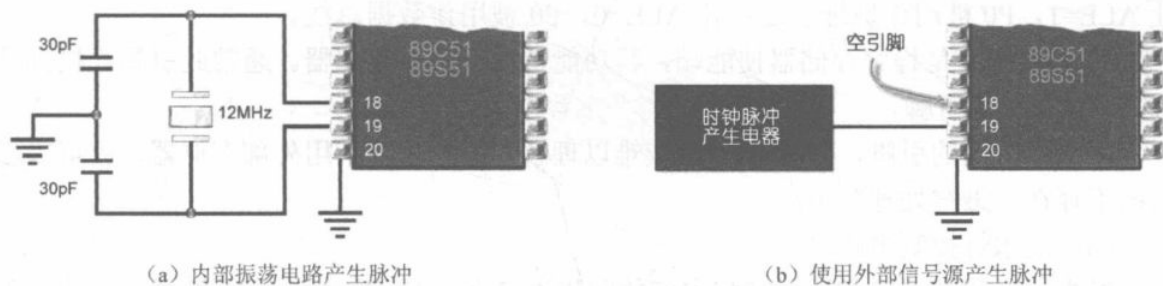


图 1-1-2 单片机时钟电路

本任务使用内部振荡器，因此在 XTAL1 和 XTAL2 之间外接 12MHz 的石英晶体和 30pF 的陶瓷电容 C_1 和 C_2 即可。

(3) 复位电路

复位是单片机的初始化操作，使 CPU 及其他功能部件都处于一个确定的初始状态，并从这个状态开始工作。除系统正常上电（开机）外，在单片机工作过程中，如果程序运行出错或操作错误使系统处于死机状态，也必须进行复位，使系统重新启动。

复位引脚是第 9 脚，此引脚连接高电平超过两个机器周期，即可产生复位的动作。以 12MHz 的时钟脉冲为例，每个时钟脉冲为 $1/12\mu\text{s}$ ，两个机器周期为 $2\mu\text{s}$ ，因此，在该引脚上产生一个 $2\mu\text{s}$ 以上的高电平脉冲，即可产生复位的动作。

复位有上电复位和按键复位两种，如图 1-1-3 所示。上电复位 [图 1-1-3 (a)] 利用复位电路电容充放电来实现；而按键复位 [图 1-1-3 (b)] 通过使 RST 端经电阻器 R 与 +5V 电源接通而实现，它兼有自动复位的功能。

电路中 R 和 C 组成一个典型的充放电电路，充放电时间 $T=1/RC$ 。根据理论计算结果可知，选择时钟频率为 12MHz，一个机器周期是 $1\mu\text{s}$ 。只要 $T>2\mu\text{s}$ ，就可以复位。本任务开发板选用 $R=10\text{k}\Omega$ ， $C=10\mu\text{F}$ 。

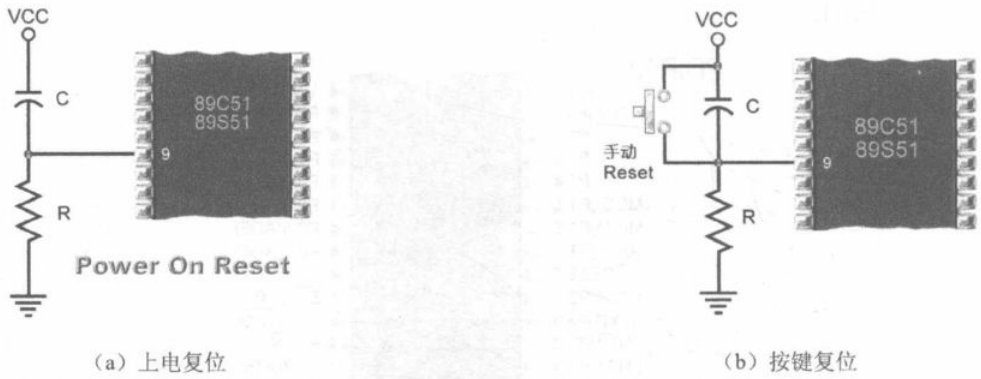


图 1-1-3 单片机复位电路

(4) 存储器设置电路

31脚EA为复用引脚。当EA为低电平时，系统使用外部存储器。当EA为高电平时，系统使用内部存储器。对于初学者而言，所写的程序比较简单，大多使用内部存储器，所以就把31脚直接接到VCC。

