国家级实验教学示范中心

“电气工程基础实验中心”系列实验教材

西南交通大学323实验室工程系列教材

单片机原理及应用实验教程

DANPIANJI YUANLI JI YINGYONG SHIYAN JIAOCHENG

赵 琳 编

西南交通大学峨眉校区国有资产及实验管理处 审

西南交通大学出版社

Http://press.swjtu.edu.cn

国家级实验教学示范中心
“电气工程基础实验中心”系列实验教材
西南交通大学“323 实验室工程”系列教材

单片机原理及应用实验教程

赵　琳　编
西南交通大学峨眉校区
国有资产及实验管理处　审

西南交通大学出版社
·成　都·

图书在版编目（CIP）数据

单片机原理及应用实验教程 / 赵琳编. —成都：西南交通大学出版社，2013.3

国家级实验教学示范中心. “电气工程基础实验中心”系列实验教材. 西南交通大学“323 实验室工程”系列教材

ISBN 978-7-5643-2264-9

Ⅰ. ①单… Ⅱ. ①赵… Ⅲ. ①单片微型计算机 – 高等学校 – 教材 Ⅳ. ①TP368.1

中国版本图书馆 CIP 数据核字（2013）第 056251 号

国家级实验教学示范中心
“电气工程基础实验中心”系列实验教材
西南交通大学“323 实验室工程”系列教材
单片机原理及应用实验教程
赵 琳 编

责任编辑	李芳芳
特邀编辑	李庞峰
封面设计	本格设计
出版发行	西南交通大学出版社 （成都二环路北一段 111 号）
发行部电话	028-87600564　028-87600533
邮政编码	610031
网　　址	http：//press.swjtu.edu.cn
印　　刷	成都蓉军广告印务有限责任公司
成品尺寸	185 mm × 260 mm
印　　张	7
字　　数	177 千字
版　　次	2013 年 3 月第 1 版
印　　次	2013 年 3 月第 1 次
书　　号	ISBN 978-7-5643-2264-9
定　　价	14.00 元

图书如有印装质量问题　本社负责退换

前　言

本书是依据电气工程专业“单片机原理及应用”课程的教学大纲要求，依托西南交通大学“323 实验工程”建设投入的软硬件实验平台，配合课程教材《单片机原理及应用》而编写的实验指导教材，供电气工程、电子信息、自动化等相关专业的学生学习使用。

单片机是电气类专业的一门重要的专业基础课程，具有很强的技术性和应用性，实验教学在课程中占有非常重要的位置。目前，国内各高校大多采用现成的实验箱进行单片机实验教学。实验箱上虽然硬件资源丰富，但高度的集成化却使得整个实验箱缺少相应的灵活性，学生只能按照规定的步骤完成有限的实验项目，很难进行个性化实验的拓展与创新，从而挫伤学生实验及研究的积极性，使实验效果大打折扣。

近年来，随着电子设计自动化(EDA)技术的飞速发展，各种 EDA 工具软件备受设计师们青睐。为了提高实验教学质量，西南交通大学峨眉校区引入先进的 Proteus 系列软件，在全校范围内建立网上虚拟仿真实验平台，激发了学生的学习热情，使学生的实验再也不受时间及空间的限制了。实验内容也更加多样化，可由一般性实验项目扩展到综合型、设计型及创新型项目上来，非常灵活方便，节约了成本，拓宽了学生的视野，锻炼了学生综合运用知识的能力，培养了学生的科创意识。

本书以 Proteus 仿真软件及相关硬件为实验平台，对传统的实验内容进行调整，让学生有更多思考、研究与开发的内容。根据单片机课程的知识体系，本书内容分为四部分：第一，单片机基础知识和实验环境介绍。主要内容包括 Keil μ Vision 集成开发环境使用方法，Proteus 仿真软件使用方法，硬件实验箱的联机调试方法，课外 DIY 实验过程及方法。这类实验配合一定的指令及程序结构来开设，可使学生在熟悉各种软、硬件开发环境的过程中，对单片机中的指令和程序结构有更深入的了解和掌握，起到非常好的教学效果。第二，单片机内部结构基础实验。主要内容包括输入/输出端口实验，外部中断实验，定时器/计数器实验及串行通信实验。这些实验都是围绕着单片机本身的资源来开设的，是对单片机各部分功能的验证，在教学当中是每个学生都必须完成的基本实验内容。第三，单片机扩展技术实验。内容包括存储器扩展实验，I/O 口扩展实验，数码管扩展实验，键盘扩展实验，D/A 转换实验及 A/D 转换实验等。这些实验内容都是在单片机的基础上进行相关功能的扩展，思路和原理基本一致。因受实验学时影响，不可能做到面面俱到，故这部分可以定为选作内容，供学生自由选择。主要是使学生学会基本的扩展原理及常用扩展芯片的使用方法，重在培养学生自行设计

