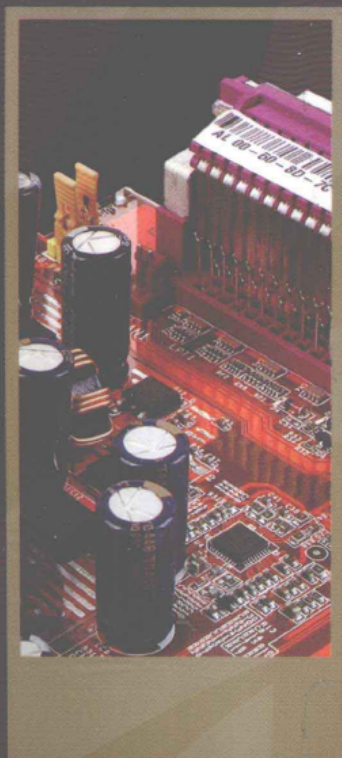


<http://www.phei.com.cn>

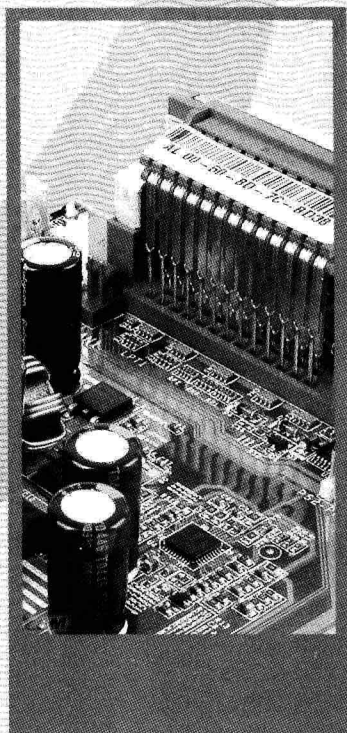


基于 Proteus 和 Keil C51 的 单片机设计与仿真

■ 杜树春 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



基于 Proteus 和 Keil C51 的 单片机设计与仿真

■ 杜树春 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是用 Proteus 软件调试和仿真 51 系列单片机及外围电路的实例集。所用软件是 Proteus 7.5 版本。本书首先介绍和 Proteus 软件配套使用的 Keil C51 单片机的集成开发环境及使用方法；然后介绍单片机交互可视化仿真软件 Proteus 的使用方法；之后，就是使用这两个集成开发工具开发单片机系统的实例，这一部分占了本书的大部分篇幅。

在本书中，对每一种单片机和芯片接口提供了两种不同的程序——汇编语言程序和 C51 语言程序，这扩大了本书的读者范围，即不论是用汇编语言编程的人，还是用 C51 语言编程的人，都可以借鉴本书。

本书适合 51 系列单片机开发的技术人员阅读，也可作为高等学校相关专业的教学用书。

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 Proteus 和 Keil C51 的单片机设计与仿真 / 杜树春编著. —北京: 电子工业出版社, 2012. 2
(职场金钥匙)

ISBN 978 - 7 - 121 - 15501 - 7

I. ① 基… II. ① 杜… III. ① 单片微型计算机 - 系统设计 ② 单片微型计算机 - 系统仿真
IV. ① TP368.1

中国版本图书馆 CIP 数据核字 (2011) 第 264398 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 刘真平

印 刷: 涿州市京南印刷厂
装 订:

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: 19 字数: 486.4 千字

印 次: 2012 年 2 月第 1 次印刷

印 数: 4 000 册 定价: 39.90 元

凡所购买电子工业出版社的图书, 如有缺损问题, 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。

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

服务热线: (010) 88258888。

前 言

目前,单片机开发的产品已被广泛应用于家电、通信、工商业、航空、航天及军事等方面。单片机的系列和型号有很多,但实际应用中最流行的是 8051 系列。这是因为它有众多制造厂商支持,有适于各种应用特点的品种和非常丰富的学习资料。8051 系列也是各大专院校(包括中专、中技和职高)单片机教学的首选机型,因为它是目前应用最广泛、使用人数最多的单片机,事实上它已成为单片机领域的实际标准。

随着电子技术的发展,单片机的开发工具也有了长足的进步。传统的开发工具是程序编译软件、仿真器和编程器(又称写入器)等。这种开发方法的特点是:在开发期间,必须有单片机及外围电路的硬件支持。近几年来开发出的电子设计自动化(EDA)软件,使单片机及其外围电路的软硬件设计和调试都能在计算机虚拟环境下进行。其开发步骤是:先把待开发单片机系统的电路原理图画好,再把配套运行的程序编好,然后就可以进入仿真调试环境。随着程序的执行,运行结果可以在计算机屏幕上的电路原理图上反映出来。比如,你要点亮电路图上画的某个发光二极管,或让电路图上画的喇叭发声,或让电路图上画的直流电动机转动,这些都可以实现。用这种方法调试完成后,再按照调试所用元器件清单把元器件购买回来,制出与调试时一致的印制电路板,组装起来,把调好的程序下载或写入到单片机里,一个单片机系统开发工作即告完成。用这种软件进行开发的好处是:避免了用传统方法开发时方案更换带来的多次购买元器件及制版的麻烦,节约了时间和经费。

