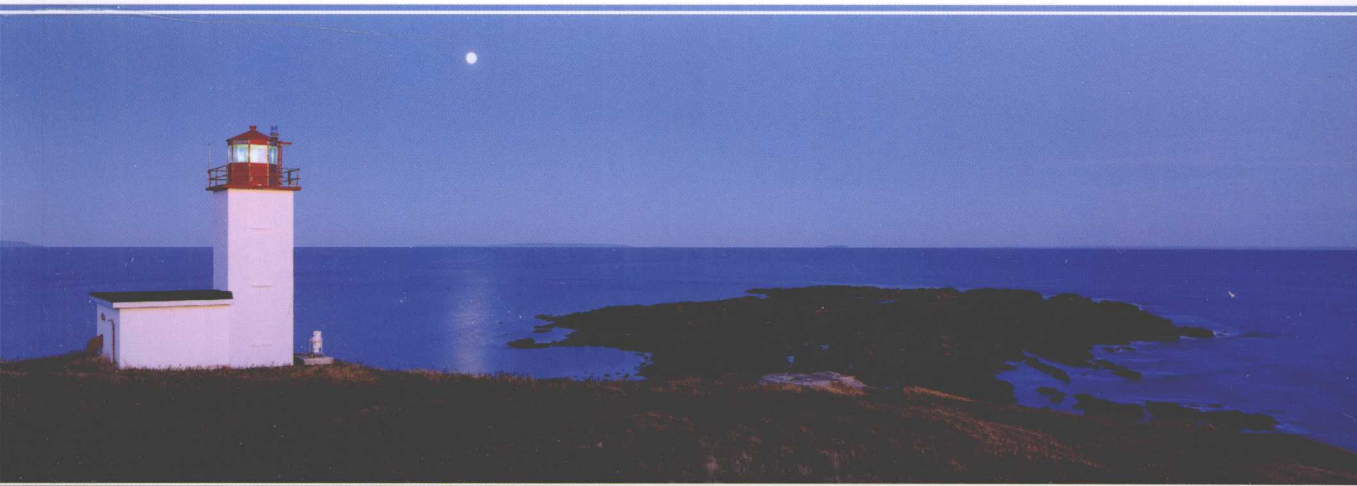


高等职业教育电子技术技能培养规划教材

Gaodeng Zhiye Jiaoyu Dianzi Jishu Jineng Peiyang Guihua Jiaocai

单片机 实用技术教程

杨宏丽 主编 王静霞 柴继红 副主编 高林 主审



An Applied Technique Course Book
for Micro Controller Unit

引入项目教学

应用中学习基础知识

培养实用技能

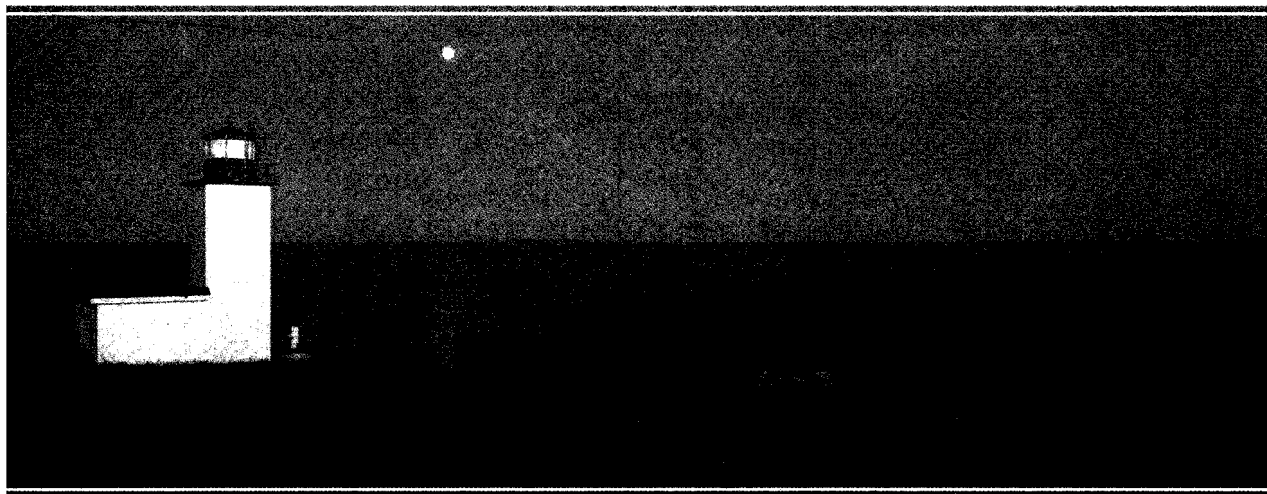


人民邮电出版社
POSTS & TELECOM PRESS

高等职业教育电子技术技能培养规划教材
Gaodeng Zhiye Jiaoyu Dianzi Jishu Jineng Peiyang Guihua Jiaocai

单片机 实用技术教程

杨宏丽 主编 王静霞 柴继红 副主编 高林 主审



An Applied Technique Course Book
for Micro Controller Unit

人民邮电出版社

北京

图书在版编目(CIP)数据

单片机实用技术教程 / 杨宏丽主编. —北京: 人民邮电出版社, 2009.10
高等职业教育电子技术技能培养规划教材
ISBN 978-7-115-20501-8