单片机扩展电路的能力。第四，综合研发型实验。这类实验内容可选取一些具有较完整功能的小制作，例如电子钟、0～5 V 直流数字电压表、交通灯控制器、信号发生器、温度监测仪等。要求学生利用自己所掌握的单片机技术，设计并完成相关产品的制作。

本书编写过程中，潘育山、许金福等老师给予了大力的支持和帮助，西南交通大学峨眉校区国有资产及实验管理处作为主审，提出了很多宝贵的指导意见，西南交通大学出版社责任编辑李芳芳为本书的顺利出版付出了大量辛勤的劳动，在此一并表示衷心的感谢！

由于编者水平有限，书中难免存在疏漏之处，敬请读者批评指正。

编 者

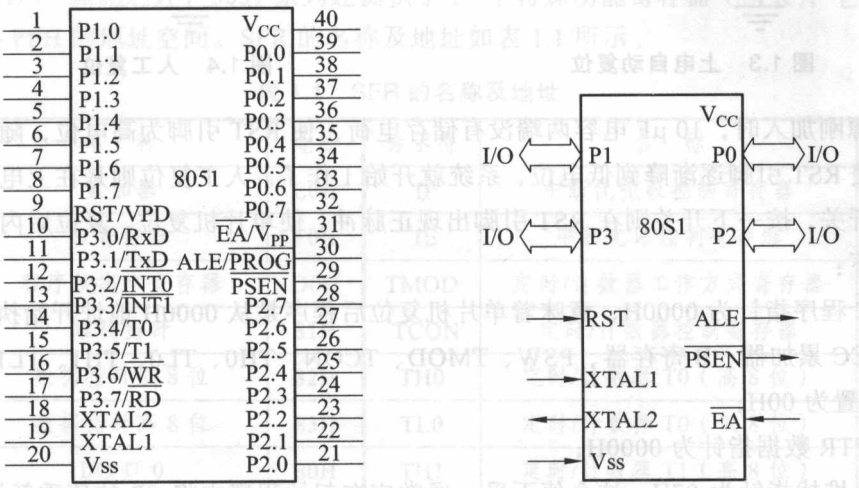
2013 年 3 月

目　录

第 1 章　单片机基础知识及实验环境训练

1.1　MCS-51 单片机的引脚功能

在 MCS-51 系列中，制造工艺为 HMOS 型单片机都采用 40 条引脚的双列直插式封装（DIP），引脚排列如图 1.1（a）所示，逻辑符号如图 1.1（b）所示。



（a）引脚排列　　（b）逻辑符号

图 1.1　MCS-51 引脚排列

1. 电源引脚

V_{CC}（40 脚）：+5 V 电源线；V_{SS}（20 脚）：接地线。

2. 时钟引脚

XTAL1（19 脚）和 XTAL2（18 脚）：片内振荡电路输入线。MCS-51 的时钟可利用内部的振荡电路产生，只要在 XTAL1、XTAL2 引脚上跨接石英晶振及两个补偿电容，就可以构成稳定的自激多谐振荡器，如图 1.2 所示。

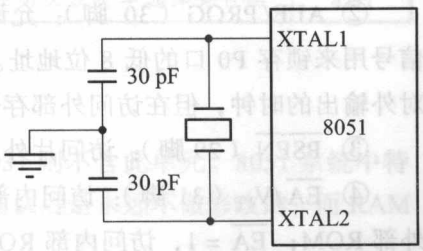


图 1.2　振荡电路

振荡频率 f_{osc} 主要由石英晶振的频率确定，典型值为 12 MHz 或 6 MHz。两个电容的作用是帮助起振，典型值为 30 pF。

3. 控制引脚

① RST/VPD（9 脚）：复位/备用电源线。

延续两个机器周期以上的高电平，复位有效。复位有上电自动复位和人工复位两种，分别如图 1.3、1.4 所示。

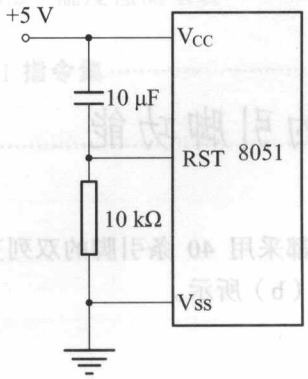


图 1.3　上电自动复位

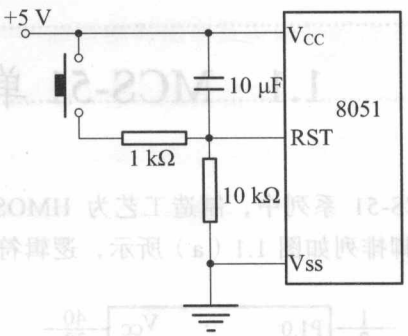


图 1.4　人工复位

当电源刚加入时，10 μF 电容两端没有储存电荷，使 RST 引脚为高电位，随即电容开始充电，促使 RST 引脚逐渐降到低电位，系统就开始工作了。人工复位则是在上电复位电路中并联一个开关，按一下开关则在 RST 引脚出现正脉冲，使单片机复位。复位后内部寄存器的初始值如下：

