

单片机硬件设计与Proteus 仿真同步

高林 徐建 著



四川大学出版社

内容简介

本书以 MCS-51 系列单片机为背景，重点突出“单片机原理”课程的实验教学，可作为在已经掌握 MCS-51 系列单片机的基本原理和硬件结构的基础上进行应用系统开发的实践性教程。本书介绍了 Keil μ Vision4 编程软件的使用方法和 C51 编程语言的基础知识，讲述了 Proteus 7.8 软件的使用及单片机仿真的基础知识，阐述了 Dais 单片机实验系统的电路单元功能及其与 Keil μ Vision4 软件进行连接的设置方法。读者可以通过对 24 个基础实验和 12 个综合性、设计性实验的学习，掌握单片机实验系统硬件实验与 Proteus 软件仿真同步进行的方法，从而熟练掌握单片机系统的软硬件开发过程。另外，书中的综合性、设计性实验还能作为“单片机课程设计”的课题来完成，可以帮助读者积累单片机系统的开发经验。

本书既可以作为“单片机原理”课程的配套实验教材和“单片机课程设计”课程的辅助教材，也可以作为希望掌握单片机系统开发技术的本、专科学生的自学教材，还可以供从事单片机应用开发的工程技术人员自学和参考。

前 言

随着嵌入式系统技术的迅速发展，单片机技术已经广泛地应用到工业控制、医疗仪器、信息家电、通信设备等众多领域。为了满足社会的需要，国内各院校都将“单片机原理”作为工科学生的必修课程，特别是对电子、电气、自控等专业而言更是非常重要的必修课。作为一门应用性课程，必须在学习过程中将软件与硬件、理论与实践结合起来，才能掌握单片机相关技术。因此，“单片机原理”课程的实验教学就占有非常重要的地位，可以毫不夸张地说，不做实验，不亲自动手编程是不可能学好单片机技术的。如果实验室条件允许，甚至可以将“单片机原理”课程全部放到实验室中进行，教师边讲解，学生边实践，能够大大提升该课程的教学效果。

几年以前在进行“单片机原理”课程教学时，采用的还是汇编语言，这种编程语言虽然能够让学生掌握单片机内部结构，但是存在程序可读性差、可移植性差，学生学习费时且不易掌握等问题，在实际应用中受到极大的限制。而后来的教学实践表明，刚开始学习单片机就用 C51 语言编程，既省时，学起来又容易，学生进步速度快，而且程序可读性好、可移植性好，在学习过程中，事实上对于单片机的内部结构也能够逐步掌握。因此，本书第 1 章直接介绍 51 单片机 C51 语言基础知识，并且介绍其常用的编程软件 Keil μ Vision4 的使用方法，让学生在“单片机原理”课程实验时，直接使用 C51 语言编程，使学生能够更加快速的入门，能够在较短时间内掌握单片机的应用。

以前在进行单片机实验教学时，是让学生先在上位机中编程，然后再下载到单片机实验系统中验证程序的结果。由于实验系统中电路已经固定，而且大多数电路在内部已经连接好，学生只需要连接少数几根线即可完成整个实验电路的搭建。因此，在实验过程中学生缺乏对单片机系统电路的理解，往往是“知其然，不知其所以然”，尤其是对单片机扩展存储器或接口芯片的译码电路，学生更是无法深入的理解。学生如果不能理解电路，就不会设计实际的应用系统电路。为了解决这一问题，本书引入 Proteus 软件进行电路仿真，该软件拥有单片机实验系统中几乎所有的元器件和虚拟仪器、仪表，能够让学生在实验系统中完成硬件实验的同时，在 Proteus 软件中对实验电路进行原理图仿真，从而实现与硬件实验完全相同的效果。通过这种教学方式能让学生分析并理解现有实验电路的功能。在此基础上，Proteus 软件还拥有很多实验系统中没有的元器件，能让学生进一步修改电路，甚至设计出各种功能的实际应用电路进行仿真，从而锻炼学生进行电路系统设计的能力。因此，本书第 2 章详细介绍 Proteus ISIS 原理图设计及仿真软件的功能和使用方法，让学生在实验室里能够结合 Proteus 软件与单片机实验系统学习单片机，如果学生不在实验室，只要有 Proteus 软件也能够学习单片机，提高他们的学习兴趣。