I. 单… II. 杨… III. 单片微型计算机—高等学校: 技术学校—教材 IV. TP368.1

中国版本图书馆CIP数据核字(2009)第057437号

内 容 提 要

本书由实际问题入手, 通过技能训练引入相关知识点。全书共分为9个项目, 每个项目由若干具体任务组成, 按照“任务分析”→“任务实现”→“相关知识”的思路编排, 使学生能够轻松掌握单片机的实用技术。

本书系统地介绍单片机的基础知识、单片机的I/O端口、定时器和中断、单片机通信系统、显示接口、键盘接口、A/D和D/A接口, 并给出交通灯控制系统及智能温度测控系统的完整设计过程。

本书可作为高职高专院校电子类、机电类等相关专业单片机课程的教材, 也可作为广大电子制作爱好者的自学用书。

高等职业教育电子技术技能培养规划教材

单片机实用技术教程

-
- ◆ 主 编 杨宏丽
 - 副 主 编 王静霞 柴继红
 - 主 审 高 林
 - 责任编辑 潘春燕
 - 执行编辑 赵慧君

 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷

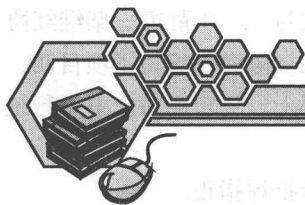
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 406千字
印数: 1—3 000册
 - 2009年10月第1版
2009年10月河北第1次印刷

ISBN 978-7-115-20501-8/TP

定价: 27.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154



前言

单片机从其诞生之日起,就以其体积小、功能强、可靠性高、应用面广等优点迅速占领了自动化控制领域,逐步取代了传统电子线路构成的控制系统,并成为电子系统智能化的最好工具,是从事工业控制、家用电器、仪器仪表、机电控制等领域工作的技术人员必须掌握的技术。为适应这一要求,全国各高职高专院校电子类及相关专业普遍开设了单片机类课程。单片机技术已经成为这类专业学生的基本技能要求,一本适合于高职高专院校培养目标的好的单片机教材也因此显得尤为重要。

作者在编写教材过程中,总结了深圳职业技术学院及其他院校不同专业的单片机技术课程的教学经验,力求在内容、结构、项目完成与相关知识的衔接方面充分体现高职高专教育的特色,同时将一些生动的实际开发案例融入到教材中,以提高学生的学习兴趣,使他们能够轻松掌握单片机的基本开发技能。本书和其他相关教材相比,具有以下特点。

- 创新体系结构。作者突破传统的教材模式,对本书的体系结构进行精心设计。各项目训练目的明确、教学重点突出、设计任务针对性强;每个项目中均设计了2~3个任务,全书的工作任务具有连贯性,符合循序渐进的原则。
- 基于工作任务的学习模式。书中每个项目均从“任务”出发,以工作任务为驱动,按照“任务分析”→“任务实现”→“相关知识”的思路编排。
- 书中穿插安排了一些“小知识”、“小提示”、“小问答”等提示总结性文字,打破了枯燥无味的叙述模式,调动了读者的学习积极性。

本书参考学时 92 学时,其中理论环节为 48 学时,实践环节为 44 学时,各部分的参考学时见下面的学时分配表。

项 目	课 程 内 容	学 时 分 配	
		讲 授	实 训
项目一	让单片机工作起来	4	4
项目二	编写应用程序控制单片机的 I/O 端口	10	8
项目三	学习使用单片机的内部资源——定时器和中断	8	6
项目四	建立单片机通信系统	6	6
项目五	学习显示接口	8	8
项目六	学习键盘接口	6	6
项目七	学习 A/D 接口和 D/A 接口	6	6
项目八	设计交通指示灯控制系统		课外完成
项目九	设计智能温度测控系统		课外完成
课时总计		48	44

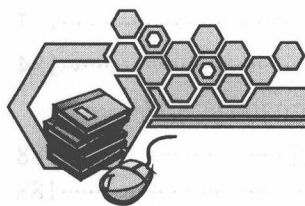


杨宏丽对本书的编写思路与任务设计进行了总体策划，并编写了项目二。王静霞、柴继红协助完成上述工作，王静霞编写了项目一，柴继红编写了项目三、项目八。陈海松编写了项目四、项目九，刘丽莎编写了项目五，何惠琴编写了项目六，韩君编写了项目七。杨宏丽对全书进行统稿。北京联合大学高林对全书进行了审定。