◇ PC 程序指针为 0000H，意味着单片机复位后程序是从 0000H 地址开始执行；

◇ ACC 累加器、B 寄存器、PSW、TMOD、TCON、TH0、TL0、TH1、TL1、SCON 被设置为 00H；

◇ DPTR 数据指针为 0000H；

◇ SP 堆栈指针为 07H，这个值不妥，通常应在起始程序中将 SP 的值重新设成 50H 或 60H 等，以免整个内部数据空间都被堆栈区占去；

◇ P0、P1、P2、P3 为 FFH，即输入的状态；

◇ IP 为 XXX00000B；IE 为 0XX00000B；PCON 为 0XXXXXXXB。

② ALE/$\overline{\text{PROG}}$（30 脚）：允许地址锁存/编程脉冲输入端。当访问片外存储器时，ALE 信号用来锁存 P0 口的低 8 位地址。注意：ALE 以 1/6 振荡频率周期性地输出信号，可作为对外输出的时钟，但在访问外部存储器时将跳过一个脉冲。

③ $\overline{\text{PSEN}}$（29 脚）：访问片外 ROM 选通线，低电平时允许读 ROM 中的指令码。

④ $\overline{\text{EA}}$/V_{PP}（31 脚）：访问内部或外部 ROM 的选择信号/编程电压输入。$\overline{\text{EA}}$ = 0，访问外部 ROM；$\overline{\text{EA}}$ = 1，访问内部 ROM，PC 指针超过 4K（0FFFH）时，自动转向外部 ROM。

4. I/O 口引脚

四个 8 位双向 I/O 端口（P0、P1、P2、P3），每一条 I/O 线都能独立地作输入或输出。

P0 口（32～39）通用 I/O 口或低 8 位地址/数据复用线；

P1 口（1～8）通用 I/O 口；

P2 口（21～28）通用 I/O 口或高 8 位地址线；

P3 口（10～17）双功能口。

1.2　MCS-51 单片机的内部存储器

1. 内部数据存储器（RAM）

8051 系列提供 128 字节可读写的数据存储单元，地址空间为 00H～7FH。按功能和用途可分为 3 个区域，即工作寄存器区（00H～1FH）、位寻址区（20H～2FH）、堆栈和数据缓冲区（30H～7FH）。除此之外，8051 系列还提供了 21 个特殊功能寄存器（SFR），它们离散的分布在 80H～FFH 的地址空间。SFR 的名称及地址如表 1.1 所示。

表 1.1　SFR 的名称及地址

标示符	名　称	地址	标示符	名　称	地址
*ACC	累加器	E0H	*IP	中断优先级控制寄存器	B8H
*B	B 寄存器	F0H	*IE	中断允许控制寄存器	A8H
*PSW	程序状态字寄存器	D0H	TMOD	定时/计数器工作方式寄存器	89H
SP	堆栈指针	81H	*TCON	定时/计数器控制寄存器	88H
DPL	数据指针低 8 位	82H	TH0	定时/计数器 T0（高 8 位）	8CH
DPH	数据指针高 8 位	83H	TL0	定时/计数器 T0（低 8 位）	8AH
*P0	I/O 口 0	80H	TH1	定时/计数器 T1（高 8 位）	8DH
*P1	I/O 口 1	90H	TL1	定时/计数器 T1（低 8 位）	8BH
*P2	I/O 口 2	A0H	*SCON	串行口控制寄存器	98H
*P3	I/O 口 3	B0H	SBUF	串行口数据缓冲器	99H
			PCON	电源控制及波特率选择寄存器	87H

注：*表示可以位寻址。

2. 内部程序存储器（ROM）

8051 系列提供 4 096 个字节（4KB）的 ROM，注意 8031 则不含此单元。8051 系统中特地将程序区和数据区分开，ROM 用于存放应用程序，一经确认后是永远不做修改的；而 RAM 是可随时读写的，用于存放程序运行中的一些变动的数据。

