

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

单片机 基础教程

杨宏丽 王静霞 主编



Basal Tutorial of Micro Controller Unit

降低理论难度，内容通俗易懂
引入项目教学，激发学习兴趣
提供设计项目，培养工作技能

 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 基础教程

杨宏丽 王静霞 主编



Basal Tutorial
of Micro Controller Unit

人民邮电出版社

北京

图书在版编目(CIP)数据

单片机基础教程 / 杨宏丽, 王静霞主编. —北京: 人民邮电出版社, 2009.3
高等职业教育电子技术技能培养规划教材
ISBN 978-7-115-19396-4

I. 单… II. ①杨…②王… 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001645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, 每个项目都按照“项目分析”→“项目相关知识”→“项目实施”→“知识拓展”的思路编排, 使学生能够轻松掌握单片机技术的基本技能和相关知识。

全书共分为10个项目, 通过14个任务介绍了单片机及其开发环境, MCS-51单片机的相关知识, 数据传送指令、算术运算指令、逻辑运算指令、控制转移指令和位操作指令的使用方法, 编写汇编语言程序的流程和方法, 单片机内部资源和外部接口的应用等内容。

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技术课程的教材, 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

单片机基础教程

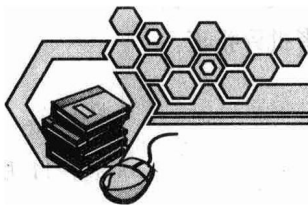
-
- ◆ 主 编 杨宏丽 王静霞
责任编辑 潘春燕
执行编辑 赵慧君
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 12.75
字数: 326千字
印数: 1—3000册
- 2009年3月第1版
2009年3月北京第1次印刷

ISBN 978-7-115-19396-4/TP

定价: 23.00元

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

反盗版热线: (010)67171154



单片机从诞生之日起,就以体积小、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。

本书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。本书与其他相关教材相比,具有以下特点。

(1) 每个项目均从任务出发,以工作任务为驱动,按照“项目分析”→“项目相关知识”→“项目实施”→“知识拓展”的思路编排内容。

本书由实际问题入手,通过技能训练引入相关知识点,理论知识体现“够用为度”的原则,突出知识的实用性;由训练项目引出相关概念、硬件设计与编程技巧,理论与实践融为一体,互相补充。

(2) 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。

本书将老师在实际教学中经常遇到的问题、在单片机系统设计中总结的经验技巧以及学生在学习常常感到困惑的地方,用言简意赅的文字,以“小知识”、“小提示”、“小问答”的形式表达出来。希望读者在学习时能够引起重视,提高学习效率。

本书参考学时为75~90学时,其中:项目1、项目2认识单片机及其开发环境、学习单片机硬件系统,参考学时为10~15学时;项目3~项目7训练汇编语言指令的运用,参考学时为20学时;项目8训练汇编语言的程序设计,参考学时为20学时;项目9学习运用单片机的内部资源,参考学时为15~20学时,项目10初步尝试操作单片机的外部接口,参考学时为10~15学时。由于完成项目制作可能要花费一定的时间,对于资料检索、准备器件、电路焊接及编程等基础性工作最好预先安排学生在课外去独立完成,既为学生营造了主动参与项目制作实践活动的锻炼机会,又能提高实训课堂技能训练效率。

在本书的编写过程中,杨宏丽对本书的编写思路与项目设计进行了总体策划,并对全书进行统稿和审稿,王静霞协助完成上述工作。其中杨宏丽编写了项目1、项目

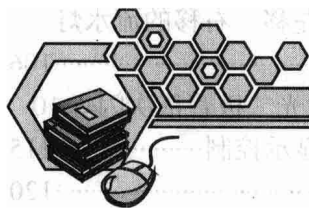


2、项目 10，王静霞编写了项目 3~项目 7，韩君编写了项目 8，柴继红编写了项目 9。

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

2008 年 11 月



项目 1 认识单片机及其开发环境..... 1

项目分析.....	1
项目相关知识.....	1
知识 1 什么是单片机.....	1
知识 2 什么是单片机应用系统.....	2
知识 3 什么是单片机开发环境.....	3
项目实施.....	5
知识拓展.....	12
知识 4 Keil 51 工具软件的使用.....	12
技能训练.....	17
项目小结.....	18