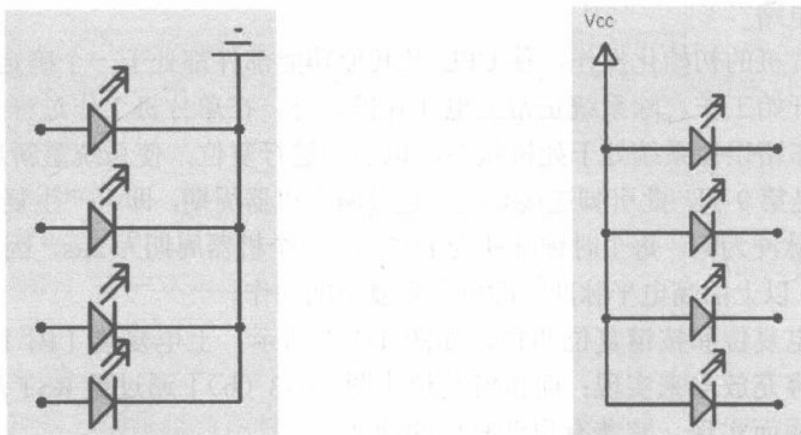
30脚ALE是地址锁存信号，其功能是在存取外部存储器时，将原本在P0的地址信号锁存到外部存储器IC，让P0口空出来，以传输数据。简单讲，当外接存储器电路时，让ALE=1，P0被用作地址总线；让ALE=0，P0被用作数据总线。

29脚PSEN是程序存储器使能端，其功能是读取外部存储器。通常此引脚连接到外部存储器的OE引脚。

相对于前面的引脚，29、30脚比较难以理解。但是只要不用外部存储器，就可以当它们不存在，悬空处理即可。

(5) 流水灯控制电路

发光二极管的连接方法：若将它们的阴极连接在一起，阳极信号受控制，即构成共阴极接法，如图1-1-4(a)所示；若将它们的阳极连接在一起，阴极信号受控制，则构成共阳极接法，如图1-1-4(b)所示。由于P2口引脚输出高电位时电压大约是5V，为保证发光二极管可靠工作，必须在发光二极管和单片机输出引脚间连接一只限流电阻。



(a) 共阴极发光二极管的接法

(b) 共阳极发光二极管的接法

图 1-1-4 发光二极管的接法

本任务选用硅型普通发光二极管，限流电阻取 220Ω 。

3. 硬件电路原理图

单片机的 P0、P1 和 P2 端口都是双向的 I/O 端口，P3 端口既可作为普通的 I/O 端口，又可用于第二功能操作中。在该任务中，选择 P2 端口作为流水灯的控制端口，实现数据的输入输出。综上所述，得到图 1-1-5 所示的 8 位流水灯的电路原理图。

