

■ 聚焦 EDA

基于 Proteus 的 51 系列单片机 设计与仿真

侯玉宝 陈忠平 李成群 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

聚焦 EDA

基于 Proteus 的 51 系列单片机设计与仿真

侯玉宝 陈忠平 李成群 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书以目前流行的软、硬件仿真软件 Proteus 为核心,从实验、实践、实用的角度,通过丰富的实例详细叙述了该软件在 51 单片机课程教学和单片机应用产品开发过程中的应用。全书共 9 章,主要介绍 51 单片机系统的设计及相关软件的使用,在 Proteus 中原理图的绘制与仿真及 PCB 的制作、Proteus 在单片机软件程序设计中的应用,Proteus 在单片机硬件系统设计中的应用。书中选择的实例都具有很强的实用性,通过阅读这些实例,读者可以在不花费硬件成本的前提下,学习和开发单片机软、硬件系统。

本书适合从事单片机应用研发的技术人员阅读,也可作为高等院校单片机课程的教学用书。

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 的 51 系列单片机设计与仿真/侯玉宝等编著. —北京:电子工业出版社,2008.9
(聚焦 EDA)

ISBN 978-7-121-07226-0

I. 基… II. 侯… III. ①单片微型计算机—系统设计—应用软件, PROTEUS ②单片微型计算机—系统仿真—应用软件, PROTEUS IV. TP368

中国版本图书馆 CIP 数据核字(2008)第 120133 号

策划编辑:张 剑

责任编辑:张燕虹

印 刷:北京市顺义兴华印刷厂

装 订:三河市双峰印刷装订有限公司

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:21.75 字数:585 千字

印 次:2008 年 9 月第 1 次印刷

印 数:5 000 册 定价:39.80 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。

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

与单片机打交道已有多多年，记得刚接触单片机的时候觉得它很神奇，只需通过修改相关代码就能进行相应控制。但是，在初学的时候感觉很难，也曾有过放弃学习它的念头，这主要是苦于条件较差，没有资金配备专门的开发工具，好不容易学习了一些理论知识，也只能在纸上编写程序（当时计算机还没现在这样普及），不能与被控制对象很好地联系起来。

随着时间的推移，单片机的应用与普及是我们始料未及的。现在，许多单片机初学者都是先学习单片机相关基础知识，然后使用计算机在一些编译软件上编写程序，通过分析寄存器的内容以检验程序设计是否符合要求，或者自己焊接硬件电路将源程序代码下载到单片机中，观察运行效果。这种学习方法很好，但是如果所设计系统较复杂，难以分析相关寄存器内容，或者由于硬件成本过高，难以制作硬件电路时，这种学习方法就有所欠缺，尤其是对于在校学生而言。

一次偶然的机会接触了 Proteus，使用之后，感觉很好，它对于单片机学习或者系统开发都能提供有力帮助。Proteus 是多功能的 EDA 软件，真正实现了虚拟物理原型功能，在目标板还没投产前，就可以对所设计的硬件系统的功能、合理性和性能指标进行充分调整，可以在没有硬件电路的情况下，进行相应的程序设计与调试，可由仿真原理图直接导出绘制成印制电路板（PCB）。虚拟物理原型，对于单片机初学者来讲可节约学习成本，提高学习积极性；对于单片机系统开发人员来讲，可缩短开发时间，提高设计效率，降低开发成本。

为使初学者能迅速入门，提高对单片机的兴趣与爱好，并能在短期内掌握单片机的应用开发要领，特编写此书。在编写过程中，编者注重题材的取舍，使本书具有以下四个特点。

1. 由浅入深，循序渐近

本书在内容编排上采用由浅入深、由易到难的原则，从最初的 51 单片机开发环境、Proteus 软件的使用，到单片机内部单元的实现，再到单片机外围扩展，直至单片机的工业控制应用。这样的编排可让读者根据自己的情况选择阅读起点。

2. 软硬结合，虚拟仿真

沿用传统单片机学习与开发经验，通过相关编译软件（如 Keil）编写程序并生成 HEX 文件，然后在 Proteus 中画好硬件电路图（这一过程相当于硬件电路的焊接），调用 HEX 文件进行虚拟仿真（这一过程相当于硬件调试）。这样，可节约学习成本，提高读者的学习积极性。

3. C 语言与汇编语言并存

为增加单片机应用系统程序的可读性和可移植性，单片机编程也从传统的汇编语言编程转向 C 语言编程。为适应这一形势的需要，本书第 3 章和第 5 章的例子全部采用汇编语言编程，第 4 章和第 6 章采用汇编语言和 C 语言两种方式编写程序，第 7 章至第 9 章主要采用 C 语言编写程序。

