

高职高专“十二五”规划教材

基于任务驱动的 单片机应用教程

主 编 程院莲 廖春蓝
副主编 卢飞跃 汪建宇



西安电子科技大学出版社
<http://www.xduph.com>

高职高专“十二五”规划教材

基于任务驱动的 单片机应用教程

主 编 程院莲 廖春蓝

副主编 卢飞跃 汪建宇

西安电子科技大学出版社

内 容 简 介

本书是按“任务驱动”模式编写的 51 单片机原理及应用技术教材。全书共设 13 个任务, 每个任务都是单片机的实际应用项目, 且均由内容与分析、实施和相关知识等部分组成, 在任务实施中详细介绍了该项目的电路原理图、程序流程图、程序代码、调试步骤及扩展功能等, 充分体现了项目的完整性、系统性和扩展性。

本书内容深入浅出, 以学生的认知规律为主线, 任务由简到繁, 循序渐进。通过 13 个任务的学习与实训, 学生可掌握单片机应用系统开发的基本知识与基本技能。

本书可作为高职高专应用电子技术、电气自动化、计算机等专业的教材, 也可作为电子技术爱好者自学单片机的参考书。

★本书配有电子教案, 需要者可登录出版社网站, 免费下载。

图书在版编目(CIP)数据

基于任务驱动的单片机应用教程/程院莲, 廖春蓝主编. —西安: 西安电子科技大学出版社, 2011.8
高职高专“十二五”规划教材

ISBN 978 - 7 - 5606 - 2623 - 9

I. ①基… II. ①程… ②廖… III. ①单片微型计算机—高等职业教育—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2011)第 126203 号

策 划 杨丕勇

责任编辑 阎 彬 杨丕勇

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2011 年 8 月第 1 版 2011 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 14

字 数 332 千字

印 数 1~3000 册

定 价 21.00 元

ISBN 978 - 7 - 5606 - 2623 - 9/TP • 1289

XDUP 2915001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

市面上以 8051 为内核的系列单片机的教材数不胜数，而以“任务驱动”模式编写的教材为数不多。

本书融入了编者多年在教学、科研实践中所获取的经验，充分汲取了近年来高职高专院校在探索培养高等技术应用型人才和教材建设方面取得的成功经验，在对单片机课程进行教学改革的基础上，以“任务”为教学单元，将各部分知识分解成一个个知识点，将不同知识点分解到不同任务中。各任务由简到繁，循序渐进，使得学习过程成为一个不断成功地完成任务的过程。这样每完成一个任务都会带给学生强烈的成就感，从而激发学生强烈的求知欲和浓厚的学习兴趣。

本书内容力求深入浅出，所选任务均是有一定趣味性及较强实用性的单片机应用项目。具体内容如下：

任务 1：控制单只 LED 闪烁发光——初识单片机与单片机开发工具使用。

任务 2：进位指示灯控制——初识编程与单片机存储器的配置。

任务 3：汽车转向灯控制——输入输出电路设计与位操作指令。

任务 4：流水灯控制——控制转移类指令与循环结构程序。

任务 5：运料小车控制——堆栈操作类指令与子程序。

任务 6：pH 值监控系统的设计——外部中断技术的应用。

任务 7：音频报警器的设计——定时器 KKD/计数器资源使用。

任务 8：数码秒表的设计——LED 数码管显示接口技术应用。

任务 9：电子密码锁的设计——键盘接口技术应用。

任务 10：远距离控制记分牌的设计——串行接口技术应用。

任务 11：简易波形发生器的设计——D/A 接口技术应用。

任务 12：5 V 直流数字电压表的设计——A/D 接口技术应用。

任务 13：液晶显示广告牌的设计——液晶显示接口技术应用。

此外，本书还具有以下特色：

典型性：目前市场上的单片机产品仍以 8051 为内核，形成了所谓的 8051 主流系列，故本书就是以 89S51 为例进行介绍的。

创新性：根据学生的认知规律，以任务为单元构建认知单元，而不是以单片机的结构或功能作为认知单元。

科学性：从基础着手，深入浅出，循序渐进，图文并茂，注重理论与实践的紧密结合。

实用性：每一单元均围绕实用性较强的任务展开，每个任务都有详细的电路图、程序流程图及程序代码等。在程序中，每条指令都会提供注解，便于快速理解程序的内涵。另外，在介绍相关知识点后，尽量结合一定实例来说明问题，且提出了一些具有启发性的问题，可以帮助读者巩固所学知识，达到举一反三的目的。

方便教学：全书层次分明、语言简练，每个任务都有小结和精选的练习题，便于教学。建议采用理论实践一体化的教学模式，学时数为 70~80 学时。

广州番禺职业技术学院程院莲、廖春蓝任本书主编，卢飞跃、汪建宇任副主编，程院莲负责全书的统稿工作。另外，广州城市职业技术学院的谢元成、三门峡职业技术学院的员青泽、武汉船舶职业技术学院的牛涛也参与了本书的编写，在此表示衷心感谢。本书在编写过程中参考了许多文献资料(列在书后参考文献中)，在此向原作者表示感谢。