由于编者水平有限，书中难免存在错误和疏漏之处，敬请广大读者批评指正。

编 者

2009年7月



目 录

项目 1 让单片机工作起来	1	秒表	64
任务 1 认识单片机的开发环境	1	任务分析	64
任务分析	2	任务实现	64
任务实现	2	相关知识	68
相关知识	9	任务 2 实现交通指示灯模拟	
任务 2 控制 8 个信号灯亮灭		控制	78
闪烁	16	任务分析	78
任务分析	16	任务实现	79
任务实现	16	相关知识	86
相关知识	19	习题	93
习题	24	项目 4 建立单片机通信系统	95
项目 2 编写应用程序控制单片机的 I/O 端口	25	任务 1 设计双机通信测试	
任务 1 设计 I/O 端口直接驱动的汽车转向灯	26	电路	95
任务分析	26	任务分析	95
任务实现	27	任务实现	96
相关知识	30	相关知识	100
任务 2 控制模拟流水灯的显示	41	任务 2 设计 PC 控制的模拟交通	
任务分析	41	指示灯	115
任务实现	41	任务分析	115
相关知识	46	任务实现	116
任务 3 显示基于 8×8 点阵的多模式广告	50	相关知识	121
任务分析	50	习题	125
任务实现	51	项目 5 学习显示接口	127
相关知识	56	任务 1 设计七段数码管广告牌	127
习题	61	任务分析	127
项目 3 学习使用单片机的内部资源——定时器和中断	63	任务实现	128
任务 1 设计定时器实现的简易		相关知识	132
		任务 2 设计正计时的电子秒表	136
		任务分析	136
		任务实现	137
		相关知识	141



任务 3 设计基于字符液晶的秒表	144
任务分析	144
任务实现	144
相关知识	149
习题	156
项目 6 学习键盘接口	157
任务 1 设计 4 人抢答器	157
任务分析	157
任务实现	158
相关知识	161
任务 2 设计数码管显示的密码开关	164
任务分析	164
任务实现	164
相关知识	167
习题	174
项目 7 学习 A/D 接口和 D/A 接口	176
任务 1 用 DAC0832 实现方波、三角波、锯齿波波形发生器	177
任务分析	177
任务实现	177
相关知识	184
习题	200
项目 8 设计交通指示灯控制系统	201
项目分析	201
项目实现	203
相关知识	213
习题	219
项目 9 设计智能温度测控系统	220
项目分析	220
项目实现	221
相关知识	231
习题	241
附录 MCS-51 指令表	242

项目 1

让单片机工作起来

【学习目标】

- (1) 掌握单片机、单片机应用系统和单片机最小系统概念
- (2) 掌握单片机引脚及功能
- (3) 掌握单片机的内部结构
- (4) 学会使用单片机开发环境调试应用程序

【训练任务】

- (1) 单片机开发环境的连接和使用
- (2) I/O 端口直接驱动的发光二极管
- (3) 如何用程序控制单片机硬件工作

【重点难点】

- (1) 单片机开发环境的使用
- (2) 单片机概念的理解和引脚的应用方法

【参考学时】 8 学时

任务 1 认识单片机的开发环境

单片机在控制、测量领域有着广泛的应用。基于单片机的应用系统一般包括硬件电路和程序设计两个部分，这两个部分通过调试最终能够实现一个完整的系统功能。在调试过程中，单片机开发环境是必不可少的工具，通过开发环境可以对电路和程序进行纠错、调试、运行。掌握单片机开发环境的使用技能是学习单片机的第一步。



任务分析

单片机的开发环境用来调试单片机应用系统的硬件和软件，以完成系统的特定功能，并可以将程序下载到系统电路中。为了更好地理解开发环境的使用，我们首先制作一块简单的单片机电路板——采用单片机控制的信号灯系统电路。电路中用一个发光二极管模拟信号灯，以单片机的 I/O 端口控制其亮灭，通过在开发环境下编程和调试程序，实现该信号灯的闪烁效果。

任务实现

步骤 1——硬件电路设计

单片机控制信号灯硬件电路如图 1.1 所示，包括单片机、复位电路、晶振电路、电源电路及用一个发光二极管模拟信号灯的控制电路。包括复位电路、晶振电路和电源电路的单片机系统是保证单片机工作起来所需要的最小的电路结构，我们称这样的电路为单片机的最小应用系统。为了实现一个信号灯的亮灭控制，在单片机最小应用系统的基础上，增加了 P1.0 端口连接的发光二极管。当 P1.0 引脚输出低电平时，发光二极管点亮；当 P1.0 引脚输出高电平时，发光二极管熄灭。

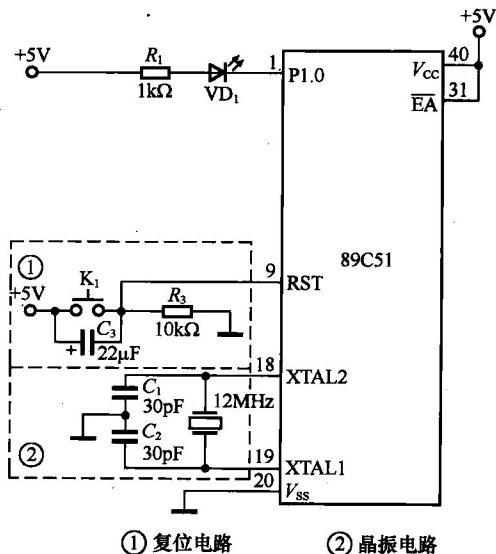


图 1.1 单片机控制信号灯硬件电路图

步骤 2——准备元器件并制作电路板

单片机控制信号灯电路的元器件清单如表 1.1 所示。



表 1.1

单片机最小应用系统的元器件清单

元器件名称	参 数	数 量	元器件名称	参 数	数 量
IC 插座	DIP40	1	按键		1
单片机	87E51 或 89C51	1	电阻	1k Ω	2
晶体振荡器	6MHz 或 12MHz	1	电阻	470 Ω	1
瓷片电容	20pF	2	电解电容	22 μ F	1
发光二极管		1			