实验 1.1 Keil μ Vision 集成开发环境的使用

一、实验目的

（1）熟悉 Keil μ Vision 集成开发环境，掌握用 Keil μ Vision 对程序进行编辑、编译及调试的方法；

（2）学会利用 Keil μ Vision 观测程序运行结果及执行时间；

（3）强化对常用指令及程序的理解。

二、实验内容及要求

（1）按照下面的步骤，熟练掌握 Keil μ Vision 的工程建立、编辑、编译及调试功能。

① 打开 Keil 软件。

点击【开始】→【程序】→Keil μ Vision2，或双击桌面上的“Keil μ Vision2 图标，即可打开 Keil μ Vision 软件，如图 1.5 所示。

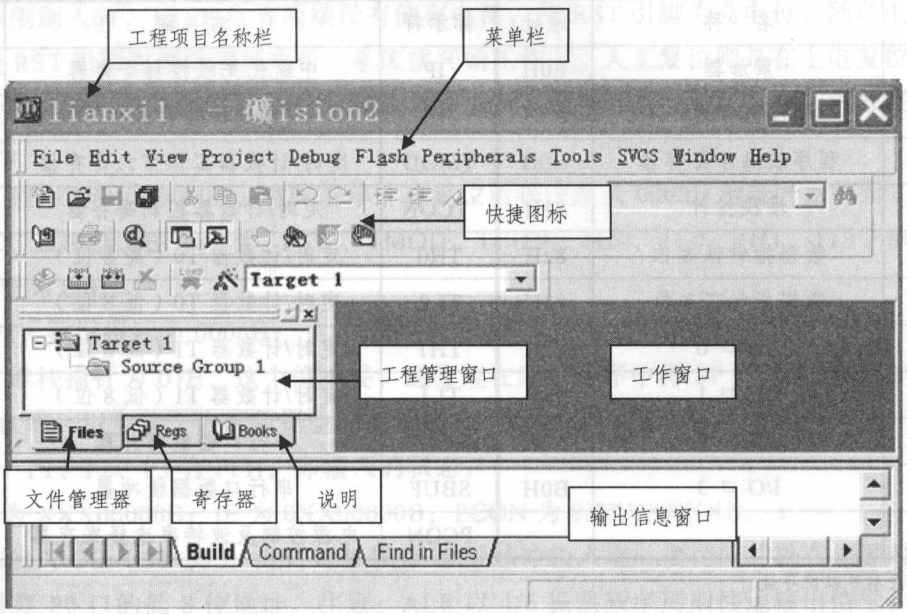


图 1.5 Keil μ Vision 工作界面

② 建立工程文件。

选择菜单“Project→New Project”，弹出建立新工程对话框，如图 1.6 所示。为工程项目选择合适的保存路径，可事先在电脑上建立一个文件夹，例如“实验一”，将自己的本次的实验内容放在此文件夹中。输入工程名，如 shiyan1，不用写扩展名，默认的文件扩展名为“.uv2”。