4. 淡化原理，注重实用

基本原理、基本实例一直是学习和掌握单片机应用技术的基本要求，本书侧重于实际应用，因此很少讲解相关理论知识，这样避免了知识重复性。为紧随技术的发展，在编写过程中还注重知识的新颖、实用性，因此本书中讲解了 SPI 总线、I²C 总线、1-Wire 总线及 RS-485 总线芯片的使用方法，使读者学习的知识能够紧随时代的发展。

由于 Proteus 软件的操作方式与一般的应用软件不同，菜单栏与工具条也不完全对应，因此本书只列出了与菜单栏相对应的一些工具条，对于与菜单栏不对应的工具条，请读者参考相关资料。为了便于读者学习和使用实际的 Proteus 软件，本书对电路图中不符合国家标准的图形及符号等未做改动（例如，在 Proteus 软件中，电容单位 μF 使用 uF 进行标记）。

参加本书编写的有侯玉宝、陈忠平、李成群、高见芳、龚亮、龙晓庆等。全书由周少华教授主审，在编写过程中还得到了李锐敏、陈建忠等众位高工、老师的大力支持及帮助，在此向他们表示衷心的感谢。由于编者知识水平和经验的局限性，书中难免存在缺点和错误，敬请广大读者给予批评指正。编者 E-mail: czpmcu@126.com

编著者

2008 年 6 月

目 录

第 1 章 80C51 单片机应用系统的设计及相关软件的使用	1
1.1 80C51 单片机应用系统的设计	1
1.2 Keil C51 的使用	1
1.2.1 创建项目	1
1.2.2 调试程序	4
1.3 仿真器	7
1.4 编程器	8
1.5 ISP 下载	9
1.6 串行调试软件	10
第 2 章 Proteus 7.1 入门	11
2.1 Proteus ISIS 的操作及电路原理图设计	11
2.1.1 Proteus ISIS 简介	11
2.1.2 Proteus ISIS 编辑环境及参数设置	11
2.1.3 Proteus ISIS 原理图设计	22
2.1.4 Proteus ISIS 元件制作	27
2.2 Proteus VSM 虚拟系统模型	30
2.2.1 激励源	30
2.2.2 Proteus VSM 虚拟仪器的使用	30
2.3 Proteus ARES 的 PCB 设计	40
2.3.1 Proteus ARES 简介	40
2.3.2 Proteus ARES 参数设置	44
2.3.3 Proteus ARES 中的 PCB 制作实例	45
第 3 章 51 系列软件程序设计与仿真	50
3.1 清零、置位程序的设计与仿真	50
3.1.1 片内清零程序的设计	50
3.1.2 片内清零程序的调试与仿真	50
3.1.3 片外清零程序的设计	52
3.1.4 片外清零程序的调试与仿真	52
3.1.5 置位程序的设计	53
3.1.6 置位程序的调试与仿真	53
3.2 拼字程序的设计与仿真	55
3.2.1 片内拼字程序的设计	55
3.2.2 片内拼字程序的调试与仿真	55
3.2.3 片外拼字程序的设计	57
3.2.4 片外拼字程序的调试与仿真	57

3.3	拆字程序的设计与仿真	58
3.3.1	片内拆字程序的设计	58
3.3.2	片内拆字程序的调试与仿真	58
3.3.3	片外拆字程序的设计	60
3.3.4	片外拆字程序的调试与仿真	60
3.4	数据块传送程序的设计与仿真	61
3.4.1	数据块传送程序的设计	61
3.4.2	数据块传送程序的调试与仿真	62
3.5	数据排序程序的设计与仿真	63
3.5.1	数据排序程序的设计	63
3.5.2	数据排序程序的调试与仿真	64
第 4 章	51 系列通用 I/O 控制	66
4.1	P1 口的应用 (一)	66
4.1.1	硬件设计	66
4.1.2	程序设计	66
4.1.3	调试与仿真	67
4.2	P1 口的应用 (二)	69
4.2.1	硬件设计	69
4.2.2	程序设计	69
4.2.3	调试与仿真	71
4.3	闪烁灯	72
4.3.1	硬件设计	72
4.3.2	程序设计	73
4.3.3	调试与仿真	75
4.4	流水灯	77
4.4.1	硬件设计	77
4.4.2	程序设计	77
4.4.3	调试与仿真	78
4.5	花样灯 (一)	79
4.5.1	硬件设计	79
4.5.2	程序设计	80
4.5.3	调试与仿真	82
4.6	花样灯 (二)	83
4.6.1	硬件设计	83
4.6.2	程序设计	84
4.6.3	调试与仿真	87
4.7	模拟交通灯	88
4.7.1	硬件设计	89
4.7.2	程序设计	89
4.7.3	调试与仿真	91
4.8	定时/计数器的应用 (一)	92