在万能板上按电路图焊接元器件，完成电路板制作，图 1.2 所示为焊接好的电路板硬件实物。

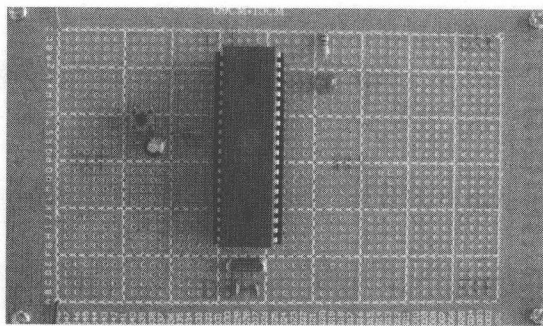


图 1.2 单片机控制信号灯电路板硬件实物



小知识

- 焊接单片机应用系统硬件电路时，为了调试方便，一般不直接将单片机芯片焊接在电路板上，而是焊接一个与单片机芯片引脚相对应的直插式插座，以方便芯片的拔出与插入。
- 晶振电路焊接时尽可能靠近单片机芯片，以减少电路板分布电容，使晶体振荡器频率更加稳定。
- 发光二极管及其他显示元器件最好分布在单片机的右侧，这样可避免在仿真调试时，被联结仿真器与用户板的仿真排线所遮挡。
- 元器件分布时，要考虑为后面不断增加的元器件预留适当的位置，且元器件引脚不宜过高。

步骤 3——准备单片机开发环境

图 1.3 所示为单片机系统开发所需的装备及其相互连接的示意图。装备包括计算机、单片机在线仿真器、开发工具软件。如果单片机开发系统具备固化程序的功能，可直接将调试好的程序代码下载到单片机应用系统的程序存储器中，若开发系统不具备固化的功能，还需配备专用的编程器。

这里以 ME-52 型仿真器和 MedWin 工具软件为例，建立单片机开发环境。如果读者使用的是不同系列的单片机开发工具，也可参考下列步骤，依照使用说明书建立单片机开发环境。

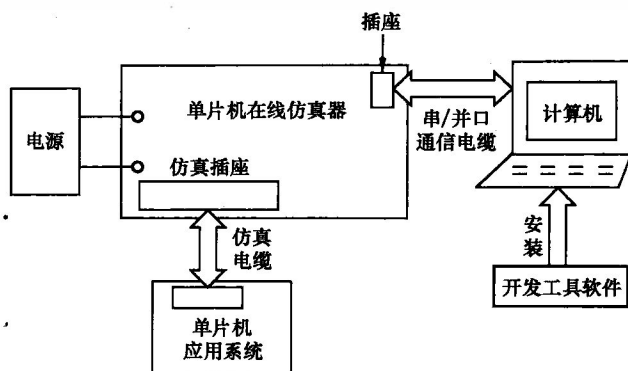


图 1.3 单片机开发装备示意图

- ① 利用并口线将计算机和单片机在线仿真器连接起来。
- ② 用带插头的仿真扁平线将单片机在线仿真器与实验室提供的试验板(箱)或自己设计制作的单片机应用系统连接在一起,管脚的对应关系一定不能搞错。
- ③ 打开在线仿真器电源。
- ④ 在计算机上启动 MedWin 工具软件,此时单片机开发环境就建立起来了。

步骤 4——编写控制程序

一个单片机应用系统由硬件系统和软件系统两部分组成,二者缺一不可。图 1.2 所示的只是单片机应用系统的硬件实物,实际上必须在单片机芯片的内部存储器中烧录预先编写好的信号灯控制程序,才能看到信号灯的闪烁效果。

信号灯闪烁源程序如下:

```

; *****信号灯控制程序*****
; 程序名: 信号灯控制程序 PM1_1.asm
; 程序功能: 控制信号灯闪烁

        ORG    0000H           ; 将程序从地址 0000H 开始存放在存储器中
START:   CLR    P1.0           ; P1.0=0, 点亮信号灯
        ACALL  DELAY          ; 调用延时子程序
        SETB   P1.0           ; P1.0=1, 熄灭信号灯
        ACALL  DELAY          ; 调用延时子程序
        AJMP   START          ; 返回, 重复闪烁过程

; *****延时子程序 DELAY*****
; 程序名: 延时子程序 DELAY
; 程序功能: 延时一段时间, 延时时间长短主要由 R3、R4 的值决定, 可尝试修改

DELAY:   MOV    R3, #0FFH      ; 延时子程序
DEL2:    MOV    R4, #0FFH
DEL1:    NOP
        DJNZ   R4, DEL1
        DJNZ   R3, DEL2
        RET                      ; 子程序返回
        END                      ; 汇编结束