英国 Labcenter 公司开发的电路分析与实物仿真软件 Proteus ISIS 就是这样一种软件。它运行于 Windows 操作系统上,可以仿真、分析(SPICE)各种模拟元器件、集成电路和众多的单片机系统。Proteus ISIS 软件既可以用于设计电路原理图,又可以画印制电路板(PCB)图,并可以实现交互可视化仿真。Proteus ISIS 软件提供了 30 多个元器件库,上万个元器件。元器件涉及电阻、电容、二极管、三极管、MOS 管、变压器、继电器、放大器、各种激励源、300 多种微控制器、TTL 和 CMOS 集成电路等。Proteus ISIS 软件提供了多个虚拟仪表,包括示波器、逻辑分析仪、定时器/计数器、虚拟终端、信号发生器、图形发生器、交直流电压表、交直流电流表等。总之,Proteus ISIS 这个交互可视化仿真软件一经使用,便给人以耳目一新的感觉。

本书是用 Proteus 软件调试和仿真 51 系列单片机及外围电路的实例集。实例部分介绍了 51 单片机和外围器件接口方法,每一实例选用一种或几种接口芯片。介绍的次序如下:首先是该单片机接口芯片功能简介,其次是该芯片与单片机连接电路原理,该芯片汇编语言程序,该芯片 C51 语言程序,该芯片的调试和仿真,最后是使用该芯片的注意事项(部分章节没有这一项)。

本书共分 23 章,第 1 章介绍 Keil C51 集成开发环境及其用法;第 2 章介绍单片机仿真软件 Proteus 的用法。第 3~15 章是单片机和外围芯片连接的例子,共 13 个。第 3 章介绍 8 位串行输入、串行或并行输出的移位寄存器 74HC595 的用法;第 4 章介绍串行 E²PROM 芯片 AT24C02 的用法;第 5 章介绍带 RAM 存储器的实时时钟芯片 PCF8583;第 6 章介绍带 RAM 存储器的串行 I²C 实时时钟芯片 DS1307;第 7 章介绍单通道 12 位串行 A/D 转换器 MAX1241;第 8 章介绍 11 路 12 位串行 A/D 转换器 TLC2543;第 9 章介绍串行 8 通道多量程

12 位 A/D 转换器 MAX127；第 10 章介绍单路 8 位串行 D/A 转换器 MAX517；第 11 章介绍单路 10 位串行 D/A 转换器 MAX515；第 12 章介绍 I²C 总线 8 位远程 I/O 扩展口芯片 PCF8574；第 13 章介绍 SPI 串行总线 E²PROM X25040；第 14 章介绍铁电存储器（FRAM）FM24C64；第 15 章介绍铁电存储器（FRAM）FM24C256。第 16 ~ 23 章是单片机和常用集成电路芯片接口的例子，共 8 个。第 16 章介绍译码器；第 17 章介绍七段显示译码器；第 18 章介绍优先编码器；第 19 章介绍触发器；第 20 章介绍锁存器；第 21 章介绍移位寄存器；第 22 章介绍数据选择器；第 23 章介绍模拟电子开关。

电子资料包中的内容仍以书中章节为单位，每一章（第 3 ~ 23 章）下都有 1 个文件夹，内有多个文件。其中，扩展名为“DSN”的文件是 Proteus 仿真原理图文件，在 Proteus 软件已安装在计算机中的前提下，双击具有“DSN”扩展名的文件，就可进入显示电路原理图画面，也就是 Proteus 的调试状态；扩展名为“ASM”的是 51 系列汇编语言程序；扩展名为“C”的是 C51 语言程序；扩展名为“UV2”的是 Keil C51 工程文件；扩展名为“HEX”的是用 Keil C51 软件对上述汇编语言程序（或 C51 语言程序）编译生成的十六进制文件。在原理图的单片机上双击鼠标左键，在弹出的“编辑元件”对话框内的“Program Files”文本框中输入“XXX.HEX”，单击“确定”按钮即完成十六进制程序文件载入，之后，就可以仿真和调试。书中的应用程序都已在 Proteus 环境下调试通过，读者既可以原封不动地执行程序，也可以边改边试。读者可以登录电子工业出版社电子技术出版分社网站 <http://yydz.phei.com.cn>，到“资源下载”栏目下载电子资料包。