本书第 3 章介绍了单片机实验系统的电路单元功能和使用方法。首先讲述了实验室现有 Dais 单片机实验系统所包含的 37 个电路单元模块的功能，其次阐述了该实验系统自带 4×8

键盘的监控操作,即该实验系统可以单独进行程序机器代码的写入和读取过程而不需要上位机。最后对该实验系统如何与PC机上的Keil μ Vision4软件进行连接进行了说明。这些都是学生在进行“单片机原理”实验时需要掌握的基础知识。

根据单片机实验系统的电路结构,本书第4章设计了24个单片机基础实验,可使学生掌握单片机内部P1口、定时器、外部中断和串口等功能,掌握扩展8255、8155、8253、ADC0809、DAC0832、数据存储器和锁存器和缓冲器的方法,学会使用发光二极管、数码管、1602液晶、16×16点阵等输出设备以及电平开关、行列式键盘等输入设备,也能够让学生掌握步进电机、直流电机、扬声器和继电器等外设的使用方法。上述内容既能在实验系统中完成,也可以通过Proteus软件进行仿真,得出相同的结果。让学生既学会单片机编程,也弄清楚单片机电路结构。

由于单片机基础实验所涉及的电路和程序比较简单且功能单一,往往不能单独满足实际应用的需要。因此,本书第5章设计了12个单片机综合性、设计性实验,让学生在解决实际问题的过程中完成更加复杂的单片机电路系统设计和程序的编写、调试过程,从而掌握用单片机解决实际问题的方法。第5章中一部分实验能够在实验系统和Proteus软件中同步完成,而另一部分实验由于现有实验系统缺少相关电路,则只能在Proteus软件中进行仿真。

本书由湖北民族学院高林和湖北民族学院科技学院徐建共同执笔,其中高林负责第4章和第5章,徐建负责第1章至第3章,全书由高林负责审校。在此感谢湖北民族学院信息工程学院提供了很好的单片机实验条件,还要感谢廖红华、黄勇、方芳、谭忠祥、孙先波,他们为本书的撰写提供了单片机方面的技术支持。

由于时间紧迫,书中错误及疏漏之处在所难免,敬请读者批评指正!

高林、徐建

2013年11月

目 录

第 1 章 Keil μ Vision4 软件使用和 C51 基础	(1)
1.1 Keil μ Vision4 软件简介	(1)
1.2 用 Keil μ Vision4 建立一个工程	(1)
1.3 编写 C51 程序并生成 HEX 文件	(7)
1.4 Keil μ Vision4 程序调试	(12)
1.5 C51 基础知识	(16)
第 2 章 Proteus 7.8 概述及单片机仿真基础	(24)
2.1 Proteus ISIS 简介	(24)
2.2 Proteus ISIS 原理图设计及仿真	(44)
第 3 章 单片机实验系统概述	(64)
3.1 硬件系统简介	(64)
3.2 系统键盘/显示使用	(78)
3.3 Dais-MCS51 单片机实验系统软件使用简介	(81)
3.4 Keil μ Vision4 软件与实验系统的连接设置	(83)
第 4 章 基础实验	(89)
4.1 P1 口霓虹灯实验	(89)
4.2 P1 口转弯灯实验	(94)
4.3 P3.3 口输入, P1 口输出实验	(99)
4.4 工业顺序控制实验	(103)
4.5 8255 A、B、C 口输出方波实验	(109)
4.6 8255 PA 口控制 PB 口实验	(116)
4.7 急救车与交通灯实验	(119)
4.8 简单 I/O 口扩展实验	(125)
4.9 数字时钟显示实验	(130)
4.10 脉冲计数实验	(135)
4.11 A/D 转换实验	(141)
4.12 D/A 输出三角波实验	(146)
4.13 电子音响实验	(153)
4.14 继电器控制实验	(160)
4.15 键盘控制时钟实验	(165)

