

仿真和集成开发环境使用

单片机应用系统以单片机为核心,设计外围电路并通过软件编程来实现系统功能。单片机应用系统的设计包括硬件电路设计和软件系统设计。本书的例程均包含仿真电路图和C程序代码,程序是在 Keil μ Vision4 集成开发环境中进行编写和编译,电路是在 Proteus 7.5 软件进行原理图绘制并加载 Keil μ Vision4 编译生成的 .hex 文件进行仿真。本章对 Keil μ Vision4 和 Proteus 7.5 的初步使用进行讲解。对于单片机工程开发项目,程序调试编译后要将 .hex 文件下载到单片机中程序才能够执行,在电路板中实现所设计开发的功能。本章介绍 51 单片机的编程器下载、AT89S51 单片机的 ISP 下载和 STC 单片机的串口下载的方法。

3.1 Keil μ Vision4 使用简介

通过 Keil μ Vision4 集成开发环境可以快速地实现程序的编辑、仿真和调试功能。本节仅对工程的建立和程序编译步骤和基本的调试功能做简要介绍,更多的高级调试技巧参阅 Keil 软件的技术手册。

3.1.1 Keil μ Vision4 建立工程及程序编译

1. 项目文件创建

选择 Project→New μ Vision Project 命令,如图 3-1 所示。

选择保存路径,输入工程文件名称 LED,如图 3-2 所示,然后单击“保存”按钮。

弹出如图 3-3 所示的界面,提示选择单片机的型号,根据开发者使用的单片机来选择,Keil C51 几乎支持所有的 51 内核的单片机,这里选择 89C51 之后,右侧的 Description 栏是对这个单片机功能的基本说明,查看所选单片机功能是否满足项目开发的要求,确认所选的单片机类型后,单击 OK 按钮。Keil 元件列表下的单片机型号并不涵盖所有的 51 内核单片机型号,如 51 单片机市场份额最大的国产宏晶单片机 STC 并不在列,因为都是采用英特尔公司 51 内核,如果使用 STC 系列单片机,选择相近型号 51 内核单片机代替即可,对后续编程不会有任何影响。

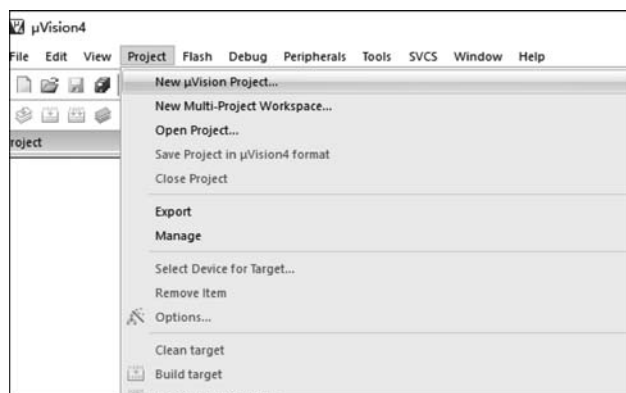


图 3-1 新建工程文件



图 3-2 新建工程文件保存路径

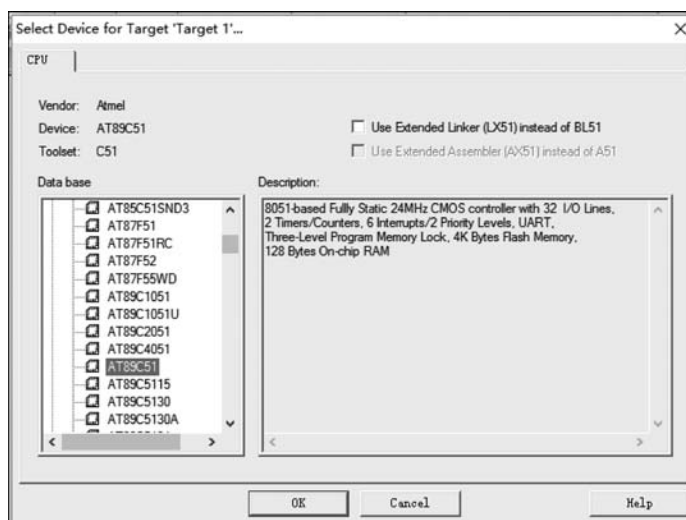


图 3-3 选择单片机型号

弹出如图 3-4 所示对话框,问是否加入启动代码到工程中。启动代码 STARTUP.A51 文件就是处理器最先运行的一段代码,操作包括清除数据存储器内容、初始化硬件及可重入堆栈指针,以上的操作均与处理器体系结构和系统配置密切相关,由汇编语言来编写,用户也可根据项目需求对启动代码进行修改。如果工程里没有加入启动代码,Keil 就会使用库文件里的默认启动代码,如果工程中加入了启动代码,Keil 就会编译并使用加入的启动代码。由于一般情况不需要修改默认的启动代码,所以这里是否加入启动代码都一样,这里选择“否”。

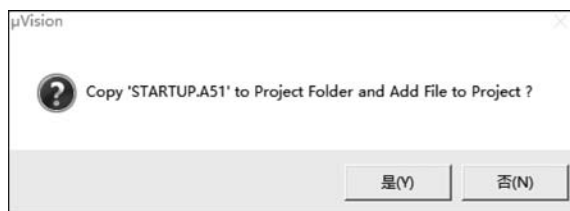


图 3-4 是否加入启动代码对话框

完成上一步后,软件界面如图 3-5 所示,此时已经建立好了一个工程文件。

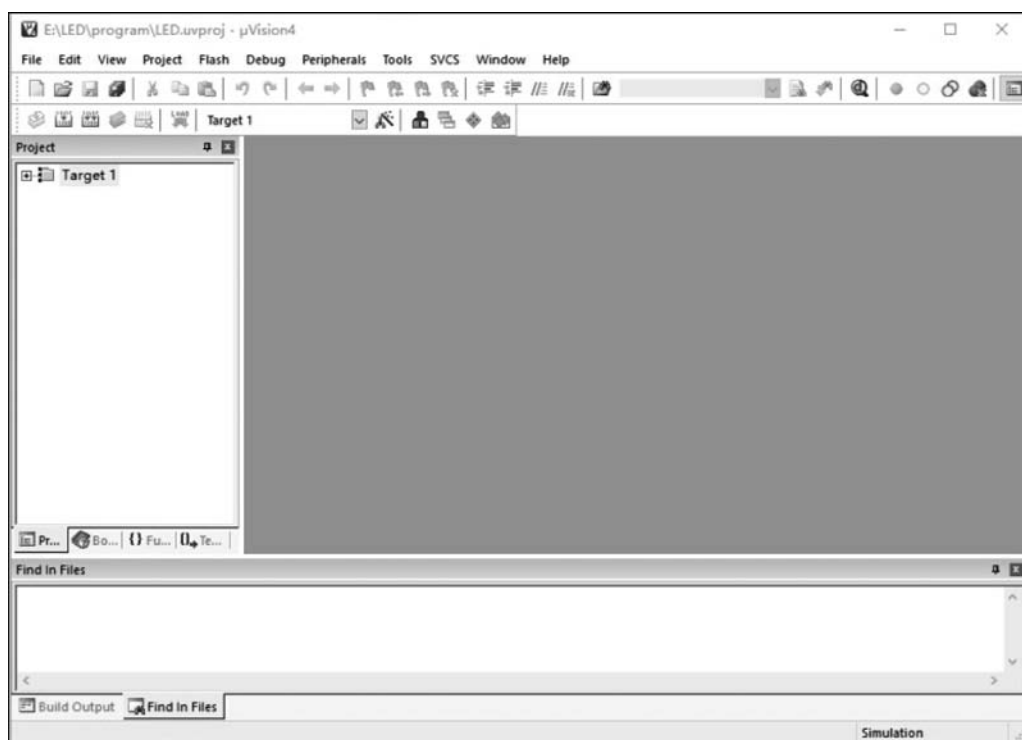


图 3-5 完成新建工程文件

2. 创建新的源程序文件并添加

现在工程文件里面还没有程序,需要新建和加入程序,一个工程可以加入多个程序文件。选择 File→New 命令,此时编辑窗口新建了一个空白文档,光标在编辑窗口里闪烁,提

示可以输入应用程序。在输入程序之前需要保存该空白文档,选择 File→Save As 命令,屏幕如图 3-6 所示,在“文件名”栏右侧的编辑框中输入欲使用的文件名,用 C 语言编写扩展名为 .c,用汇编语言编写扩展名为 .asm,创建头文件扩展名为 .h。在这里输入 main.c,然后单击“保存”按钮,如图 3-6 所示。

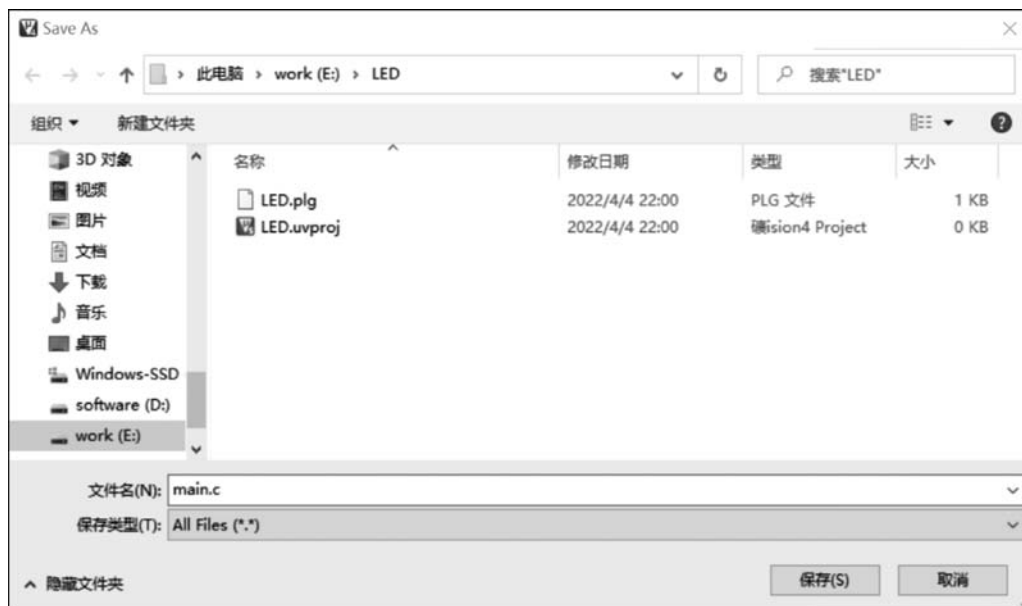


图 3-6 保存新建文档

回到编辑界面后,单击 Target 1 前面的“+”号,然后在 Source Group 1 上右击,弹出菜单如图 3-7 所示。

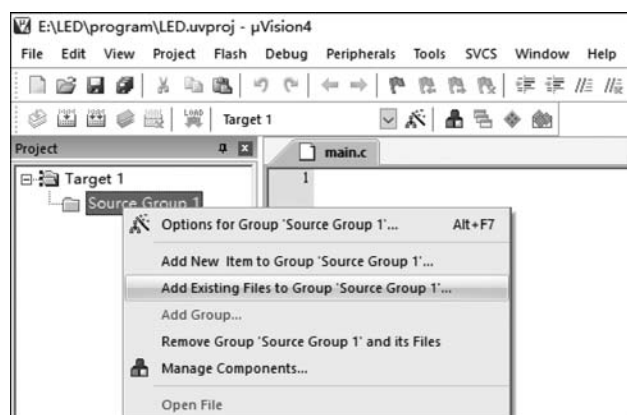


图 3-7 打开添加文档界面

在此菜单栏中,向工程中添加程序文件,单击 Add Existing Files to Group ‘Source Group 1’. 选中刚才建立的 main.c 文件,然后单击 Add 按钮,如图 3-8 所示。