4.8.1 硬件设计	92
4.8.2 程序设计	92
4.8.3 调试与仿真	94
4.9 定时/计数器的应用(二)	95
4.9.1 硬件设计	95
4.9.2 程序设计	95
4.9.3 调试与仿真	97
4.10 中断系统的应用(一)	98
4.10.1 硬件设计	98
4.10.2 程序设计	98
4.10.3 调试与仿真	101
4.11 中断系统的应用(二)	102
4.11.1 硬件设计	102
4.11.2 程序设计	102
4.11.3 调试与仿真	105
4.12 两个单片机串行通信	106
4.12.1 硬件设计	106
4.12.2 程序设计	108
4.12.3 调试与仿真	110
4.13 串行口扩展应用	111
4.13.1 硬件设计	111
4.13.2 程序设计	112
4.13.3 调试与仿真	113
第5章 音乐的应用	115
5.1 单片机唱歌	115
5.1.1 单片机产生音调的基础知识	115
5.1.2 音乐软件的设计	117
5.1.3 歌曲的设计	117
5.1.4 调试与仿真	120
5.2 电子琴	122
5.2.1 电子琴的基础知识	122
5.2.2 电子琴软件的设计	122
5.2.3 调试与仿真	127
第6章 LED数码管与键盘的应用	130
6.1 LED数码管的应用	130
6.1.1 LED数码管的结构及分类	130
6.1.2 LED数码管的显示方式	131
6.1.3 串行口驱动1位LED数码管的设计	131
6.1.4 共阴极LED和共阳极LED的应用	135
6.1.5 0~99计数器的设计	138
6.1.6 59s计时的设计	144

6.1.7	电子钟的设计	148
6.1.8	MAX7219 串行驱动 LED 数码管	151
6.2	键盘的应用	161
6.2.1	键盘的工作原理	161
6.2.2	查询式键盘的设计	162
6.2.3	矩阵式键盘的识别 (一)	166
6.2.4	矩阵式键盘的识别 (二)	171
6.2.5	8255A 并行 I/O 端口扩充键盘	177
第 7 章	数/模转换器和模/数转换器的应用	182
7.1	数/模转换器的应用	182
7.1.1	TLC5615 的基础知识	182
7.1.2	方波发生器	184
7.1.3	锯齿波发生器	187
7.2	模/数转换器的应用	190
7.2.1	ADC0808 的基础知识	190
7.2.2	数字电压表的设计	191
第 8 章	显示器的应用	196
8.1	LED 点阵显示器的应用	196
8.1.1	LED 点阵显示器的基础知识	196
8.1.2	一个 5×7 点阵字符显示	196
8.1.3	一个 8×8 点阵字符串显示	199
8.1.4	两个 8×8 点阵字符串显示	204
8.1.5	两个 8×8 点阵滚动显示	211
8.1.6	一个 16×16 点阵汉字显示	214
8.1.7	两个 16×16 点阵汉字显示	220
8.1.8	两个 16×16 点阵汉字分批显示	225
8.2	LCD (液晶显示器) 的应用	228
8.2.1	LCD 的基础知识	228
8.2.2	字符式 LCD 的应用	230
8.2.3	汉字式 LCD 的应用	235
8.2.4	汉字式 LCD 移位显示	253
8.2.5	汉字式 LCD 滚动显示	264
第 9 章	工业控制	270
9.1	SPI 总线 DS1302 实时时钟控制	270
9.1.1	DS1302 的基础知识	270
9.1.2	DS1302 采用 1 位 LED 显示时钟的设计	271
9.1.3	DS1302 采用 MAX7219 控制 8 位 LED 显示时钟的设计	278
9.2	I ² C 总线 24C04 开启次数统计控制	288
9.2.1	24CXX 的基础知识	289
9.2.2	24C04 开启次数统计的设计	290

9.3 RS-485 在单片机多机通信中的应用	297
9.3.1 RS-485 接口标准简述	297
9.3.2 RS-485 在单片机多机通信中的应用设计	298
9.4 1-Wire Bus DS18B20 温度测量的设计	304
9.4.1 DS18B20 的基础知识	304
9.4.2 DS18B20 测量温度的设计	307
9.5 电动机转速控制	313
9.5.1 步进电动机转速控制	313
9.5.2 直流电动机转速控制	315
9.6 电气模拟控制	319
9.6.1 步进电动机的启动、停止控制	319
9.6.2 直流电动机的启动、停止控制	321
9.6.3 步进电动机的正、反转控制	323
9.6.4 直流电动机的正、反转控制	326
9.6.5 电动机的多地控制	328
附录 A 单片机指令速查表	331
附录 B Proteus 的常用快捷键	335
参考文献	336

第 1 章 80C51 单片机应用系统的设计及 相关软件的使用

单片机应用系统以单片机为核心，同时配以相应的外围电路及软件来完成某种或几种功能的系统。它包括硬件和软件两部分，硬件是系统的躯体，软件是系统的灵魂。

1.1 80C51 单片机应用系统的设计

单片机应用系统的设计从两方面入手：硬件电路的设计和软件系统的设计。

在设计单片机应用系统的硬件前，先明确开发内容及技术要求；然后，确定系统的总体设计方案及相应的功能模块，如信号测量、控制、人机交互接口、通信功能模块等；再根据功能的需求，选择合适的单片机、对关键元件选型；最后，制作相应的印制电路板。