4.16	步进电机控制实验	(174)
4.17	8253 输出方波实验	(182)
4.18	串并转换实验	(187)
4.19	单片机与 PC 机串行通信实验	(192)
4.20	小直流电机调速实验	(202)
4.21	16×16 点阵显示实验	(210)
4.22	8155 并行扩展 3×3 键盘、4 位 LED 显示实验	(219)
4.23	1602 液晶显示接口实验	(226)
4.24	片外数据存储扩展实验	(232)
第 5 章	综合性、设计性实验	(240)
5.1	多功能数字钟实验	(240)
5.2	模拟真实交通灯实验	(243)
5.3	1602 液晶显示电子琴实验	(247)
5.4	单片机与单片机之间双机串行通信实验	(250)
5.5	键盘控制步进电机实验	(254)
5.6	简易波形信号发生器实验	(257)
5.7	数字频率计实验	(261)
5.8	数字音乐盒实验	(267)
5.9	八位竞赛抢答器实验	(272)
5.10	LCD 显示的指针式电子钟实验	(274)
5.11	多机串行通信实验	(276)
5.12	基于热敏电阻的数字温度计实验	(279)
参考文献		(283)

第 1 章 Keil μ Vision4 软件使用和 C51 基础

本章详细介绍了 51 单片机编程软件 Keil μ Vision4 的用法，从 Keil 工程的建立、程序输入、编译、生成 HEX 文件到程序调试这个完整过程进行说明。然后对 C51 语言基础知识进行介绍，包括其基本数据类型、扩充数据类型、常用头文件、运算符以及流程控制语句等内容。使初学者能够掌握 Keil μ Vision4 软件的使用方法，同时也学会用 C51 编写单片机程序。

1.1 Keil μ Vision4 软件简介

不管是用汇编语言还是 C51 语言编程，都要使用编译器，以便将源程序编译为 HEX 程序文件，然后将其存入单片机内。Keil μ Vision4 是众多单片机应用开发软件中最优秀的软件之一，其支持众多不同公司的 MCS-51 架构的单片机，集编辑、编译、仿真等于一体，其界面和常用的 VC++ 软件的界面相似，界面友好、易学易用，在调试程序、软件仿真方面也有很强大的功能。因此很多开发 51 系列单片机应用的工程师或普通的单片机爱好者，都对其十分喜欢。

2009 年 2 月发布的 Keil μ Vision4 比起 μ Vision3 或是 μ Vision2 的界面更加友好一些，新的用户界面可以更好地利用屏幕空间和更有效地组织多个窗口，提供一个整洁、高效的环境来开发应用程序。新版本还添加了一些其他新功能，但是不管其如何更新换代，如果已经学会使用早期版本，学起 Keil μ Vision4 来会更加容易。

目前使用 Keil μ Vision4 的产品有 Keil MDK-ARM、Keil C51、Keil C166 和 Keil C251，本书主要讲解其中专门针对 51 单片机的 Keil C51。

1.2 用 Keil μ Vision4 建立一个工程

与 VC++ 编程软件类似，51 单片机的 C51 语言程序以工程 (Project) 的形式打包存放，便于统一管理。因此最好先建立一个空文件夹，把工程文件全部放到里面，以避免和其他文件弄混。

首先在“我的文档”中创建一个名为“例 1.1”的文件夹，如图 1.2.1 所示。

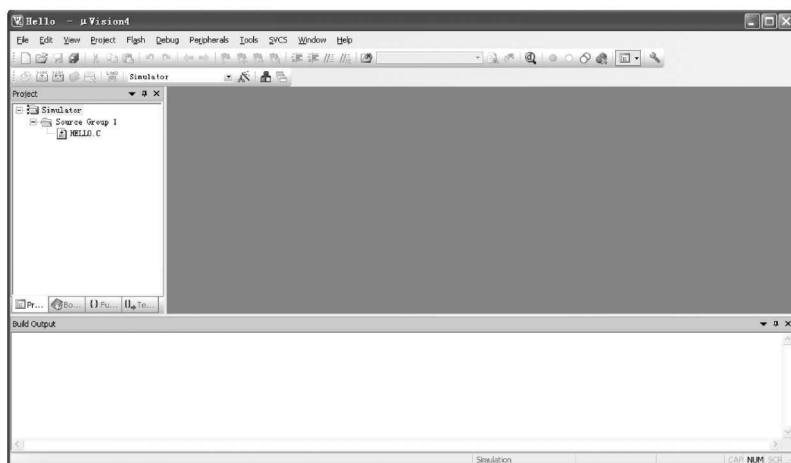


图 1.2.1 在“我的文档”中创建一个名为“例1_1”的文件夹

其次双击 Windows 桌面上的 Keil μ Vision4 图标，出现启动画面，如图 1.2.2 所示。紧接着进入软件编辑界面，如图 1.2.3 所示。接下来开始建立一个工程。