此时 Source Group 1 文件夹中多了一个子项 main.c,子项的多少与所增加的源程序的多少相同。

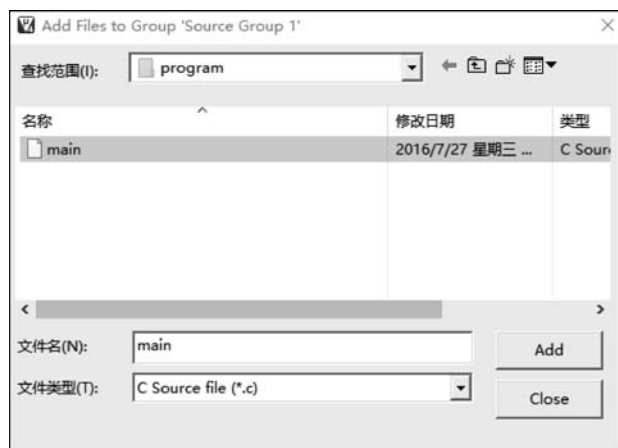


图 3-8 向工程中添加程序文件

3. 目标文件设置

下面需要对目标文件进行设置,本节仅讲后续例题编程必要的设置,更多的设置可参照 Keil μ Vision4 使用指南。

选择 Project→Options for Target Target1 命令,或者直接单击工具栏上  图标。在此进行两项设置:第一个是设置晶振频率,根据单片机最小系统所选用的外部晶振频率进行设置,这里将外部晶振频率设为 12.0MHz,如图 3-9 所示,第二项设置生成 HEX 文件,选择 Output 选项,出现如图 3-10 所示设置界面,勾选 Create HEX File 即可。HEX 文件格式是可以烧写到单片机中,被单片机执行的一种文件格式,单片机程序经 Keil 编译后,生成可供单片机执行的 HEX 文件,若未勾选此项,编译后将不产生 HEX 文件,单片机程序烧写和仿真必须加载 HEX 文件,因此必须勾选该项。



图 3-9 晶振频率设置

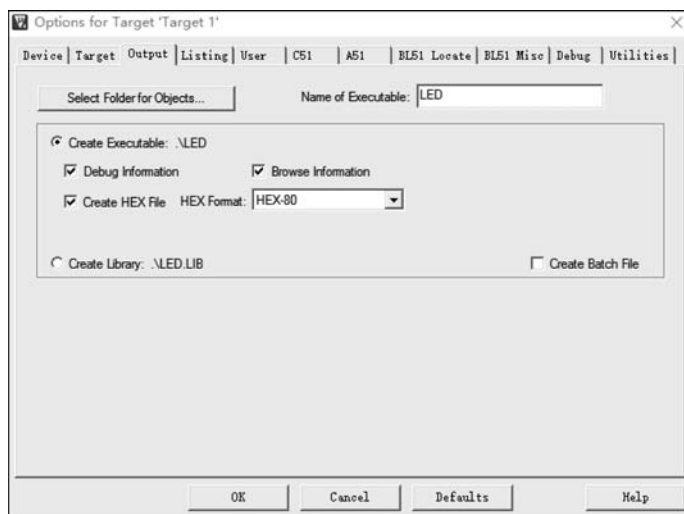


图 3-10 勾选生成 HEX 文件

4. 输入程序、编译项目

进行了上述设置以后就可以在程序界面中输入程序，本例输入功能为用单片机 P1.0 口控制 LED 闪烁的程序。程序输入完毕后就可以对源程序进行编译。单击图标  或在 Project 下拉菜单中选择 Build Target 命令就可以编译源程序并生成应用。当所编译的程序有语法错误时，将会在输出窗口的编译页中显示警告信息。在下面的程序中，故意在“LED=~LED”语句后遗漏输入“;”，则编译提示错误，如图 3-11 所示。编译提示“MAIN.C(19): error C141: syntax error near 'delay'”。

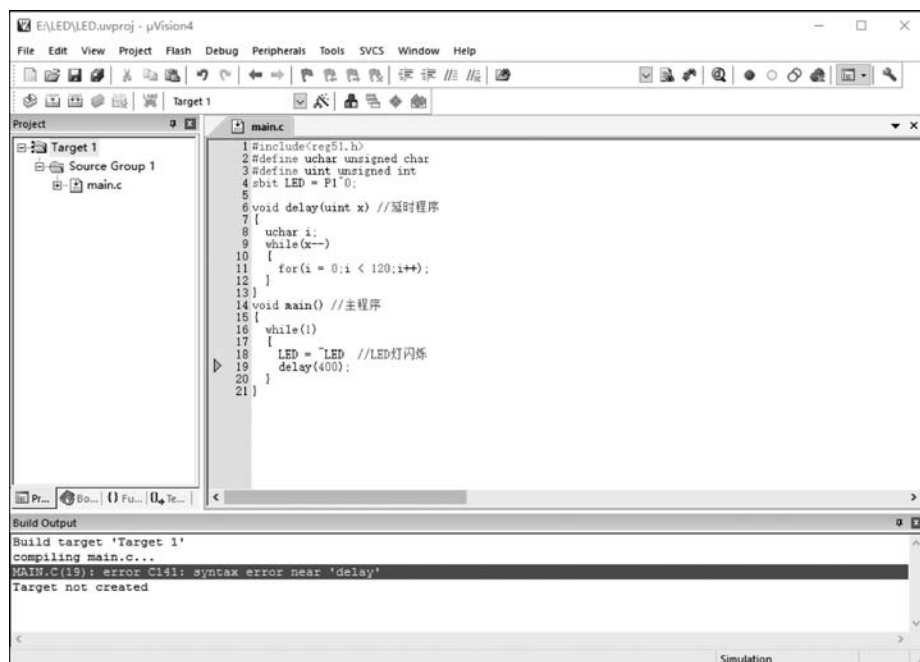


图 3-11 编译有误界面

C(19): error c141: syntax error near 'delay'",提示在 delay 语句附近有语法错误,双击提示栏中的错误信息,光标就指向出现错误的源程序位置上,在“LED=~LED”语句后输入“;”修改错误,再次编译,如图 3-12 所示,显示编译成功。在存放程序的文件夹下,生成 LED.hex 文件。在图 3-12 中可以看到 Keil 软件界面有菜单栏、工具栏、工程管理区、程序代码输入区和编译信息输出窗口。

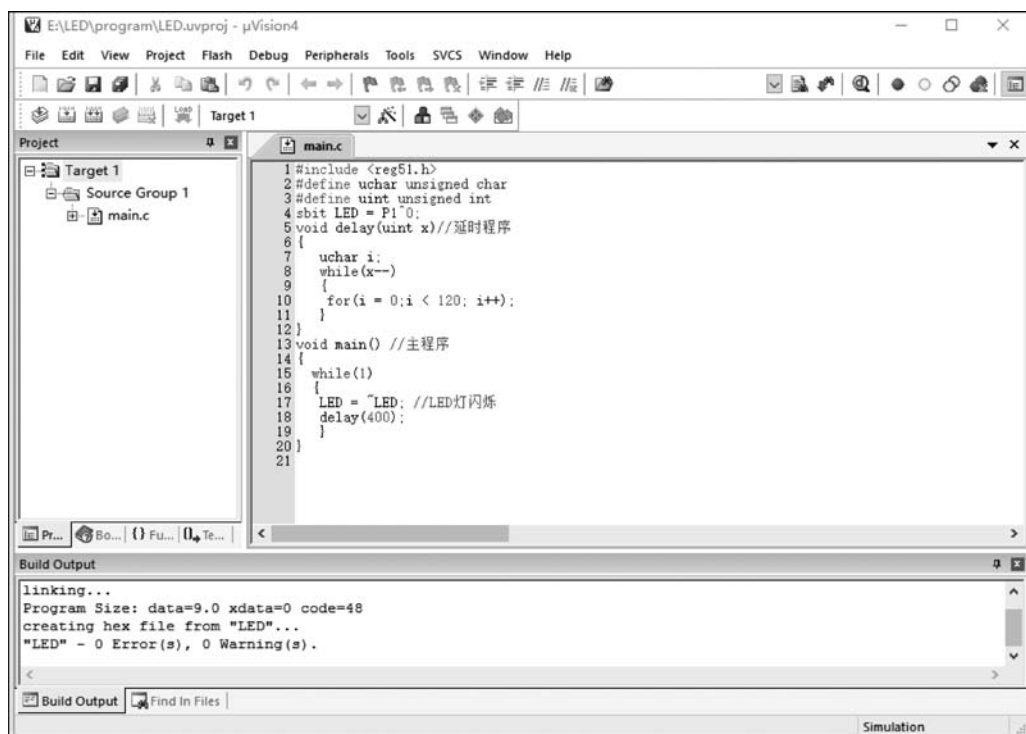


图 3-12 编译成功界面

编译信息输出窗口还提供了“Program Size: data=9.0 xdata=0 code=48”的信息。此信息提示编译后的程序占用单片机系统存储的容量,本信息提示片内数据存储器 RAM 占用 9 字节,片外数据存储器 RAM 占用 0 字节,程序存储器 ROM 占用 48 字节。此提示信息可供编程者在优化程序时作为参考,以及在选择不同存储容量型号的单片机时作参考。

3.1.2 Keil μVision4 调试程序和仿真

使用 μVision4 调试器可对源程序进行测试,μVision4 提供了两种仿真操作模式,这两种模式可以在 Option for Target‘Target1’对话框的 Debug 栏中进行选择,如图 3-13 所示。Use Simulator 为软件仿真模式,将调试器配置成纯软件仿真,能够仿真 51 单片机绝大多数功能模块(如并口、串行口、定时器等)而不需要任何硬件目标板,仿真结果可通过 μVision4 的串行窗口、观察窗口、存储器窗口等窗口直接显示。μVision4 提供了多种目标硬件调试驱动,界面右边 Use 后的下拉菜单可选择不同的硬件仿真器进行硬件仿真,需要连接仿真器和目标电路板进行仿真。硬件仿真在程序调试运行中可以观察到目标硬件的实时执行效果,有利于分析和排除各种软硬件故障。

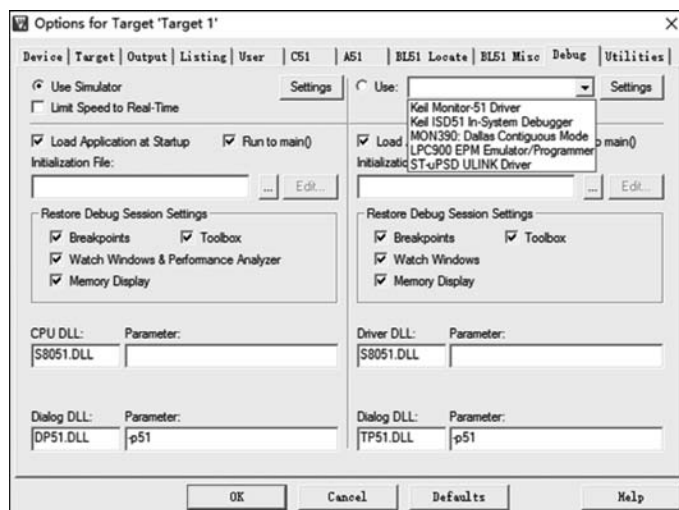


图 3-13 仿真模式设置

完成编译、链接、仿真调试设置后,可以对刚才建立项目进行调试。单击  或选择 Debug→Start/Stop Debug Session 命令,启动或退出调试器。启动调试器后进入仿真调试状态,调试界面如图 3-14 所示。

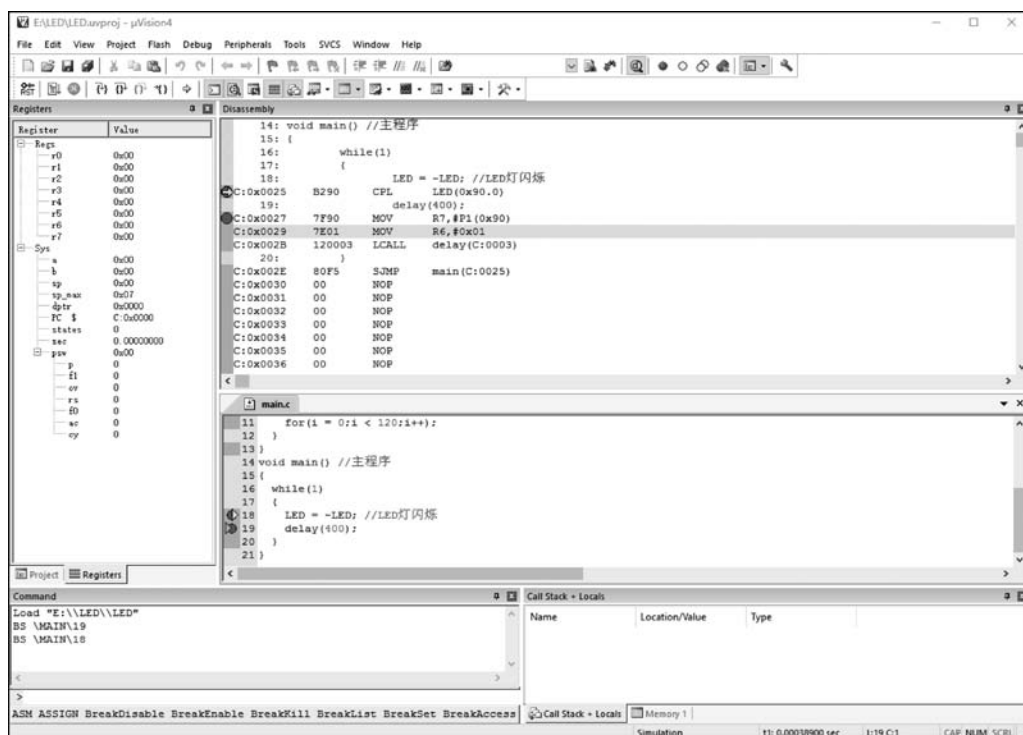
图 3-14 Keil μ Vision4 仿真调试窗口

图 3-14 左上角的 Registers 窗口显示了寄存器 r0~r7,a,b,sp,psw 等特殊功能寄存器的值,程序的运行时间 sec,这些值随着程序的执行发生变化。Disassembly 窗口为反汇编