本书适合 51 系列单片机开发的技术人员阅读，也可作为高等学校相关专业的教学用书。

在编写过程中，得到了电子工业出版社张剑编辑的指导和帮助，同时，也得到杜菲、姜鸿斌等同志的协助。在此，向以上单位和同志表示衷心感谢。

由于编著者水平有限且时间仓促，书中难免存在缺点和错误，恳请读者批评指正。作者 E-mail: dushuchun@263.net。

编著者
2011 年 7 月 29 日

目 录

| | |
|--|----|
| 第 1 章 单片机开发软件 Keil C51 的使用 | 1 |
| 1.1 Keil C51 简介 | 1 |
| 1.2 安装 Keil C51 | 2 |
| 1.3 建立一个工程 | 4 |
| 1.4 单片机选型 | 5 |
| 1.5 创建源程序 | 6 |
| 1.6 把新创建的源程序加入工程文件中 | 8 |
| 1.7 工程的设置 | 9 |
| 1.8 编译 | 14 |
| 1.9 调试 | 15 |
| 1.10 调试窗口 | 16 |
| 第 2 章 单片机仿真软件 Proteus 的使用 | 19 |
| 2.1 进入 Proteus ISIS | 19 |
| 2.2 工作界面 | 20 |
| 2.3 Proteus ISIS 原理图设计 | 26 |
| 2.4 Proteus ISIS 原理图设计中若干注意事项 | 34 |
| 2.5 Proteus 软件的仿真调试方法 | 36 |
| 2.6 Proteus VSM 仿真工具简介 | 48 |
| 第 3 章 串入并出移位寄存器芯片 74HC595 | 53 |
| 3.1 74HC595 简介 | 53 |
| 3.2 74HC595 与单片机 AT89C52 的连接 | 54 |
| 3.3 汇编语言程序 1 | 55 |
| 3.4 汇编语言程序 2 | 56 |
| 3.5 C51 语言程序 | 57 |
| 3.6 用 Proteus 软件仿真和调试 | 58 |
| 3.7 使用 74HC595 的注意事项 | 60 |
| 第 4 章 串行 E²PROM 芯片 AT24C02 | 61 |
| 4.1 AT24C02 简介 | 61 |
| 4.2 AT24C02 与单片机 AT89C52 的连接 | 62 |
| 4.3 汇编语言程序 | 62 |
| 4.4 C51 语言程序 | 67 |

| | | |
|--------------|--|------------|
| 4.5 | 用 Proteus 软件仿真和调试 | 72 |
| 4.6 | 使用 AT24C02 的注意事项 | 74 |
| 第 5 章 | 带 RAM 存储器的实时时钟芯片 PCF8583 | 75 |
| 5.1 | PCF8583 简介 | 75 |
| 5.2 | PCF8583 和单片机 AT89C52 的连接 | 77 |
| 5.3 | 汇编语言程序 | 78 |
| 5.4 | C51 语言程序 | 82 |
| 5.5 | 用 Proteus 软件仿真和调试 | 87 |
| 5.6 | 使用 PCF8583 的注意事项 | 89 |
| 第 6 章 | 带 RAM 存储器的串行 I²C 实时时钟芯片 DS1307 | 90 |
| 6.1 | DS1307 简介 | 90 |
| 6.2 | DS1307 和单片机 AT89C52 的连接 | 93 |
| 6.3 | 汇编语言程序 | 93 |
| 6.4 | C51 语言程序 | 98 |
| 6.5 | 用 Proteus 软件仿真和调试 | 104 |
| 6.6 | 使用 DS1307 的注意事项 | 105 |
| 第 7 章 | 单通道 12 位串行 A/D 转换器 MAX1241 | 107 |
| 7.1 | MAX1241/MAX1240 简介 | 107 |
| 7.2 | MAX1241 和单片机 AT89C52 的连接 | 108 |
| 7.3 | 汇编语言程序 | 110 |
| 7.4 | C51 语言程序 | 111 |
| 7.5 | 用 Proteus 软件仿真和调试 | 113 |
| 7.6 | 使用 MAX1241 的注意事项 | 115 |
| 第 8 章 | 11 路 12 位串行 A/D 转换器 TLC2543 | 116 |
| 8.1 | TLC2543 简介 | 116 |
| 8.2 | TLC2543 和单片机 AT89C52 的连接 | 119 |
| 8.3 | 汇编语言程序 | 119 |
| 8.4 | C51 语言程序 | 121 |
| 8.5 | 用 Proteus 软件仿真和调试 | 123 |
| 8.6 | 使用 TLC2543 的注意事项 | 124 |
| 第 9 章 | 串行 8 通道多量程 12 位 A/D 转换器 MAX127 | 125 |
| 9.1 | MAX127 简介 | 125 |
| 9.2 | MAX127 和单片机 AT89C52 的连接 | 129 |
| 9.3 | 汇编语言程序 | 130 |
| 9.4 | C51 语言程序 1 | 133 |