图 1.2.2 Keil μ Vision4 启动画面

图 1.2.3 Keil μ Vision4 编辑界面

(1) 新建一个工程。顺序单击菜单“Project→New μ Vision Project”选项，如图 1.2.4 所示。

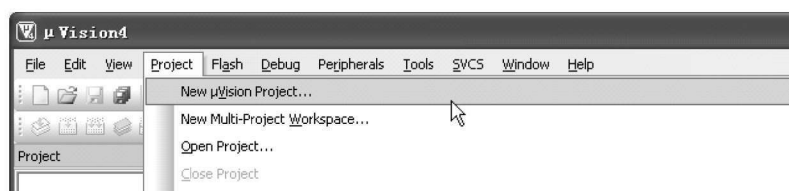


图 1.2.4 新建工程

(2) 选择工程保存的路径并输入工程名。将工程存放在前面创建的名为“例 1.1”的文件夹中，工程名可以自定，此处仍然以“例 1.1”作为工程文件的名字，如图 1.2.5 所示，然后单击“保存”按钮。注意今后生成的 HEX 文件名将自动与工程名相同。工程建立后，工程名为“例 1.1.uvproj”。注意默认的工程后缀与 μ Vision3 及 μ Vision2 版本不同了，为“uvproj”。



图 1.2.5 保存工程

(3) 选择单片机的型号。此时会弹出一个对话框，要求用户根据需要选择单片机。Keil C51 几乎支持所有 51 内核的单片机，并且内核基本通用，所以推荐选择 Atmel 的 AT89C51 或者 AT89C52 单片机（后者比前者多一个定时器 T2，多一个中断源，并且 RAM 和 FLASH 空间更大，但其他相同），如图 1.2.6 所示。此处选择 AT89C51 单片机，对话框右边的“Description”是对该型单片机的简要说明。了解一下该型单片机的基本功能后，单击“OK”按钮。

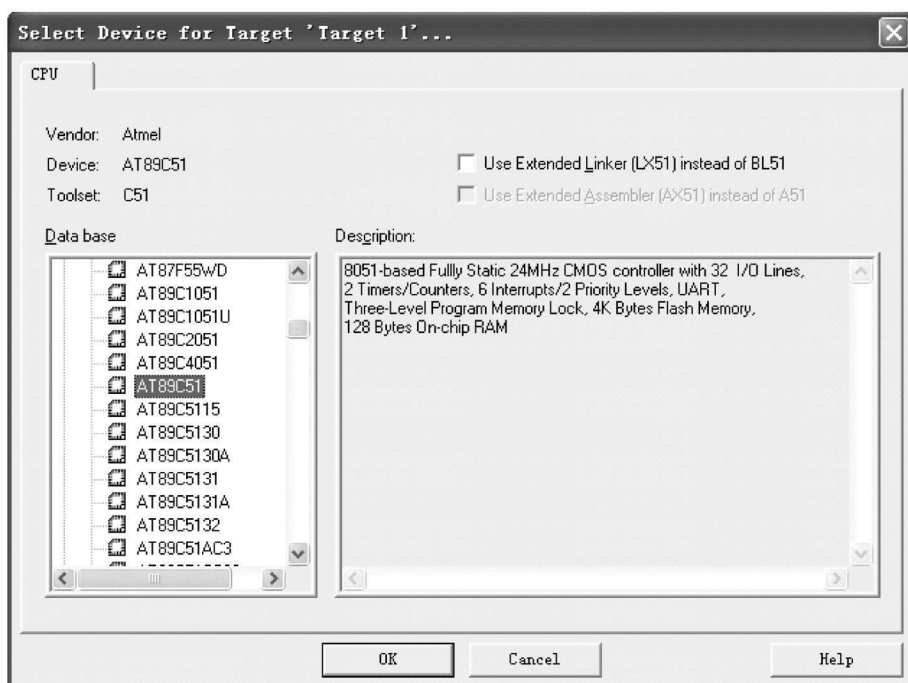


图 1.2.6 选择单片机型号

(4) 添加 “STARTUP.A51” 文件到工程中。此时会弹出一个对话框，提示 “是否复制标准 8051 启动代码到工程文件夹并添加文件到工程？”，如图 1.2.7 所示。