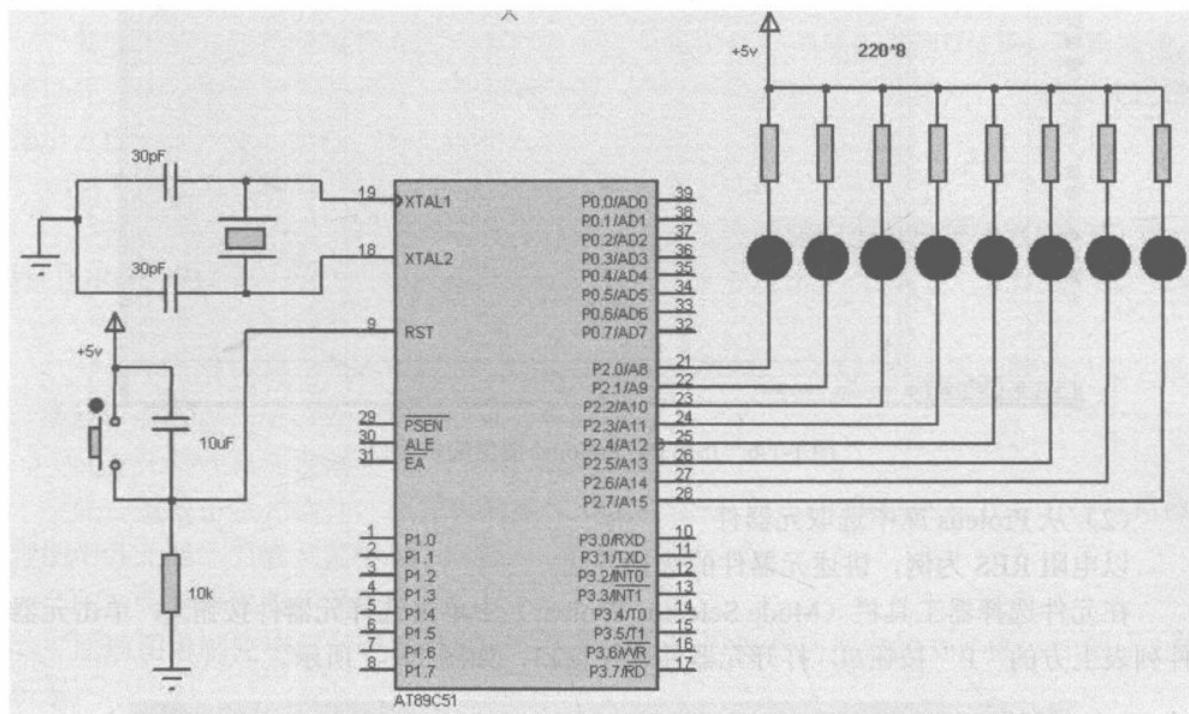


图 1-1-5 8 位流水灯的电路原理图

根据电路原理图，确定本任务所需要的元器件清单，见表 1-1-1。

表 1-1-1 海珠桥灯饰元器件清单

序号	名称	型号	数量(个)
1	单片机	AT89C51	1
2	发光二极管: LED	-	8
3	电阻: RES	220Ω	8
		$10k\Omega$	1
4	电容: CAP	$10\mu\text{F}$	1
		30pF	2
5	晶体振荡器: CRYSTAL	12MHz	1
6	按钮: BUTTON	不带自锁	1

4. 在 Proteus 仿真软件上绘制流水灯电路原理图

(1) 打开软件

选择“程序”→“Proteus 7 Professional”→“ISIS 7 Professional”命令，启动 Proteus 仿真软件，出现 ISIS Professional 图像编辑窗口，如图 1-1-6 所示。

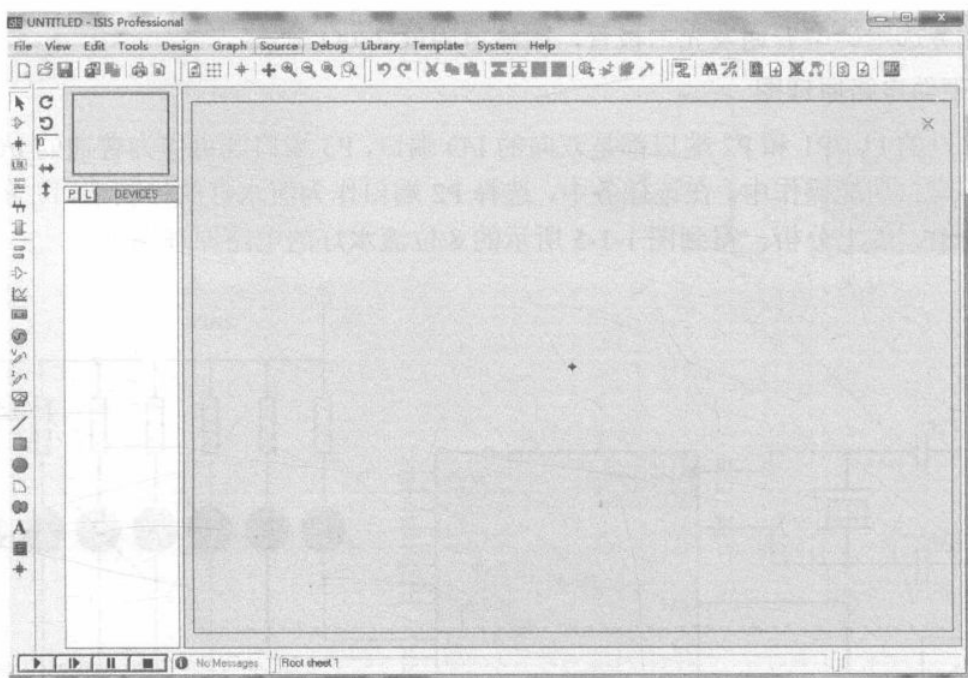




图 1-1-6 ISIS Professional 图像编辑窗口

(2) 从 Proteus 库中选取元器件

以电阻 RES 为例，讲述元器件的选择方法。

在元件选择器工具栏 (Mode Selector Toolbar) 中单击选择元器件按钮, 单击元器件列表上方的“P”按钮, 打开元器件选择窗口, 如图 1-1-7 所示。