```



步骤5——输入、调试并运行程序

① 启动中文版 MedWin 软件，计算机屏幕上会出现如图 1.4 (a) 所示的启动窗口；如果计算机与仿真器之间没有连接好，或仿真器还没有开启电源，则出现如图 1.4 (b) 所示的窗口，此时只能进入“模拟仿真”状态。

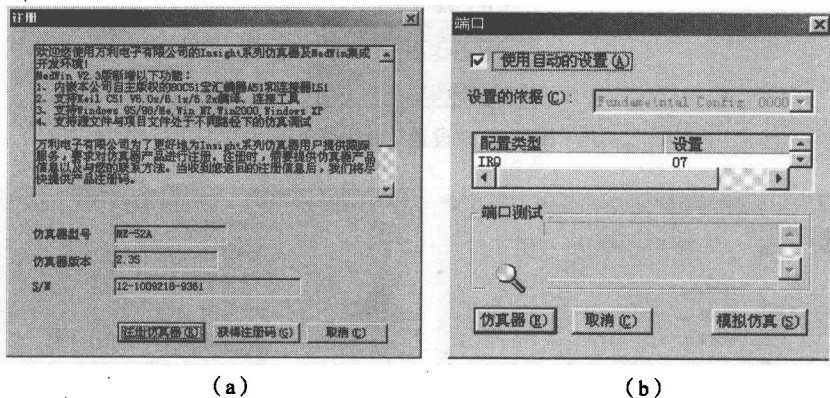


图 1.4 MedWin 工具软件窗口

在图 1.4 (a) 所示的窗口中单击“取消”按钮或在图 1.4 (b) 所示的窗口中单击“模拟仿真”按钮进入 MedWin 集成开发环境窗口，如图 1.5 所示。

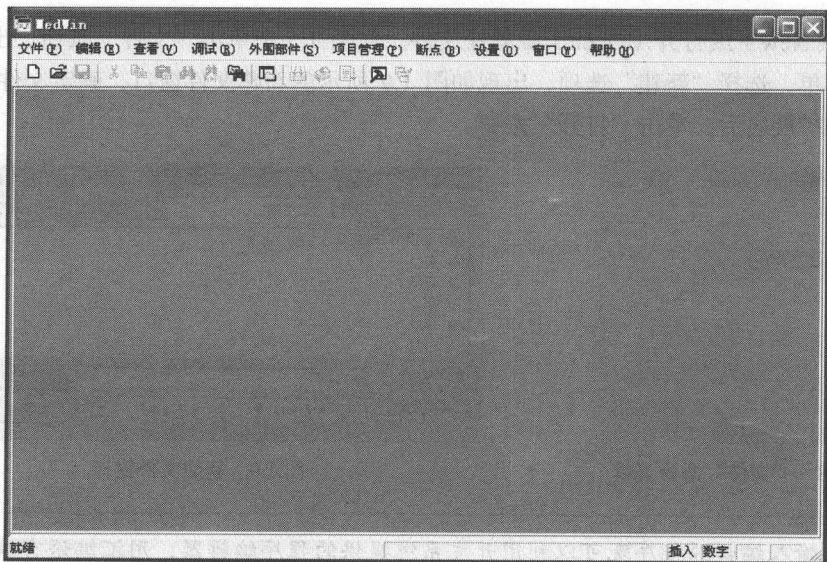


图 1.5 集成开发环境窗口

② 如果 MedWin 软件是第一次在计算机中安装运行，需先进行编译、汇编、连接配置，以后使用时就不需再配置了。在图 1.5 所示的窗口中单击“设置”按钮，出现如图 1.6 所示的命令行菜单，选择“设置向导”选项，出现如图 1.7 (a) 所示的编译、汇编、连接配置窗口。单击“下一步”按钮，出现如图 1.7 (b) 所示的窗口，在该窗口中可设置系统头文件路径和系统库文件路径。



选择源程序扩展名,若采用汇编语言编写源程序,应选择 ASM,然后单击“完成”按钮,返回如图 1.5 所示的窗口。

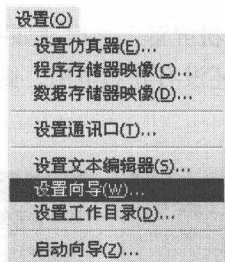


图 1.6 “设置”命令菜单

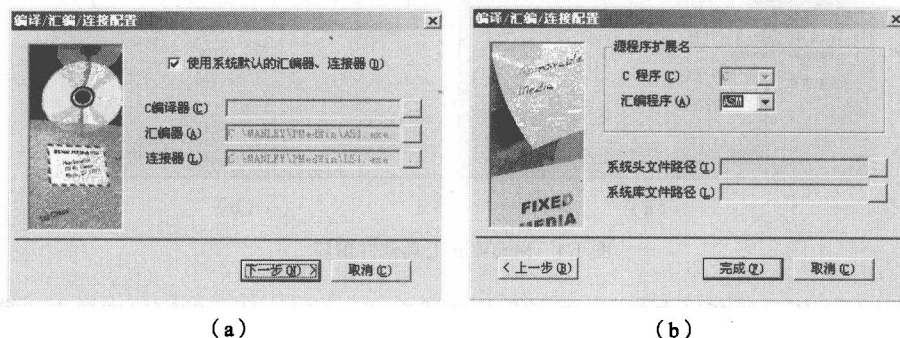


图 1.7 编译、汇编、连接配置窗口

③ 新建 (New) 或打开 (Open) 源程序文件。在图 1.5 中单击“文件”按钮,出现如图 1.8 所示的命令菜单,选择“新建”选项,出现如图 1.9 所示的新建文件窗口,确定文件存放路径,输入文件名和扩展名后,单击“打开”按钮。