| | | |
|---------------|--|------------|
| 9.5 | C51 语言程序 2 | 137 |
| 9.6 | 用 Proteus 软件仿真和调试 | 141 |
| 9.7 | 使用 MAX127 的注意事项 | 143 |
| 第 10 章 | 单路 8 位串行 D/A 转换器 MAX517 | 145 |
| 10.1 | I ² C 总线的特点及基本通信协议 | 145 |
| 10.2 | MAX517 简介 | 146 |
| 10.3 | MAX517 的工作时序 | 146 |
| 10.4 | MAX517 和单片机 AT89C52 的连接 | 147 |
| 10.5 | 汇编语言程序 | 148 |
| 10.6 | C51 语言程序 | 149 |
| 10.7 | 用 Proteus 软件仿真和调试 | 151 |
| 10.8 | 使用 MAX517 的注意事项 | 153 |
| 第 11 章 | 单路 10 位串行 D/A 转换器 MAX515 | 154 |
| 11.1 | MAX515 简介 | 154 |
| 11.2 | MAX515 和单片机 AT89C52 的连接 | 155 |
| 11.3 | 汇编语言程序 | 157 |
| 11.4 | C51 语言程序 | 158 |
| 11.5 | 用 Proteus 软件仿真和调试 | 160 |
| 11.6 | 使用 MAX515 的注意事项 | 161 |
| 第 12 章 | I²C 总线 8 位远程 I/O 扩展口芯片 PCF8574 | 162 |
| 12.1 | PCF8574 简介 | 162 |
| 12.2 | PCF8574 作为输出口 | 165 |
| 12.3 | PCF8574 作为输入口 | 173 |
| 12.4 | 使用 PCF8574 的注意事项 | 180 |
| 第 13 章 | SPI 串行总线 E²PROM X25040 | 182 |
| 13.1 | X25040 简介 | 182 |
| 13.2 | X25040 和单片机 AT89C52 的连接 | 185 |
| 13.3 | 汇编语言程序 | 186 |
| 13.4 | C51 语言程序 | 194 |
| 13.5 | 用 Proteus 软件仿真和调试 | 198 |
| 13.6 | 使用 X25040 的注意事项 | 199 |
| 第 14 章 | 铁电存储器 (FRAM) FM24C64 | 201 |
| 14.1 | FM24C64 简介 | 201 |
| 14.2 | FM24C64 和单片机 AT89C52 的连接 | 204 |
| 14.3 | 汇编语言程序 | 205 |

| | | |
|---------------|------------------------------------|------------|
| 14.4 | C51 语言程序 | 211 |
| 14.5 | 用 Proteus 软件仿真和调试 | 215 |
| 14.6 | 使用 FM24C64 的注意事项 | 217 |
| 第 15 章 | 铁电存储器 (FRAM) FM24C256 | 218 |
| 15.1 | FM24C256 简介 | 218 |
| 15.2 | FM24C256 和单片机 AT89C52 的连接 | 219 |
| 15.3 | 汇编语言程序 | 220 |
| 15.4 | C51 语言程序 | 226 |
| 15.5 | 用 Proteus 软件仿真和调试 | 230 |
| 15.6 | 使用 FM24C256 的注意事项 | 232 |
| 第 16 章 | 译码器 | 233 |
| 16.1 | 74LS138 3—8 译码器的应用 | 233 |
| 16.2 | 74LS139 2—4 译码器的应用 | 236 |
| 第 17 章 | 七段显示译码器 | 240 |
| 17.1 | 74LS48 共阴极七段显示译码器的应用 | 240 |
| 17.2 | CD4543 共阳极七段显示译码器的应用 | 243 |
| 第 18 章 | 优先编码器 | 246 |
| 18.1 | 74HC148 优先编码器的应用 | 246 |
| 18.2 | 74HC147 优先编码器的应用 | 249 |
| 第 19 章 | 触发器 | 254 |
| 19.1 | 74LS273 触发器的应用 | 254 |
| 19.2 | 74LS378 触发器的应用 | 256 |
| 第 20 章 | 锁存器 | 259 |
| 20.1 | 74HC573 8D 锁存器的应用 | 259 |
| 20.2 | 74HC373 8D 锁存器的应用 | 261 |
| 第 21 章 | 移位寄存器 | 265 |
| 21.1 | 74LS165 移位寄存器的应用 | 265 |
| 21.2 | CD4094 移位寄存器的应用 | 268 |
| 第 22 章 | 数据选择器 | 271 |
| 22.1 | 74HC251 数据选择器的应用 | 271 |
| 22.2 | 74HC257 数据选择器的应用 | 274 |

| | |
|-----------------------------|-----|
| 第 23 章 模拟电子开关 | 278 |
| 23.1 CD4051 模拟电子开关的应用 | 278 |
| 23.2 CD4053 模拟电子开关的应用 | 281 |
| 23.3 CD4066 模拟电子开关的应用 | 284 |
| 附录 A 51 单片机指令集 | 287 |
| 参考文献 | 291 |