在设计硬件电路时，要考虑元件的驱动及带负载能力，有时还要考虑系统的扩展性及通用性。在制作印制电路板时，要考虑模拟电路、数字电路、高频电路、低频电路、高压电路、低压电路的布线规则、方法，印制电路板导线宽度及所能承受的电压、电流、抗干扰能力等。

单片机应用系统的软件设计，要充分考虑硬件系统。软件编程时，使用汇编语言或单片机 C 语言书写源程序。编写好源程序后，对其进行调试、仿真并生成.HEX 文件。将生成的.HEX 文件烧写到单片机中或相应的程序存储器中。最后，将目标芯片插入电路板进行脱机运行，从而完成单片机系统的设计。

1.2 Keil C51 的使用

在哪里编写单片机的源程序？在哪里将其调试并生成.HEX 文件？这些工作可用单片机的一些编译软件完成。单片机程序的编译调试软件比较多，如 51 汇编集成开发环境、伟福仿真软件、Keil 单片机开发系统等。

Keil C51 是当前使用最广泛的基于 80C51 单片机内核的软件开发平台之一，由德国 Keil Software 公司推出。 μ Vision2 是 Keil Software 公司推出的 51 系列单片机开发工具， μ Vision2 集成开发环境 IDE 是一个基于 Windows 的软件开发平台，集编辑、编译、仿真于一体，支持汇编语言和 C 语言的程序设计。一般来说，Keil 51 和 μ Vision2 指的是 μ Vision2 集成开发环境。

Keil C51 可以从相关网站下载并安装。安装好后，双击桌面上快捷图标或在“开始”菜单中选择 Keil μ Vision2，启动 Keil μ Vision2 集成开发环境，启动后的画面如图 1-1 所示。

1.2.1 创建项目


Keil μ Vision2 中有 1 个项目管理器，用于对项目文件进行管理。它包含了程序的环境变量和编辑有关的全部信息，为单片机程序的管理带来了很大的方便。创建 1 个新项目的操作步骤如下：

- (1) 启动 μ Vision2，创建 1 个项目文件，并从器件数据库中选择 1 款合适的 CPU。

- (2) 创建 1 个新的源程序文件，并把这个源文件添加到项目中。
- (3) 为该单片机芯片添加或配置启动程序代码。
- (4) 设置工具选项，使之适合目标硬件。
- (5) 编译项目并创造 1 个可供 PROM 编程的 HEX 文件。

下面分别介绍每一步的具体操作。

1. 启动 μ Vision2 并创建 1 个项目文件

μ Vision2 是 1 个标准的 Windows 应用程序，直接在桌面上双击  就可启动它。单击 μ Vision2 菜单中的“Project”选项，在下拉菜单中单击“New Project”选项，弹出“Create New Project”对话框，要求键入项目名称。建议每个项目使用 1 个独立的文件夹。

键入新建项目名，单击“确定”按钮后，弹出如图 1-2 所示的“Select Device for Target 'Target 1'”对话框。在此对话框中，根据需要选择合适的单片机型号。在“Project”下拉菜单中，单击“Select Device for Target”选项也会弹出如图 1-2 所示的对话框。