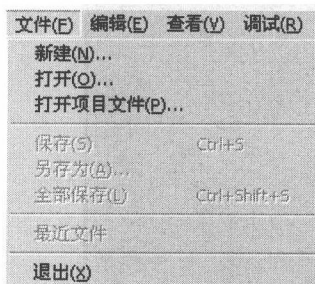


图 1.8 “文件”命令菜单

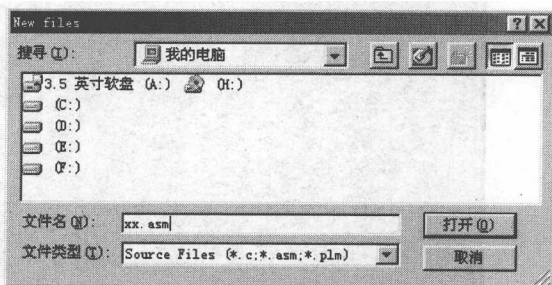


图 1.9 新建文件窗口



小提示

输入编辑源程序既可以利用开发系统提供的程序编辑器,用汇编语言编辑扩展名为 .ASM 的源程序,也可以将在 Windows 或 DOS 环境下编辑的源程序复制过来。在编制源程序时,可在每条指令的后面加必要的中英文注释,但须用分号将注释与指令间隔开来。编写程序要在西文状态下编辑,如果在中文状态下编辑源程序,在进行汇编时会带来不必要的麻烦。

④ 对源程序进行编译/汇编。就是将编好的源程序利用开发工具提供的编译/汇编功能将其转



换成由机器语言构成的目标程序，在图 1.5 所示的窗口中单击“项目管理”按钮，出现如图 1.10 所示的命令菜单。选择“编译/汇编”选项（或使用快捷键 Ctrl+F7）即可完成对当前源程序的“编译/汇编”。

⑤ 排除错误。程序经“编译/汇编”后，在屏幕下方的消息窗口，会出现纠错信息，提示是否存在错误及错误出现的位置、类型和数量等，根据信息提示对源程序的错误进行纠正，再重新进行“编译/汇编”，直至错误信息数量为“0”。

⑥ 产生代码并装入仿真器。在图 1.10 所示的“项目管理”命令菜单中选择“产生代码并装入”选项，可将生成的文件源代码装入（Load）单片机开发系统的仿真 RAM 中，如图 1.11 所示。计算机屏幕显示的源程序的前面会出现小圆点及黄色箭头，表示该程序是可执行程序并指示下一条要执行的指令。

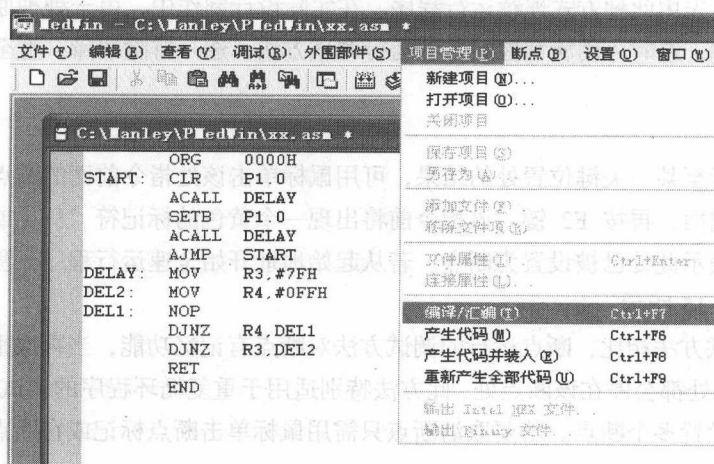


图 1.10 “项目管理”命令菜单

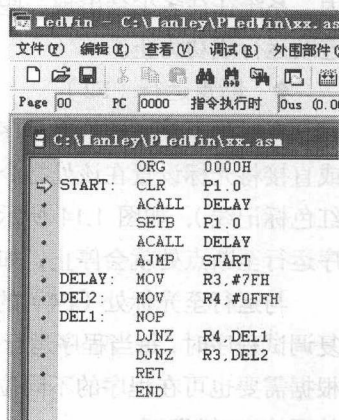


图 1.11 代码装入后的程序窗口

⑦ 运行程序。运行程序有多种操作方法，包括单步运行（F8）、跟踪运行（F7）、运行至光标处（F4）、连续运行（F9）、断点运行（F2）等，可以使用快捷键，也可在“调试”命令菜单中选择。

● 单步运行（F8）

每按一次 F8 键，黄色箭头向下移动一条指令，表示上一条指令已执行完毕。观察程序计数器 PC（Program Counter）的地址显示，其值也随之改变，如图 1.12 所示，它始终指示下一条要执行指令的地址。