第 1 章 单片机开发软件

Keil C51 的使用



1.1 Keil C51 简介

常用的单片机及嵌入式系统编程语言有两种，即汇编语言和 C 语言。汇编语言的机器代码生成效率很高，但其可读性却并不强，复杂一点的程序就更是很难读懂。C 语言在大多数情况下，其机器代码生成效率和汇编语言相当，但可读性和可移植性却远远超过汇编语言，而且 C 语言还可以嵌入汇编语言来解决高时效性的代码编写问题。就开发周期来说，用 C 语言编写中大型软件的开发周期通常要比用汇编语言编写的短很多。与汇编语言相比，C 语言在功能性、结构性、可读性、可维护性上有明显的优势，因而易学易用。由此可见，使用 C 语言编写程序是一种非常好的选择。

使用 C 语言肯定要用到 C 编译器，以便把写好的 C 程序编译为机器码，这样单片机才能执行编写好的程序。用过汇编语言后再使用 C 语言来开发，体会更加深刻。Keil μ Vision3 是众多单片机应用开发软件中最优秀的软件之一，它支持众多不同公司的 MCS51 架构的芯片，它集编辑、编译、仿真等功能于一体，同时还支持 PLM、汇编和 C 语言的程序设计，它的界面与常用的微软 VC++ 的界面相似，界面友好，易学易用。

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。Keil C51 软件提供了丰富的库函数和功能强大的集成开发调试工具，全 Windows 界面。

另外，只要看一下编译后生成的汇编代码，就能体会到 Keil C51 生成目标代码的效率非常高，多数语句生成的汇编代码很紧凑，容易理解。在开发大型软件时，更能体现采用高级语言的优势。

Keil C51 标准 C 编译器为 8051 微控制器的软件开发提供了 C 语言环境，同时保留了汇编代码高效、快速的特点。C51 编译器的功能不断增强，使用户可以更加“贴近”CPU 本身及其他的衍生产品。C51 已被完全集成到 μ Vision3 的集成开发环境中，这个集成开发环境包括编译器、汇编器、实时操作系统、项目管理器和调试器， μ Vision3 IDE 可以为它们提供单一而灵活的开发环境。C51V808A 版本是目前版本较高的 8051 开发平台，它可以支持所有 8051 的衍生产品，也可以支持所有兼容的仿真器，同时也支持其他第三方开发工具。

C51 是一种专门为 8051 单片机设计的高级语言 C 编译器，支持符合 ANSI 标准的 C 语言程序设计，同时针对 8051 单片机的自身特点做了一些特殊扩展。

C51 默认值不支持函数递归调用，需要进行递归调用的函数必须声明为再入函数。再入函数的局部数据和参数被放入再入栈中，从而允许进行递归调用。

μ Vision3 支持所有的 Keil 80C51 的工具软件，包括 C51 编译器、宏汇编器、链接器/定位器和目标文件至 Hex 格式转换器， μ Vision3 可以自动完成编译、汇编、链接程序等操作。 μ Vision3 的特点如下。

1) **C51 编译器和 A51 汇编器** 由 μ Vision3 IDE 创建的源文件可以被 C51 编译器或 A51 汇编器处理,生成可重定位的 object 文件。Keil C51 编译器支持 ANSI C 语言标准,也支持 C 语言的所有标准特性。另外,还增加了几个可以直接支持 80C51 结构的特性。Keil A51 宏汇编器支持 80C51 及其派生系列的所有指令集。

2) **LIB51 库管理器** LIB51 库管理器可以把由汇编器和编译器创建的目标文件建立目标库。这些库是按规格格式排列的目标模块,可被链接器使用。当链接器处理一个库时,仅使用库中程序使用的目标模块,而不是全部加以引用。

3) **BL51 链接器/定位器** BL51 链接器/定位器使用从库中提取出来的目标模块,以及由编译器、汇编器生成的目标模块,创建一个绝对地址目标模块。绝对地址目标文件或模块包括不可重定位的代码和数据。所有的代码和数据都被固定在具体的存储器单元中。

4) **μ Vision3 软件调试器** μ Vision3 软件调试器能十分理想地进行快速、可靠的程序调试。调试器包括一个高速模拟器,用户可以使用它来模拟整个 80C51 系统,包括片上外围器件和外部硬件。当用户从器件数据库选择器件时,这个器件的属性会被自动配置。

5) **μ Vision3 硬件调试器** μ Vision3 硬件调试器向用户提供了几种在实际目标硬件上测试程序的方法。

6) **RTX51 实时操作系统** RTX51 实时操作系统是针对 80C51 微控制器系列的一个多任务内核。RTX51 实时内核简化了需要对实时事件进行反应的复杂应用的系统设计、编程和调试。这个内核完全集成在 C51 编译器中,使用非常简单。任务描述表和操作系统的一致性由 BL51 链接器/定位器自动进行控制。

此外, μ Vision3 还具有极其强大的软件环境、友好的操作界面和简单快捷的操作方法,主要表现在以下几点:

- 丰富的菜单栏。
- 可以快速选择命令按钮的工具条。
- 一些源代码文件窗口。
- 对话框窗口。
- 直观明了的信息显示窗口。



1.2 安装 Keil C51

要使用 Keil C51 软件,首先必须把它安装在计算机中。Keil C51 是由 Keil Software 公司推出的一个商业软件,有正版和评估版两个版本,评估版可从网上免费下载,评估版的目标代码有小于 2KB 的限制,不过,这对一般的个人学习和小型应用已经够用了。