由于编者水平有限，书中难免有不妥之处，恳请读者批评指正。如有疑问和建议，请联系编者(E-mail: chengyl@gzpyp.edu.cn)。

编 者
2011 年 3 月

目 录

任务 1 控制单只 LED 闪烁发光 1	
1.1 任务内容与分析..... 1	
1.2 任务实施..... 2	
1.3 任务相关知识..... 11	
1.3.1 微型计算机与单片机..... 11	
1.3.2 单片机产品分类..... 12	
1.3.3 8051 的基本结构..... 13	
1.3.4 89S51 的封装与引脚..... 13	
1.3.5 单片机最小应用系统..... 14	
1.3.6 单片机的特点与应用..... 16	
1.3.7 8051 应用系统的开发制作 过程与工具..... 17	
小结..... 19	
实练..... 19	
任务 2 进位指示灯控制 20	
2.1 任务内容与分析..... 20	
2.2 任务实施..... 21	
2.3 任务相关知识..... 25	
2.3.1 单片机中常用的基本术语..... 25	
2.3.2 8051 存储器的基本结构..... 27	
2.3.3 有关指令与程序的基本概念..... 31	
2.3.4 数据传送类指令..... 34	
2.3.5 算术运算类指令..... 37	
2.3.6 伪指令..... 41	
2.3.7 程序设计基本步骤..... 44	
2.3.8 基本程序结构 1——顺序结构..... 45	
小结..... 46	
实练..... 46	
任务 3 汽车转向灯控制 48	
3.1 任务内容与分析..... 48	
3.2 任务实施..... 49	
3.3 任务相关知识..... 53	
3.3.1 认识 8051 的输入/输出端口 (I/O 口)..... 53	
3.3.2 8051 输出电路设计..... 56	
3.3.3 8051 输入电路设计..... 59	
3.3.4 位操作类指令..... 61	
3.3.5 基本程序结构 2——分支结构..... 63	
小结..... 64	
实练..... 64	
任务 4 流水灯控制 66	
4.1 任务内容与分析..... 66	
4.2 任务实施..... 67	
4.3 任务相关知识..... 70	
4.3.1 逻辑运算与循环类指令..... 70	
4.3.2 控制转移类指令..... 72	
4.3.3 基本程序结构 3——循环结构..... 76	
小结..... 77	
实练..... 77	
任务 5 运料小车控制 79	
5.1 任务内容与分析..... 79	
5.2 任务实施..... 79	
5.3 任务相关知识..... 82	
5.3.1 空操作指令..... 82	
5.3.2 基本程序结构 4——子程序 及其调用..... 82	
小结..... 87	
实练..... 87	
任务 6 pH 值监控系统的设计 89	
6.1 任务内容与分析..... 89	
6.2 任务实施..... 90	
6.3 任务相关知识..... 92	
6.3.1 8051 的中断系统..... 92	
6.3.2 外部中断应用实例..... 98	

小结	99	10.3.2 单片机的串行接口	160
实练	99	10.3.3 单片机与 PC 机的通信	166
任务 7 音频报警器的设计	101	小结	169
7.1 任务内容与分析	101	实练	169
7.2 任务实施	102	任务 11 简易波形发生器的设计	171
7.3 任务相关知识	103	11.1 任务内容与分析	171
7.3.1 定时器/计数器的基本概念	103	11.2 任务实施	172
7.3.2 定时器/计数器的基本结构与		11.3 任务相关知识	173
工作原理	104	11.3.1 D/A 转换器基础知识	174
7.3.3 定时器/计数器计数功能的应用——		11.3.2 读写片外 RAM 指令	175
大会发言限时器	109	11.3.3 并行 D/A 转换器 DAC0832	176
小结	111	小结	179
实练	111	实练	179
任务 8 数码秒表的设计	113	任务 12 5 V 直流数字电压表的设计	181
8.1 任务内容与分析	113	12.1 任务内容与分析	181
8.2 任务实施	114	12.2 任务实施	182
8.3 任务相关知识	117	12.3 任务相关知识	187
8.3.1 LED 数码管	117	12.3.1 A/D 转换器基础知识	187
8.3.2 ROM 查表指令	120	12.3.2 并行 A/D 转换器 ADC0809	189
8.3.3 数码管静态显示和动态显示	121	12.3.3 串行 A/D 转换器 ADC0832	192
小结	129	小结	197
实练	129	实练	197
任务 9 电子密码锁的设计	130	任务 13 液晶显示广告牌的设计	198
9.1 任务内容与分析	130	13.1 任务内容与分析	198
9.2 任务实施	131	13.2 任务实施	199
9.3 任务相关知识	138	13.3 任务相关知识	202
9.3.1 键盘接口	138	13.3.1 字符型液晶显示器的基本知识	202
9.3.2 独立式键盘的工作原理	140	13.3.2 1602 字符型 LCD 简介	203
9.3.3 矩阵式键盘的工作原理	142	13.3.3 1602 字符型 LCD 的使用	204
小结	147	13.3.4 1602 字符型 LCD 的设计实例	209
实练	147	小结	212
任务 10 远距离控制记分牌的设计	148	实练	212
10.1 任务内容与分析	148	附录 A ASCII 码表	213
10.2 任务实施	149	附录 B MCS 51 指令速查表	214
10.3 任务相关知识	154	部分参考答案	217
10.3.1 串行通信概述	154	参考文献	218