窗口,反汇编窗口同时显示了目标程序、编译的汇编程序和二进制文件。

Debug 工具栏用于控制程序运行的操作,可以打开各类窗口观察程序运行中的各类寄存器和运行状态的实时值,以便于进行程序分析。Debug 工具栏按钮功能如表 3-1 所示。

表 3-1 Debug 工具栏按钮功能说明

按 钮	名 称	说 明
	Reset	复位 CPU。单击此按钮程序复位,即程序回到 main 函数的开头处
	Run	全速执行。程序全速运行,程序连续全速运行,若设置断点,运行到断点处停止
	Stop	程序停止运行
	Step	单步运行。程序单步执行,若运行到子函数,则进入子函数
	Step Over	过程单步。运行到子函数不进入,将子函数当作一个整体运行
	Step Out	执行完当前子程序。程序在子程序内部运行,跳出当前子程序
	Run to Cursor Line	运行到光标处
	Show next statement	显示下一条语句
	Command Windows	命令窗口
	Disassembly Windows	反汇编窗口
	Symbols Windows	符号窗口
	Registers Windows	寄存器窗口
	Call Stack Windows	调用堆栈窗口
	Watch Windows	监视窗口
	Memory Windows	内存窗口
	Serial Windows	串口窗口
	Analysis Windows	逻辑分析窗口
	Trace Windows	追踪窗口
	System Viewer Window	系统查看器窗口
	Toolbox	工具箱

还可通过 Peripherals 菜单栏下的 Interrupt、I/O-Ports、Serial、Timer 菜单查看中断寄存器、I/O 口、串口和定时器在程序运行中的实时值。

程序调试时,一些程序行必须满足一定的条件才能被执行(如程序中某变量达到一定的值、按键被按下、串口接收到数据、有中断产生等),这些条件往往是异步发生或难以预先设定的,这类问题使用单步执行的方法很难调试,需使用程序调试中的另一种非常重要的方法——断点设置。断点设置的方法有多种,常用的是在某一程序行设置断点,设置好断点后可以全速运行程序,一旦执行到该程序行即停止,可在此观察有关变量值,以确定问题所在。在程序行设置/移除断点的方法是将光标定位于需要设置断点的程序行,选择 Debug→Insert/Remove Breakpoint 命令,移除断点(也可以双击该行实现同样的功能);选择 Debug→Enable/Disable Breakpoint 命令,开启或暂停光标所在行的断点功能;选择 Debug→Disable All Breakpoint 命令,暂停所有断点;选择 Debug→Kill All Breakpoint 命令,清除所有的断点设置。这些功能也可以用工具条上的快捷按钮进行设置。

仿真验证程序功能是否达到预期功能,在“LED=~LED;”语句处设置一个断点。单击全速运行,程序在此处停下,观察 P1 寄存器窗口,再次单击全速运行,观察 P1 寄存器窗口。如图 3-15(a)所示,可见 P1.0 首次全速运行,值为 1,下一次全速运行到此,值为 0,如此循环往复,调试仿真界面如图 3-15(b)所示,仿真表明实现预期功能,即让 P1.0 不断地取反,实现了 LED 闪烁的功能。



图 3-15 LED 闪烁 Keil μ Vision4 仿真界面

在 Keil 仿真和调试中要充分利用各种窗口进行调试,Keil 还提供外围接口、性能分析、变量来源分析、代码作用分析等辅助工具,帮助编程者了解程序的性能,查找程序中的隐藏错误。

3.2 Proteus ISIS 使用初步

Proteus 软件是英国 Lab Center Electronics 公司出版的 EDA 工具软件,能够实现原理图设计、电路仿真到 PCB 设计的一站式作业,真正实现了电路仿真软件、PCB 设计软件和虚拟模型仿真软件的三合一。它不仅具有其他 EDA 工具软件的仿真功能,还能仿真单片机及外围器件,是目前比较好的仿真单片机及外围器件的工具,受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。

Proteus 的特点如下:

(1) 完善的电路仿真和单片机协同仿真。具有模拟、数字电路混合仿真,单片机及其外围电路的仿真,拥有多样的激励源和丰富的虚拟仪器。

(2) 支持主流单片机类型多,目前支持的单片机类型有 68000 系列、8051 系列、ARM 系列、AVR 系列、PIC10 系列、PIC12 系列、PIC16 系列、PIC18 系列、PIC24 系列、DSPIC33 系列、MPS430 系列、HC11 系列、Z80 系列以及各种外围芯片。

(3) 提供代码的编译与调试功能,自带 8051、AVR、PIC 的汇编器,支持单片机汇编语言的编辑、编译,同时支持第三方编译软件进行高级语言的编译和调试。

(4) 智能、实用的原理图与 PCB 设计。在 ISIS 环境中完成原理图的设计后可以一键进入 ARES 环境进行 PCB 设计。

本节主要介绍 Proteus ISIS 的工作环境和基本操作。

3.2.1 Proteus ISIS 的工作界面

Proteus ISIS 的工作界面是标准的 Windows 界面(图 3-16),包括:屏幕上方的标题栏、

菜单栏、标准工具栏,屏幕左侧的绘图工具栏、对象选择按钮、预览对象方位控制按钮、仿真进程控制按钮、预览窗口、对象选择器窗口、屏幕下方的状态栏、屏幕中间的图形编辑窗口。下面简单介绍界面上的菜单栏和工具栏的功能。

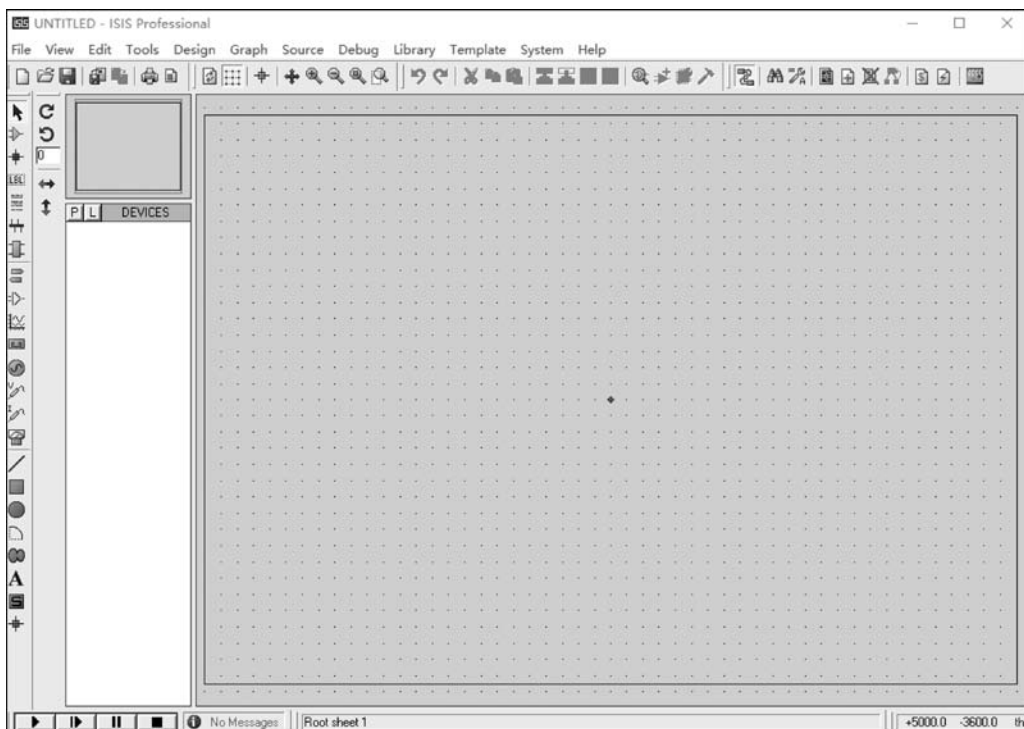


图 3-16 Proteus ISIS 工作界面

1. 菜单栏

Proteus ISIS 共有 12 级菜单,每项都有下一级菜单。

File: 文件菜单,用于新建、保存、导入位图、导出图形、打印等文件操作。

View: 查看菜单,可进行原理图编辑窗口定位、栅格尺寸调整及图形缩放等操作。

Edit: 编辑菜单,可进行原理图编辑窗口中元件剪切、复制、粘贴、撤销、恢复等操作。

Tools: 工具菜单,可进行实时标注、自动布线、材料清单、网络表编辑等操作。

Design: 设计菜单,具有编辑设计属性、编辑面板属性、编辑设计注释、配置电源线、设计浏览等功能。

Graph: 图形菜单,具有编辑仿真图形、增加跟踪曲线、查看日志、导出数据、图形一致性分析等功能。

Source: 源文件菜单,具有添加/删除源文件、设置编译、设置外部文件、编辑器和全部编译等功能。

Debug: 调试菜单,具有调试、运行/停止调试、断点运行等功能。

Library: 库菜单,用于元件库的操作,具有选择元件/符号、制作元件、制作符号、封装工具、分解元件、编译到库、验证封装、库管理等功能。

Template: 模板菜单,具有设置图形颜色、设置文本格式、设置连接点等功能。
System: 系统菜单,具有系统信息、文本预览、设置系统环境、设置路径等功能。
Help: 帮助菜单,为用户提供帮助文档。

2. 主工具栏

主工具栏包括 File Toolbar(文件工具条)、View Toolbar(查看工具条)、Edit Toolbar(编辑工具条)、Design Toolbar(设计工具条)四个工具栏。

3. 工具箱

原理图绘制主要通过工具箱下的工具按钮进行,表 3-2 列出了常用工具按钮的功能。

表 3-2 常用工具按钮功能

按 钮	名 称	说 明
	Selection Mode	选择按钮,可选中任意元件并编辑元件属性
	Components Mode	元件按钮,可拾取元器件并结合“P”按钮选择元件
	Junction Dot Mode	节点按钮,可在原理中放置连接点
	Wire Label Mode	网络标号按钮,可在元件引脚和线段上标注网络标号
	Text Script Mode	文本脚本按钮,可在电路中输入文本脚本
	Buses Mode	总线按钮,在电路中绘制总线
	Subcircuit Mode	子电路按钮,用于绘制子电路
	Terminals Mode	终端按钮,在对象选择器中列出各种终端,其中 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
	Device Pins Mode	元件引脚按钮,在对象选择器中列出各种引脚(如普通引脚、时钟引脚、反电压引脚和短接引脚等)
	Graph Mode	图表按钮,列出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)
	Tape Recorder Mode	录音机按钮,当对设计电路分割仿真时采用此模式
	Generator Mode	信号源按钮,可选择各种激励源(如直流激励源、正弦激励源、脉冲激励源、指数激励源等)
	Voltage Probe Mode	电压探针按钮,在电路仿真时可显示探针处的实时电压值
	Current Probe Mode	电流探针按钮,在电路仿真时可显示探针处的实时电流值
	Virtual Instruments Mode	虚拟仪器按钮,有多种虚拟仪器供用户选择(如示波器、逻辑分析仪、计数器、SPI 总线调试器、I ² C 总线调试器、信号发生器等)

4. 仿真按钮

 为仿真按钮,用于仿真运行控制,依次为运行、单步运行、暂停、停止。

3.2.2 Proteus ISIS 的虚拟仿真调试工具

激励源为电路提供仿真所需要的输入信号,在 Generator Mode 工具按钮下可选择输入信号类型,用户可对激励源的参数进行设置,如表 3-3 所示。

表 3-3 Proteus 中的激励信号源

激励源符号	英文名称	中文名称
	DC	直流信号发生器
	SINE	正弦波信号发生器
	PULSE	脉冲信号发生器
	EXP	指数脉冲发生器
	SFFM	单频率调频波发生器
	PWLIN	任意分段线性激励源
	FILE	文件数据信号发生器
	AUDIO	音频信号发生器
	DSTATE	单稳态逻辑电平发生器
	DEDGE	单边沿信号发生器
	DPULSE	单周期数字脉冲信号发生器
	DCLOCK	数字时钟信号发生器
	DPATTERN	数字模式信号发生器
	SCRIPTABLE	可脚本化波形发生器