现在介绍安装评估版软件的方法。

单击从网上下载的评估版软件 C51V808A.exe,计算机屏幕就会出现 Keil C51 软件安装对话框。根据安装对话框中的提示,做正确的选择,逐步执行下去,很快就能完成 Keil C51 软件的安装。安装完毕,计算机桌面上就会有一个 Keil μ Vision3 图标。

单击计算机桌面上的“Keil μ Vision3”图标,就会出现如图 1-1 所示的启动界面。

进入 Keil μ Vision3 集成开发环境后,有一个问题不得不提,就是在编辑 C 语言或汇编语言程序时,中文注释中汉字的下面部分会缺失,而且每一行的最后一个字总要有重叠的现

象，给人以不舒服的感觉，如图 1-2 所示。

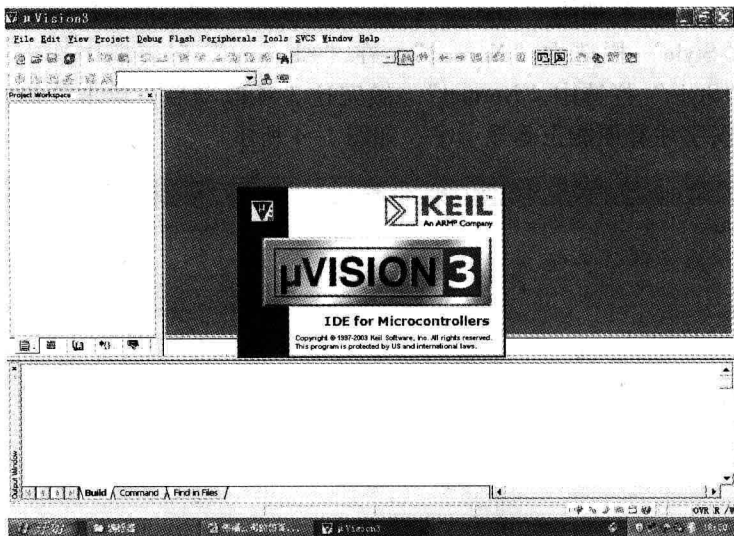


图 1-1 Keil μVision3 的启动界面

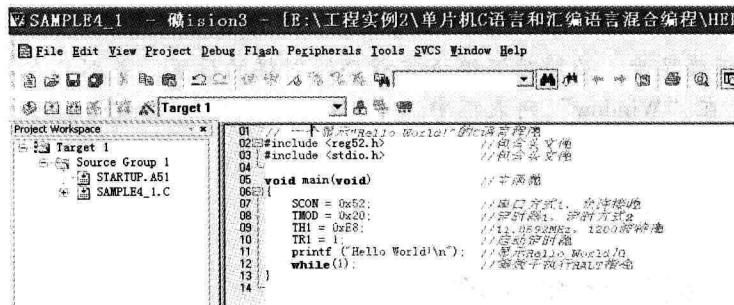


图 1-2 程序中文注释每一行的最后一个字有重叠现象

解决的办法为：执行菜单命令“Edit”→“Configuration...”，弹出“Configuration”对话框，如图 1-3 所示。在“Configuration”对话框中选中“Colors & Fonts”选项卡，在“Window”

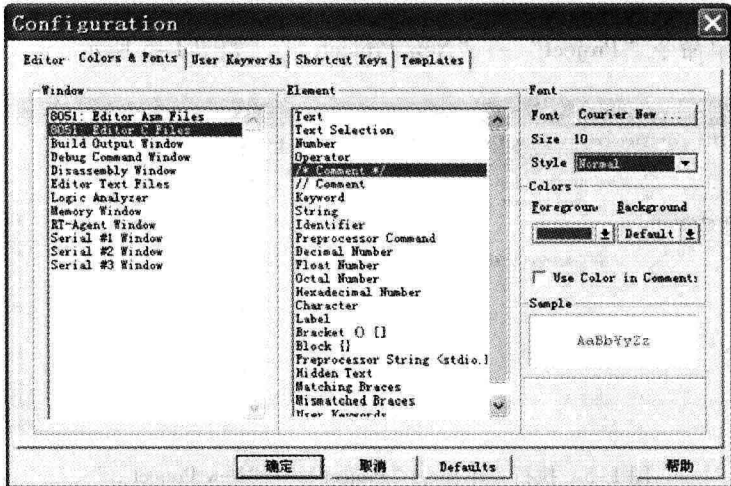


图 1-3 “Configuration”对话框

列表框中选中“8051: Editor C Files”，在“Element”列表框中选中“/* Comment */”，在“Font”区域的“Style”栏选中“Normal”；在“Element”列表框中选中“//Comment”，在“Font”区域的“Style”栏选中“Normal”；在“Element”列表框中选中“Keyword”，在“Font”区域的“Style”栏选中“Normal”。做完这3项改动后，单击“确定”按钮，退出配置设置，这样汉字注释便能正确显示了，如图1-4所示。