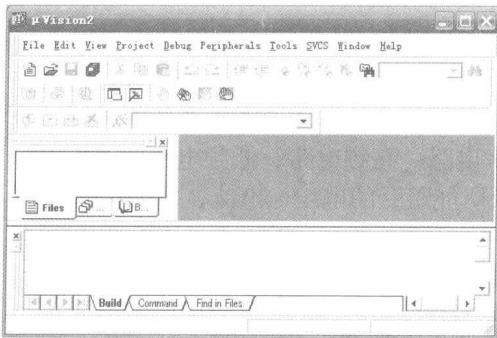


图 1-1 启动 Keil μ Vision2 后的画面

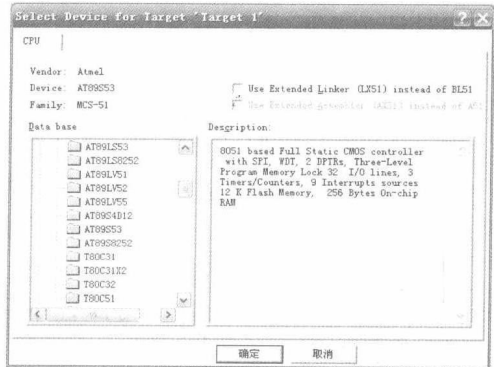



图 1-2 “Select Device for Target 'Target 1'”对话框

在图 1-2 的“Data base”栏中列出了各厂商名及其产品，“Description”栏是对选中的单片机的指标说明。

2. 创建新的源程序文件

使用图标  或在“File”菜单中选择“New”命令就可以创建 1 个源程序文件。该命令会打开 1 个空的编辑器窗口，在该窗口中键入源代码，如图 1-3 所示。用汇编语言或单片机 C 语言键入源代码后，在“File”菜单中选择“Save as”或“Save”命令对源程序进行保存。在保存时，文件名可以是字符、字母或数字，并且一定要带扩展名（使用汇编语言编写的源程序，扩展名为.A51 或.ASM；使用单片机 C 语言编写的源程序，扩展名为.C）。保存好源程序后，源程序窗口中的关键字呈彩色高亮度显示。

源程序文件创建好之后，可以把这个文件添加到项目中。在 μ Vision2 中，添加的方法有多种，例如在图 1-1 中左边“Project”的窗口“File”页上单击文件组，然后单击鼠标右键，弹出如图 1-4 所示的文件组快捷菜单，选中“Add Files to Group 'Source Group 1'”选项，在弹出的对话框中选择刚才创建的源程序文件即可。

3. 添加并配置启动程序代码

startup.a51 文件是大部分 8051CPU 及其派生产品的启动程序，启动程序的操作包括清除数据存储内容、初始化硬件及可重入堆栈指针。一些 8051 派生的 CPU 需要初始化代码以使配置符合硬件上的设计。例如，Philips 的 8051RD+片内 xdata RAM 需通过在启动程序

中的设置才能使用。按照目标硬件的要求来创建相应的 startup.a51 文件，或直接将它从安装路径的\C51\LIB 文件夹中复制到项目文件中并按照需要进行更改。

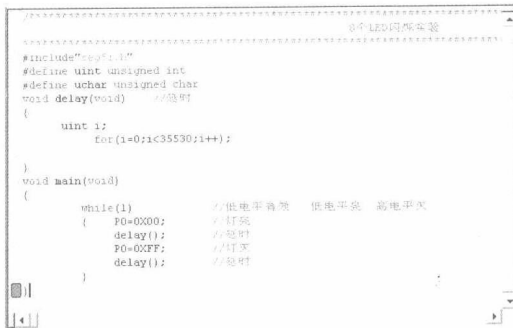


图 1-3 编辑器窗口

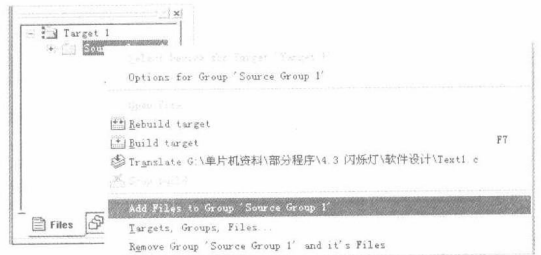


图 1-4 文件组快捷菜单

为 CPU 配置文件也可以通过一个新文件组来实现。在“Project”下拉菜单中，选择“Targets, Groups, Files”命令，打开如图 1-5 所示的对话框，在对话框的“Group to Add”栏中键入“System Files”后，单击“Add”按钮，将在“Available Groups”栏中添加了“System Files”项。选中该项后，单击“Add Files to Group”按钮，弹出“Add Files to Group 'System Files'”对话框，将 startup.a51 文件添加到项目中。

在项目窗口的“File”页中添加了“System Files”，单击“STARTUP.A51”后，就可以在编辑器窗口中按照“CPU AND C STARTUP CODE”所述的方式编辑启动代码，如图 1-6 所示。使用片内 RAM 时，启动代码中的内容必须与“Options for Target 'Target 1'”对话框中的“Target”页中的内容相符。

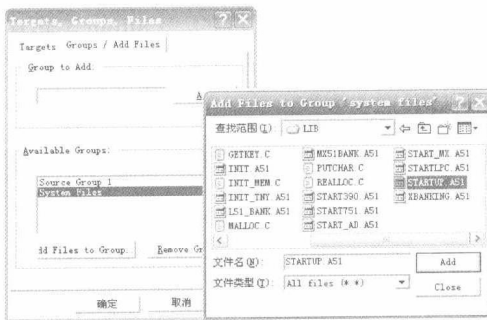


图 1-5 将 System Files 组添加到目标项目中

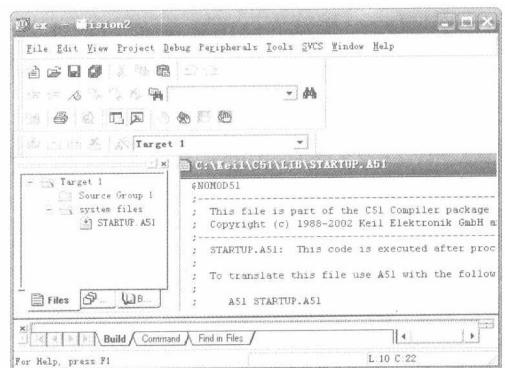



图 1-6 编辑启动代码

4. 为目标设定工具选项

单击图标或在“Project”下拉菜单中选择“Options for Target”命令，将会出现“Options for Target 'Target 1'”对话框，如图 1-7 所示。在“Target”页中可以对目标硬件及所选器件片内部件进行参数设定，表 1-1 描述了“Target”对话框的选项。

