



电气自动化技能型人才系列

DIANQIZIDONGHUA
JINENGXINGRENCAL
XILIE

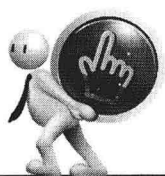
C51单片机

应用实训

主 编 陈 雷
副主编 陈 爽



中国电力出版社
CHINA ELECTRIC POWER PRESS



电气自动化技能型人才系列

C51单片机

应用实训

主 编 陈 雷

副主编 陈 爽

参 编 王冬云 韩宝如



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书是《电气自动化技能型人才系列》中的一本。

本书分两篇。第1篇为基础知识,共3章,主要介绍了 Keil μ Vision3 使用快速入门、Proteus ISIS 使用快速入门和 51 系列单片机内部资源编程,旨在使读者掌握 C51 单片机设计应用的基础知识。第2篇常用模块设计实例,共8章,分别介绍了显示模块、A/D 及 D/A 转换、矩阵键盘、日历时钟、E²PROM 存储器 AT24C02、单总线数字温度传感器 DS18B20、电动机控制技术和红外遥控技术的原理及设计应用实例。所有这些内容都给出了完整的电路图和 C 语言程序代码,部分内容给出了汇编语言程序代码。

本书可作为高等院校电子信息、自动化等相关专业学生学习单片机技术的实践型教材或培训教材,也可供相关专业学生课程设计、毕业设计参考,还可以供单片机开发技术人员及爱好者阅读参考。

图书在版编目(CIP)数据

C51 单片机应用实训/陈雷主编. —北京:中国电力出版社, 2011.5

(电气自动化技能型人才系列)

ISBN 978-7-5123-1657-7

I. ①C… II. ①陈… III. ①单片微型计算机
IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2011)第 081386 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

航远印刷有限公司印制

各地新华书店经售

*

2011 年 7 月第一版 2011 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 11 印张 289 千字

印数 0001—3000 册 定价 24.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前言

随着技术的进步,各种智能化的仪器仪表以及智能控制系统在生产和生活中得到了广泛的应用,而单片机在其中起到了举足轻重的作用。各种单片机应用系统的设计与开发,自 20 世纪 90 年代以来就一直是技术的热点,社会对相关人才的需求也一直十分迫切,很多学校的相关专业也都开设了单片机课程。针对单片机课程实践性很强的特点,我们在课程设置方面专门增加了“单片机应用训练”,目的是进一步提高学生的单片机应用能力,培养设计能力和分析、解决问题的能力。为了更好地进行单片机训练课程的教学,达到培养技能型人才的目标,编写了本教材。

本书主要在 Keil μ Vision3 开发软件和 Proteus ISIS 仿真平台的基础上讲解 51 系列单片机开发的相关知识,包括硬件电路的设计和软件程序的编写。在读者掌握 Keil 和 Proteus 软件以及 51 系列单片机内部资源开发知识的基础上,以专题训练的形式讲解了 51 系列单片机常用外围芯片及器件的接口电路设计和程序设计,每一专题都列举了大量实例,同时给出了完整的电路和程序,并对程序进行了详细的注释,所有实例程序都经过 Keil μ Vision3 编译成功和 Proteus 仿真通过,并且在硬件电路上实际运行成功。此外,本书每一节后都给出了相应的练习题目,希望读者能即学即用,动脑思考,促使读者举一反三,让所学知识得以进一步巩固,同时满足教师教学和学生自学的需要。

本书涉及到的程序源代码和仿真电路电子版以及相关工具软件和芯片资料可发电子邮件 lianxiangqhd@126.com 与作者联系索取。

本书第 1 章由秦皇岛职业技术学院王冬云、河北科技师范学院数信学院陈爽合编,第 2 章由海南软件职业技术学院韩宝如编写,第 3~11 章及附录由东北石油大学秦皇岛分校陈雷编写。

限于编者水平和时间,书中内容难免有错误和不足之处,敬请广大读者批评指正或提出修改意见。

编 者

2011 年 6 月

目 录

前言

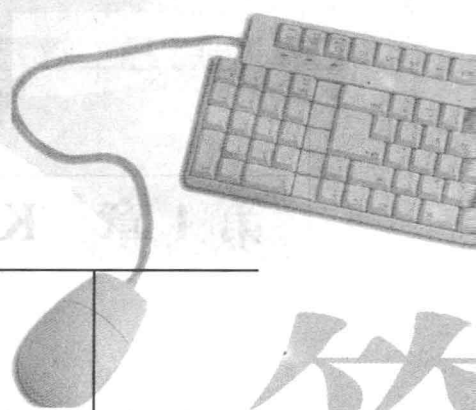
第1篇 基础知识

第1章	Keil μVision3 使用快速入门	2
1.1	建立工程	2
1.1.1	新建工程	2
1.1.2	添加文件	3
1.2	配置工程	4
1.2.1	Target 选项卡的配置	4
1.2.2	Output 选项卡的配置	6
1.2.3	Debug 选项卡的配置	7
1.3	编译链接	7
1.4	程序调试	7
1.4.1	单步执行调试	8
1.4.2	存储器窗口的使用	8
1.4.3	观察和调用堆栈窗口的使用	9
1.4.4	基于 Peripherals 菜单的仿真调试	9
第2章	Proteus ISIS 使用快速入门	12
2.1	Proteus ISIS 的工作环境	12
2.1.1	编辑窗口	13
2.1.2	绘图工具栏	13
2.1.3	仿真进程控制	13
2.1.4	预览对象方位控制	14
2.1.5	预览窗口	14
2.1.6	器件选择器窗口	14
2.2	Proteus ISIS 的单片机仿真	14
2.2.1	绘制原理图	14
2.2.2	加载目标程序	16
2.2.3	仿真调试	16
2.3	Proteus ISIS 与 Keil 的连接调试	18