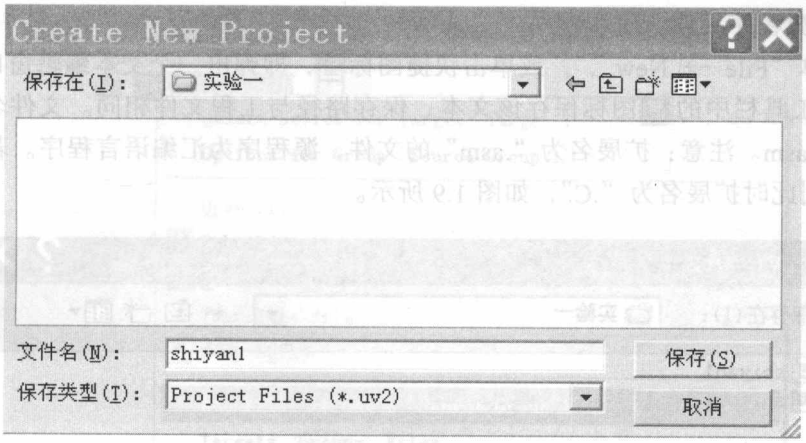


图 1.6　对新建的工程项目进行命名和保存

③ 选择单片机型号。

单击图 1.6 的【保存】按钮后，弹出单片机型号选择对话框，如图 1.7 所示。

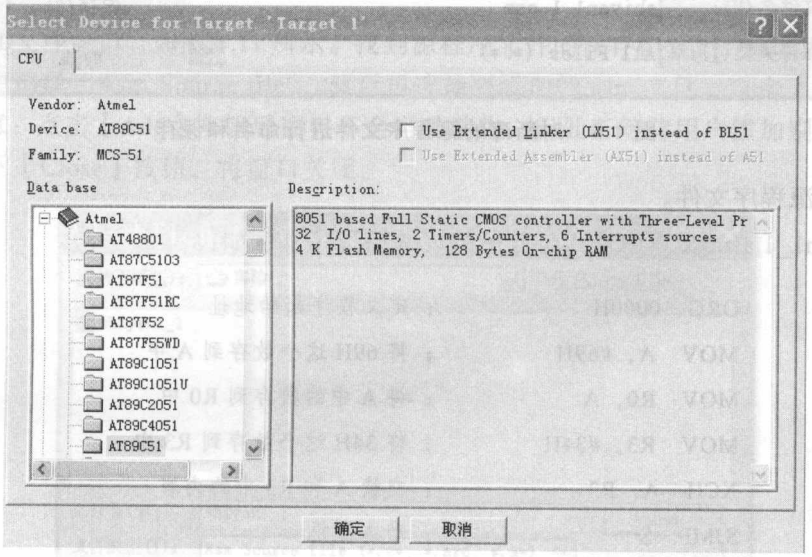


图 1.7　单片机型号选择对话框

选择 CPU 型号为 Atmel → AT89C51，单击【确定】按钮。注意：如果此时弹出图 1.8 所示的对话框，询问是否将初始化代码一起加入工程，单击【否】按钮即可。

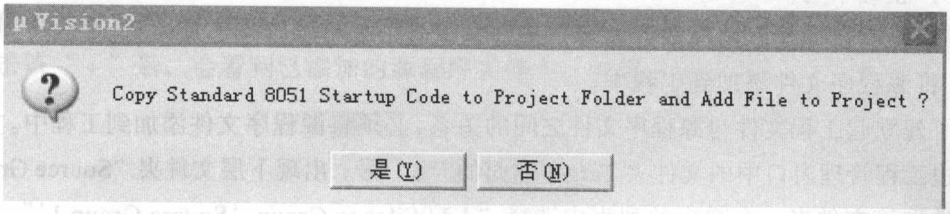


图 1.8　是否将初始化代码一起加入工程

④ 新建源程序文件。

选择菜单“File → New...”，或单击快捷图标，即弹出一个文本编辑窗口，默认名为Text1。单击工具栏中的图标保存该文本，保存路径与工程文件相同。文件名可任取，例如 shiyan1_1.asm。注意：扩展名为“.asm”的文件，源程序为汇编语言程序。若采用 C 语言编写程序，则此时扩展名为“.C”，如图 1.9 所示。

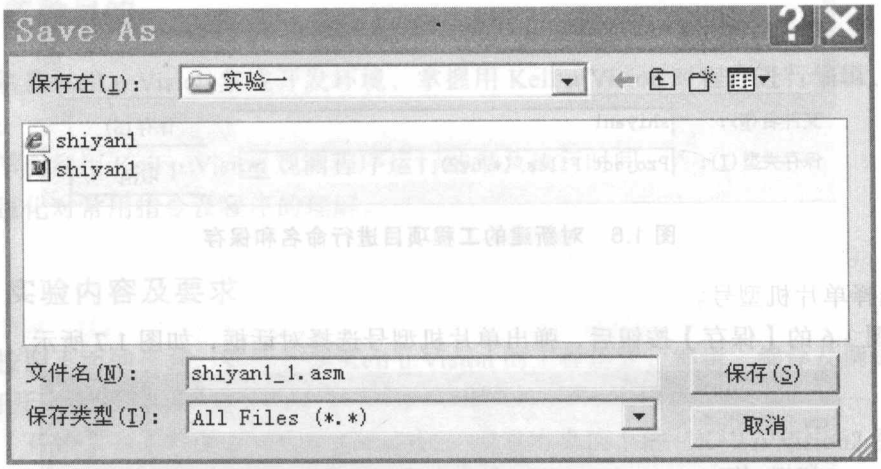


图 1.9　对新建的源程序文件进行命名和保存

⑤ 编辑源程序文件。

在 shiyan1_1.asm 文本中输入自己所编写的源程序。例如：

```
ORG   0000H          ; 定义程序起始地址
MOV   A, #69H        ; 将 69H 这个数存到 A 中
MOV   R0, A          ; 将 A 中的数存到 R0 中
MOV   R3, #34H       ; 将 34H 这个数存到 R3 中
XCH   A, R3          ; 交换 A 和 R3 中的内容
SJMP  $              ; 动态停机
END                  ; 程序结束
```

注意：

◇ 若该文本在编辑之前已经保存为.asm 文件，则写程序时关键字会变色，用于提醒用户正确书写。

◇ 程序应在英文半角模式下书写，特别是逗号、分号等符号必须注意。

⑥ 将源程序文件添加到工程中。

为了建立起工程文件与源程序文件之间的关系，必须将源程序文件添加到工程中。方法是：单击左边工程管理窗口中的文件夹 Target 前面的“+”号，出现下层文件夹“Source Group 1”，用右键选取该文件夹，在弹出的列表中选择“Add Files to Group ‘Source Group 1’”，操作过程如图 1.10 所示。