项目 2 学习单片机硬件系统..... 19

项目分析.....	19
项目相关知识.....	20
知识 1 认识 MCS-51 单片机的引脚.....	20
知识 2 单片机的 I/O 口.....	21
项目实施.....	22
知识拓展.....	24
知识 3 了解 MCS-51 系列单片机的内部结构.....	24
知识 4 熟悉单片机的存储器结构.....	25
知识 5 通过项目认识单片机汇编语言指令.....	31
技能训练.....	31
项目小结.....	32

项目 3 运用数据传送指令..... 34

项目分析.....	34
项目相关知识.....	35

知识 1 数据传送指令中的寻址方式.....	35
------------------------	----

知识 2 内部 RAM 和 SFR 的数据传送指令.....	39
--------------------------------	----

项目实施.....	43
-----------	----

知识拓展.....	46
-----------	----

知识 3 堆栈操作指令及实例.....	46
---------------------	----

知识 4 其他数据传送类指令.....	48
---------------------	----

技能训练.....	49
-----------	----

项目小结.....	51
-----------	----

项目 4 运用算术运算指令..... 54

项目分析.....	54
-----------	----

项目相关知识.....	54
-------------	----

知识 1 加法减法指令及实例.....	54
---------------------	----

项目实施.....	57
-----------	----

知识拓展.....	58
-----------	----

知识 2 乘法、除法指令及实例.....	58
----------------------	----

知识 3 其他运算指令.....	59
------------------	----

技能训练.....	60
-----------	----

项目小结.....	62
-----------	----

项目 5 运用逻辑运算指令..... 63

项目分析.....	63
-----------	----

项目相关知识.....	63
-------------	----

知识 1 循环移位指令及实例.....	63
---------------------	----

项目实施.....	65
-----------	----

知识拓展.....	67
-----------	----

知识 2 逻辑“与”、“或”、“非”和“异或”指令及实例.....	67
-----------------------------------	----

技能训练.....	69
-----------	----

项目小结.....	72
-----------	----



项目 6 运用控制转移类指令	73	任务 3 依次左移、右移的流水灯控制	106
项目分析.....	73	任务 4 8 个发光二极管依次闪烁 10 次的显示控制	115
项目相关知识.....	73	技能训练.....	120
知识 1 无条件转移指令和条件转移指令及实例.....	73	项目小结.....	122
项目实施.....	77	项目 9 运用单片机的内部资源	124
知识拓展.....	78	任务 1 每隔 0.5s 移位一次的流水灯控制.....	124
知识 2 比较转移指令和循环转移指令及实例.....	78	任务 2 把流水灯变为实用的交通灯控制系统.....	145
知识 3 子程序的调用和返回.....	79	技能训练.....	162
知识 4 散转指令及实例.....	81	项目小结.....	164
技能训练.....	81	项目 10 操作单片机的外部接口	165
项目小结.....	83	项目分析.....	165
项目 7 运用位操作指令	84	项目相关知识.....	166
项目分析.....	84	知识 1 认识八段 LED 数码管.....	166
项目相关知识.....	84	知识 2 用单片机驱动一个 LED 静态显示.....	167
知识 1 了解位寻址区和位操作的概念.....	84	项目实施.....	169
项目实施.....	86	知识拓展.....	174
知识拓展.....	87	知识 3 用单片机驱动多个 LED 动态显示.....	174
知识 2 位操作指令及实例.....	87	知识 4 单片机与按键的接口.....	177
知识 3 常用伪指令.....	89	技能训练.....	187
技能训练.....	90	项目小结.....	191
项目小结.....	91	附录 MCS-51 指令表	192
项目 8 编写汇编语言程序	92	参考文献	197
任务 1 将压缩 BCD 码 43H 拆为两个 BCD 码 04H、03H 并显示.....	92		
任务 2 用开关来控制 8 个信号灯的显示方式.....	98		

项目 1

认识单片机及其开发环境

本项目从对单片机芯片的认识开始,手把手教读者建立单片机的开发环境;通过项目实例引领读者进入开发环境,体验开发环境的使用方法。在实际操作中让读者由内到外、由浅到深,逐步认识单片机及其开发环境。

项目分析

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实现一个完整的系统功能。在调试过程中,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。本项目通过调试一个单片机应用程序来介绍单片机的基本知识及其开发环境。