第3章	51 系列单片机内部资源编程	20
3.1	延时程序的编写	20
3.1.1	常用延时程序的编写	20
3.1.2	Keil 中延时程序的调试	22
3.2	并行 I/O 口	22
3.2.1	I/O 口驱动蜂鸣器	23
3.2.2	I/O 口驱动电磁继电器	23
3.2.3	51 系列单片机 I/O 口程序设计实例	24
3.3	外中断	28
3.4	定时器/计数器	30
3.4.1	定时器/计数器的工作方式	31
3.4.2	定时器/计数器的初始化及定时初值的计算方法	31
3.4.3	定时器/计数器的应用	32
3.5	串行通信	38
3.5.1	串行通信的基本知识概述	38
3.5.2	51 单片机串行口通信原理	39
3.5.3	51 单片机串行口通信程序设计实例	41
3.5.4	基于 Proteus 的串行口通信仿真方法	45
 第2篇 常用模块设计实例		
第4章	显示模块设计实例	50
4.1	液晶显示应用技术	50
4.1.1	字符液晶显示器 LCD1602 的应用	50
4.1.2	带汉字库的液晶显示器 LCD12864 的应用	57
4.2	数码管显示应用技术	72
4.2.1	数码管的基本工作原理	72
4.2.2	数码管动态扫描显示原理	73
4.2.3	数码管动态扫描电路设计方案与程序设计实例	74
第5章	A/D 及 D/A 转换设计实例	82
5.1	A/D 转换器的应用技术	82
5.1.1	A/D 转换的基本知识	82
5.1.2	A/D 转换器的应用实例	84
5.2	D/A 转换器的应用技术	99
5.2.1	D/A 转换的基本知识	99
5.2.2	D/A 转换器的应用实例	100
第6章	矩阵键盘设计应用实例	104
第7章	日历时钟设计应用实例	107
7.1	DS1302 简介	107

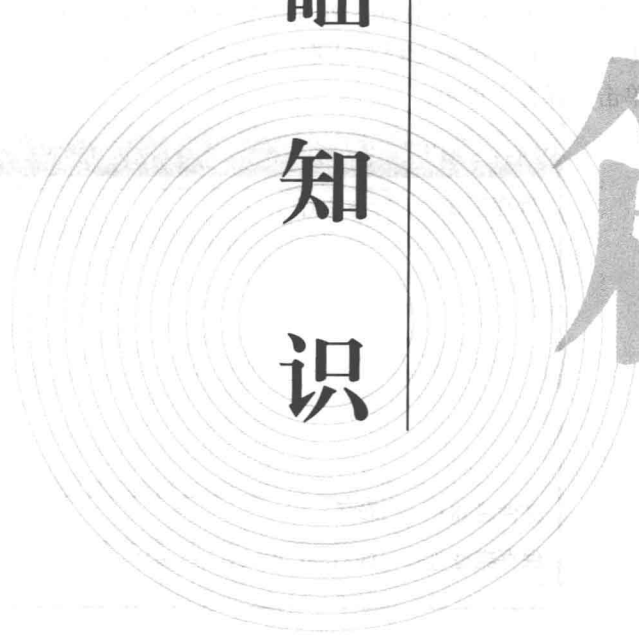
7.2	DS1302 的控制方法	108
7.3	DS1302 的程序设计实例	111
7.3.1	DS1302 的驱动程序设计	111
7.3.2	基于 DS1302 的电子钟设计	113
第 8 章	E²PROM 存储器 AT24C02 的设计应用实例	117
8.1	AT24Cxx 系列芯片简介	117
8.2	IIC 总线时序与驱动程序	118
8.3	AT24C02 的读写时序及程序设计	120
8.4	Proteus 中对 AT24C02 的仿真调试	125
第 9 章	单总线数字温度传感器 DS18B20 的设计应用实例	128
9.1	DS18B20 概述	128
9.2	DS18B20 的内部结构	128
9.3	DS18B20 的操作命令	131
9.4	DS18B20 的单总线协议及驱动程序	132
9.5	DS18B20 的数据处理方法	134
9.6	DS18B20 的应用实例	135
9.6.1	配置 DS18B20	135
9.6.2	单个 DS18B20 测温	136
9.6.3	读取 DS18B20 序列号	138
9.6.4	用 DS18B20 实现多点测温	139
第 10 章	电动机控制技术	141
10.1	直流电动机的控制	141
10.1.1	直流电动机简介	141
10.1.2	直流电动机正反转控制方法	141
10.1.3	直流电动机的转速控制方法	141
10.1.4	常用直流电动机的驱动电路及控制程序设计	142
10.2	步进电动机的控制	149
10.2.1	步进电动机的基本知识	149
10.2.2	四相五线步进电动机 28BYJ-48 的驱动	151
第 11 章	红外遥控技术	157
11.1	红外遥控发射及其编码	157
11.2	红外遥控接收及其解码	158
11.3	红外遥控解码设计实例	159
附录		164
附录 A	51 系列单片机几个常用的特殊功能寄存器	164
附录 B	C51 单片机中断编程注意事项	166
参考文献		167