Proteus ISIS 为用户提供了多种虚拟仪器,在 Virtual Instruments Mode 按钮下,可选的虚拟仪器如图 3-17 所示。

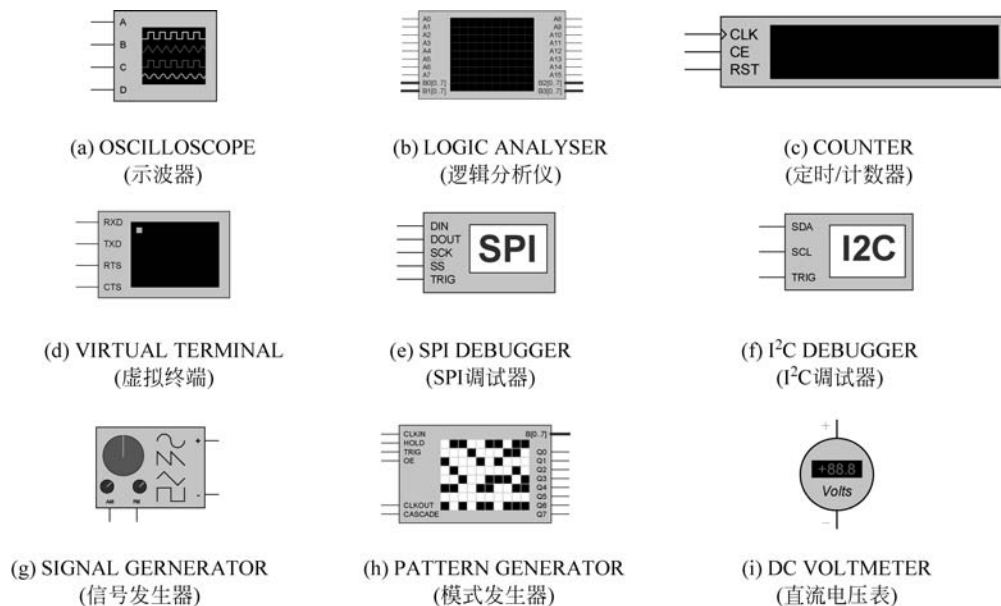


图 3-17 PROTEUS 中的虚拟仪器



图 3-17 (续)

下面简单介绍单片机应用系统调试中常用的几种虚拟仪器,详细的使用和设置可参考相关书籍和软件帮助文件。

1. 虚拟示波器

虚拟示波器界面如图 3-18 所示,将频率 50Hz,幅值 5V 的脉冲信号和正弦信号分别接入虚拟示波器的 A 通道和 B 通道,信号就显示在虚拟示波器面板上。

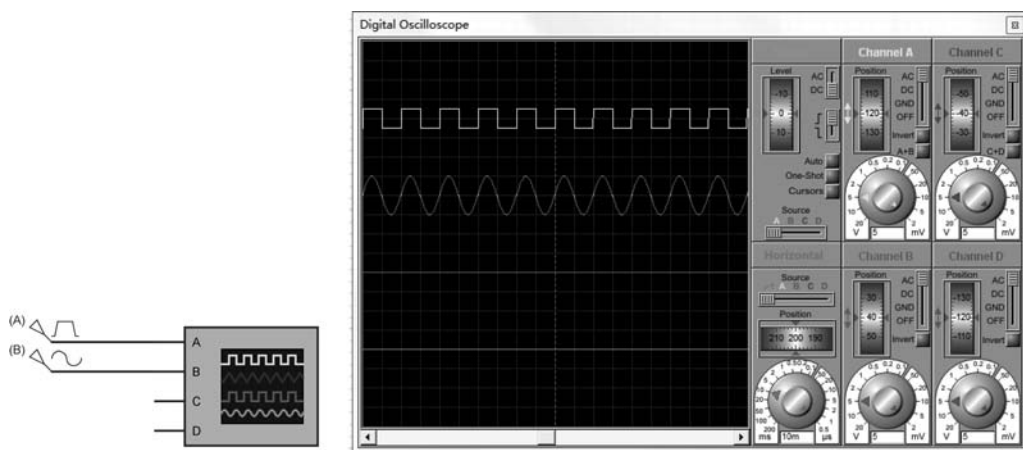


图 3-18 虚拟示波器接法及示波器面板界面

示波器的界面共分为六部分:四个通道区、触发区、水平区。

Channel A、Channel B、Channel C、Channel D 对应于四个通道区,每个通道区的操作功能一样,主要有两部分,Position 用来调整波形的垂直位移,旋钮用来调整波形的 Y 轴增益,白色区域的刻度表示波形显示区每格对应的电压值,内旋钮式微调,外旋钮式粗调。在图形区读电压时,把内旋钮顺时针调到最右端。

Trigger 为触发区,其中 Level 用来调节水平坐标,水平坐标只在调节时才显示,Auto 按钮一般为红色选中状态。Cursors 按钮被选中后,可以在波形显示区标注横坐标和纵坐标,读取波形的电压和周期。

Horizontal 为水平区,Position 用来调整波形的左右位置,旋钮用于调整扫描频率,当波形太密或太疏时,可调整此按钮。注意,当读周期时,需要把内环的微调旋钮顺时针旋转到底。

2. 虚拟逻辑分析仪

逻辑分析仪是通过将连续记录的输入信号存入到大的捕捉缓冲器进行工作的,这是一个采样过程,具有可调的分辨率,用于定义可以记录的最短脉冲。在触发期间,驱动数据捕

捉处理暂停,并监测输入数据,触发前后的数据都可以显示。具有可存放 10 000 个采样数据的捕捉缓冲器,因此支持放大/缩小显示和全局显示。同时,用户还可以移动测量标记,对脉冲宽度进行精确定时。

虚拟逻辑分析仪的原理图符号如图 3-19 所示,其中 A0~A15 为 16 位数字信号输入,B0~B3 为总线输入,每条总线支持 16 位数据,主要用于接单片机的动态输出信号。运行后,可以显示 A0~A15、B0~B3 的数据输入波形。

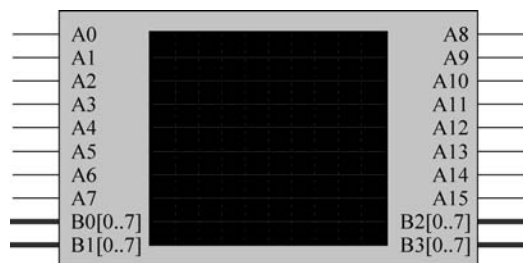


图 3-19 虚拟逻辑分析仪的原理图符号

3. 虚拟终端

Proteus 提供的虚拟终端相当于键盘和屏幕的双重功能,方便用户在串行通信仿真调试时用来替代上位机系统的仿真模型。

虚拟终端的原理图符号和设置对话框如图 3-20 所示。虚拟终端共有四个接线端,RXD 为数据接收端,TXD 为数据发送端,RTS 为请求发送信号,CTS 为清除传送信号,是对 RTS 的响应信号。虚拟终端属性设置对话框设置的主要参数如下:

- (1) Baud Rate: 波特率。
- (2) Data Bits: 传输的数据位数。
- (3) Parity: 奇偶校验位,可选择奇校验、偶校验或无校验位。
- (4) Stop Bit: 停止位。
- (5) Send XON/XOFF: 发送允许/禁止(第 9 位)。

4. SPI 调试器

SPI 是一种常用的单片机与外围器件总线通信协议。Proteus 仿真可通过 SPI 调试器查看 SPI 总线发送和接收的数据,方便在 SPI 通信接口编程的调试。SPI 调试器的原理图符号和设置对话框如图 3-21 所示。SPI 调试器共有五个接线端,DIN 为接收数据端,DOUT 为输出数据端,SCK 为连线总线时钟端,SS 为从模式选择端,TRIG 为输入端,能够把下一个存储序列放到 SPI 的输出序列中。

双击 SPI 原理图符号,打开属性设置对话框,设置的主要参数如下:

(1) SPI Mode: SPI 模式选择位,有三种工作模式可选择,Monitor 为监控模式,Master 为主模式,Slave 为从模式。

(2) Master clock frequency in Hz: 主模式的时钟频率(Hz)。

(3) SCK Idle state is: SCK 空闲状态为高或低,选择一个。

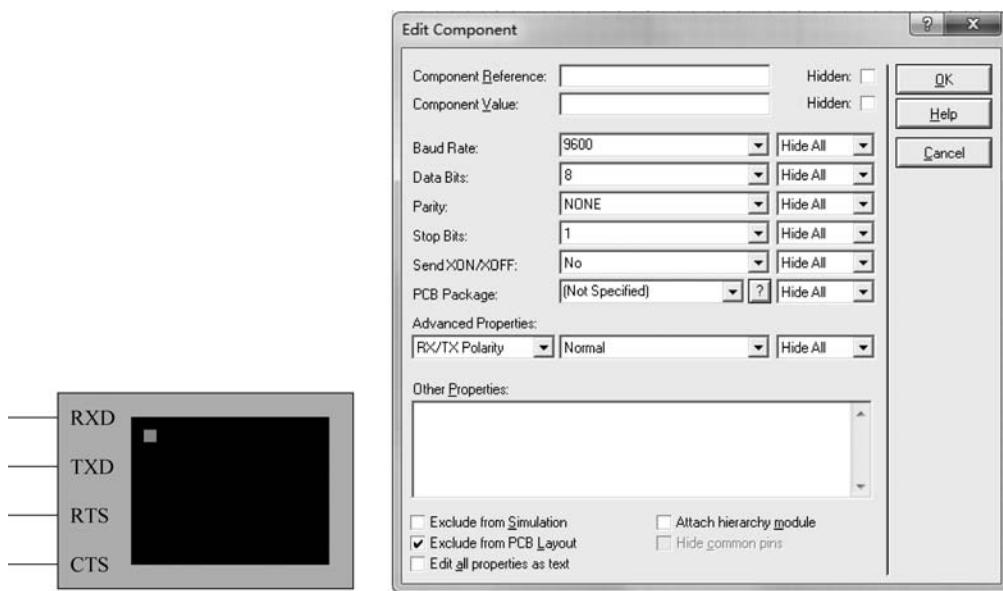


图 3-20 虚拟终端的原理图符号和设置对话框

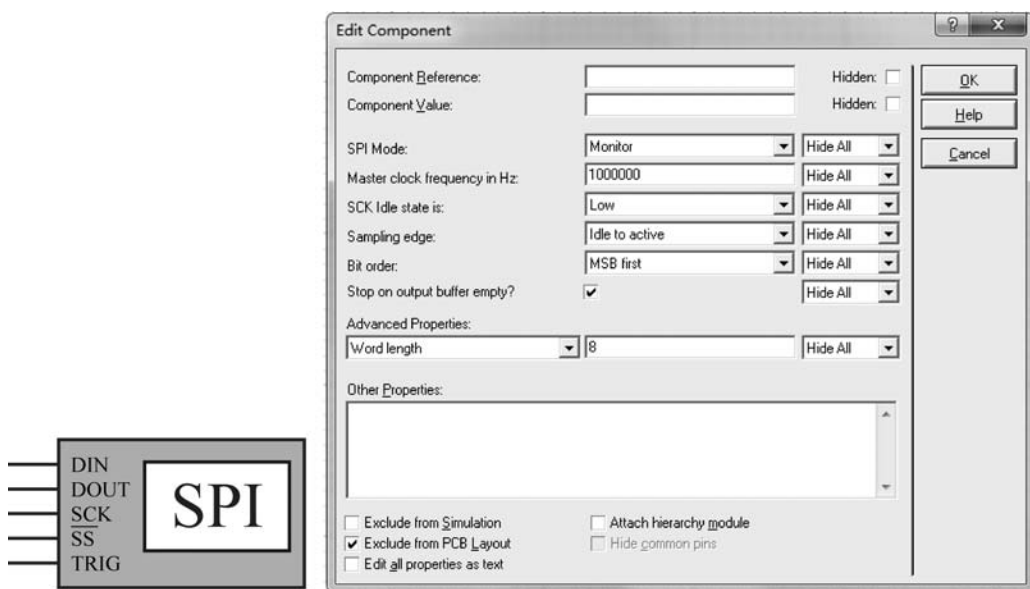


图 3-21 SPI 调试器的原理图符号和设置对话框

(4) Sampling edge: 采样边沿, 指定 DIN 引脚采样的边沿, 选择当 SCK 从空闲到激活状态, 或从激活到空闲状态时。

(5) Bit order: 选择 MSB(高位在前)或 LSB(低位在前)。

5. I²C 调试器

I²C 是飞利浦公司开发的一种简单、双向二线制同步串行总线, 只需要两根线即可在连接于总线上的器件之间进行通信, I²C 是一种多向控制总线, 也就是说多个芯片可以连接到