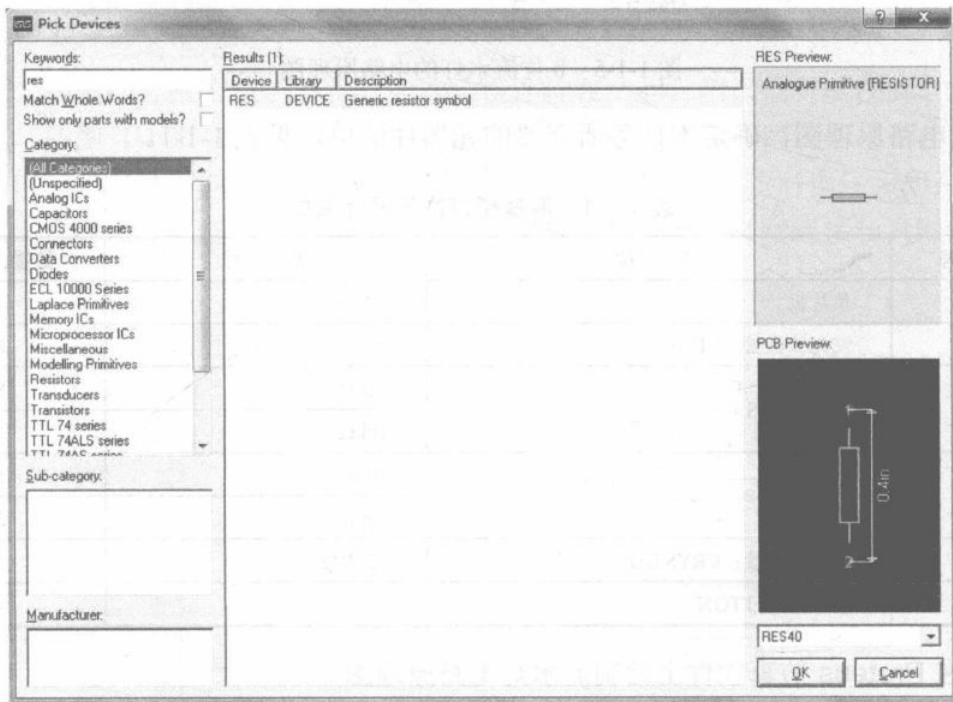






图 1-1-7 元器件选择窗口

在元器件选择窗口左上角的关键字栏中输入关键字，例如需要电阻就输入“res”，




从元件库中选取元器件。以此类推，可以选取单片机、电容、发光二极管、按钮、晶振等元器件。

(3) 放置元器件

在对象选择器中单击要放置的元器件（蓝色高亮条表示目前选取的元器件），然后在编辑窗口中合适的位置单击就放置了一个元器件。依次把各元器件放入编辑区中的适当位置。

若要改变元器件的放置方向，可以右击选中元器件后再单击按钮或。若要镜像，可以先右击选中元器件，再单击按钮或。若要多个元器件一起转向，可先按住左键拖出方框选中多个元器件，再单击相应的操作按钮。