● 电气自动化技能型人才系列
C51 单片机应用实训



第1篇

基础知识



第 1 章 Keil μ Vision3 使用快速入门

Keil 软件是基于 Windows 的 51 系列单片机集成开发平台，是目前最流行的 51 系列单片机开发软件。Keil 软件提供了包括编译器、宏汇编、链接定位、库函数维护和项目管理、软件模拟调试、硬件仿真调试等一系列完整开发方案，提供了丰富的库函数和功能强大的集成开发调试工具。通过 Keil 软件可以对 C 语言程序进行编译，对汇编语言程序进行汇编，链接目标模块与库模块产生目标文件，生成 HEX 文件，其生成的目标代码的效率非常高。掌握该软件的使用方法对学习 51 系列单片机是十分必要的，目前普及使用的版本是 Keil μ Vision3。为使读者快速上手，本章对 Keil μ Vision3 使用过程中常用的主要步骤和关键设置进行介绍。

1.1 建 立 工 程

1.1.1 新建工程

启动 Keil μ Vision3 软件，进入开发界面，单击菜单栏中的“Project→New Project...”，弹出如图 1-1 所示的创建新工程对话框，在对话框上部的下拉列表选择好保存路径，并输入工程名称，单击“保存”即可。

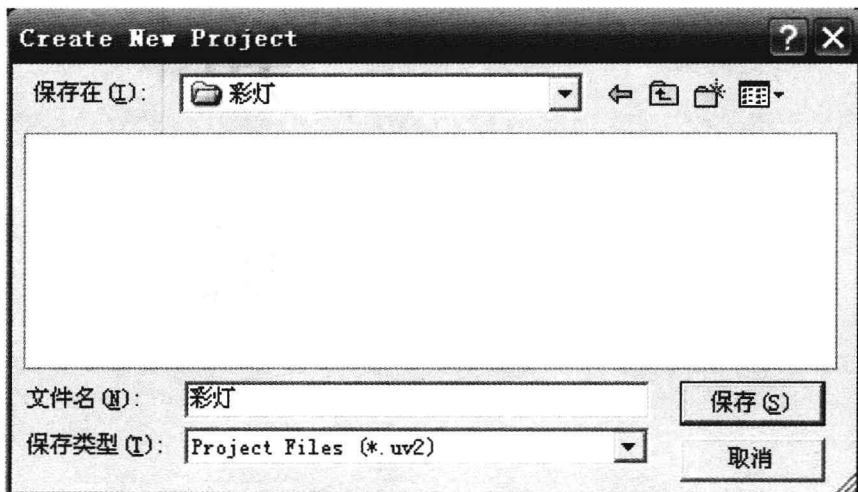


图 1-1 创建新工程对话框

保存后弹出如图 1-2 所示的选择目标器件对话框，Keil 支持几百种基于 51 内核的单片机系列，可以根据需要选择相应型号，这里展开 Data base 栏中的 Atmel，单击选中 AT89S52，在右边的 Description 区域给出了关于该器件的基本硬件资源描述。然后，单击“确定”按钮回到主界面。

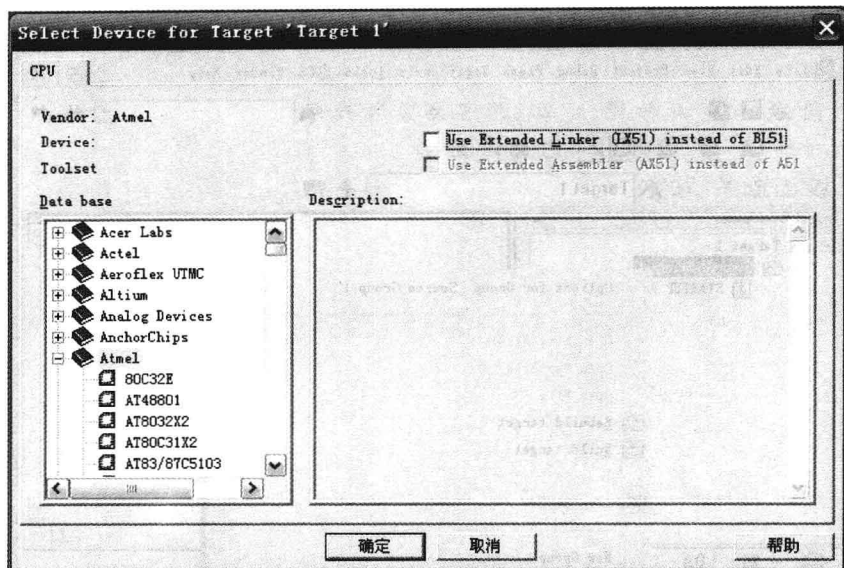


图 1-2 选择目标器件对话框

1.1.2 添加文件

新建工程后，需要为工程添加文件，单击菜单栏中的“File→New...”或单击快捷图标，弹出空白文本文件界面，这时可以新建一个 C 语言或汇编语言源文件。单击菜单“File→Save”或单击弹出如图 1-3 所示的保存文件对话框，输入文件名并单击保存即可。注意：这里必须输入扩展名，若采用汇编语言，扩展名为 .ASM；若采用 C 语言，扩展名为 .C。