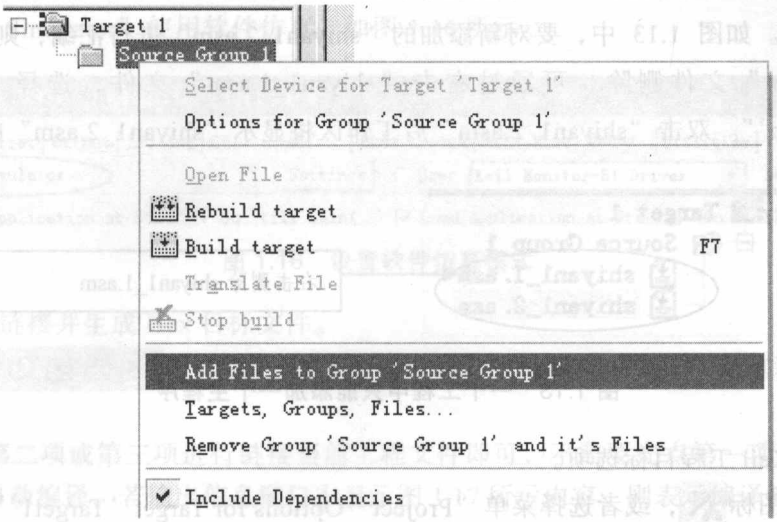


图 1.10　打开工程管理文件夹

弹出添加文件窗口，如图 1.11 所示。找到源程序文件的保存位置，并在窗口下部的文件类型列表框中选择“Asm Source file”，然后再选择要添加的.asm 文件，点击【Add】按钮即可完成。注意：点击【Add】按钮后窗口不会自动消失，仍处于等待用户添加别的.asm 文件，这时，应按下【Close】按钮，将窗口关闭。

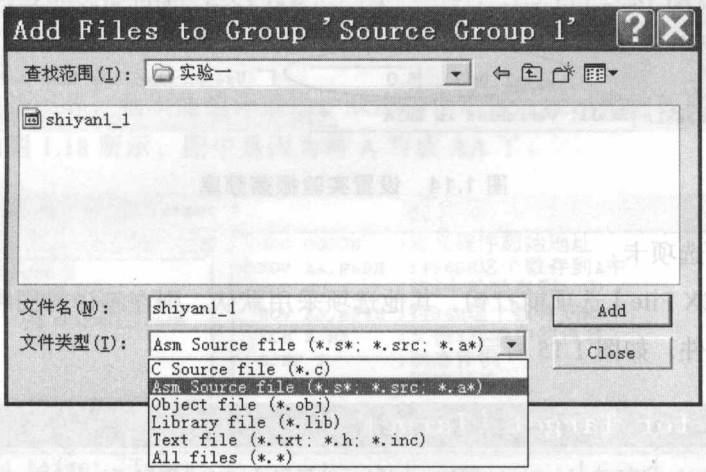


图 1.11　添加源程序文件到工程中

文件添加成功后，左边的工程管理窗口中的“Source Group 1”文件夹前会出现一个“+”号，单击该“+”后，会看到已添加的源程序文件名，如图 1.12 所示。

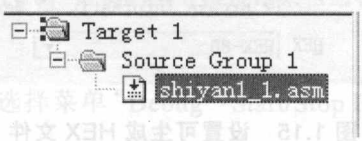


图 1.12　已添加好源程序的工程管理窗口

注意：一个工程中只能添加一个主程序，若要利用此工程再汇编其他程序，则要将原有的源程序删掉。如图 1.13 中，要对新添加的“shiyan1_2.asm”进行汇编，则必须将原有的“shiyan1_1.asm”文件删除，可通过右击“shiyan1_1.asm”文件，选择“Remove File ‘shiyan1_1.asm’”。双击“shiyan1_2.asm”后工作区将显示“shiyan1_2.asm”的程序内容。