任务1 控制单只 LED 闪烁发光

学习目标:

1. 了解单片机的概念、种类、特点、应用及开发流程等。
2. 掌握 51 单片机的最小应用系统。
3. 熟悉单片机仿真软件和开发工具的使用。

关键点:

1. AT89S51 芯片及引脚功能。
2. 单片机时钟电路及复位电路的连接方法。
3. 单片机仿真软件 Keil μ Vision3 的使用。
4. 仿真器、编程器的认识与使用。

1.1 任务内容与分析

1. 任务内容

在日常生活中经常看到有些电器上带有的 LED(发光二极管)指示灯有节奏地闪动,任务1就以单片机控制一只 LED 闪烁(简称为闪烁灯)为例,简单介绍单片机。

2. 任务分析

要用单片机控制 LED 闪烁,那么,LED 必须和单片机的某个引脚相连,否则单片机就无法控制它,那么应和哪个引脚相连呢? 89S51 单片机有 40 个引脚,设计该芯片的 Intel 公司已经给每个引脚起好名字,且每个引脚都有具体功能,可以将 LED 和 89S51 单片机的 1 脚(P1.0)相连,如图 1-1 所示,图中, R1 为限流电阻。当 P1.0 输出低电平时,使发光二极管正向偏置,就会点亮 LED; P1.0 输出高电平时,LED 就熄灭。如果 P1.0 输出电平在高低电平之间不停转换,则 LED 会产生闪烁。也就是说,先点亮 LED 一段时间之后熄灭 LED,再延时一段时间后点亮 LED,如此反复。

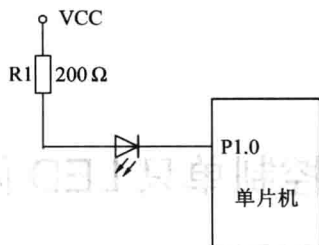


图 1-1 闪烁灯示意图

那如何让 P1.0 脚输出高电平或低电平呢？现实生活中叫人做事，说一声就可以，这叫发布命令。同样，要计算机做事，也得向计算机发布命令。计算机能听得懂的命令，称之为计算机的指令。在 89S51 单片机中，让一个引脚输出高电平的指令是 SETB，让一个引脚输出低电平的指令是 CLR。因此，要 P1.0 输出高电平，只要写 SETB P1.0；要 P1.0 输出低电平，只要写 CLR P1.0 即可。但计算机怎样执行指令呢？第一，计算机看不懂 SETB、CLR 之类的指令，因此要把指令先翻译成计算机能懂的方式，再让计算机去读。计算机只能懂二进制数字 0、1，因此，要把 SETB P1.0 变为(D2H, 90H)，把 CLR P1.0 变为(C2H, 90H)。至于为什么是这样的数字，是由 51 芯片的设计者——Intel 公司规定的，暂时先不去研究。第二步，在得到这些数字后，要把这些数字写进单片机的内部程序存储器中，这要借助于硬件工具——编程器。将编程器与电脑连好，运行编程器的软件，然后在编辑区内输入图 1-2 所示的目标代码，或者用仿真软件生成 HEX 文件(目标代码)再导入。取下单片机，将单片机插入做好的电路板，并接通电源。正常情况下，可见 LED 在闪烁。图 1-2 中地址栏后的 c: 表示存储器，OX 表示十六进制，00 表示空白。

存储器	
地址:	c:0x000
C:0x0000:	01 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C:0x0010:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C:0x0020:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C:0x0030:	D2 90 11 3A C2 90 11 3A 01 30 7F C8 7E 7B 00 DE
C:0x0040:	FD DF F9 22 00 00 00 00 00 00 00 00 00 00 00 00
C:0x0050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C:0x0060:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

图 1-2 闪烁灯程序目标代码

1.2 任务实施

下面一起实现闪烁灯这个任务吧！即将要解决的问题有电路的连接、程序的编写、程序的汇编、程序的调试、程序的下载运行及程序的烧写等，重点熟悉单片机仿真软件 Keil μ Vision3 的使用。