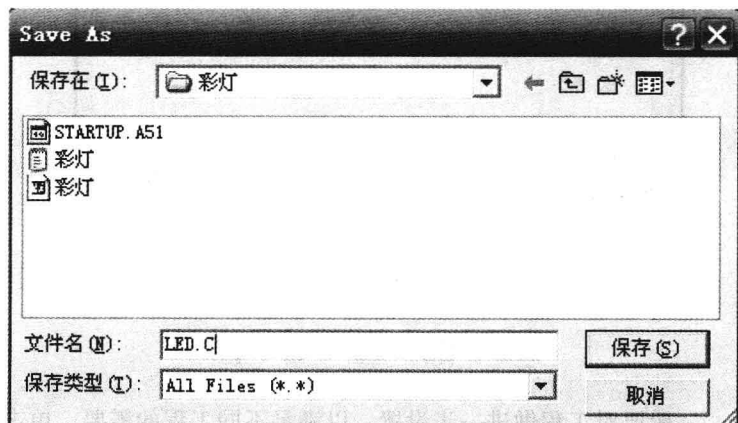


图 1-3 保存文件对话框

这时需要把刚才建立并保存的源文件添加到工程中，如图 1-4 所示，右键单击左侧项目工作区列表框中 Target 1 下的 Source Group 1，在出现的快捷菜单中单击“Add Files to Group ‘Source Group 1’”，出现添加源文件对话框，如图 1-5 所示。Keil 默认加入 C 程序源文件，若加入的文件为其他类型，如汇编语言源文件，则需要在文件类型的下拉列表中选择相应的选项。双击要加入的文件，或单击要加入的文件后单击 Add 按钮，可将该文件加入到工程中，然后单击 Close 关闭该对话框。

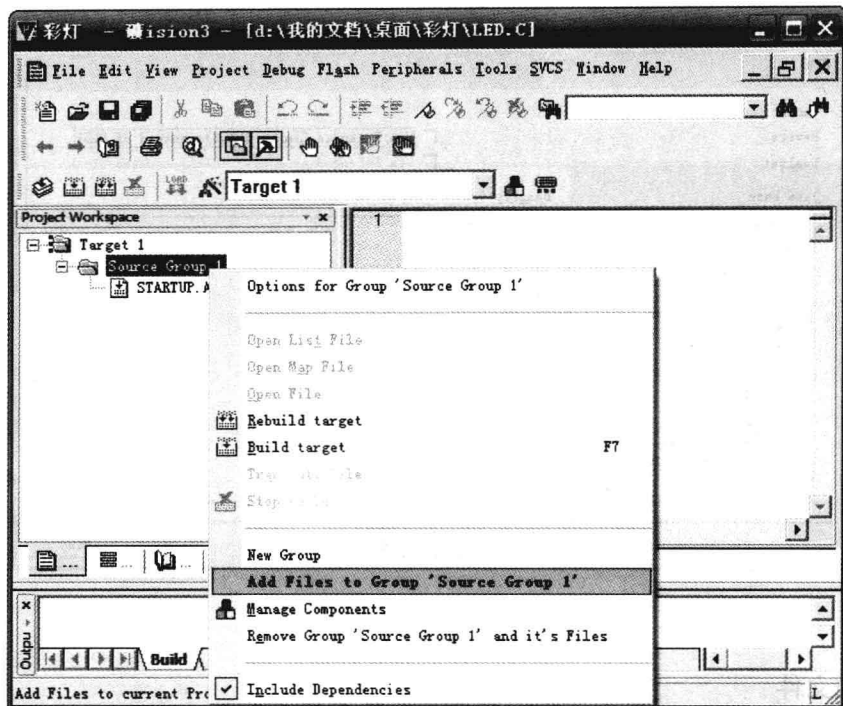


图 1-4 添加源文件到工程

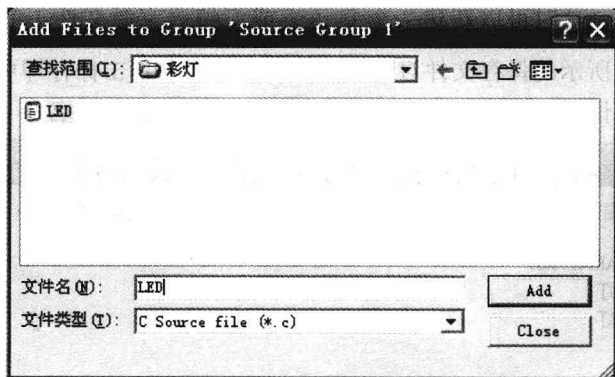


图 1-5 添加源文件对话框

1.2 配置工程

工程建立好后,一般要对工程做进一步设置,以满足不同工程的需要。单击左侧项目工作区列表框中的 Target 1 使其高亮显示,然后单击菜单栏中的“Project→Options for Target ‘Target 1’”,会弹出目标工程设置界面,该界面的大部分内容选默认设置即可,下面对部分常用设置进行说明。