同一总线结构下,同时每个芯片都可以作为实时数据传输的控制源。这种方式简化了信号传输总线接口,很多外围器件采用 I²C 接口。I²C 调试器的原理图符号和设置对话框如图 3-22 所示。I²C 调试器共有三个接线端,SDA 为双向数据线,SCL 为双向输入端,时钟连接,TRIG 为触发输入,能引起存储序列被连续地放置到输出队列中。

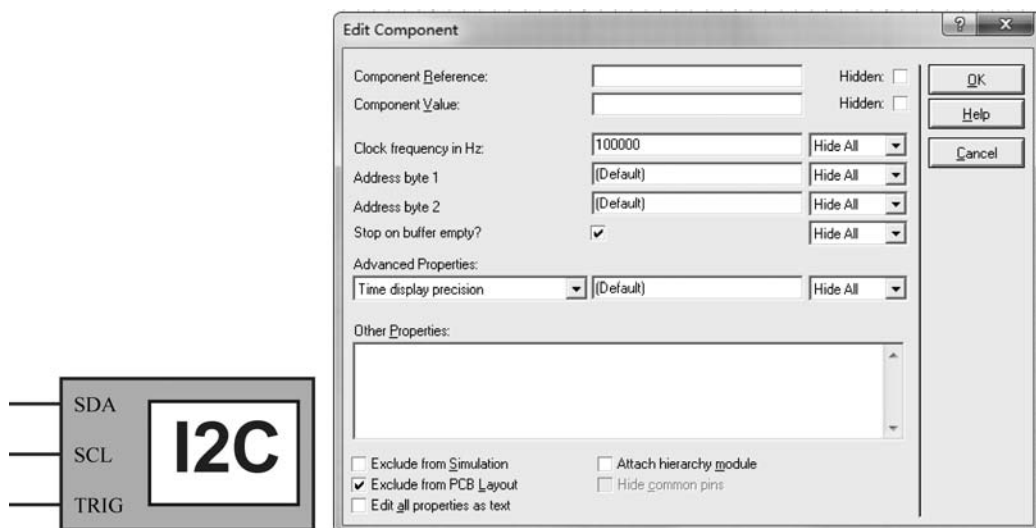


图 3-22 I²C 调试器的原理图符号和设置对话框

双击 I²C 原理图符号,打开属性设置对话框,设置的主要参数如下:

(1) Address byte 1: 地址字节 1,如果使用此终端仿真一个从元件,这一属性指定从器件的第一个地址字节。

(2) Address byte 2: 地址字节 2,如果使用此终端仿真一个从元件并期望使用 10 位地址,这一属性指定从器件的第二个地址字节。

3.2.3 Proteus ISIS 原理图绘制步骤

Proteus 的菜单和操作较多,本书只介绍常用的基本菜单和基本操作。下面以流水灯电路为例,讲解 Proteus 原理图绘制步骤,流水灯程序将在第 4 章进行讲解。本例只进行仿真,不进行原理图纸设置等操作。选择 File→Save Design As 命令,将文件命名为“LED”并保存在程序的文件夹下。

1. 选择元器件

先将需要用到的元器件加载到对象选择器窗口。在 ISIS 工作界面上单击 P 按钮,弹出 Pick Device 界面,选择 Category→Microprocessor ICs 命令,再单击 8051 Family。找到 AT89C51,双击 AT89C51,这样在左侧的对象选择器就有了 AT89C51 元件。如果知道元件的名称或者型号,可以在 Keywords 输入元件名称,系统在对象库中进行搜索查找,并将搜索结果显示在 Results 中。接着开始放置晶振,在 Keywords 中输入 CRY,在“Results”的列表中,双击 CRYSTAL 将晶振加载到对象选择器窗口内。继续加入其他元器件,流水灯电路所需的元器件见表 3-4。为方便绘制仿真电路图时查找元器件,附录 A 给出了 Proteus

提供的仿真元件分类及子类中英文对照。

表 3-4 流水灯电路所需元器件

名 称	元件名	数 量/个	所在子类
单片机	AT89C51	1	Microprocessor ICs/8051 Family
晶振	CRYSTAL	1	Miscellaneous
独石电容	CAP	2	Capacitors/Generic
电解电容	CAP-ELEC	1	Capacitors/Generic
电阻	RES	10	Resistors/Generic
按键	BUTTON	1	Switches & Relays
发光二极管	LED-RED	8	Optoelectronics

2. 放置元器件并修改参数

在对象选择器窗口内选中 AT89C51,如果元器件的方向不方便绘图,可使用预览对象方向控制按钮进行操作。用按钮对元器件进行顺时针旋转,用按钮对元器件进行逆时针旋转,用按钮对元器件进行左右反转,用按钮对元器件进行上下反转。元器件方向方便绘图后,将光标置于图形编辑窗口元器件需要放置的位置,单击并一直按住鼠标左键,出现紫色的元器件轮廓符号,拖动鼠标可对元器件的放置位置进行调整,拖动到适合的位置后释放鼠标左键,元器件被放置。放置元器件后,如还需调整方向,单击需要调整的元器件,再右击进行调整。同样的操作将晶振、电容、电阻、发光二极管放置到图形编辑窗口。

将元器件编号并修改参数。修改的方法是在图形编辑窗口中双击元器件,在弹出的 Edit Component 对话框中进行修改。以电阻为例进行说明,如图 3-23 所示。

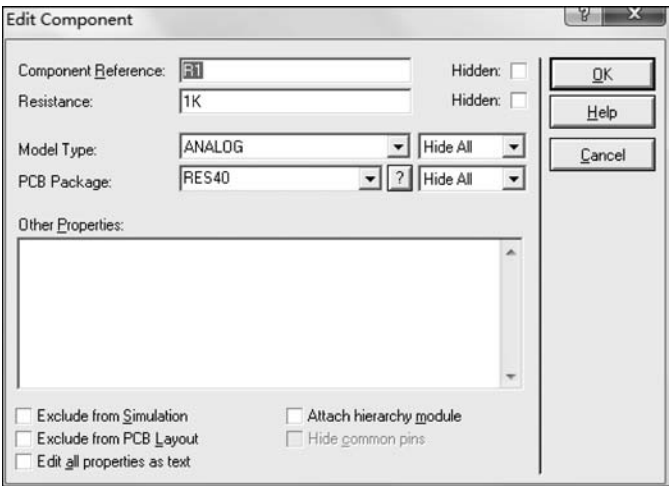


图 3-23 电阻参数修改

把 Component Reference 中的 R? 改为 R1,把 Resistance 中的 10k 改为 1k。修改好后单击 OK 按钮,这时编辑窗口就有了一个编号为 R1、阻值为 1k Ω 的电阻。重复以上步骤就可对其他元件的参数进行设置。

3. 元器件的电气连接

Proteus 具有自动线路功能(wire auto router),当光标移动至连接点时,光标指针处出现一个虚线框,单击并移动光标至下一个连接点,出现虚线框时,单击完成连线。在此过程中可以按下 Esc 键或者右击放弃连线。

4. 放置电源端子。

单击绘图工具栏的  按钮,使之处于选中状态。选择 POWER 放置两个电源端子,选择 GROUND 放置一个接地端子,放置好后完成接电和接地的连线。

5. 绘制总线并设置网络标号

如果连线较多,则可以使用总线简化电路图的连线。在复杂的电路图中使用总线,可以清晰快速地理解多连线元件间的接线关系。单击绘图工具栏的  按钮,使之处于选中状态。将光标置于图形编辑窗口,单击,确定总线的起始位置,移动光标,屏幕出现一条蓝色的粗线,选择总线的终点位置,双击,这样就绘制好一条总线。

绘制与总线连接导线时,为了和一般的导线区分,通常习惯画斜线来表示分支线。此时需要自己决定走线路径,只需在想要拐点处单击即可。在绘制斜线时需要关闭自动线路功能。可通过使用工具栏里的 WAR 命令按钮  关闭。

绘制完总线和分支线,但是还不能表示对应元件引脚之间的电气连接,需使用网络标号(net label)。网络标号是一种具有电气连接属性的标号,一般由字母或数字组成,具有相同网络标号的电气连接线、引脚及网络是连接在一起的。单击绘图工具栏的网络标号按钮  使之处于选中状态。将光标置于欲放置网络标号的导线上,这时会出现一个“×”,表明该导线可以放置网络标号。单击,弹出 Edit Wire Label 对话框,在 String 输入网络标号名称(如 D1),单击 OK 按钮,完成该导线的网络标号的放置。接着放置其他导线的标号。注意:在放置导线网络标号的过程中,相互接通的导线必须标注相同的标号,相同标号采用的大小写符号也需要完全一致。

按照以上步骤绘制电路图后,8 路流水灯电路就绘制完毕,完成的电路图如图 3-24 所示,本电路图的单片机没有接电和接地的引脚,这是因为 Proteus 软件为了用户仿真电路更加方便快捷,默认接好了电源和地。如需要显示电源和地的引脚,可在菜单 Template 下 Set Design Defaults 的界面左下角勾选 Show hidden pins,单片机元件上就出现电源和地的引脚。为使电路图更加简洁明了,采用系统默认设置,不要勾选此项。在 Proteus 绘制单片机仿真电路原理图时,单片机已经默认可以正常工作,单片机最小系统所需的外围器件和电路均可省略,不影响仿真运行,本书为保证电路原理图的完整性,画出了最小系统。

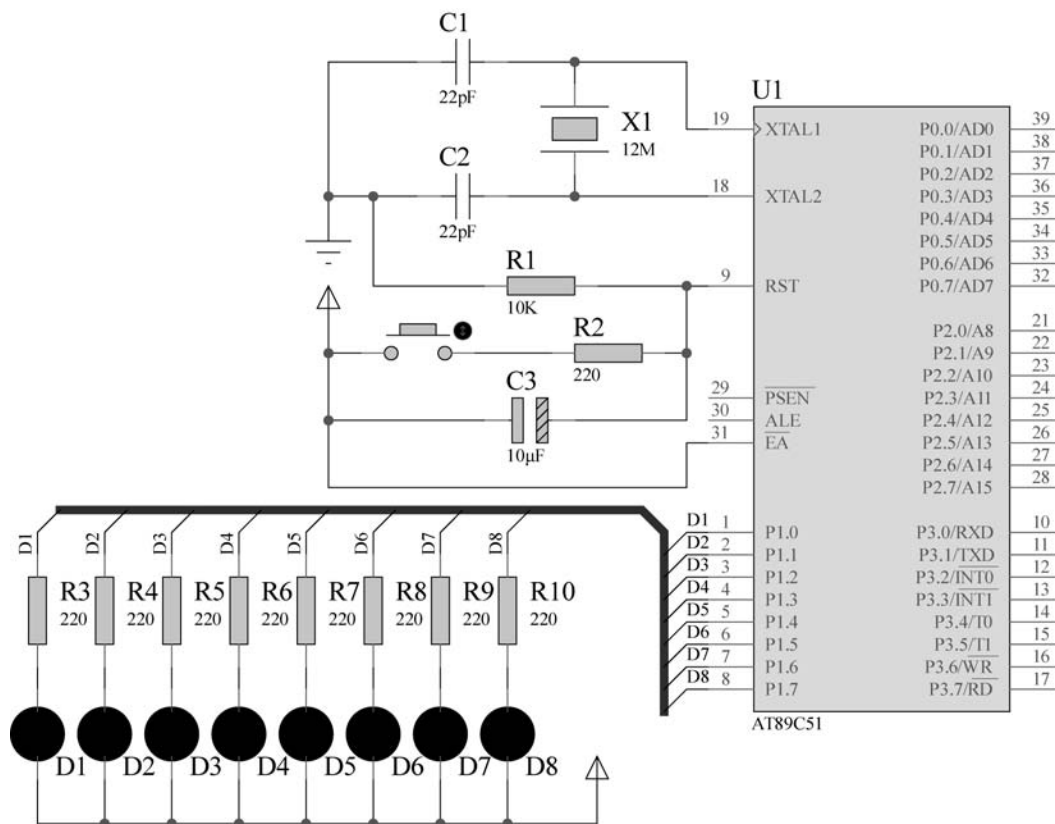


图 3-24 8 路流水灯电路图

3.3 第一个仿真电路和程序

有了上述操作基础之后,来做第一个仿真电路和程序。下面以 Keil 软件调试过程中的 LED 闪烁为例来做一个完整的仿真的电路和程序。

例 3-1 编制程序,以及单片机控制 LED 点亮,以及单片机控制一个 LED 闪烁。

如图 3-25 所示,LED 驱动电路是将发光二极管阴极接到单片机 I/O 口上,发光二极管阳极通过一个限流电阻接到 V_{CC} 上。这样只需要给 P1.0 一个低电平,发光二极管导通就可以点亮 LED,这种点亮 LED 的方式称为灌电流方式,即从外部向 I/O 口流入电流。接限流电阻是因为 LED 的正常工作电流为 10~20 mA。如果不接限流电阻,工作电流过大,将会烧坏 LED。LED 压降一般为 1.8V,限流电阻两端压降则为 $5-1.8=3.2(V)$ 。取工作电流 $15mA=0.015A$,则限流电阻阻值为 $3.2/0.015=213(\Omega)$,这里选用 220Ω 的限流电阻。