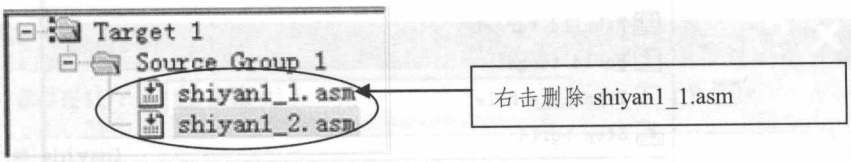


图 1.13 一个工程中只能添加一个主程序

⑦ 设置 Keil 工程目标选项。

单击快捷图标，或者选择菜单“Project→Options for Target ‘Target1’”，打开工程设置对话框，进行下列设置。

a.“Target”选项卡：

在“Xtal”栏中填写振荡频率，如 6.0，即实验采用的晶振是 6 MHz，如图 1.14 所示。

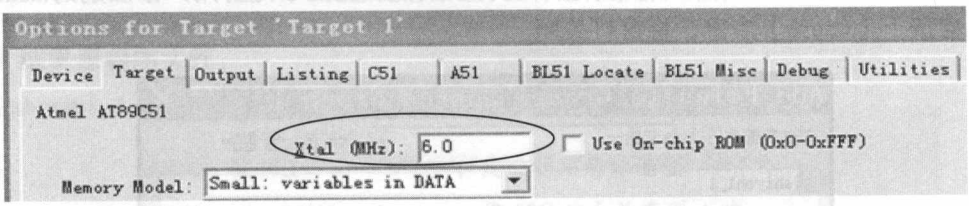


图 1.14 设置实验振荡频率

b.“Output”选项卡：

在【Creat HEX File】选项前打钩，其他选项采用默认，则在编译的同时生成与工程文件名同名的 HEX 文件，如图 1.15 所示。

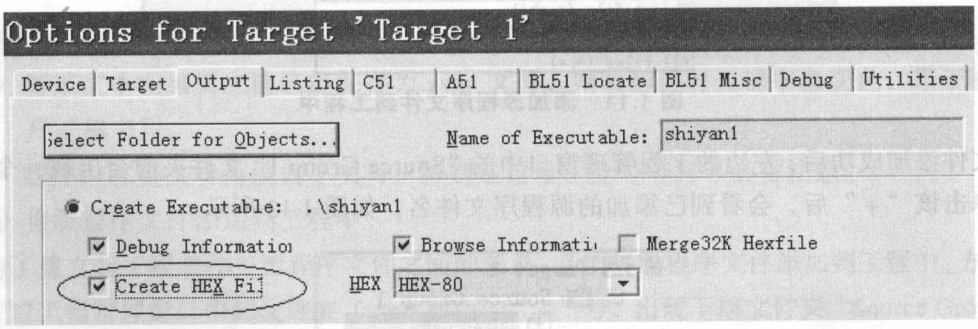


图 1.15 设置可生成 HEX 文件

c. “Debug” 选项卡：

选择 “Use Simulator” 使用软件仿真，如图 1.16 所示。

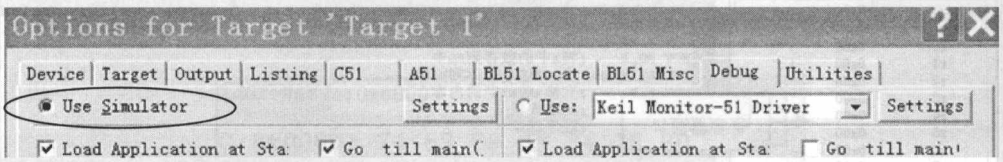
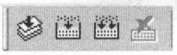


图 1.16　设置软件仿真模式

⑧ 编译、链接并生成 Hex 目标文件。

在工具条  中，第一项是编译，第二项是链接，第三项是重新链接全部工程文件。

直接点击第二项或第三项进行链接当前工程文件即可，不需要先点第一项编译了，因为链接时机器会自动编译。若输出信息窗口中显示图 1.17 所示内容，则表示编译链接没有出错，并成功生成了 hex 目标代码。