1.2.1 Target 选项卡的配置

选择工程设置界面中的 Target 选项卡,如图 1-6 所示。

(1) Xtal: 设置单片机的工作频率,该值与最终的目标代码无关,仅在软件模拟调试时显示程序执行时间时使用。

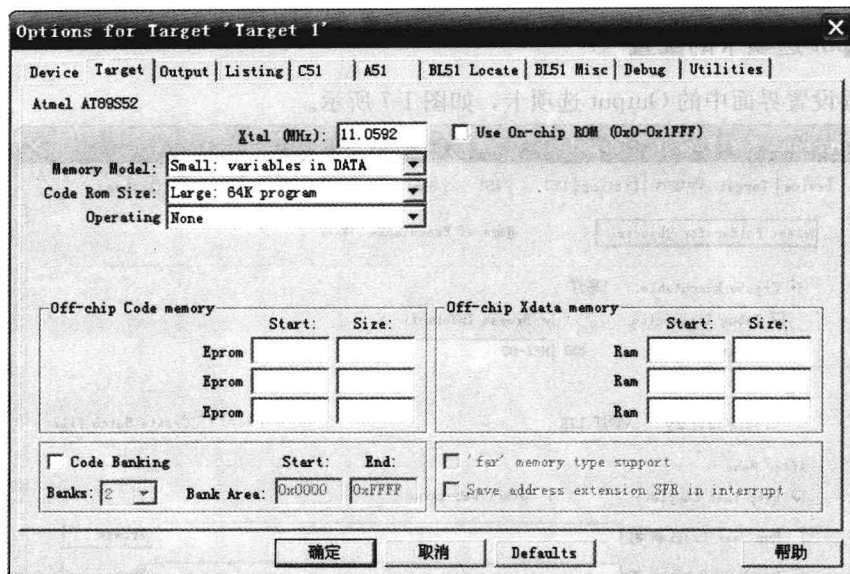


图 1-6 Target 选项卡

(2) Use On-chip ROM: 用于确认单片机是否使用片内 ROM, 选中表示使用片内 ROM, 但该选项与最终的目标代码无关。

(3) Memory Model: 用于设置 RAM 的使用, 该栏的下拉列表有三个选项。

Small: 变量优先存在内部 RAM 里。

Compact: 变量优先存在外部 RAM 里, 可使用 8 位间接寻址访问一页外部 RAM。

Large: 变量优先存在外部 RAM 里, 可使用 16 位间接寻址访问全部外部 RAM。

对于 AT89S52 而言, 这三种模式都支持内部 256B 和外部 64KB 的 RAM, 一般使用 Small 来存储变量, 这时单片机优先将变量存于内部 RAM 中, 内部 RAM 不够时才会存到外部 RAM, 由于使用内部 RAM 的运算速度要比使用外部 RAM 快得多, 产生的代码更加紧凑、高效, 因此大部分应用选择 Small 模式。

使用 Small 模式时, 若要将变量存于外部 RAM, 需要在程序中特别指定, 如将变量 m 存于外部 RAM 中, 可以用语句 “unsigned char xdata m” 实现。

(4) Code Rom Size: 用于设置 ROM 的使用, 该栏的下拉列表也有三个选项。

Small: 用于低于 2KB 的程序空间, 如 AT89C2051 只有 2KB 的程序空间, 编译时会使用 ACALL 或 AJMP 等短跳转指令, 而不会使用 LCALL 或 LJMP 等跳转空间超过 2KB 的指令。

Compact: 单个子函数的代码不能超过 2KB, 整个程序可以有 64KB 的代码。

Large: 单个子函数和整个程序都可以达到 64KB。选择该模式速度不会比前两种模式慢很多, 一般都选择 Large 模式。

(5) Operating: 用于设置是否使用操作系统, 以及使用何种操作系统。

(6) Off-chip Code memory: 表示片外扩展 ROM 的地址范围, 若没有片外 ROM, 则不需填任何数据。

(7) Off-chip Xdata memory: 表示片外扩展 RAM 的地址范围, 若没有片外 RAM, 则不需填任何数据。

(8) Code Banking: Keil 可以支持程序代码超过 64KB 的情况, 最大可以有 2MB 的程序代码。若代码超过 64KB, 须选中 Code Banking, Code Banking 支持自动切换 Bank。