还有一种接法是将 LED 的阳极接单片机的 I/O 口,阴极接地,单片机 I/O 口输出高电平,LED 点亮。由于 51 单片机驱动电流不足,无法驱动 LED,此时需接上拉电阻从外部补充电流。51 单片机 I/O 口默认为高电平,这样单片机复位后 LED 就会点亮,这不一定是我们想要的结果,如果初始状态为 LED 熄灭,还需要通过程序让其熄灭。不推荐这样驱动 LED,当然这也要根据具体设计情况而定。

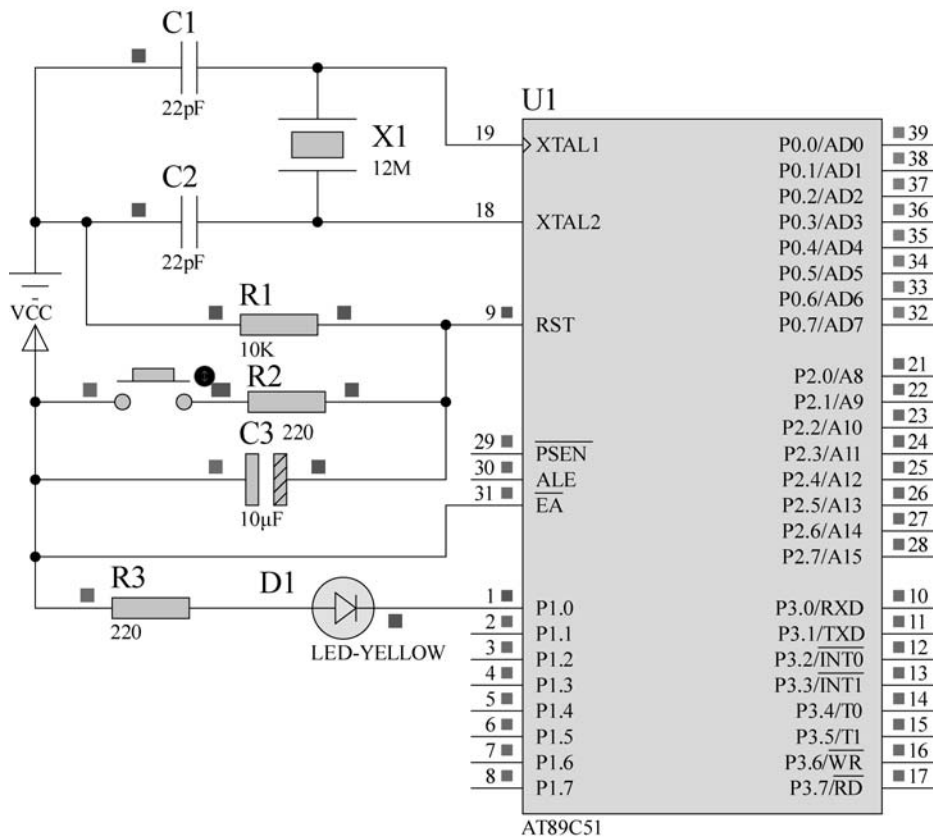


图 3-25 控制一个 LED 亮灭电路图

根据以上分析,LED 由 P1.0 口控制,要点亮 LED,只需要在程序中让 P1.0 口为 0 就可以。程序和程序说明如下:

```
#include <reg51.h>①
sbit D0 = P1^0; ②
void main()
{
    D0 = 0; ③ //点亮 LED
    while(1) //循环,挂起 ⑥
    {
        ; ④
    }
}
```

程序说明:

- ① 引用含有 SFR 符号定义的头文件。
- ② 定义符号 D0 为 P1 口的第 0 位, D0 可以换为其他自定义符号,作用相同,如 sbit LED = P2^2;。
- ③ 点亮发光二极管的功能语句。
- ④ while(1)无限循环,用于程序的反复执行或者程序挂起。
- ⑤ 使用 Tab 键或者空格,使程序结构清晰,便于阅读。
- ⑥ 适当加入注释,使编程思路清晰,便于阅读和后期维护。

上述程序实现了一个 LED 点亮。让 LED 不断闪烁,需要通过 I/O 口来控制 LED 的亮灭,即是让 P1.0 口输出 1 和输出 0,不断取反,如果仅仅是不断取反,闪烁频率太快,人眼不能分辨,因此在闪烁之间需要有一段延时,应编制一段延时子程序。在 Keil μ Vision4 中建立工程文件,在编程窗口中输入以下代码:

```

#include <reg51.h>
#define UCHAR unsigned char
#define UINT unsigned int
sbit D0 = P1^0;
void delay(UINT x)                //延时子程序
{
    UCHAR t;
    while(x-- ) for(t = 0; t<120; t++);
}
void main( )                      //主程序
{
    while(1)
    {
        D0 = ~D0;                //LED 闪烁
        delay(200);
    }
}

```

程序输入完毕后,需要进行编译,生成可执行代码,供仿真电路运行。在菜单 Project 下单击 Build Target,或者按快捷键 F7,即可进行编译。如果语法错误,会在 Build Output 对话框中有错误提示,按照错误指示进行修改正确后才可编译成功。如果语法正确,编译成功会提示语法无错误并且编译成功。这样就成功输入和编译了第一个程序。

前面绘制了仿真电路图,如果进行仿真,则需要在单片机中加载程序编译生成的可执行代码。在仿真电路图双击 AT89C51 单片机,弹出如图 3-20 所示 Edit Component 对话框。在弹出的对话框里单击 Program File 的“打开文件”按钮,在所建工程文件的路径下找到刚才编译得到的 HEX 文件并打开,然后单击 OK 按钮就可以模拟。单击调试控制按钮 ,进入仿真。仿真过程中能清楚地看到 LED 闪烁。在仿真电路图上也可以清楚看到每个引脚的电平状态,红色代表高电平,蓝色代表低电平。注意:单片机仿真电路运行的时钟频率是以图 3-26 中 Clock Frequency 栏中设置的时钟频率为准。

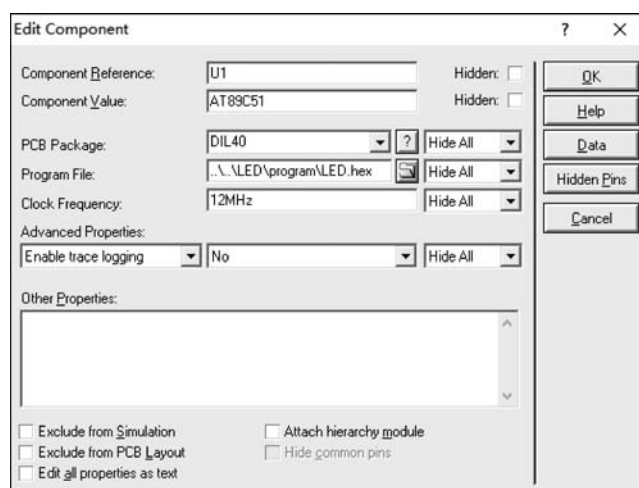


图 3-26 单片机加载可执行文件

3.4 Proteus 原理图与 Keil 环境联机仿真调试

Proteus 支持与 Keil 集成开发环境的联机仿真调试,需要安装 Keil 与 Proteus 的链接文件 vdmagdi.exe,然后进行以下设置:

1. Keil 软件联机调试设置

选择 Project→Options for Target Target1,或者直接单击工具栏上  图标。弹出窗口,单击 Debug 按钮,在出现的对话框右上栏的下拉菜单里选择 Proteus VSM Simulator,如图 3-27 所示。

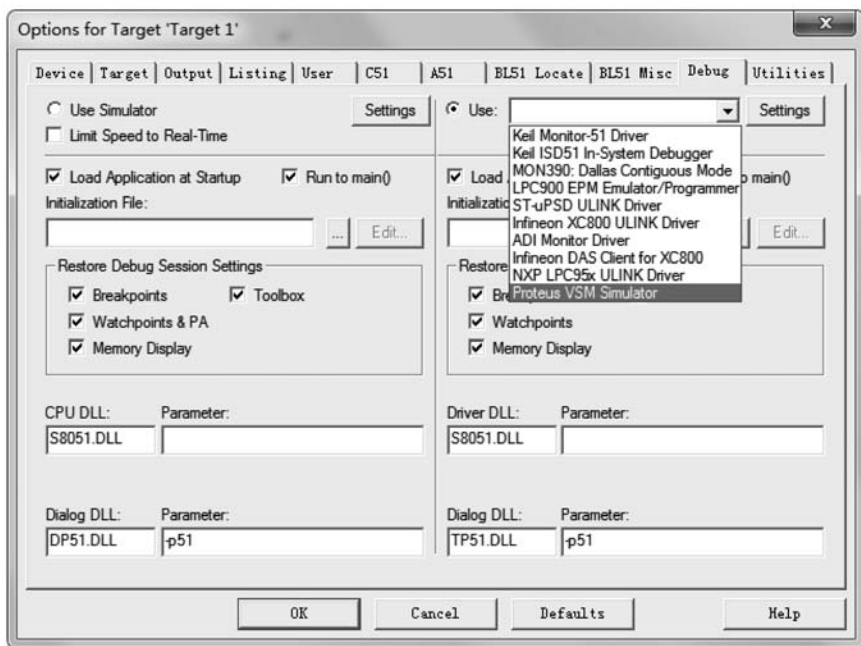


图 3-27 Keil 软件设置与 Proteus 软件联机调试

单击 Settings 按钮还可以设置通信接口,如图 3-28 所示,这里采用默认设置,若使用的不是同一台计算机,还需在这里填上另一台计算机的 IP 地址。

单击 OK 按钮设置完毕,将工程编译,进入调试状态。

2. Proteus 软件联机调试设置

在 Proteus ISIS 中,选择 Debug→use remote debugger monitor 命令,如图 3-29 所示,即可实现 Keil 与 Proteus 的联机调试。

进行完以上设置后便可以进行 Proteus 与 Keil 的联机调试,两个软件进行联机调试,Keil 的软件需要是已编译成功的,Proteus 要打开对应的电路图并已经加载 Keil 软件编译后生成的 .hex 文件进入仿真运行模式。在 Keil 软件单击  或选择 Debug→Start/Stop Debug Session 命令,启动调试器,通过 Debug 工具栏按钮控制程序的运行,便可在 Proteus