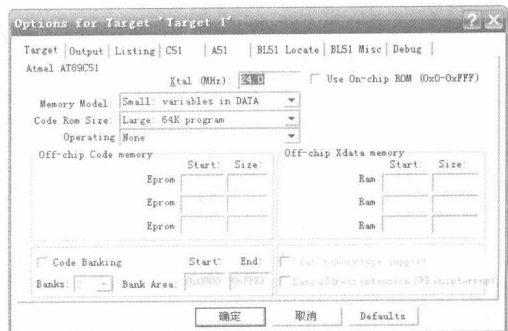



图 1-7 为目标设定工具选项

表 1-1 对话框的选项说明

对话框项	说 明
Xtal	指定器件的 CPU 时钟频率，在多数情况下，它的值与 Xtal 的频率相同
Use On-chip ROM	使用片上自带的 ROM 作为程序存储器
Memory Model	指定 C51 编译器的存储模式，在开始编辑新应用时，默认 SMALL
Code Rom Size	指定 ROM 存储器的大小
Use On-chip XRAM	使用片上自带的 XRAM 存储器
Off-chip Code memory	指定目标硬件上所有外部地址存储器的地址范围
Off-chip Xdata memory	指定目标硬件上所有外部数据存储器的地址范围
Code Banking	指定 Code Banking 参数

标准的 80C51 的程序存储器空间为 64KB，若程序空间超过 64KB 时，可在“Target”页中对“Code Banking”栏进行设置。Code Banking 为地址复用，可以扩展现有的 CPU 程序存储器寻址空间。选中“Code Banking”复选框，用户根据需求在“Banks”栏中选择合适的块数。在 Keil C51 中，用户最多能使用 32 块 64KB 的程序存储空间，即 2MB 的空间。

5. 编译项目并创建 .HEX 文件

在“Target”页中设置好工具后，就可对源程序进行编译。单击图标或在“Project”下拉菜单中，选择“Build Target”命令，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， μ Vision2 将会在输出窗口（Output Window）的编译页（Build）中显示错误和警告信息，如图 1-8 所示。双击某一条信息，光标将会停留在 μ Vision2 文本编辑窗口中出现该错误或警告的源程序位置上。

若成功创建并编译了应用程序，就可以开始调试。当程序调试好之后，要求创建一个 .HEX 文件，生成的 .HEX 文件可以下载到 EPROM 编程器或模拟器中。

若要创建 .HEX 文件，必须将“Options for Target 'Target 1'”对话框的“Output”页中的“Create HEX File”复选框选中，如图 1-9 所示。在“Run User Program #1”中指定 PROM 编程工具后，可以在 Make 处理后启动 PROM 编程工具。

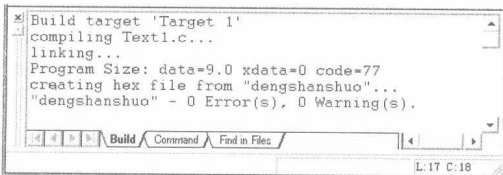


图 1-8 错误和警告信息

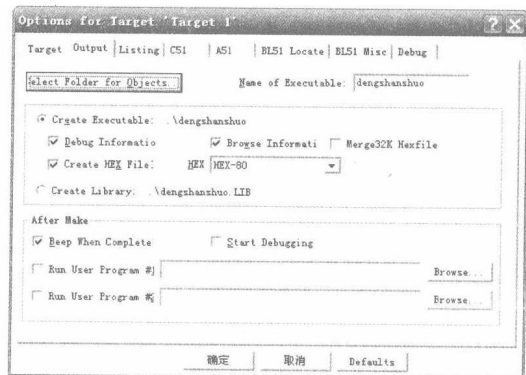


图 1-9 选中“Create HEX File”复选框

1.2.2 调试程序

使用 μ Vision2 调试器可对源程序进行测试， μ Vision2 提供了两种操作工作模式，这两种模式可以在“Option for Target 'Target 1'”对话框的“Debug”页中选择，如图 1-10 所示。