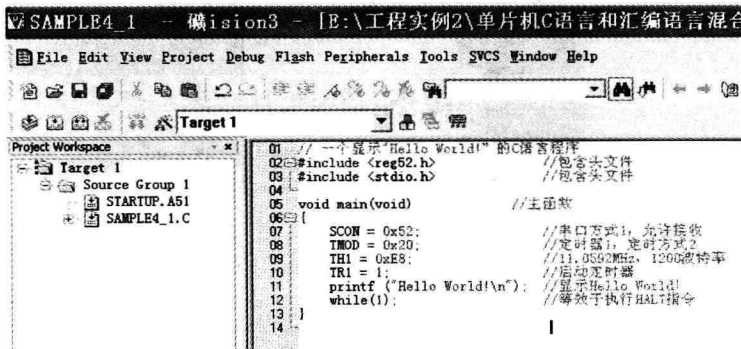


图1-4 程序汉字注释正确显示

说明 上述让程序中汉字注释正确显示是针对C语言而言的。如果要想让汇编语言程序中汉字注释也能正确显示，必须对汇编语言程序做同样的设置。差别只在打开“Configuration”对话框后，在“Window”列表框中选中“8051: Editor Asm Files”，之后的设置同C语言。



1.3 建立一个工程

Keil C51 是 Windows 版的软件，无论是用汇编语言还是 C 语言，无论是只有一个文件的程序还是有多个文件的程序，都要建立一个工程文件。没有工程文件，就不能进行编译和仿真。建立一个新的工程文件的步骤如下。

(1) 执行菜单命令“Project”→“New Project...”，如图1-5所示。

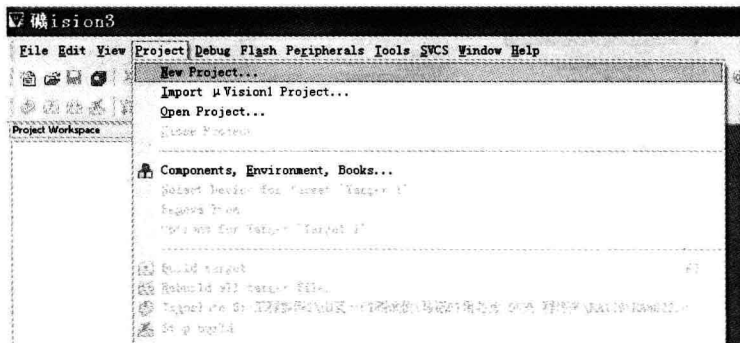


图1-5 执行菜单命令“Project”→“New Project...”

(2) 选择要保存的路径, 输入工程文件的名字。例如, 保存到 Keil 目录里, 工程文件的名字为“TEST1.UV2”, 然后单击“保存”按钮, 这样一个新的工程就建好了, 如图 1-6 所示。

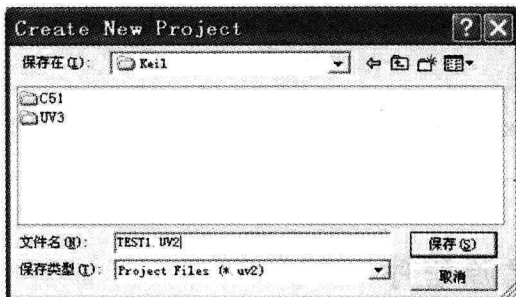


图 1-6 “Create New Project”对话框



1.4 单片机选型

在“Create New Project”对话框中单击“保存”按钮, 弹出“Select Device for Target ‘Target 1’”对话框, 在此可以选择自己使用的单片机的型号。Keil C51 几乎支持所有的 51 内核的单片机。如果读者设计的是华邦的 W77E58, 可以选择“Winbond”→“W77E58”。在此还是以大家用得比较多的 Atmel 的 AT89C52 来说明, 选择“AT89C52”后, 在“Description”栏会显示对 AT89C52 的基本说明, 然后单击“确定”按钮即可, 如图 1-7 所示。