图 1.2.7 添加 “STARTUP.A51” 文件到工程的提示对话框

如果单击 “否”，则不添加 “STARTUP.A51” 文件到工程中，从而对 RAM 清零等操作将采用默认的方式。如果要改变 RAM 清零区域，可以单击 “是”，这样 “STARTUP.A51” 文件的一个副本将添加到工程中，可以根据需要修改此文件。

此处单击 “是”，则窗口界面如图 1.2.8 所示。单击左边 “Project” 选项卡中 “Target 1” 前面的 “+” 号，然后再单击 “Source Group 1” 前面的 “+” 号，能看到一个名为 “STARTUP.A51” 的文件。“STARTUP.A51” 文件是 51 单片机上电初始化文件，包含了 C51 目标程序的启动代码。一些 8051 派生的 CPU 产品要求初始化 CPU 来满足设计中的相应硬件，如果用户需要可以对该文件进行修改，此处保持默认。

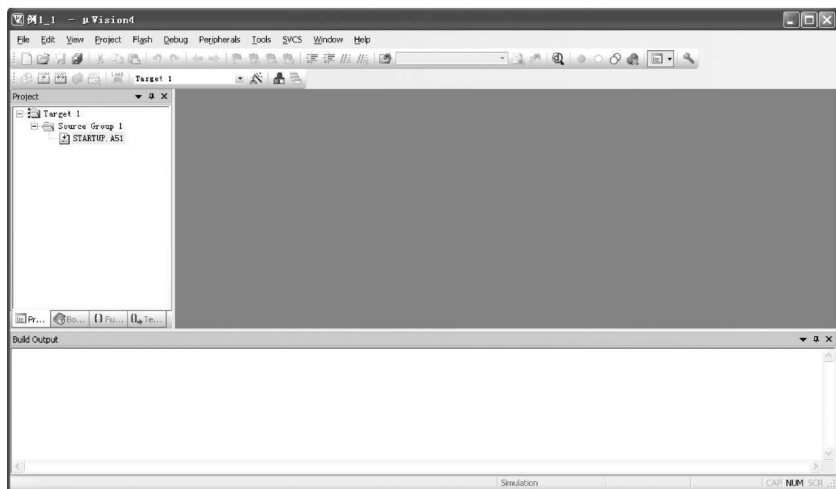


图 1.2.8 工程建好以后的窗口界面

到此为止，已经建立好一个工程。但是工程中还没有任何文件和代码，接下来需要进行添加。

(5) 新建文件。顺序单击菜单“File→New”选项，或者单击工具栏的  按钮来新建文件，如图 1.2.9 所示。新建文件以后的窗口界面如图 1.2.10 所示。

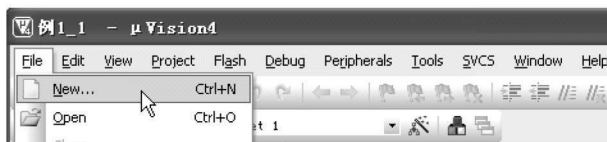


图 1.2.9 新建文件

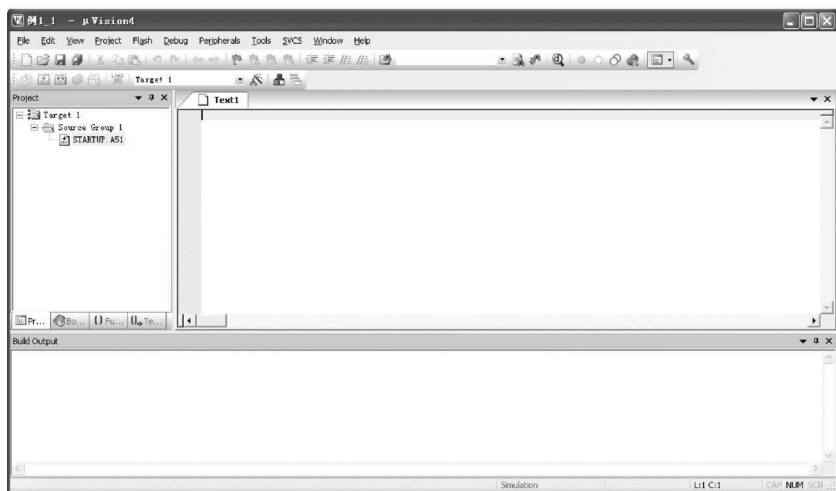


图 1.2.10 新建文件后的窗口界面

该新建文件没有任何扩展名，即没有文件类型。顺序单击菜单“File→Save”选项，或者单击工具栏的  按钮来保存文件。如果用 C51 语言编程，那么扩展名为“.c”；如果用汇编语言编程，那么扩展名为“.asm”。这里的文件名由用户自定，但是不要忘记后面必须加