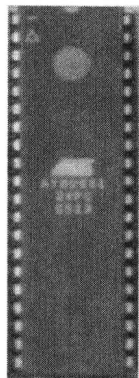
项目相关知识

知识 1 什么是单片机

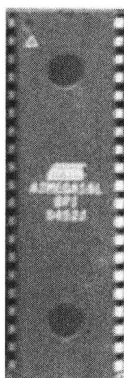
单片机(Micro Controller Unit, MCU)实际上是集成在一个芯片上的微型计算机,其芯片内部集成了中央处理器(Central Processing Unit, CPU)、随机存储器(Random Access Memory, RAM)、只读存储器(Read-only Memory, ROM)、输入/输出(Input/Output)接口电路、定时器/计数器(Time/Count)等功能部件。目前市场上广泛使用的单片机有



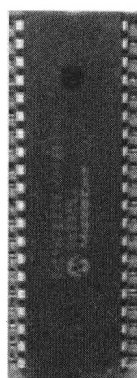
多种系列，图 1.1 所示为 3 种常用单片机芯片。本书主要涉及 MCS-51 系列单片机其最通用的结构形式是 40 引脚直插式芯片，如图 1.1 (a) 所示。



(a) MCS-51 单片机



(b) AVR 单片机



(c) PIC 单片机

图 1.1 单片机芯片实物图

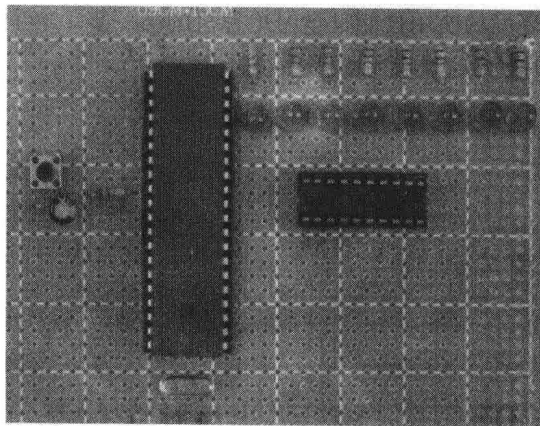


小知识

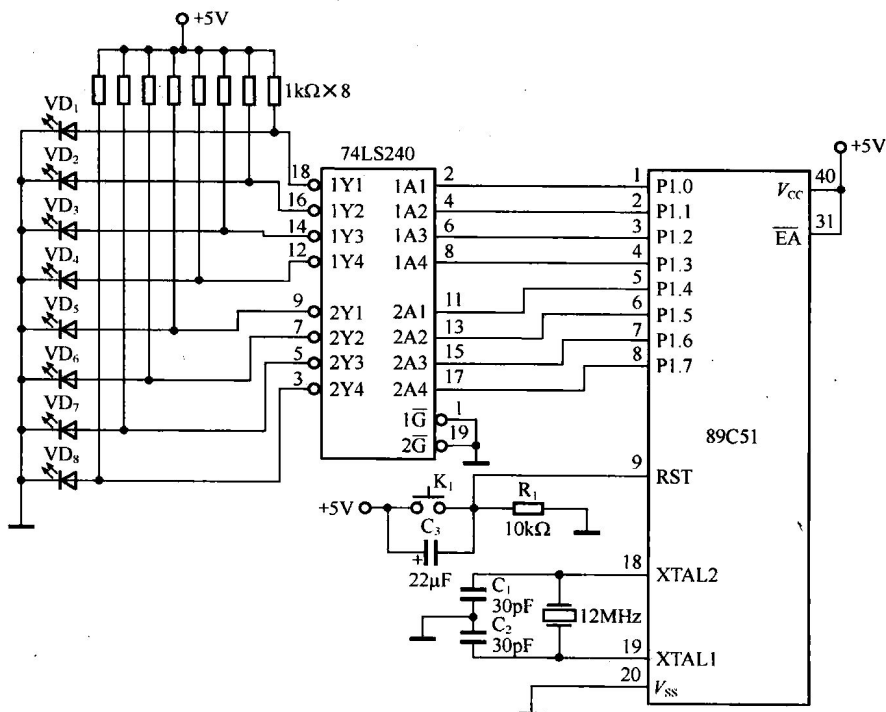
单片微型计算机（简称单片机）把组成微型计算机的各种功能部件，包括 CPU、RAM 和 ROM、基本 I/O 接口电路、定时器/计数器等，制作在一块集成芯片上，用以实现微型计算机的基本功能。因此，单片机实质上就是一块芯片。

知识 2 什么是单片机应用系统

在各类电子产品中，利用单片机实施控制的系统被称作单片机应用系统。图 1.2 所示为以单片机为核心构成的流水灯控制硬件系统，图 1.2 (a) 所示为完成后的实物，图 1.2 (b) 所示为系统电路图，该系统可以对流水灯实施多种方式的亮灭控制。