(4) 放置电源和地

单击元件选择器工具栏中的端子按钮，在对象选择器中选取电源（POWER）、地（GROUND），分别放置于编辑窗口的合适位置上。

(5) 连线

分别单击要连线（元器件引脚、终端、线）的起点和终点，在这两点间会自动生成一条线。若终点在空白处，双击即可结束画线。

(6) 元器件属性设置

先左键双击各元器件，在弹出的属性编辑对话框（Edit Component）中，按电路原理图中各元器件的值设置相应的属性。

(7) 绘制原理图并保存设计

原理图绘制完毕后，单击“File”→“Save as”保存到指定的文件夹，如图 1-1-8 所示。

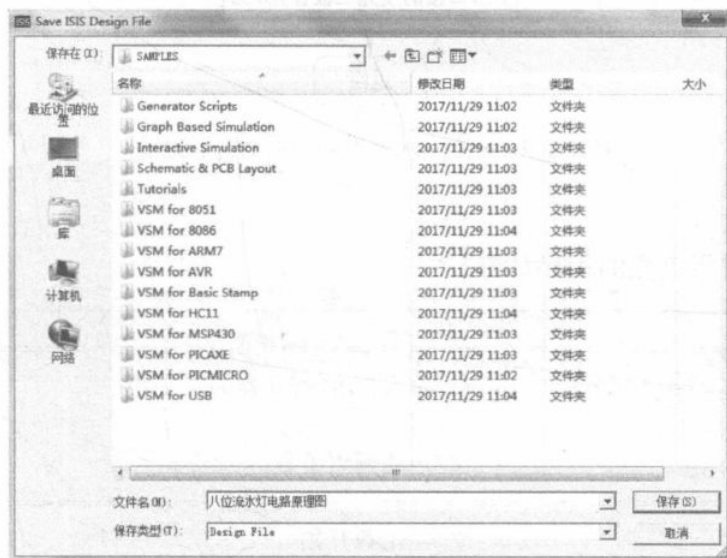


图 1-1-8 保存“八位流水灯电路原理图”

二、软件程序设计与调试

1. 软件设计思路

利用单片机的 P2 口来控制 8 个 LED，让这 8 个 LED 依次点亮，其设计步骤如下。

当 P2 口的引脚输出低电平 (0) 时,其所连接的 LED 呈现正向偏压而发亮;当引脚输出高电平 (1) 时,其所连接的 LED 呈现反向截止而熄灭。因此,我们的程序设计就要让 P2.0 口接的灯亮,输出为 11111110,以十六进制表示为“FE”;延时一段时间后,P2.1 口接的灯亮,输出为 11111101,以十六进制表示为“FD”。以此类推,周而复始。

2. 绘制程序流程图

有了设计思路后,可以将思路转换成流程图,如图 1-1-9 所示。



图 1-1-9 8 位流水灯循环点亮程序流程图

3. 编写程序

8 位流水灯循环点亮的程序如下:

```

//=====声明区=====
#include <reg51.h> //定义8051寄存器的头文件
#define led P2
void delay(int); //声明延迟函数
//=====主程序=====
main( ) //主函数开始
{ led=0xff; //流水灯初始状态,全熄灭
while(1) //无限循环
{led=0xfe; //P2.0口的灯亮
delay(500); //延时0.5s
led=0xfd; //P2.1口的灯亮
delay(500); //延时0.5s
led=0xfb; //P2.2口的灯亮
}
}
  
```

```

delay(500);           //延时0.5s
led=0xf7;            //P2.3口的灯亮
delay(500);           //延时0.5s
led=0xef;            //P2.4口的灯亮
delay(500);           //延时0.5s
led=0xdf;            //P2.5口的灯亮
delay(500);           //延时0.5s
led=0xbf;            //P2.6口的灯亮
delay(500);           //延时0.5s
led=0x7f;            //P2.7口的灯亮
delay(500);           //延时0.5s
}                       //while循环结束
}                       //主程序结束
//=====延迟函数=====
void delay(int x)      //延迟函数开始, x=延迟次数
{int i, j;             //声明整数变量i, j
for (i=0;i<x;i++)     //计数x次, 延迟x*1ms
for (j=0;j<120;j++); //计数120次, 延迟1ms
}                       //延迟函数结束