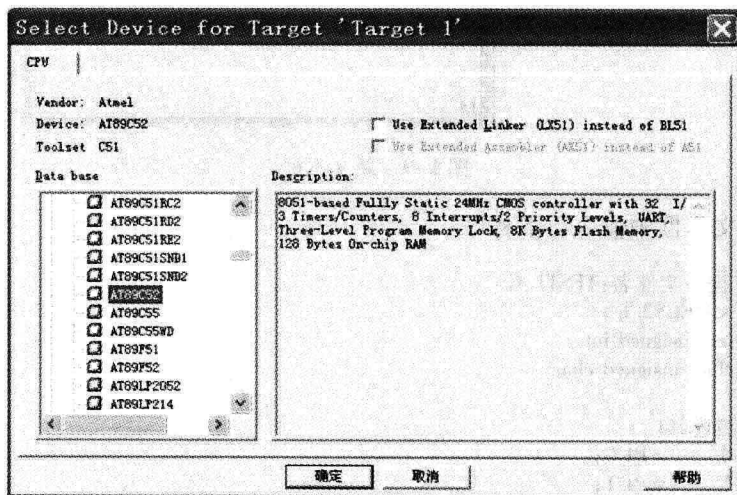


图 1-7 选择要使用的单片机

选择单片机厂商和型号后, Keil C51 系统会进一步询问是否将 STARTUP.A51 文件复制到设计项目中, 如图 1-8 所示。STARTUP.A51 文件原来是放在 \ Keil \ C51 \ LIB \ 文件夹中的, 它提供了 C51 用户程序执行前必须先执行的一些初始语句, 如堆栈区的设置、程序执行首地址, 以及 C 语言中定义的一些变量和数组的初始化等。若单击“是”按钮, 系统

就会把 STARTUP.A51 文件复制到项目文件中；若单击“否”按钮，就不会复制此文件。

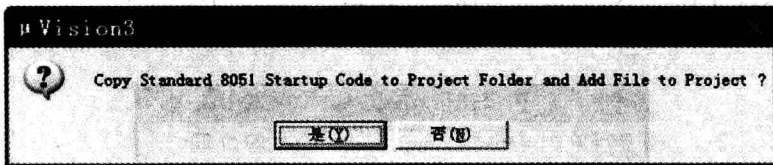


图 1-8 选择是否将 STARTUP.A51 文件复制到设计项目中



1.5 创建源程序

现在新建一个源程序文件（汇编或 C 文件）。如果已经有源程序文件，可以忽略这一步。执行菜单命令“File”→“New”，即出现如图 1-9 所示的新文本框。

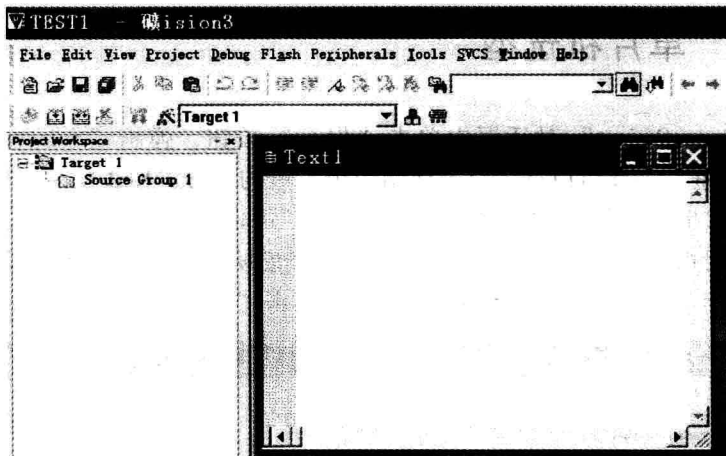


图 1-9 新文本框

(1) 在新的文本框中输入以下程序：

```
//C 语言程序文件名:TEST1.C
#include <REG52.h>
#define uint unsigned int
#define uchar unsigned char

//定义控制端口
sbit P10 = P1^0;
sbit P11 = P1^1;
sbit P12 = P1^2;
sbit P13 = P1^3;
sbit P14 = P1^4;
sbit P15 = P1^5;
sbit P16 = P1^6;
sbit P17 = P1^7;

sbit P00 = P0^0;
```



```
sbit P01    = P0^1;
sbit P02    = P0^2;
sbit P03    = P0^3;
sbit P04    = P0^4;
sbit P05    = P0^5;
sbit P06    = P0^6;
sbit P07    = P0^7;

//函数声明
void Delay(uint);    //延时函数
/ *****
//延时
*****/
void delay(uint x)
{
    uchar tw;
    while (x-->0) {
        for (tw=0;tw<125;tw++) {};
    }
}

/ *****
//主函数
*****/
void main(void)
{
    Delay(200);
    P10=0;
    P11=1;
    P12=0;
    P13=1;
    P14=0;
    P15=1;
    P16=0;
    P17=1;

    P00=0;
    P01=1;
    P02=0;
    P03=1;
    P04=0;
    P05=1;
    P06=0;
    P07=1;
    Delay(200);
}
```

输入程序后的窗口如图 1-10 所示。

(2) 执行菜单命令“File”→“Save”，弹出“Save As”对话框，如图 1-11 所示。

(3) 选择要保存的路径，在“文件名”栏输入文件名。注意：一定要输入扩展名，如果是 C 程序文件，扩展名为 .C；如果是汇编文件，扩展名为 .ASM；如果是其他文件类型，如注解说明文件，则可以保存为 .TXT 的扩展名。在此要存储一个 C 语言源程序文件，所以输入扩展名“.C”，保存文件名为“TEST1.C”（也可以保存为其他名字），单击“保存”按钮。