步骤 1：把电路连接妥当，如图 1-3 所示。P1.0 输出低电平(指令为 CLR P1.0)时，所连接的 LED 点亮；输出高电平(指令为 SETB P1.0)时，所连接的 LED 熄灭。如果 P1.0 输出电平在高低电平之间不停转换，则 LED 便会产生闪烁。

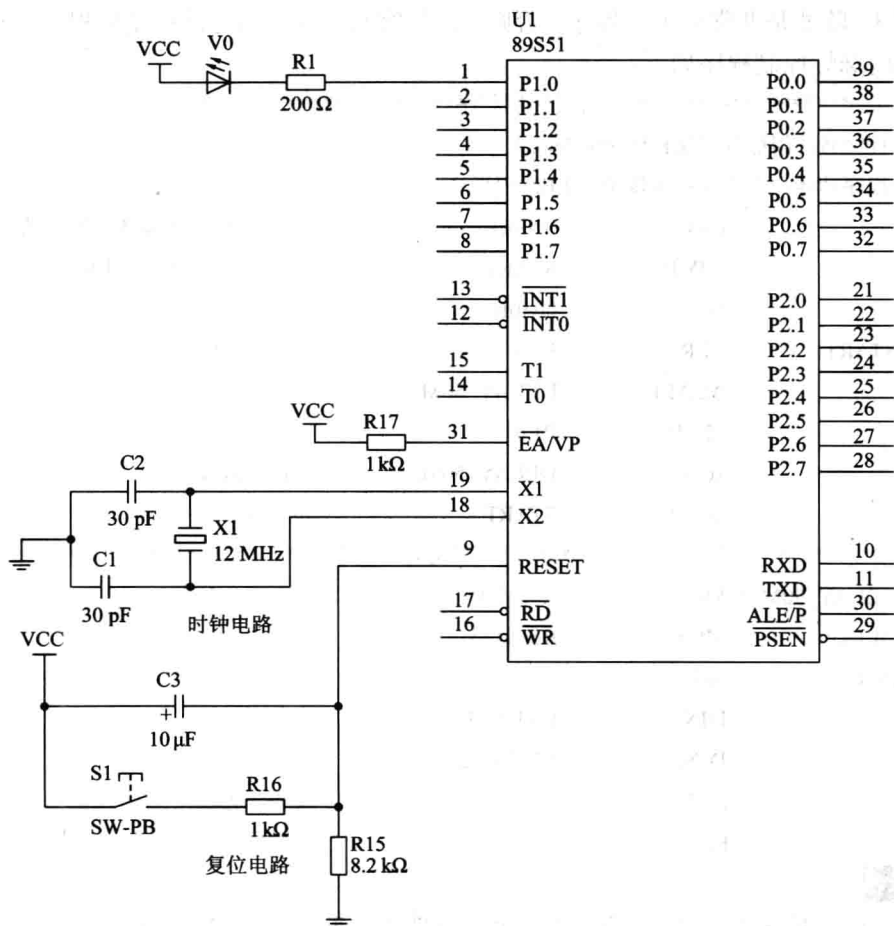


图 1-3 闪烁灯电路原理图

步骤 2: 有了电路, 又有了思路, 即可将思路画成流程图, 如图 1-4 所示。其中延时用延时子程序来实现。在“任务 5”中将详细讨论子程序的应用。

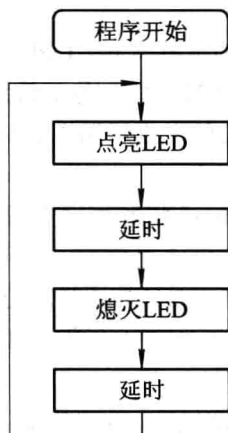


图 1-4 闪烁灯程序流程图

步骤 3: 除非是非常简单的程序, 否则, 比较容易且保险的方法还是要根据流程图来编写程序。闪烁灯控制程序如下:

```

;*****闪烁灯控制程序*****//
;程序名: 闪烁灯控制程序 xiaodeng.a
;程序功能: 用单片机实现单只 LED 闪烁

                ORG      0000H          ; 程序从地址 0000H 开始存放
                AJMP     START          ; 低端地址单元留给中断处理用
                ORG      0030H
START:          CLR      P1.0          ; 点亮 LED
                ACALL    DELAY_50MS    ; 延时 50 ms
                SETB     P1.0          ; 熄灭 LED
                ACALL    DELAY_50MS    ; 延时 50 ms
                AJMP     START          ; 重复闪烁

; *****延时子程序*****
DELAY_50MS: MOV     R7, #200          ; 延时子程序开始
DEL2:          MOV     R6, #123
DEL1:          NOP
                DJNZ     R6, DEL1
                DJNZ     R7, DEL2
                RET              ; 子程序返回
                END              ; 汇编程序结束

```

小提示:

① LED 闪烁实际上就是 LED 交替亮灭的过程, 单片机运行一条指令的时间只有几个微秒, 由于闪烁速度太快, 眼睛无法分辨, 反而看不到闪烁效果。因此, 用单片机控制 LED 闪烁时, 需要增加一定的延时时间(一般要大于等于 50 ms)。

② 在程序中添加注释是编程的良好习惯, 标注程序名和程序功能有助于增加程序的可读性和可移植性。

③ 指令经汇编后将产生可执行的目标代码, CPU 读取代码并译码后产生控制信号, 从而完成指令的特定功能。但 ORG 和 END 指令属于伪指令, 又称指示性指令, 汇编时不产生可执行的目标代码, 只是在汇编过程中为汇编工具提供某种控制或汇编信息, 从而指示并引导汇编程序进行正确汇编。

步骤 4: 通过仿真软件 Keil μ Vision3 将源程序生成 HEX 格式的目标文件。

① 工程的建立。在项目开发中, 并不是仅有一个源程序就行了, 还要选择 CPU, 确定编译/汇编、链接的参数及指定调试方式等。为了管理方便, Keil 软件使用工程(Project)将所需设置的参数和所有文件都加在一个工程中, 只能对工程而不能对单一的源程序进行编译/汇编和链接等操作。

首先点击 Keil μ Vision3, 进入 Keil μ Vision3 界面。点击工具栏“项目”选项中的“新项目”, 准备开始建立自己的项目, 如图 1-5 所示。

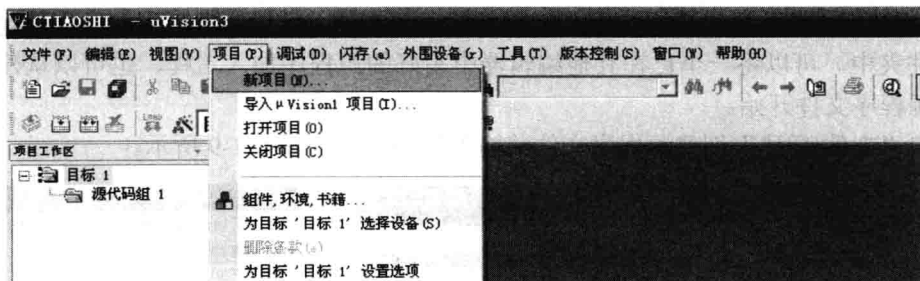


图 1-5 建立项目

给工程起一个名字，并选择保存工程文件的目录，然后单击“保存”即可，如图 1-6 所示。

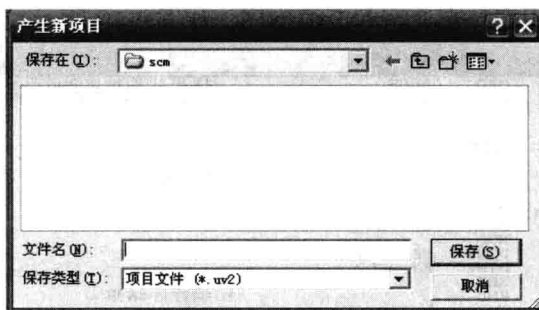


图 1-6 创建新工程

进入下一步，选择目标 CPU。Keil 支持 400 余种以 80C51 为内核的单片机系列，这里选择 Atmel 公司生产的 AT89S51。双击“Atmel”，找到“AT89S51”后点击，然后选择“确定”，如图 1-7 所示。

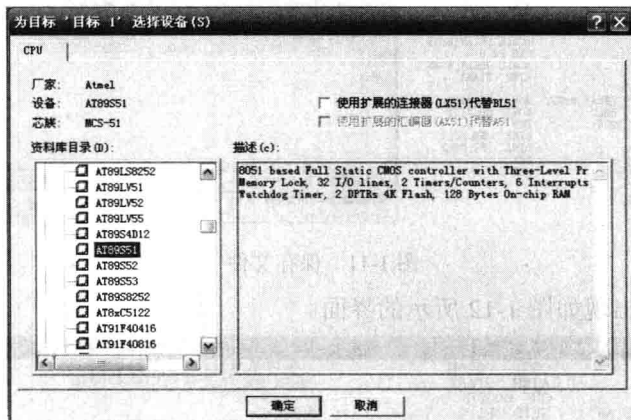


图 1-7 选择目标 CPU

返回到主界面，此时会出现下面的对话框，询问是否要将 8051 标准启动代码的源程序拷贝到工程所在文件夹，并将这一文件加入到项目中。这是新版本 Keil 软件新增加的功能。在使用 C 语言编程且要修改启动代码时可以选择“是”，在使用汇编语言进行编程时可以选择“否”，如图 1-8 所示。选择好后回到主界面，此时工程已建立。