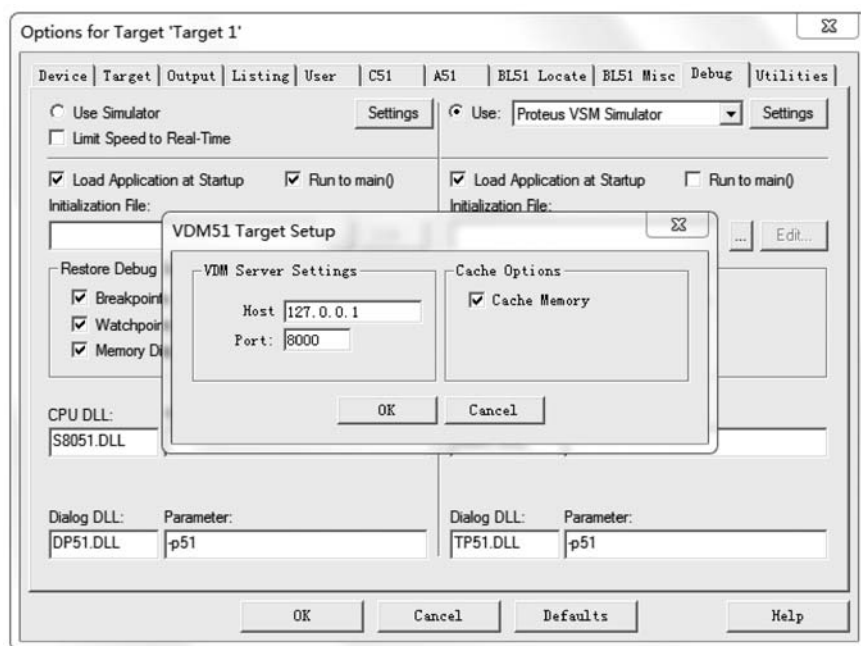


图 3-28 Keil 软件设置网络节点和端口

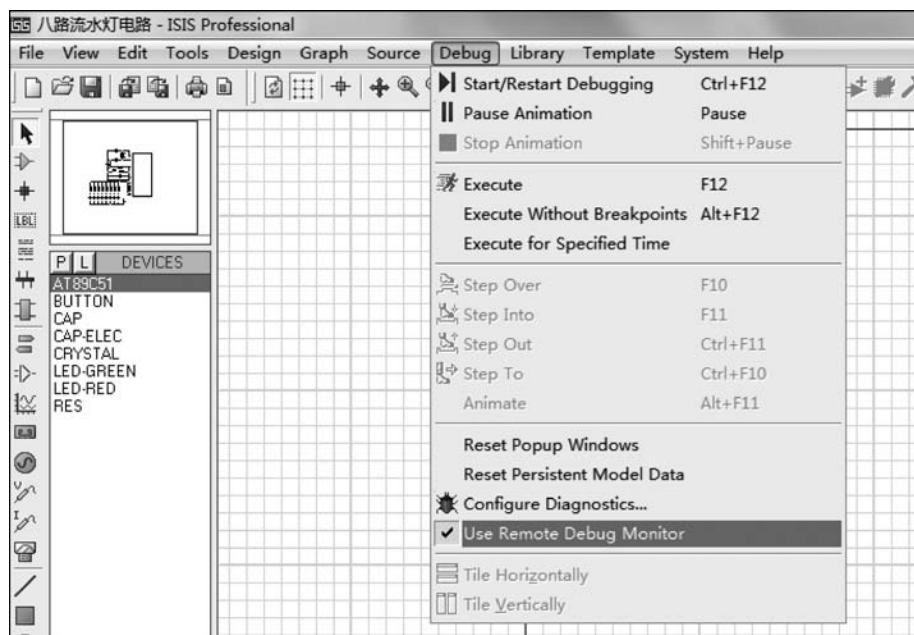


图 3-29 Proteus 软件设置与 Keil 软件联机调试

仿真电路上看到执行相应程序语句所产生的电路运行效果。如图 3-30 所示,对 8 路流水灯电路和程序进行联机调试,在 Keil 调试器中运行单步运行,便可看到 LED 流水灯流动点亮的每一步运行的效果。

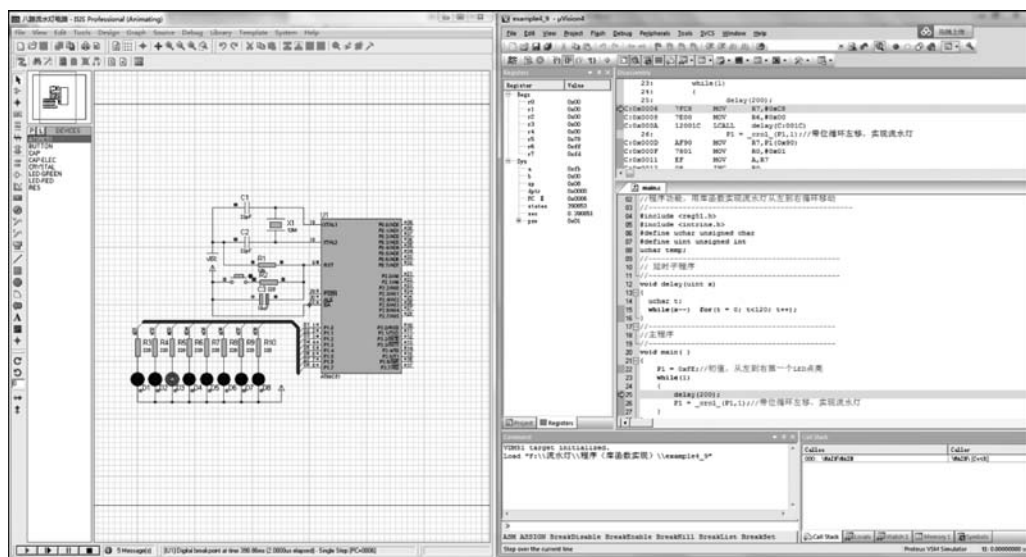


图 3-30 Proteus 软件与 Keil 软件联机调试工作界面

3.5 单片机程序下载方式简介

对于单片机工程开发项目,程序调试编译后要将 .hex 文件下载到单片机中,程序才能够执行,在电路板中实现相应的功能。单片机程序的下载方式有多种,下面介绍编程器下载、ISP(串口下载)、SW、JTAG 常用的下载方式。STC51、LPC11C14、STM32F103C8T6 单片机都可以用 ISP 方式下载,LPC11C14 可以用 SW 方式下载,STM32F103C8T6 可以用 JTAG 方式下载。

3.5.1 编程器下载程序

编程器也叫烧写器或下载器,早期的 OTP 芯片是烧断芯片内的熔丝,故得名烧写器。编程器下载就是将程序数据通过编程器下载进单片机或芯片的 ROM 中。英特尔公司的 8051 是厂家掩膜批量生产的,8751 是在单芯片上集成了当时昂贵的 EPROM,也是需要编程器下载,当时一般是并行编程器,例如 8 位单片机一般先送出 8 位地址再写入 8 位数据,按照一定的时序将程序“烧录”到芯片上。简单的编程器仅用 PC 的并口通过控制引脚的时序即可实现,复杂的编程器使用专门的主控芯片和大量外围电路。有的编程器操作烦琐和价格昂贵,一般需几百元甚至上千元,专业量产的脱机编程器价格高达上万元。常规编程器对纯储存类芯片(非单片机类)如 EPROM/EEPROM/Flash 芯片也可以下载,目前编程器支持高速下载、有检验、全驱引脚驱动、智能升级等。

VS4000P 是一款经济型通用编程器,支持常用的 MCU、EPROM、EEPROM、NorFlash 以及少量的 GAL 器件的烧录,支持器件总数量达 2 万余种。VS4000P 编程器编程软件启动界面如图 3-31 所示。

对 51 单片机芯片烧录按照以下五个步骤。



图 3-31 VS4000P 编程器编程软件启动界面

1. 选择器件

单击图 3-31 界面的“选择器件”按钮，弹出如图 3-32 所示选择器件对话框。有两种途径选择：一是从器件类型选择制造商，然后选择对应型号；二是输入器件型号关键字模糊搜索。选择器件的信息是被保留的，当用户选择好某个型号器件之后，下次打开软件时，自动默认为之前的选择状态。这里选择 ATMEL 公司的 AT89C51 单片机。

2. 加载文件

单击主界面的“打开文件”按钮，弹出 Load File 对话框，如图 3-33 所示。单击右上角的“浏览”按钮，找到需要加载的数据文件。然后单击“确定”按钮，即可加载数据文件。加载文件不会保存，下次打开软件或重新选择器件后需要重新加载。这里选择例程编译好的 LED.hex 文件。

单击主界面左上角的“设置加密、配置参数”，设置加密方案或配置参数，在弹出的窗口进行设置，如图 3-34 所示。这一步是否需要取决于所选择的器件。通常单片机都需要这类器件，单纯存储器则不一定需要。这里选择默认的加密模式。

3. 设置“一键写入”所包含的操作项目

选择不同的器件，会有不同的默认的包含项目。一般不用改变，如果想改变，有以下两种方法：



图 3-32 选择器件



图 3-33 载入程序

(1) 勾选右上角的相应选项。

(2) 选择“高级用户→设置一键完成项目”命令，打开窗口后，右边栏是默认的项目排列，而左边是可供选择的项目。

如果能确认器件是空白的，例如新的器件，则去除擦除操作和查空操作，有利于节约烧写时间，本设置将被自动保存直到重新选择器件。

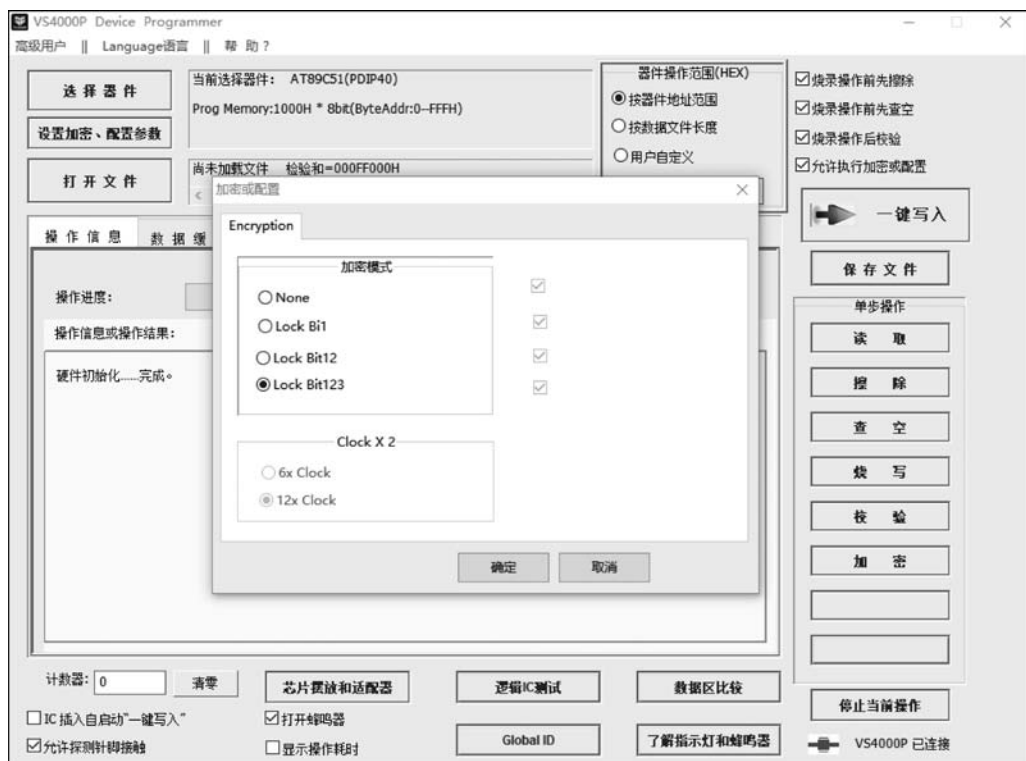


图 3-34 加密模式设置

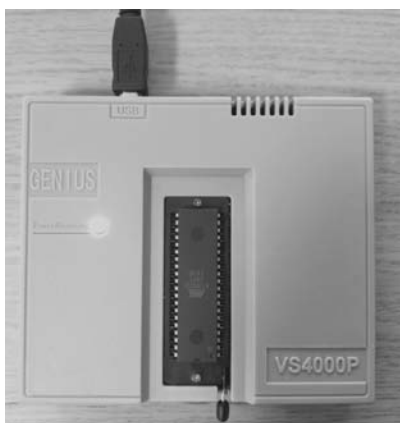


图 3-35 VS4000P 插入单片机芯片烧写程序

单片机烧写程序如图 3-35 所示。

如果需烧录数量较多,可以启用“IC 插入自动启动‘一键写入’”功能。注意:对于 OTP 器件,最好不要启用这个功能。勾选或去除勾选主窗口底部的“IC 插入自动启动‘一键写入’”多选框,可以启用/禁用此功能。

以上步骤,所选择的器件,以及加密与配置参数,当软件关闭时被记忆。下次再打开软件时,只要加载文件就可以烧写。图 3-36 为烧写成功后的软件界面。

采用编程器下载均需要将芯片取出放在编程器的 IC 插座上,下载不够方便,也需要编

4. 烧录芯片

以上三步都是烧录之前的准备工作,下面开始烧录芯片。烧录过程只有放置(更换)芯片,单击“一键写入”按钮这两个动作,极为简单。多个烧录步骤在单击“一键完成”按钮之后,按步骤顺序自动完成。只要注意观察烧录完成之后的结果显示即可。