(a) 硬件实物



(b) 硬件电路图

图 1.2 流水灯控制硬件系统

图 1.2 (a) 中看到的只是单片机应用系统的硬件电路部分, 实际上在单片机芯片的内部存储器中已经烧录了预先编写好的信号灯控制程序。因此, 一个单片机应用系统由硬件系统和软件系统两部分组成, 如图 1.3 所示, 二者缺一不可。

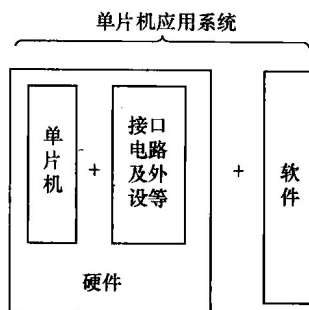


图 1.3 单片机应用系统的组成



实际的单片机应用系统, 通常要比上述电路复杂很多, 一般还包括显示、键盘、前向通道的模拟输入、后向通道的输出电路等, 这些知识将随着今后的学习逐渐介绍。

知识 3 什么是单片机开发环境

单片机应用系统的开发过程一般分为原理图设计、电路制作、编写应用程序、联机运行调试



和脱机运行调试 5 个步骤。其中，联机运行调试是指在单片机开发环境中运行程序，利用仿真器在模拟仿真状态下对硬件和软件进行综合调试，对存在的问题逐一排查和改进。

图 1.4 所示为单片机开发环境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、开发工具软件等。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程序的功能，还需配备专用的编程器。

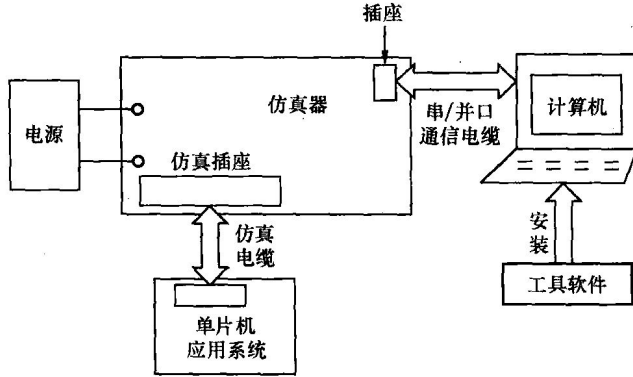


图 1.4 单片机开发环境装备及连接示意图

单片机在线仿真器有很多种，这里选用 Insight 系列 ME-52 型仿真器和中文版的 MedWin 工具软件，Insight 系列 ME-52 型仿真器外观如图 1.5 所示，下面将以该仿真器和 MedWin 工具软件为例进行介绍。

图 1.6 所示为单片机应用系统与仿真器联机运行状态，在这种状态中，系统尚未插入单片机芯片，其运行程序与硬件资源来源于仿真器。

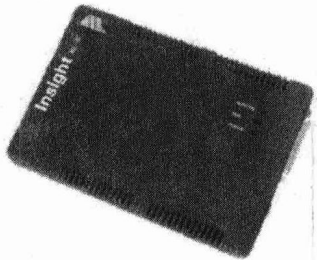


图 1.5 Insight 系列 ME-52 型仿真器

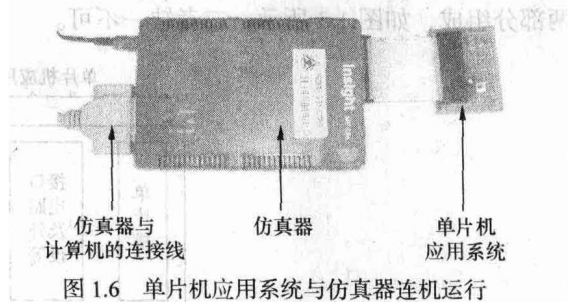


图 1.6 单片机应用系统与仿真器联机运行

脱机运行调试是指联机运行调试完成后，将源程序代码固化到单片机芯片内部的程序存储器中，并将带有程序的单片机芯片插入到单片机应用系统硬件电路的插座中，上电后系统就能直接运行单片机中的程序，脱机运行调试的单片机应用系统的硬件实物如图 1.2 (a) 所示。



单片机应用系统的调试过程，一般都是先通过联机调试的方式查找硬件故障和软件执行方面的不足，在联机调试通过后，再将程序下载到单片机芯片内部的程序存储器中，完成脱机运行。