“Use Simulator”：软件仿真模式，将 μ Vision2 调试器配置成纯软件产品，能够仿真

8051 系列产品的绝大多数功能而不需要任何硬件目标板，如串行口、外部 I/O 和定时器等，这些外围部件设置是在器件数据库选择 CPU 时选定的。

“Use”：硬件仿真。例如 TKS Debugger，用户可以直接把这个环境与仿真程序或 Keil 监控程序相连。

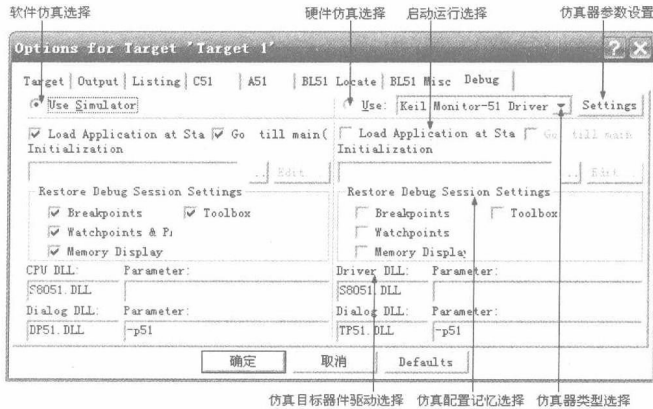



图 1-10 仿真设置

1. CPU 仿真


μ Vision2 仿真器可以模拟 16MB 的存储器，该存储器被映射为读、写或代码执行访问区域。除了将存储器映射外，仿真器还支持各种 80C51 派生产品的集成外围器件。在“Debug”页中可以选择和显示片内外围部件，也可通过设置对话框的内容来改变各种外设的值。

2. 启动调试

源程序编译好后，选择相应的仿真操作模式，可启动源程序的调试。单击图标或在“Debug”下拉菜单中，选择“Star/Stop Debug Session”命令可启动 μ Vision2 的调试模式，调试界面如图 1-11 所示。

3. 断点的设定

在编辑源程序的过程中或程序尚未编译前，用户可以设置执行断点。在 μ Vision2 中，可用以下不同的方法来定义断点：

(1) 在文本编辑框中或反汇编窗口中选定所在行，然后单击“File Toolbar”断点按钮或单击图标。


(2) 在文本编辑窗口或反汇编窗口中单击右键，打开快捷菜单进行断点设置。

(3) 利用“Debug”下拉菜单，打开“Breakpoint”对话框，在这个对话框中可以查看定义或更改断点的设置。

(4) 在“Output Window”窗口的“Command”页可以使用 BreakSet、BreakKill、BreakList、BreakEnable 和 BreakDisable 命令选项。

4. 目标程序的执行

目标程序的执行可使用以下方法：

(1) 在“Debug”下拉菜单中，单击“GO”命令或直接单击图标。

(2) 在文本编辑窗口或反汇编窗口中单击右键，在弹出的快捷菜单上选择“Run till Cursor line”命令。

(3) 在“Output Window”窗口的“Command”页中可以使用 Go、Ostep、Pstep、Tstep 命令。

5. 反汇编窗口

在进行程序调试及分析时，经常会用到反汇编。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，如图 1-12 所示。

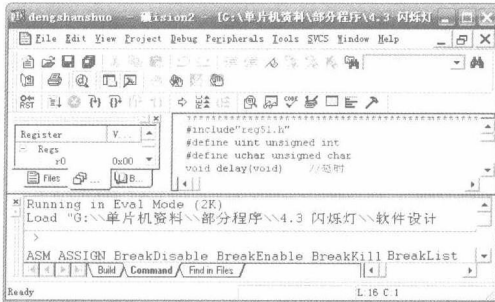


图 1-11 调试界面

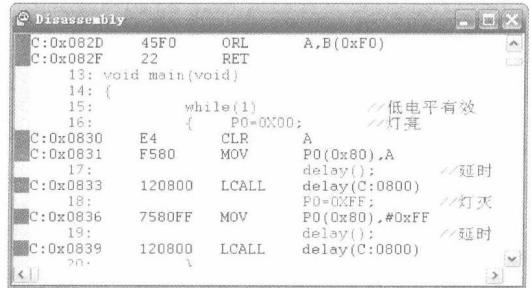


图 1-12 反汇编窗口

已经执行了的指令都可以在单击“Debug”下拉菜单中的“View Trance Records”选项后显示。要想有历史记录，在“Debug”下拉菜单中选择“Enable/Disable Trance Records”选项即可。

当反汇编窗口作为当前活动窗口时，若单步执行指令，所有的程序将按照 CPU 指令，即汇编来单步执行，而不是 C 语言的单步执行。

6. CPU 寄存器窗口

在“Debug”下拉菜单中，单击“Start/Stop Debug Session”选项后，在“Project Windows”窗口的“Page”页中会显示 CPU 寄存器内存，CPU 窗口如图 1-13 所示。