1.2.2 Output 选项卡的配置

选择工程设置界面中的 Output 选项卡，如图 1-7 所示。

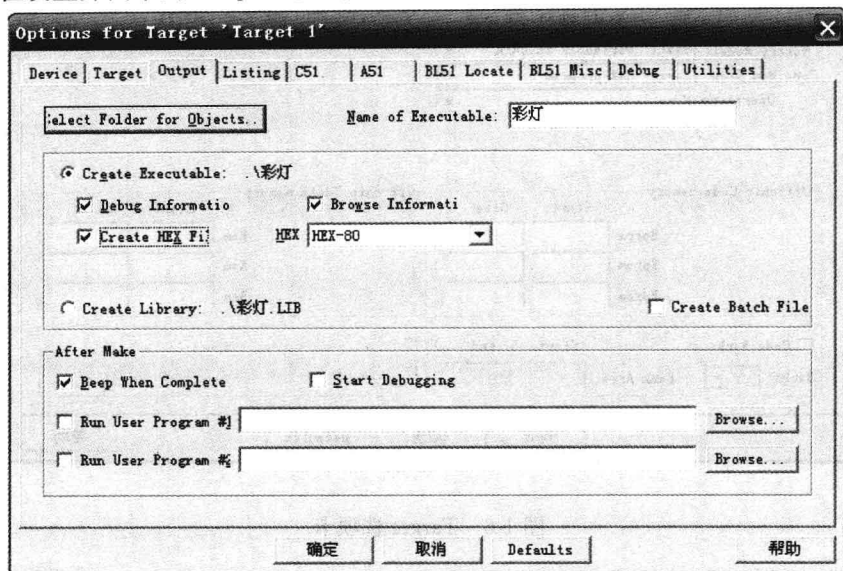


图 1-7 Output 选项卡

(1) Select Folder for Objects: 选择编译后目标文件的存储目录，默认与工程文件存储在同一个目录下。一般不需重新设置。

(2) Name of Executable: 设置生成的目标文件的名称，默认与工程文件同名。

(3) Create Executable:

Debug Information: 选中后，产生调试信息。

Browse Information: 选中后，产生浏览信息。

Create HEX File: 用以生成 HEX 文件，要向单片机写入程序，必须选中该项，默认是不选中的。

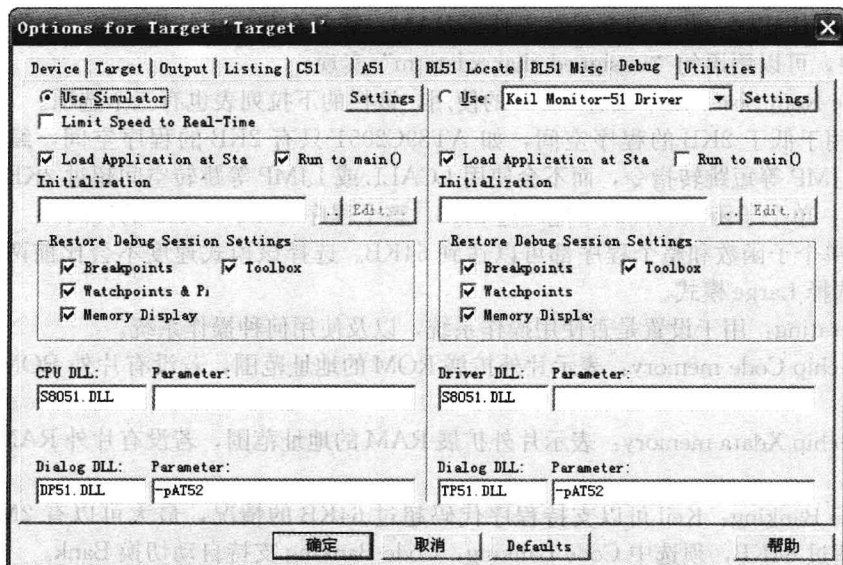


图 1-8 Debug 选项卡

(4) Create Library: 选中该项生成 lib 库文件, 一般不需要选中。

(5) After Make: 用于设置编译完成之后的动作。

Beep When Complete: 选中后, 在编译完成后发出声响。

Start Debugging: 选中后, 在编译完成后自动进入调试状态。

Run User Program #1 和 Run User Program #2: 用于设置编译完成后自动完成某个程序。

1.2.3 Debug 选项卡的配置

选择工程设置界面中的 Debug 选项卡, 如图 1-8 所示。可以选择两类仿真形式, Use Simulator 为软件仿真, Use 后选择相应的下拉项, 可以进行带目标仿真器的仿真。

1.3 编译链接

程序编写好后, 单击菜单 “Project→Rebuild all target files” 或单击相应的快捷图标  开始编译链接。若有语法错误, 则会生成错误报告。并提示出错的位置, 双击错误报告行, 可以定位到出错的源程序相应行。若编译通过, 则给出相应的信息, 如图 1-9 所示, 图中下部的编译链接输出窗口的第 5 行, 表示使用的内部 RAM (data) 为 9Byte, 外部 RAM (xdata) 为 0Byte, 程序目标代码量为 92Byte, 第 6 行提示建立 HEX 文件, 若没有此行, 则说明在 Output 选项卡中没有选中 Create HEX File, 第 7 行给出了错误和警告的数量。



图 1-9 编译链接界面

1.4 程序调试

对工程的编译链接成功后, 单击菜单栏中的 “Debug→Start/Stop Debug Session” 或使用快捷键 Ctrl+F5 进入调试环境。此时窗口左侧自动跳转到寄存器页面卡, 显示工作寄存器 R0~R7、累加器 A、寄存器 B、堆栈指针 SP、程序运行时间、程序状态字 PSW 的内容等信息, 如图