上对应的扩展名。此处因为使用 C51 语言编程，故将文件命名为“main.c”，然后单击“保存”按钮，如图 1.2.11 所示。

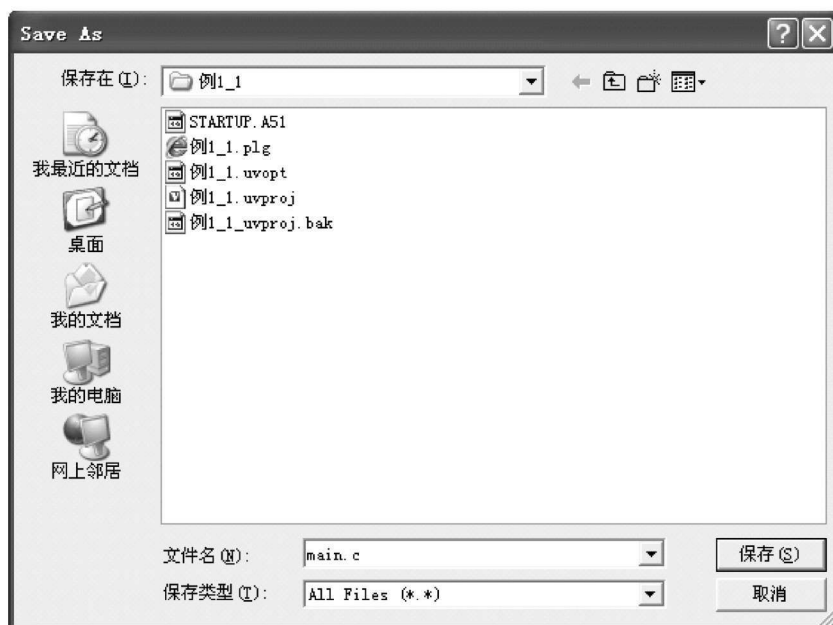


图 1.2.11 保存文件

(6) 添加文件到工程中。源文件虽然建好，但是跟工程还没有直接的联系，此时需要将该文件添加到工程中。单击左边“Project”选项卡中“Target 1”前面的“+”号，然后在“Source Group 1”上单击右键，弹出如图 1.2.12 所示的菜单。然后单击“Add Files to Group ‘Source Group 1’”选项，弹出如图 1.2.13 所示对话框。