7. 存储器窗口

在存储器窗口中，最多可以通过 4 个不同的页观察 4 个不同的存储区，每页都能显示存储器中的内容，存储器窗口如图 1-14 所示。

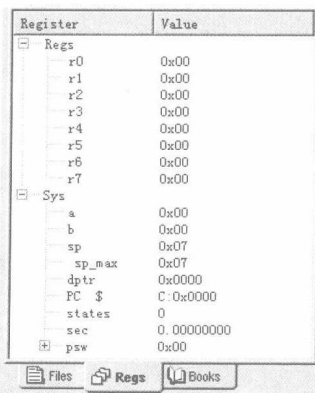


图 1-13 CPU 窗口

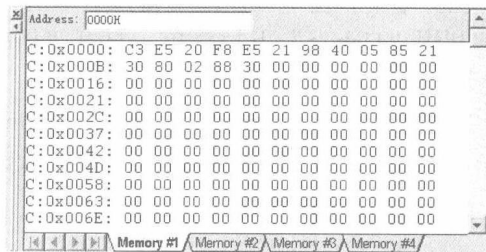


图 1-14 存储器窗口

在“Address”栏中输入地址值后，显示区域直接显示该地址的内容。若要更改地址中的内容，只需在该地址上双击鼠标并输入新的内容。

8. 串行窗口

μVision2 提供了 2 个专门用于串行调试输入和输出的窗口，被模拟仿真的 CPU 串行口数据输出将在该窗口中显示，输入串行窗口中的字符将会被输入到模拟的 CPU 中。

在“View”下拉菜单中，选择“Serial Window #1”或“Serial Window #2”命令即可打开串行调试窗口。

1.3 仿真器

编写好源程序后，进行程序调试时需对其进行仿真。仿真是使用可控的手段来模仿真实的情况，在单片机嵌入式系统的设计中，仿真主要是指对程序的仿真。仿真分为两类：软件仿真和硬件仿真。

软件仿真是使用计算机软件来模拟单片机的实际运行，而用户不需要搭建硬件电路就可以对程序进行验证，但是软件仿真无法完全仿真与硬件相关的部分。硬件仿真是使用附加的硬件来替代用户系统的单片机并完成单片机全部或大部分的功能。它能直接反映单片机的全部或部分实际运行控制功能，是开发过程中所必需的。

能进行硬件仿真的器件称为仿真器，常用的仿真器有南京伟福公司生产的伟福仿真器和广州周立功公司生产的 TKS 系列仿真器。TKS 系列仿真器在硬件上采用了 HOOKS/Bondout 仿真技术，可以实时在线仿真 Philips 公司生产的 80C51 系列单片机及 Atmel、Winbond、HY 等公司生产的兼容 MCS-51 内核的标准 80C51 单片机。TKS 仿真器的外形如图 1-15 所示。

TKS 系列仿真器除了可以使用本身自带的仿真软件外，还可以嵌入到 Keil C51 中进行硬件调试仿真，其加载方法是：

(1) 将随机提供的 TKS 仿真器驱动文件 TKS_DEB.DLL 复制到 Keil 的安装目录 C51\Bin 下。

(2) 打开 Keil 安装目录下的 Took.ini 文件，在几个分类中找到[C51]，并加入语句“TDRV3=C:\Keil\C51\bin\TKS_DEB.dll (“TKS Debugger”)”。

TDRV3 是驱动 DLL 的序号，其数值可改变。C:\Keil\C51\bin 为安装目录，在此假设 Keil 软件安装在 C 盘根目录下。

将 TKS_DEB.DLL 加载到 Keil C51 中后，在图 1-10 中先选中“Use (硬件仿真选择)”单选钮，并且选中合适的仿真器类型，单击“Settings (仿真器参数设置)”单选钮，将弹出如图 1-16 所示的窗口。在该窗口中，进行相应的仿真硬件配置。

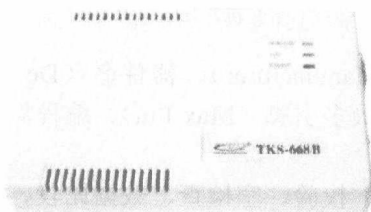


图 1-15 TKS 仿真器的外形

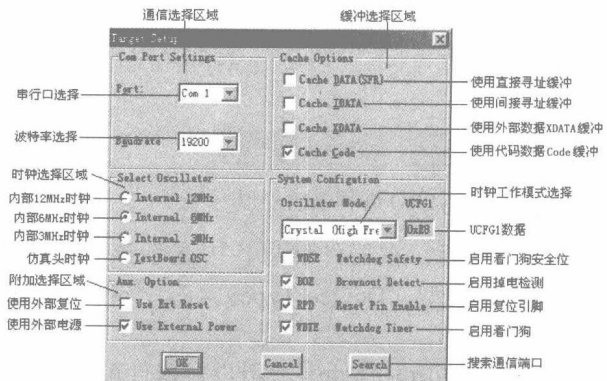


图 1-16 仿真硬件配置