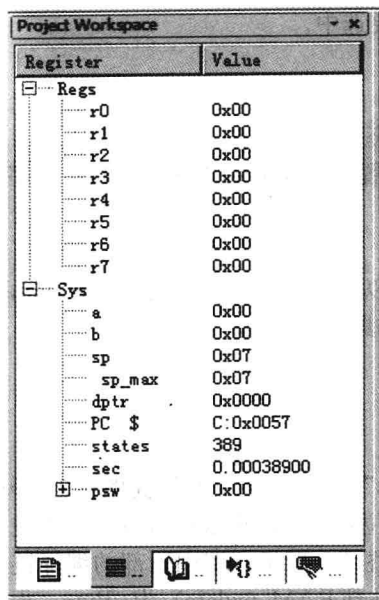


图 1-10 寄存器页面卡

1-10所示。调试过程中，可以看到左侧各寄存器内容的变化。

此时 Debug 菜单下的很多原来不能用的调试命令已经可以使用，该菜单下的内容对应的快捷工具条图标如图 1-11 所示。从左到右依次是复位、运行、停止、单步、过程单步、执行完当前子程序、执行到当前光标行、显示下一状态、运行跟踪、观察跟踪、反汇编窗口、观察和调用堆栈窗口、代码报告窗口、1 号串口、存储器窗口、性能分析窗口、逻辑分析窗口、符号窗口、工具栏。值得一提的是，Keil 提供了 Peripherals 菜单，该菜单中的选项可以非常直观地显示和控制各个寄存器的运行。下面对相关内容进行说明。

1.4.1 单步执行调试

单步执行是调试过程中常用到的，启动调试状态后，使用快捷键 F11 可以单步执行程序，使用快捷键 F10 将以过程单步执行程序，过程单步是指将子程序或子函数作为一个语句来执行。



图 1-11 调试快捷工具条

1.4.2 存储器窗口的使用

要观察调试过程中存储器内容的变化，可以单击“存储器窗口”快捷图标，调出存储器窗口。通过在 Address 后的文本框输入“字母：地址”的形式显示相应存储器的内容，字母可以是 C（程序存储空间）、D（直接寻址的片内 RAM 存储空间）、I（间接寻址的片内 RAM 存储空间）、X（扩展的外部 RAM 空间）。在该窗口中单击鼠标右键，在弹出的菜单中可以选择数据的显示形式，默认以无符号字节形式显示。

例如：输入“D: 0x00”之后，按回车键，即可显示从 0x00 开始的片内 RAM 单元值，如图 1-12 所示。输入“C: 0x30”之后，按回车键，即可显示从 0x30 开始的 ROM 单元值，如图 1-13 所示。

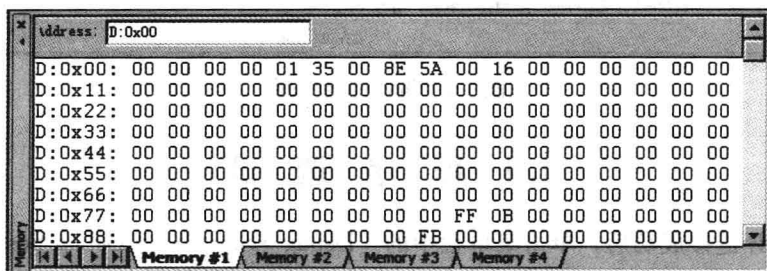


图 1-12 存储器窗口显示片内 RAM 单元值

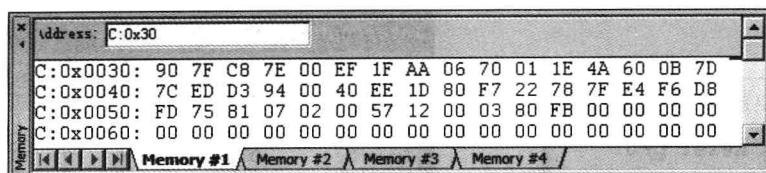
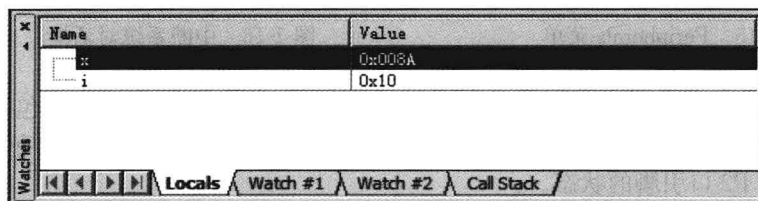