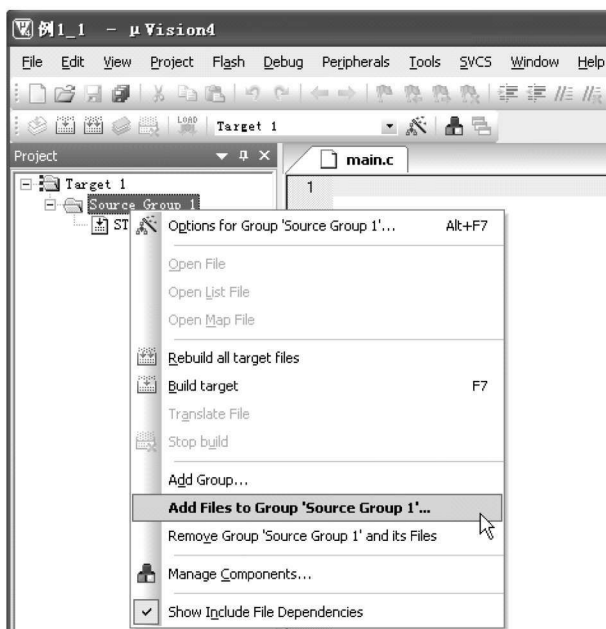


图 1.2.12 将文件加入工程的菜单

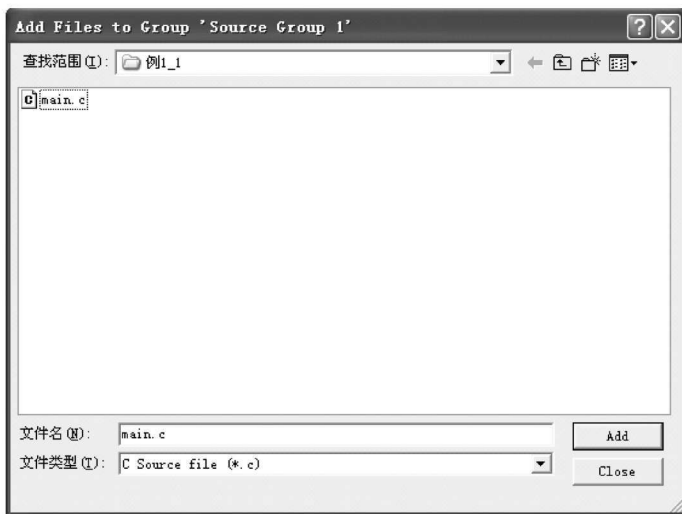


图 1.2.13 添加文件对话框

单击“main.c”文件，接着单击“Add”按钮，再单击“Close”按钮，然后会发现左侧“Source Group 1”下面增加了文件“main.c”，如图 1.2.14 所示。当一个工程中有多个文件时，都需要用同样的方法添加，这样文件才能与工程联系起来。

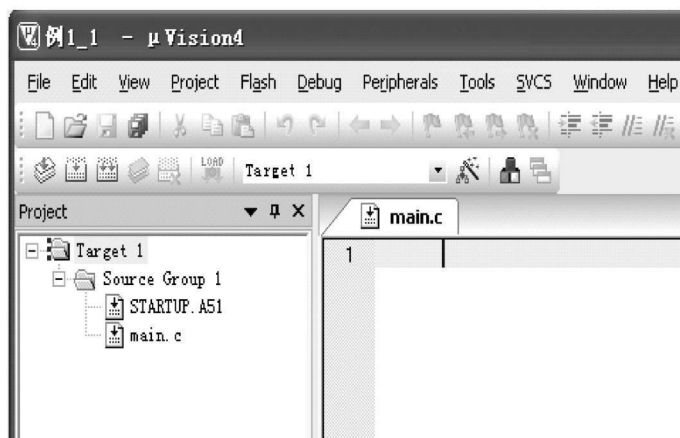


图 1.2.14 将文件添加到工程后的窗口界面

通过以上 6 步，在 Keil μ Vision4 中建立了一个完整的工程。接下来需要编写 C51 语言程序并生成 HEX 程序文件。

1.3 编写 C51 程序并生成 HEX 文件

1.3.1 程序举例及分析

下面以一个完整的 C51 程序为例，说明如何在 Keil μ Vision4 中编程并生成 HEX 文件。

例 1.3.1 编写程序，要求从单片机 P1 口输出 8 位数据，高 4 位为高电平，低 4 位为低电平。

程序清单如下：

```
#include <reg51.h>    //51系列单片机头文件
void main() //主函数
{
    P1=0xf0;    /*P1口输出8位数,高4位为高电平,低4位为低电平*/
    while(1);    /*程序无限循环*/
}
```

程序功能说明如下：

(1) #include <reg51.h> 功能：包含“reg51.h”头文件，即将此头文件内容放到引用头文件的位置。该头文件定义了 51 单片机内部所有特殊功能寄存器（sfr）和控制标志位（sbit）的地址。

例如：（摘抄 reg51.h 头文件中一部分）

```
sfr P1=0x90;
sfr P3=0xb0;
sbit RD=P3^7;
sbit TXD=P3^1;
```

包含该头文件以后，后面编程时只需输入该特殊功能寄存器或控制标志位的名字即可，不需要知道其地址。

(2) C51 语言中注释的两种写法:

//..... ——只能注释一行

/** / ——可注释任意行

加注释的好处是增加程序可读性。

(3) main 主函数的写法:

格式: void main()

特点: 无返回值, 无参数。

任何一个 C51 程序有且仅有一个 main 函数, 它是整个程序开始执行的入口。

例: void main()

```
{  
    程序;  
}
```

(4) P1=0xf0; 语句功能:

P1——51 单片机 8 位准双向并口, 其地址已经在 reg51.h 头文件中定义, 此处可直接引用。注意字母“P”一定要大写, 小写则错误, 因为 C51 语言中字母大写与小写表示两个不同字符。