如果单片机开发系统不具备固化程序的功能，还需配备专用的编程器，图 1.7 所示为 LabTool-48XP 型编程器。

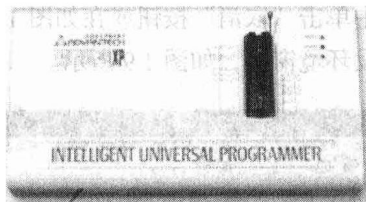


图 1.7 LabTool-48XP 型编程器

项目实施

步骤 1——建立单片机开发环境

现在以 ME-52 型仿真器和 MedWin 工具软件为例,建立单片机开发环境。使用其他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。

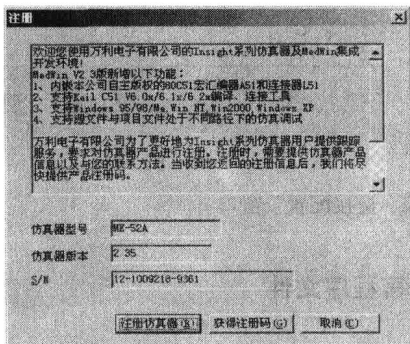
- ① 利用并口线将计算机和单片机在线仿真器连接起来。
- ② 用带插头的仿真扁平线将单片机在线仿真器与实验室提供的实验板(箱)或自己设计制作的单片机应用系统连接在一起,管脚的对应关系一定不能搞错。
- ③ 打开在线仿真器电源。
- ④ 在计算机上启动 MedWin 工具软件,此时单片机开发环境就建立起来了。



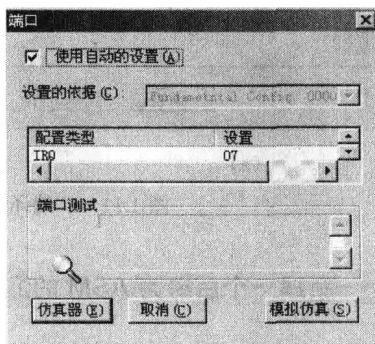
- 连接单片机开发系统各个部分时,须断电连接;连接好之后,再接通电源。
- 用带插头的仿真扁平线连接单片机在线仿真器和应用系统时,务必注意管脚的对应,通常扁平线的引脚 1 会用鲜明的颜色(例如红色)表示出来,以确保该引线连接到芯片的引脚 1 上。

步骤 2——MedWin 工具软件的启动和有关设置

① 启动中文版 MedWin 软件后,计算机屏幕上会出现如图 1.8(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 1.8(b)所示的窗口,此时只能进入“模拟仿真”状态。



(a)



(b)

图 1.8 MedWin 工具软件窗口



在如图 1.8 (a) 所示的窗口中单击“取消”按钮或在如图 1.8 (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 1.9 所示。

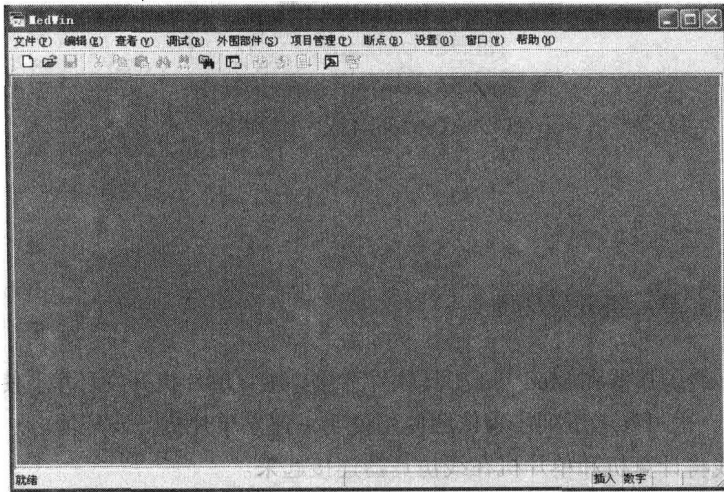


图 1.9 集成开发环境窗口

② 如果 MedWin 软件是第 1 次在计算机中安装运行，则需先进行“编译、汇编、连接配置”，以后使用时就不需再配置了。在如图 1.9 所示的界面中单击“设置”菜单，出现如图 1.10 所示的命令菜单，选择“设置向导”，出现如图 1.11 (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 1.11 (b) 所示窗口，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，则应选择 ASM。然后单击“完成”按钮返回如图 1.9 所示的窗口。