● 跟踪运行（F7）

与单步运行调试相似，每按一次 F7 键，系统就执行一条指令。在前面的信号灯闪烁程序 PM1_1.asm 中含有一小段子程序，分别用单步、跟踪两种方式运行 ACALL 指令，观察程序运行过程有什么区别。

● 运行至光标处（F4）

如果想有针对性地快速观察程序运行到某条指令处的结果，可预先将光标调到该条指令处，再按 F4 键，程序将从当前 PC 所指示的位置全速运行到光

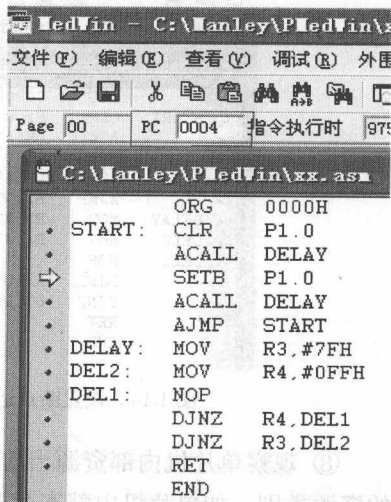


图 1.12 程序单步运行窗口



标处。此方法可加快调试程序的速度。

● 连续运行 (F9)

这种方法可完全模拟单片机应用系统的真实运行状态, 当按 F9 键时, 程序将从当前 PC 指示的地址处开始连续运行程序, 并出现如图 1.13 所示的程序运行状态指示窗口, 单击“停止”按钮可终止程序的运行。该方法便于观察程序连续运行状态下相关显示及控制过程的动态变化。因系统处于连续运行程序工作状态, 所以无法观察某条指令或某段指令的运行结果, 只能根据系统运行中所完成的显示及控制过程的变化结果来判断程序运行的正确与否。一般在程序编写完成后, 为了尽快观察程序的运行结果, 可先用此种方式连续运行程序。在实际设计制作中, 由于种种原因, 系统往往多少会出现一些故障, 这时可用所掌握的各种运行调试方法, 逐一排除故障, 直至连续运行成功为止。

● 断点运行 (F2)

有时为了快速地检查程序运行至某一关键位置处的结果, 可用鼠标单击该处指令前面的圆点或直接将光标设置在该处指令的前面, 再按 F2 键, 该指令前将出现一个黄色的标记符“!”(或红色标记线), 如图 1.14 所示, 表示此处已被设置为断点。若从起始地址开始全速运行程序, 程序运行至断点处就会停止, 如图 1.15 所示。

与运行至光标处 (F4) 的调试方法相比, 断点运行的调试方法对断点有记忆功能, 当再次重复调试程序时, 每当程序运行到此处都会停在该断点处, 此方法特别适用于重复循环程序的调试。根据需要也可在程序的不同位置设置多个断点。若要取消断点只需用鼠标单击断点标记或在断点处再按 F2 键即可。



图 1.13 程序全速运行窗口

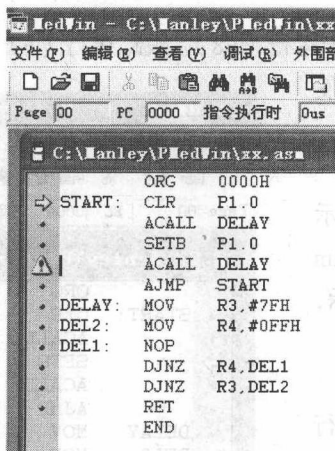


图 1.14 设置断点运行

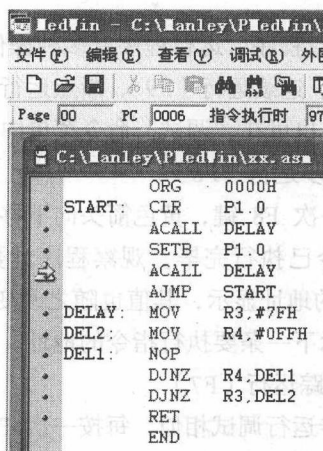


图 1.15 程序运行至断点处停止

⑧ 观察单片机内部资源当前状况。如图 1.16 所示, 在“查看”命令菜单中, 选择要查看的资源类别, 如单片机内部寄存器、特殊功能寄存器、数据缓冲区 RAM 等。图 1.17 所示为内部 RAM、寄存器和特殊功能寄存器查看窗口, 既可查看其中的内容, 也可输入新的数值对其内容进行修改。