② 新建、添加文件。接下来再为工程新建一个文件。一般应把工程与源文件建在同一个文件夹中。可以将一个已在其他编辑器中写好的源程序加入工程，也可以从建立一个空白的源程序文件开始。

点击“文件/新建”创建源程序文件并输入程序代码，如图 1-9 所示。



图 1-8 询问界面

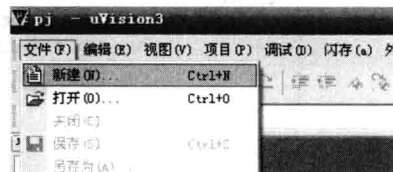


图 1-9 创建源程序文件

在文本框中输入源程序。在此写入汇编程序，如图 1-10 所示。

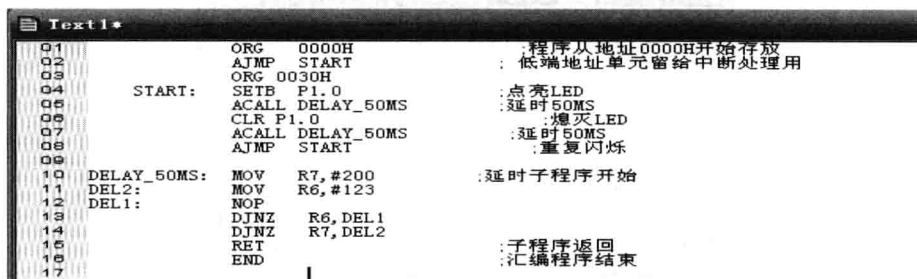


图 1-10 输入源程序

然后点击“文件/另存为”保存文件，为文件起一个名字，这里命名为“xiaodeng.a”。注意文件的扩展名不能省略，且后缀必须是“.a”，如图 1-11 所示。

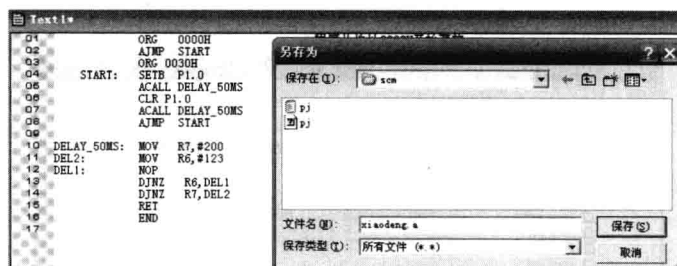


图 1-11 保存文件

点击“保存”，出现如图 1-12 所示的界面。



图 1-12 保存后的源文件

点击“视图”菜单栏中的“项目窗口”选项，弹出项目工作区(若已有项目工作区，则不用点击“视图”菜单栏中的“项目窗口”选项)，将项目工作区中“目标1”前面的“+”号展开，在其下面的字符“源代码组1”上点击鼠标右键，再点击“添加文件到组‘源代码组1’”，如图1-13所示。

弹出如图1-14所示的对话框，在文件类型中点击“Asm 源文件”，找到刚才新建的“xiaodeng.a”文件，然后点击“添加”。



图 1-13 为工程添加文件



图 1-14 加入文件对话框

只需要加入一次就够了，如果再次重复加入，将出现如图1-15所示的提示框，这时直接点击“确定”即可。

添加文件完毕，文件夹“源代码组1”前面就有了一个“+”号，点击该“+”号展开后，下面就出现了一个名为“xiaodeng.a”的文件，如图1-16所示，说明已经成功加入文件。