0x——C51 语言中十六进制数前缀。

f0——此十六进制数用二进制表示等于 1111 0000B, 其高 4 位是高电平, 低 4 位是低电平。

(5) 用“while (1);”语句的原因: 这是一个无限循环语句。单片机只要一通电, 晶振起振, 就不会停止工作, 故程序必须不停运行。

1.3.2 输入程序、编译并生成 HEX 文件

(1) 输入程序。将上述 C51 程序输入到 Keil 中, 会自动识别关键字, 并以不同颜色提示用户。注意输入法应该为英文半角状态, 如果为中文全角输入, 其标点符号将会被认作非法字符。

Keil μ Vision4 中加强了对大括号的监控力度, 当软件发现程序中大括号不成对出现时, 会在单独的大括号上以红色高亮显示, 如图 1.3.1 所示。这样会使用户少犯错误, 能提高编程效率。注意这种错误提示出现的前提是源文件已经保存成 C51 类型。

当程序正确输入完毕以后, 如图 1.3.2 所示。然后顺序单击菜单“File→Save”选项, 或者单击工具栏的  按钮来保存文件。

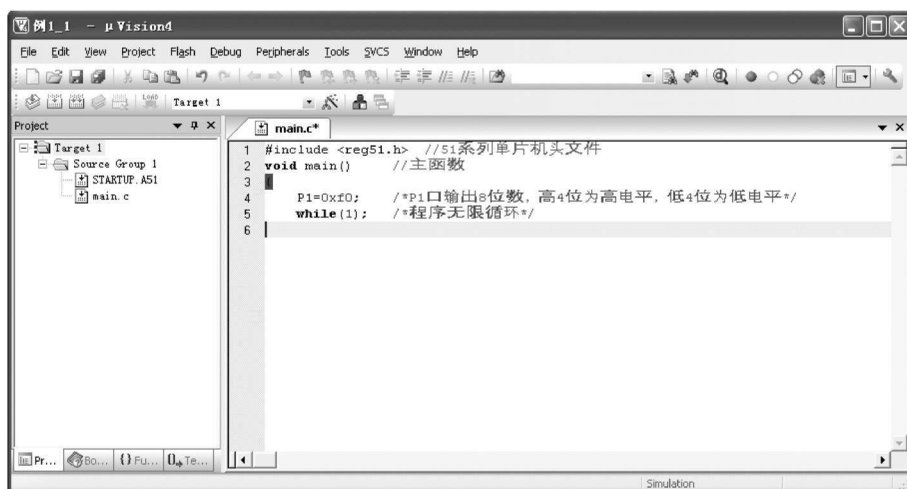


图 1.3.1 忘记输入 main 函数最后一个大括号的出错提示

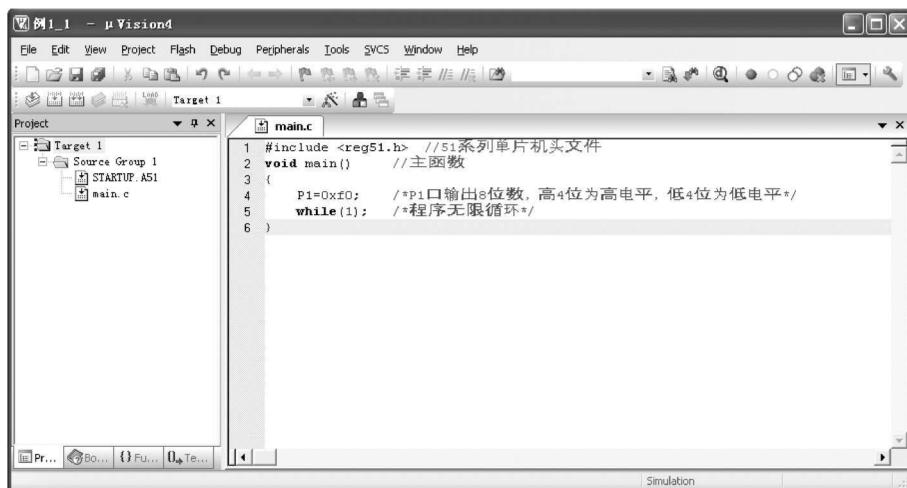


图 1.3.2 正确输入代码以后的编辑界面