```

4. 在 Keil μ Vision3 集成开发环境中新建工程和文件, 编写流水灯程序

(1) 在“开始”菜单中, 选择“程序”→“Keil μ Vision3”选项, 即可进入集成开发环境, 如图 1-1-10 所示。

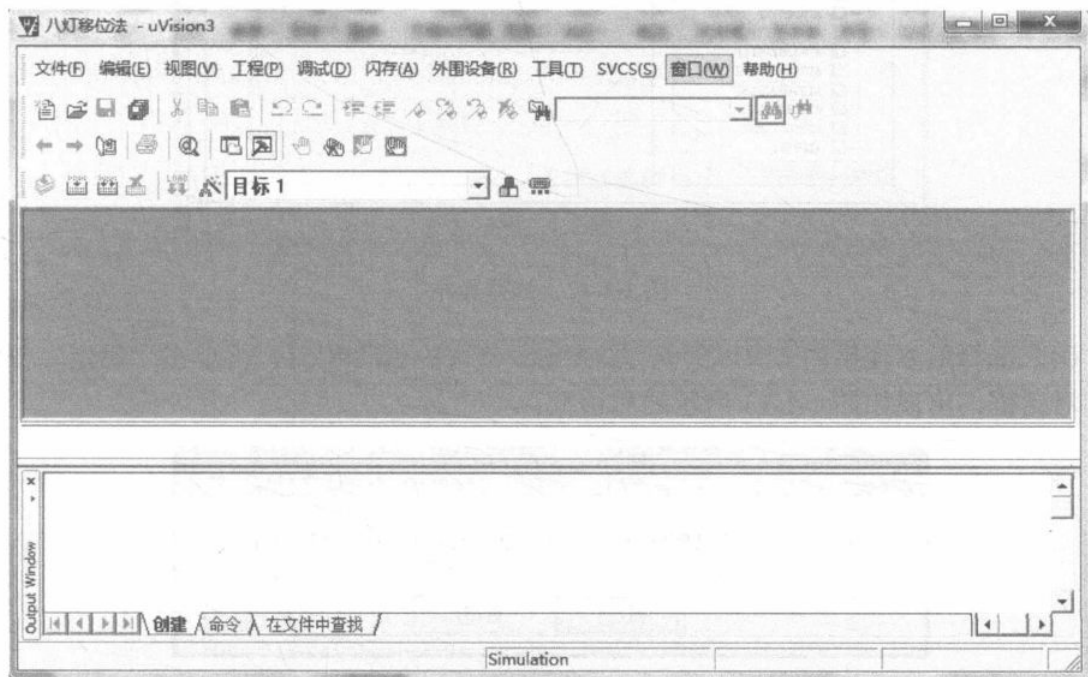


图 1-1-10 Keil μ Vision3 集成开发环境

(2) 打开一个项目, 启动“工程”菜单下的新建工程命令, 出现如图 1-1-11 所示的对话框。

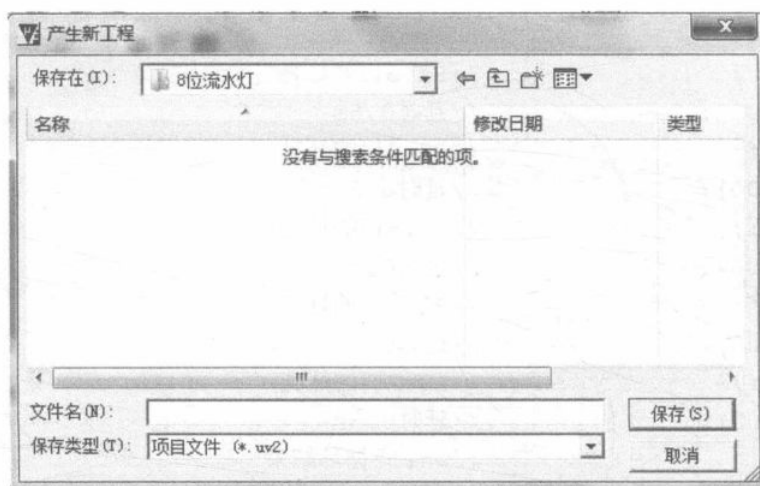


图 1-1-11 保存项目

(3) 在“文件名”栏中填写要新增的项目名称，再单击“保存”按钮，出现如图 1-1-12 所示的对话框。

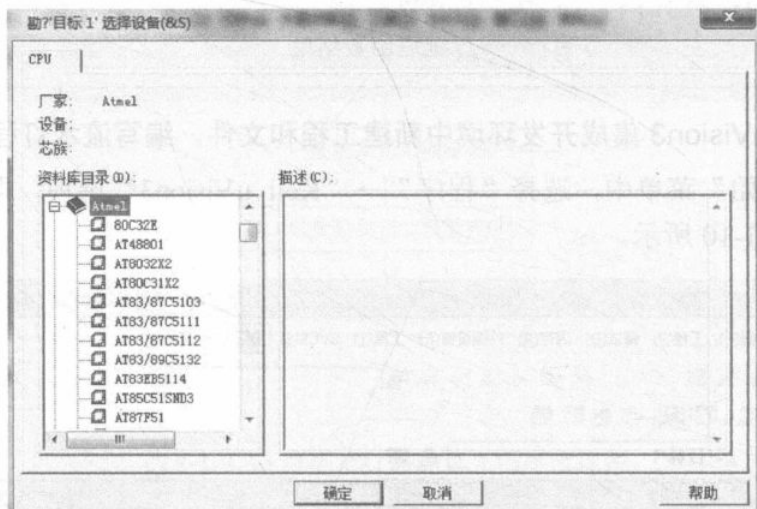


图 1-1-12 选择器件

(4) 选择所要使用的 CPU 芯片，如 Atmel 公司的 AT89C51，再单击“确定”按钮，关闭对话框，出现如图 1-1-13 所示的对话框。