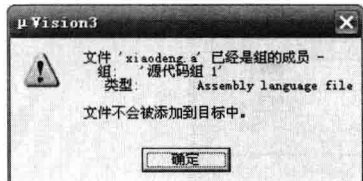


图 1-15 重复加入文件的提示

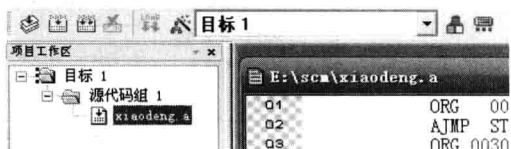


图 1-16 成功加入文件

③ 工程的设置。工程建立好后，还要对工程进行进一步的设置，以满足每个工程的个性化要求。

其具体做法是：将鼠标移到“目标1”上，点击右键，再点击“为目标‘目标1’设置选项”(如图1-17所示)，弹出如图1-18所示的窗口。

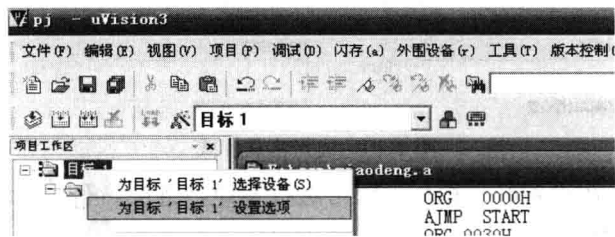


图 1-17 为目标设置选项

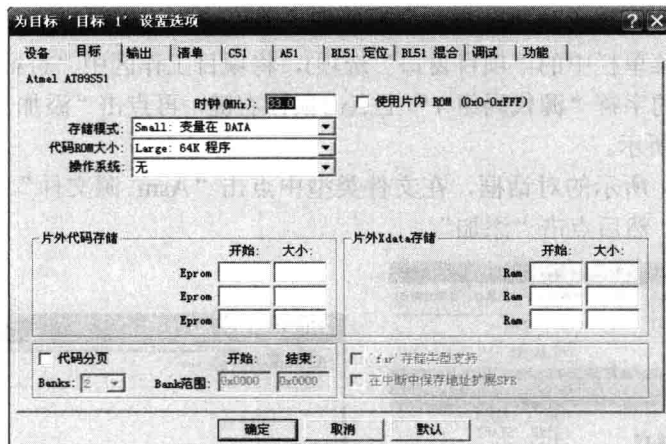


图 1-18 为目标设置选项的界面

点击“输出”菜单,在新弹出的如图1-19所示的窗口中,一定要确保“产生HEX文件”前面的小方格内有“√”,即选中该项。在“执行的名字”方框中为生成的 HEX 文件起名字,在此输入与源文件一样的名字“xiaodeng”,然后点击“确定”。

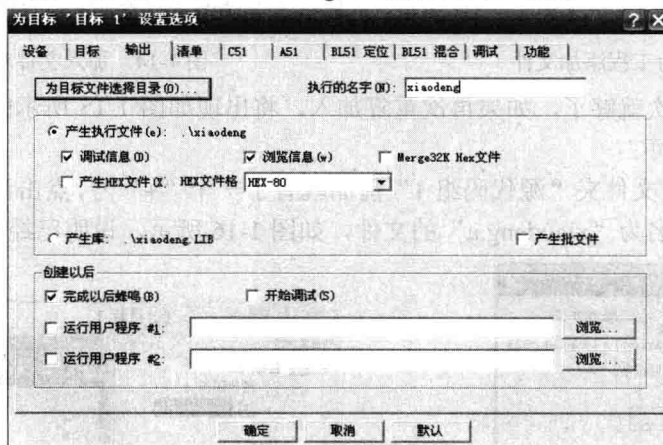


图 1-19 “输出”菜单的设置

点击“调试”菜单,如图 1-20 所示。在此菜单中可选择是使用硬件仿真,还是软件仿真。

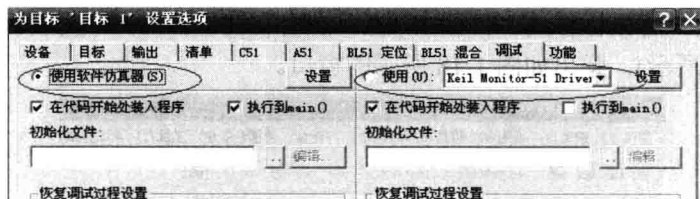


图 1-20 选择仿真模式

④ 编译、链接。在设置好工程后,即可进行编译、链接。有关编译、链接、工程设置的工具栏按钮如图 1-21 所示。

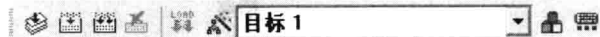


图 1-21 有关编译、链接、工程设置的工具栏

图 1-21 中, 各按钮说明如下:

编译/汇编当前文件: 使用 A51 汇编器或 C51 编译器对源程序进行编译/汇编处理, 得到可浮动地址的目标代码。

建立目标文件: 根据编译/汇编得到的目标文件调用有关库模块, 链接产生绝对地址的目标文件。如果在上次编译/汇编过后, 又对源程序做了修改, 应先对源程序进行编译/汇编, 然后再链接。

重建全部: 对工程中的所有文件先进行重新编译/汇编处理, 再进行链接以产生目标代码。使用该按钮可以防止由于一些意外情况(如系统日期不正确)造成的源文件与目标代码不一致。

停止建立: 在建立目标文件的过程中, 可以单击该按钮停止这一工作。

点击 对项目文件进行编译时, 将使用编译链接装载功能对源程序进行编译和语法检查, 编译过程中的信息将出现在输出窗口中的 Build 页中。如果源程序中有语法错误, 则会出现错误报告。双击错误报告行, 可以定位到错误的源程序的相应行。应注意, 若有错误时必须按提示排错。当出现图 1-22 所示的画面时, 说明编译成功, 目标文件 “xiaodeng.hex” 已经生成, 结果中还给出了内部 RAM 使用量(8 B)、外部 RAM 使用量(0 B)、目标代码量的大小(68 B)。