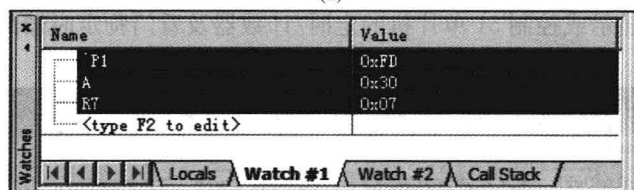
图 1-13 存储器窗口显示 ROM 单元值

1.4.3 观察和调用堆栈窗口的使用

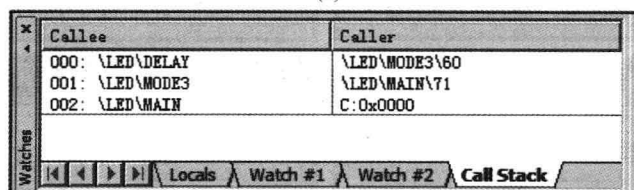
要观察调试过程中一些变量的变化,就要使用观察和调用堆栈窗口,单击菜单“View→Watch & Call Stack Window”或快捷图标后,显示如图 1-14 所示的窗口。



(a)



(b)



(c)

图 1-14 观察和调用堆栈窗口

(a) Locals; (b) Watch; (c) Call Stack

(1) Locals 标签内容显示程序调试过程中当前局部变量的使用情况,图 1-14 (a) 中显示的就是当前局部变量 x 和 i 的值。

(2) Watch #1/#2 显示程序中的变量或表达式在调试过程中的当前值,如图 1-14 (b) 所示。使用方法是在 Watch 窗口用鼠标单击<type F2 to edit>,然后按 F2 激活该处文本框,输入要观察的变量名即可,需要删除时,可以鼠标单击要删除的变量名,按 Delete 键即可。

(3) Call Stack 显示当前调试过程中对子程序的调用情况,如图 1-14 (c) 所示。

1.4.4 基于 Peripherals 菜单的仿真调试

为了更加直观地了解调试运行过程中单片机的定时器、中断、I/O 端口、串行口等内部资源及相关寄存器的状态,可以通过 Peripherals 菜单下的相关选项进行观察和控制,如图 1-15 所示。该菜单的选项内容会根据不同型号的单片机而有所变化。

(1) Interrupt: 用于显示或控制 51 系列单片机中断系统的状态,如图 1-16 所示。

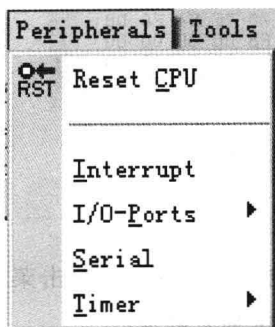


图 1-15 Peripherals 菜单

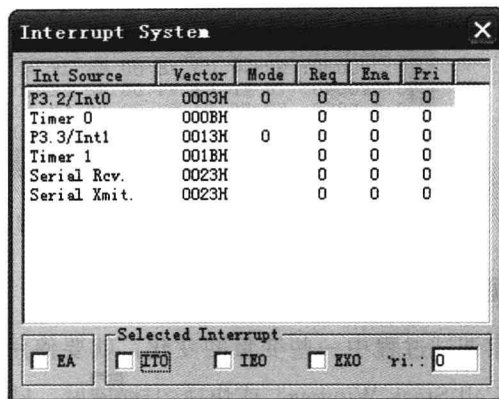


图 1-16 中断系统对话框

(2) I/O-Ports: 用于显示或控制 51 系列单片机各个 I/O 口的运行状态, 包括 I/O 口锁存器的状态和引脚状态, 如图 1-17 所示, 以 Port 2 为例, 第一行 P2 模拟特殊功能寄存器 P2 的状态, 第二行 Pins 模拟 P2 口引脚的状态。

(3) Serial: 用于显示或控制 51 单片机的串行口运行状态, 如图 1-18 所示。

(4) Timer: 用于显示或控制 51 单片机的定时/计数器及看门狗定时器的运行状态, 如图 1-19 所示。

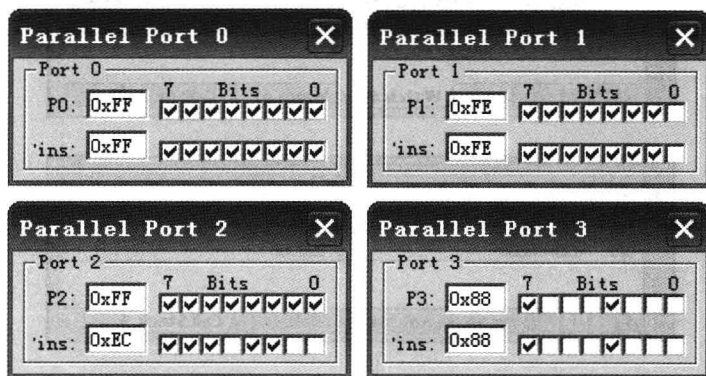


图 1-17 并行 I/O 口对话框

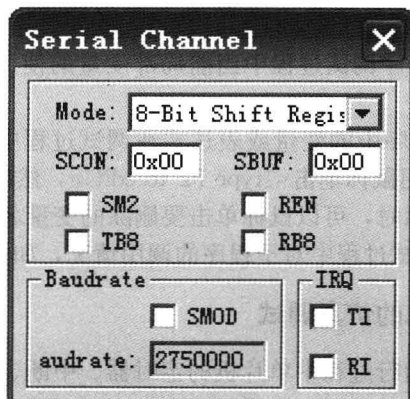


图 1-18 串行口对话框