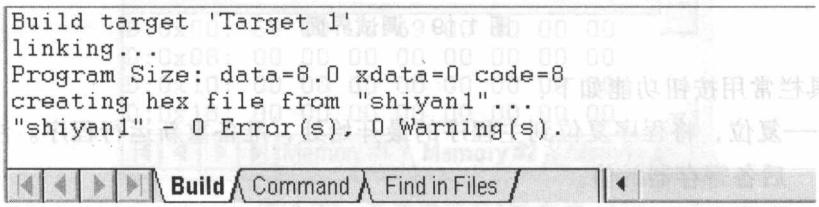


图 1.17　正确的编译链接结果信息框

若程序有语法错误等，则不能编译通过。双击错误信息后在出错的程序行上会出现一个绿色的小箭头，如图 1.18 所示，图中是因为将 A 写成 AA 了。

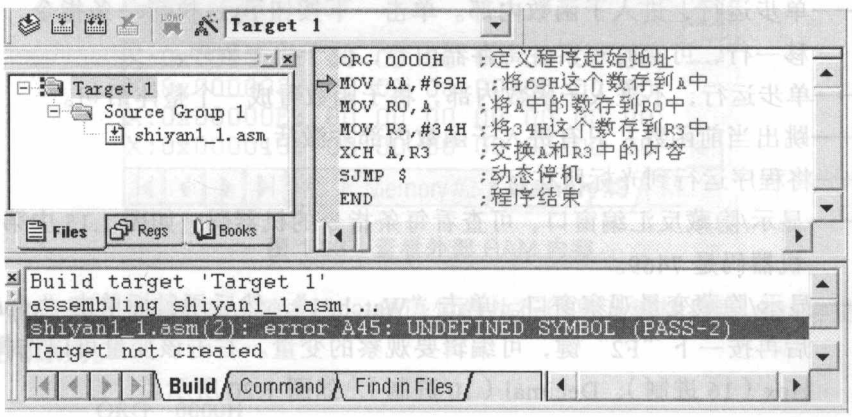


图 1.18　错误的编译链接结果信息框

⑨ 运行调试。

点击调试按钮图标  ，或选择菜单 “Debug→Start/Stop Debug Session”，进入调试状态，如图 1.19 所示。

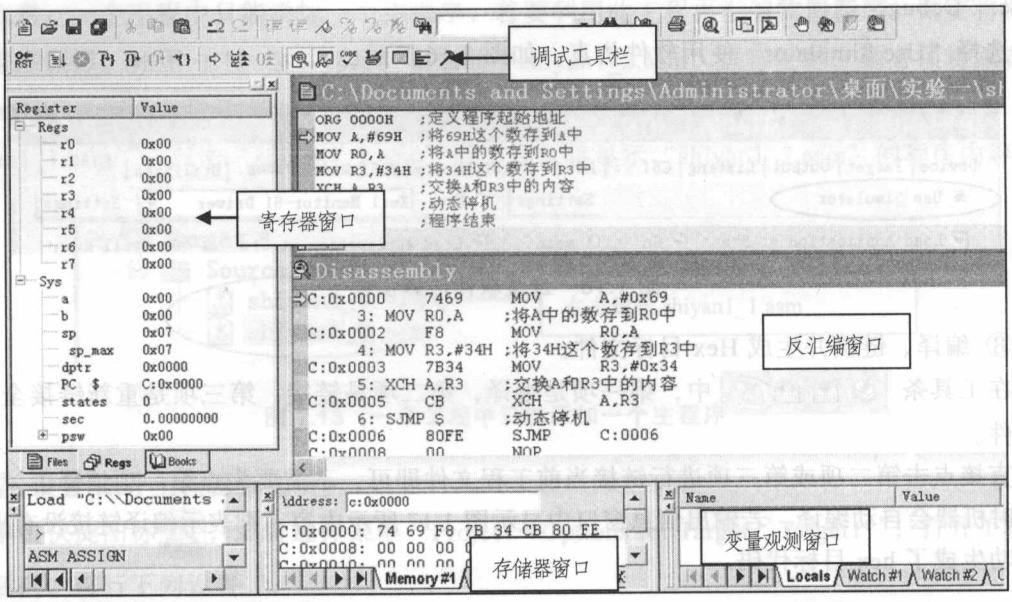


图 1.19 调试界面

调试工具栏常用按钮功能如下：

a. RST——复位，将程序复位到主程序的最开始处，准备重新运行程序。注意观察复位后各寄存器的值。

b. ——全速运行，不间断地执行程序指令。本例全速运行，最后反复执行 SJMP $，只有通过点击停止全速运行按钮才能终止运行。运行终止后，各寄存器显示程序最终执行结果。

c. ——停止全速运行。

d. ——单步运行，进入子函数内部。单击一下按钮，执行一条指令，黄色箭头下移一行，可实时观察各寄存器内容，便于检查程序。

e. ——单步运行，不进入子函数内部，将子函数看成一个整体语句。

f. ——跳出当前函数，只有进入子函数内部才激活。

g. ——将程序运行到光标所在行。

h. ——显示/隐藏反汇编窗口，可查看每条指令的机器码。如图 1.15 中第一条指令的机器码是 7469。

i. ——显示/隐藏变量观察窗口。单击“Watch#1”，然后用鼠标单击“type F2 to edit”后再按一下“F2”键，可编辑要观察的变量。右击该变量可以选择显示方式：Hex（16 进制）、Decimal（10 进制），如图 1.20 所示。

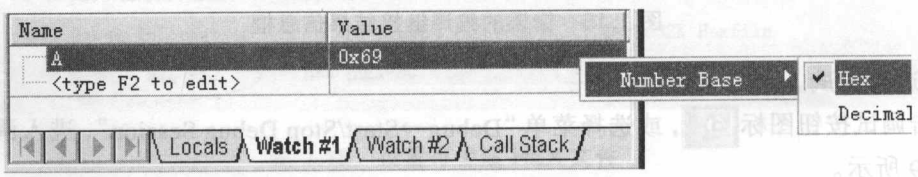


图 1.20 变量观察窗口