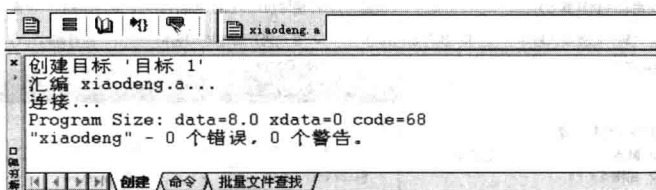


图 1-22 编译、链接后得到正确的结果

在这一过程中, 还会生成一些其他文件, 产生的目标文件被用于 Keil 软件的仿真与调试, 此时可进入下一步调试工作。

步骤 5: 程序固化(芯片烧录)。通过编译链接装载后形成的*.HEX 目标文件, 可通过编程器烧写到单片机中, 然后将烧写好的单片机插到电路板上, 一接通电源就可以工作了。

小知识:

一般来讲, 调试单片机硬件电路和汇编程序时很难一次成功。因此, 必须通过反复调试, 不断修改硬件和软件, 直到运行结果完全符合要求为止。若用编程器反复烧写程序, 这显然是很麻烦的。另外, 编程器是一次写入完整的程序, 写入后不能进行单步、断点、指定到运行处等调试。而使用仿真器联机在线调试, 可实现单步、断点、全速等操作, 还可随时观察各个寄存器的内部状态等, 从而在开发过程中大大提高调试程序的效率。等程序调试成功后, 再用编程器将程序烧写到单片机中, 这样会节省不少时间。

步骤 6: 联机仿真调试。参照图 1-23, 将单片机仿真器、单片机应用系统、(目标板)及 PC 机连接起来。

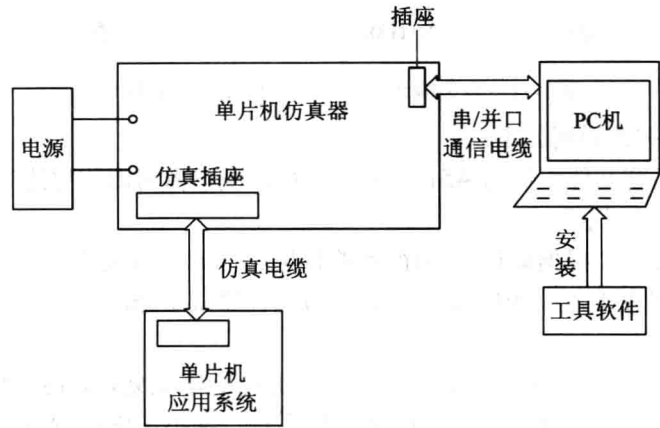


图 1-23 仿真器的连接方法

在完成步骤 4 的基础上，即程序编译装载无误后，就可以进行程序的仿真调试运行了。

在“为目标‘目标 1’设置选项”窗口中，点击“调试”菜单，可选择仿真模式，即硬件仿真或软件仿真。在此，因要连接目标板，故选择硬件仿真。点击硬件仿真选项后面的“设置”选项，在此对话框中选择串口和波特率。串口根据所连接的电脑来决定，如图 1-24 所示。

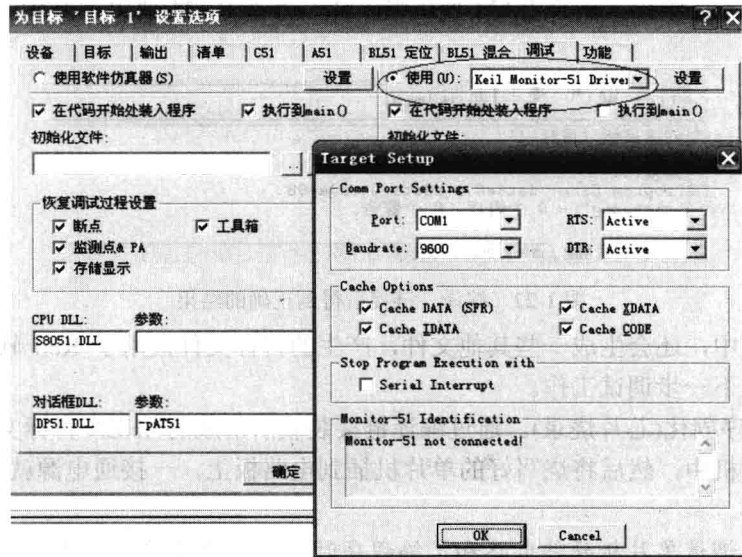


图 1-24 硬件仿真模式的设置

点击“调试”菜单下的“启动/停止调试”，进入调试界面，如图 1-25 所示。

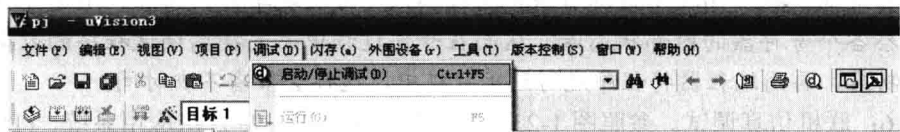


图 1-25 启动调试界面