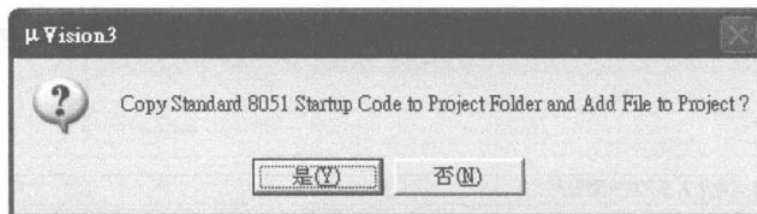


图 1-1-13 添加启动代码

(5) 这时系统询问我们要不要将汇编语言的启动代码放入所编辑的项目文件，在此单击“否”按钮关闭此对话框，则在左边产生“目标 1”项目，如图 1-1-14 所示。

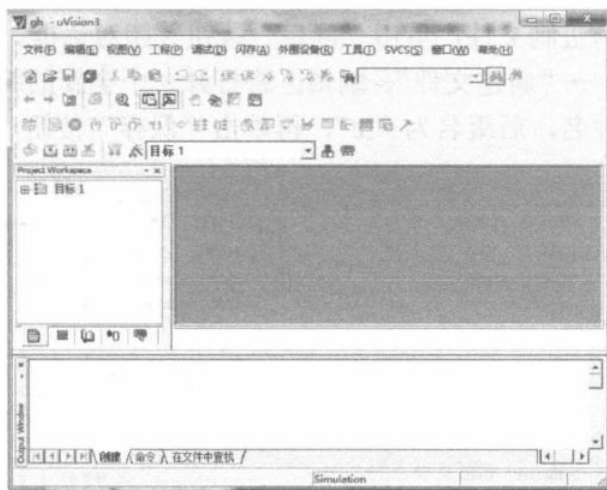


图 1-1-14 新建项目界面

(6) 单击“工程”→“为目标 1 设置选项”，出现如图 1-1-15 所示的对话框。

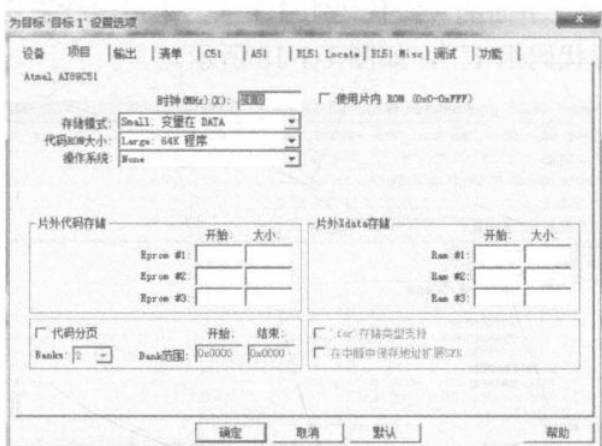


图 1-1-15 选择晶振频率

(7) 在此对话框中设置此芯片的工作频率为 12MHz，然后单击“输出”选项卡，如图 1-1-16 所示。

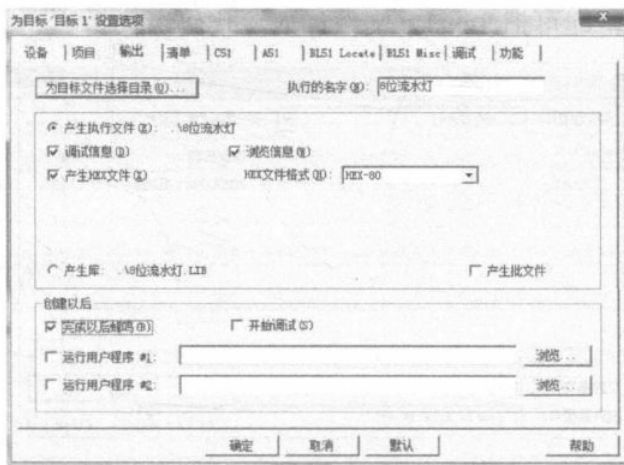


图 1-1-16 选择产生十六进制文件