操作要点:放置器件要按照插放图要求。需要用适配器的,要保证适配器的转换关系正确。如不清楚,单击软件界面底部的“芯片摆放和适配器”可查看插放方式和适配器。VS4000P 插入

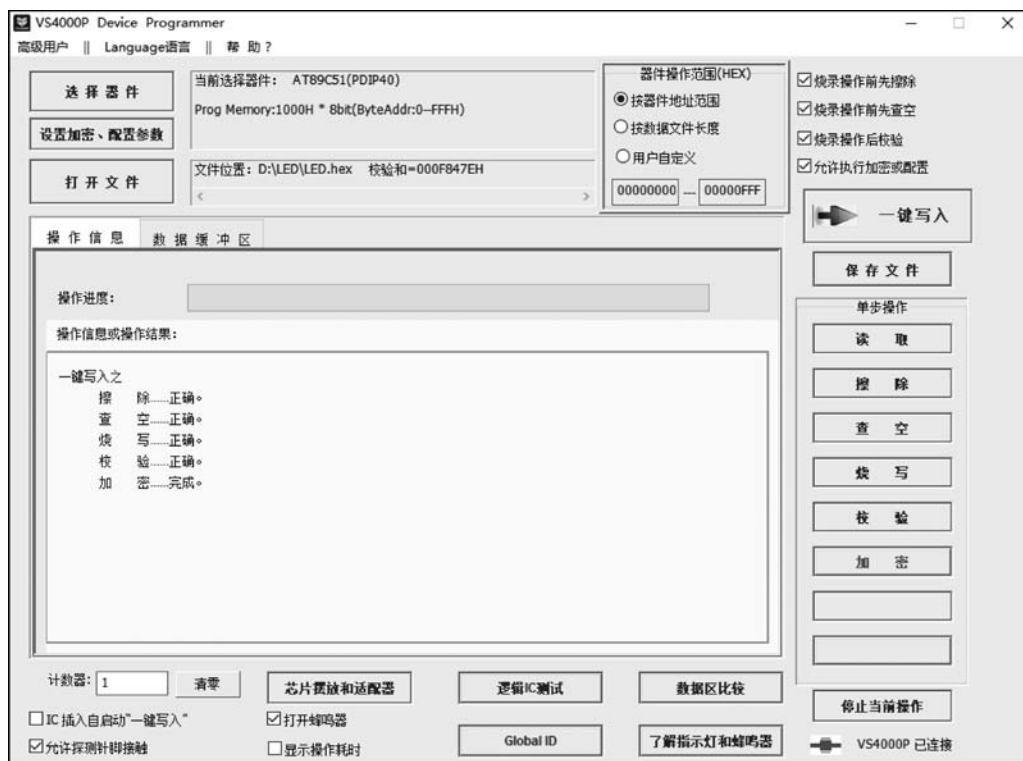


图 3-36 烧写成功后的软件界面

程器硬件投入。

3.5.2 ISP 下载

用编程器烧写程序,每修改一次源程序就要将单片机芯片从目标板上取出,再将更新后的目标代码重新固化到单片机芯片中,烧写会频繁地插拔单片机芯片可能对芯片引脚和电路板造成损坏。ISP 是一种无须将单片机从电路板上取出就能对其进行编程的方法。在线系统编程需要在目标板上有额外的电路完成编程任务。ISP 的实现比较简单,芯片内部的程序存储器可以由上位机的软件通过 SPI 来进行改写,对于单片机来说可以通过 SPI 或其他串行接口接收上位机传来的数据并写入程序存储器中。ISP 技术的优势是不需要编程器也可以进行单片机的实验和开发,既节省了单片机开发的成本,又免去了调试时频繁插拔芯片的麻烦。

ATMEL 公司的 AT89S 系列单片机(包括 AT89S51、AT89LS51、AT89S52、AT89LS52、AT89S53、AT89LS53、AT89S8252、AT89LS8252、AT89S8253、AT89158253 等型号)支持 ISP 下载功能。AT89S 系列单片机内部都是在标准 80C51 的基础上额外设计了一个实现“串行编程接口逻辑”的硬件功能,即 RST 引脚处在高电平的情况下,利用 P1.5/MOSI(串行数据输入端)、P1.6/MISO(串行数据输出端)、P1.7/SCK(同步时钟信号输入端)三个引脚的数据设置或传送实现程序下载的功能。

3.5.3 STC 单片机串口下载

STC 单片机支持串口下载,目标电路板连接前需安装 USB 转串口的驱动程序,本实验电路板采用的是 CH340 芯片的串口转 USB 接口电路,因此需要安装 CH340 驱动程序。安装好驱动程序后,设备管理器的端口能找到电路板设备,如图 3-37 所示,端口下的 USB-SERIAL CH340 即为安装好驱动的电板。将控制板接通电源,用 USB 线将控制板与计算机相连。STC 单片机实验板如图 3-38 所示。

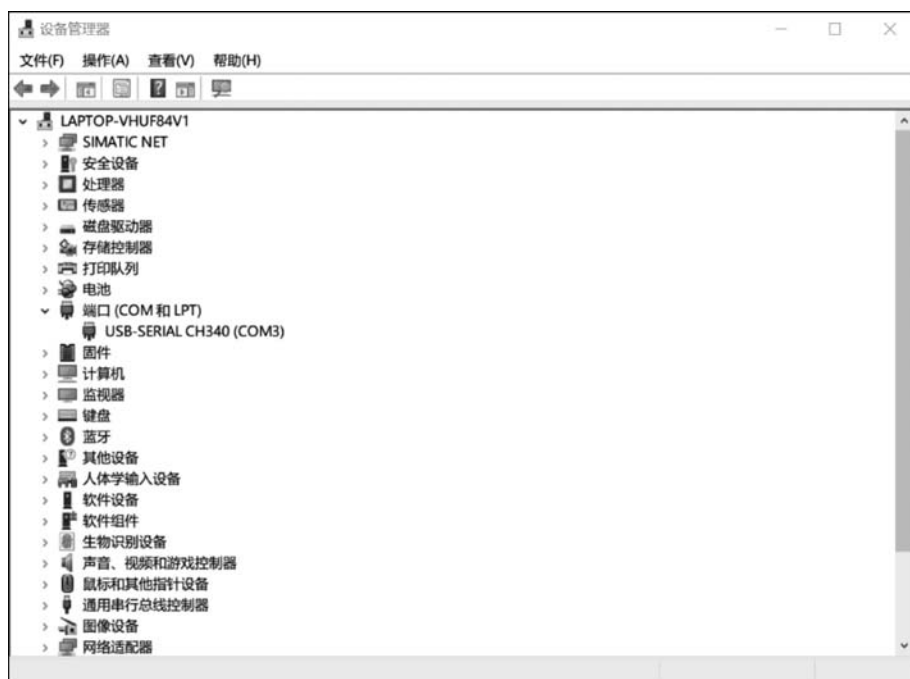


图 3-37 设备管理器中查看电路板设备

打开 STC 单片机下载软件,下载软件界面如图 3-39 所示,界面操作非常人性化,所需步骤和设置均设置在主界面上。

下面按照软件界面的步骤进行设置和烧写程序:

单击 MCU Type 下拉菜单,选择待烧写的芯片型号,这里选择目标电路板上单片机型号 STC89C52;单击打开程序文件,选择要下载的 .hex 文件,这里载入之前例程中的 LED.hex;选择串行口和最高波特率,串行口 com 号到设备管理器去查看。最高和最低波特率采用默认设置。设置选项也采用默认设置,如需其他设置,可自行更改。

以上设置完成后开始下载程序,单击 Download 按钮,断开电路板总电源,使单片机彻底失电,再接通控制板总电源,使芯片重新上电,软件开始下载程序,并提示下载完成。STC 串口 ISP 下载的方式下载程序方便,但也有一定的弱点:①串口下载速度相对高速编程器来说有点慢;②STC 单片机 ISP 下载对串口电路的信号质量要求较高,若串口电路处理不良(如 USB 转串口、232 电平转换),容易造成下载不成功;③每次下载均需重新启动(当然,可通过软件或硬件自动启动处理)。针对上述问题 STC-ISP 下载软件一直在更新改进,

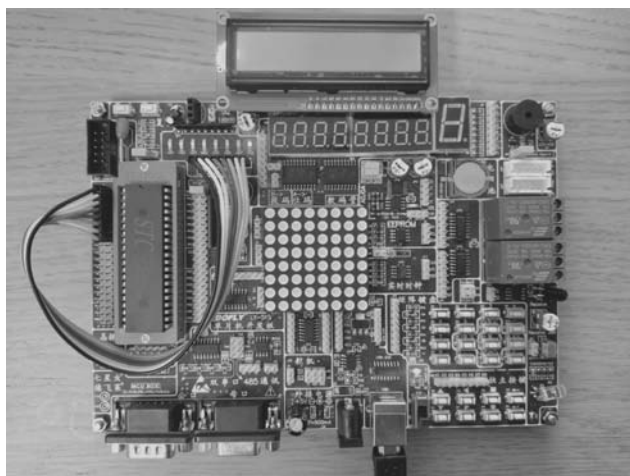


图 3-38 STC 单片机实验板



图 3-39 STC-ISP 软件界面

STC 最新的版本 STC-ISP V6.88D 支持自动识别 COM 接口,下载速度加快,稳定性好,无须重新启动电路板电源,烧写程序更加方便快捷,可到 STC 宏晶科技官网 <https://www.stcmcudata.com/> 下载最新的版本。如图 3-40 为 STC-ISP 最新版本烧写单片机成功界面。

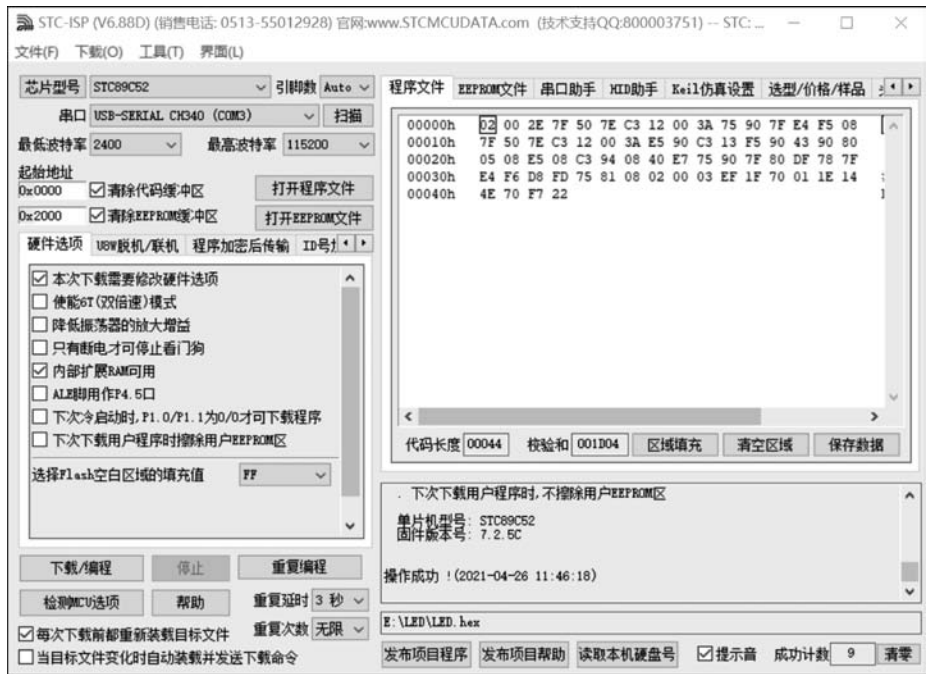


图 3-40 STC-ISP 烧写单片机成功界面

习题

一、填空

- 1. 在 Keil 中,用 C 语言编写的程序扩展名为_____,用汇编语言编写的程序扩展名为_____,经编译器编译后生成的可供单片机运行的可执行文件的扩展名为_____。
- 2. 在 Proteus 中  按钮的功能是_____。

二、单项选择

在用 Keil 调试程序时,观察函数内部指令执行的结果,常采用_____方法。

- A. 单步调试(F8)
- B. 跟踪调试(F7)
- C. 快速运行到光标处调试(F4)
- D. 断点调试(F2)

三、思考题

利用 ISIS 和 Keil 仿真单片机应用系统的主要步骤是什么?

四、绘制电路及编程练习

- 1. 绘制 8 路流水灯电路图。
- 2. 绘制单片机控制一个 LED 闪烁电路图,将程序在 Keil 里输入和编译,仿真实现功能。