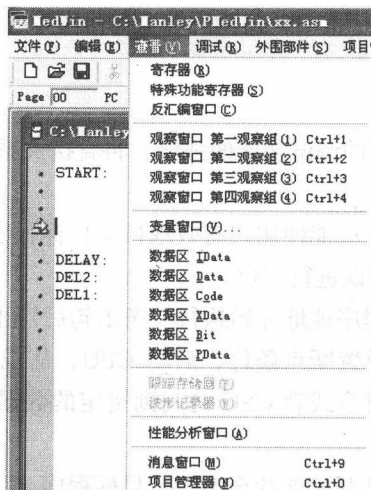


图 1.16 “查看”命令菜单

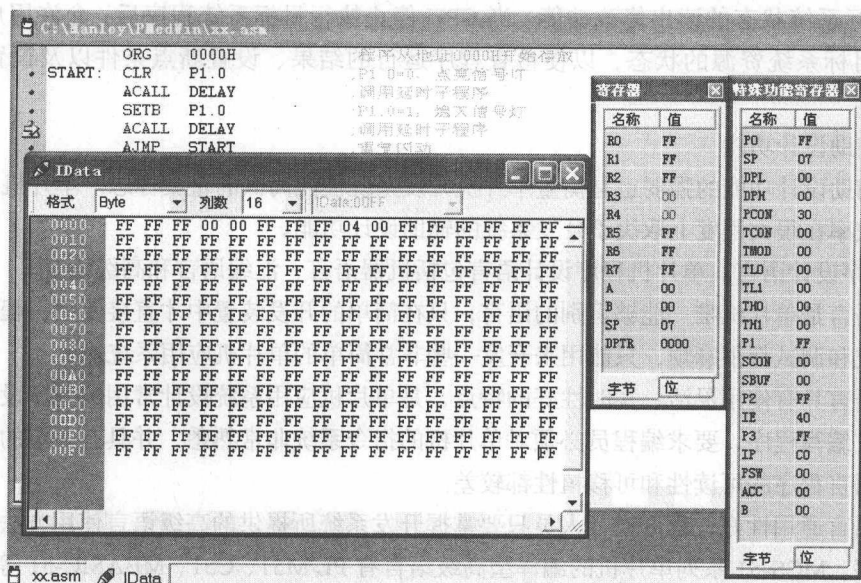


图 1.17 观察单片机内部资源

相关知识

知识 1 单片机开发系统及功能

单片机开发系统是单片机应用系统设计的必需工具，包括计算机、单片机在线仿真器、工具软件、编程器等。

单片机开发系统功能包括在线仿真、调试、软件辅助设计、目标程序固化等。

(1) 在线仿真功能

在线仿真器 (In Circuit Emulator, ICE) 是由一系列硬件构成的设备，它能仿真用户系统中的



单片机,并能模拟用户系统的 ROM、RAM 和 I/O 端口,在在线仿真状态下,用户系统的运行环境和脱机运行的环境完全“逼真”。

(2) 调试功能

开发系统对用户系统软、硬件调试功能的强弱,将直接关系到开发的效率。性能优良的单片机开发系统应具有下列调试功能。

① 运行控制功能。开发系统应能使用户有效地控制目标程序的运行,以便检查程序运行的结果,对存在的硬件故障和软件错误进行定位。

单步运行: CPU 从任意的程序地址开始执行一条语句后停止运行。

断点运行: 允许用户任意设置断点条件,启动 CPU,从规定地址开始运行,当碰到断点条件(程序地址和指定断点地址符合或者 CPU 访问到指定的数据存储器单元等条件)符合以后停止运行。

全速运行: CPU 从指定地址开始连续全速运行目标程序。

跟踪运行: 类似单步运行过程,但可以跟踪到函数内部运行。

② 目标系统状态的读出修改功能。当 CPU 停止执行目标系统程序后,允许用户方便地读出或修改目标系统资源的状态,以便检查程序运行的结果、设置断点条件以及设置程序的初始参数。

(3) 辅助设计功能

软件辅助设计功能的强弱也是衡量单片机开发系统性能高低的重要标志。单片机应用系统软件开发的效率在很大程度上取决于开发系统的辅助设计功能。

① 程序设计语言。单片机程序设计语言包括机器语言、汇编语言和高级语言。

机器语言是单片机唯一能够识别的语言,只在简单的开发装置中才直接使用,程序的设计、输入、修改和调试都很麻烦,只能用来开发一些非常简单的单片机应用系统。

汇编语言具有使用灵活、实时性好的特点,是单片机应用系统设计常用的程序设计语言。采用汇编语言编写程序,要求程序员必须对单片机的指令系统非常熟悉,并具有一定的程序设计经验。汇编语言程序的可读性和可移植性都较差。

高级语言通用性好,程序设计人员只要掌握开发系统所提供的高级语言使用方法,就可以直接编写程序。MCS-51 系列单片机的编译型高级语言有 PL/M51、C51、MBASIC-51 等。高级语言对不熟悉单片机指令系统的用户比较适用,且具有较好的可移植性,是目前单片机编程语言的主流,本书采用的是 C51 编程语言。

② 程序编译。几乎所有的单片机开发系统都能与 PC 连接,允许用户使用 PC 的编辑程序编写汇编语言或高级语言,生成汇编语言或高级语言的源文件。然后利用开发系统提供的交叉汇编或编译系统,将源程序编译成可在目标机上直接运行的目标程序,再通过 PC 的串口或并口直接传输到开发机的 RAM 中。

一些单片机的开发系统还提供反汇编功能,并可提供用户宏调用的子程序库,以减少用户软件研制的工作量。

(4) 程序固化功能

当系统调试完毕,确认软件无故障时,应把用户应用系统的程序固化到程序存储器中脱机运行,编程器就是完成这种任务的专用设备,它也是单片机开发系统的重要组成部分。