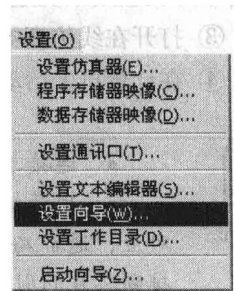
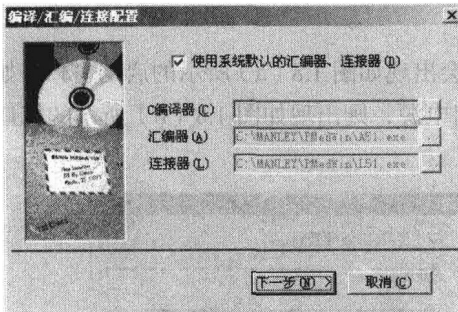
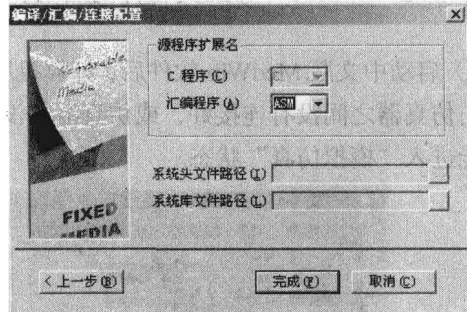


图 1.10 “设置”命令菜单



(a)



(b)

图 1.11 “编译、汇编、连接配置”窗口

步骤 3——新建一个后缀为.ASM 的汇编语言程序文件

① 在如图 1.9 所示的窗口中单击“文件”菜单，出现如图 1.12 所示的命令菜单，选择“新建”，出现如图 1.13 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。这里新建一个文



件名为 test1.asm 的新文件。

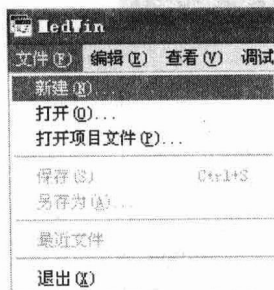


图 1.12 “文件” 命令菜单

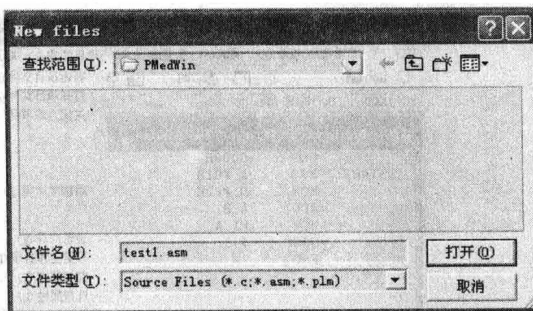


图 1.13 新建文件窗口



文件名不能用中文字符，也不能出现*/-等字符，否则在汇编、调试程序的时候会出现错误提示。

② 单击“打开”按钮，出现如图 1.14 所示的窗口，现在可以在这个窗口中输入图 1.14 中的程序 test1.asm。

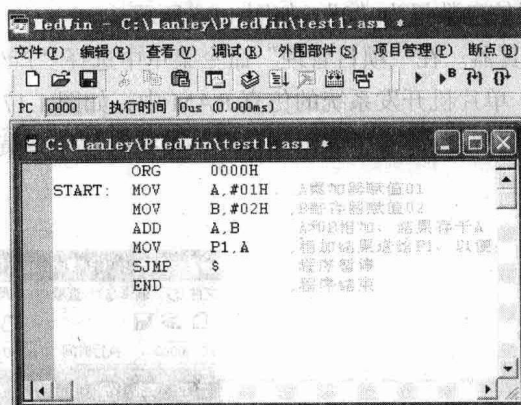


图 1.14 程序输入窗口



● 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。在编辑源程序时，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。

● 源程序一定要在西文状态下编辑。如果在中文状态下编辑源程序，则在进行汇编时会带来不必要的麻烦。特别要注意的是，其中的标点“，”“；”“：”必须是西文字符，否则汇编将产生出错信息。

步骤 4——编译/汇编程序

对源程序进行编译/汇编，就是将编辑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，为程序运行做准备。

① 在如图 1.14 所示的窗口中单击“项目管理”菜单，出现如图 1.15 所示的命令菜单。单击



“编译/汇编”（或按 Ctrl+F7 组合键）既可完成对当前源程序的“编译/汇编”。

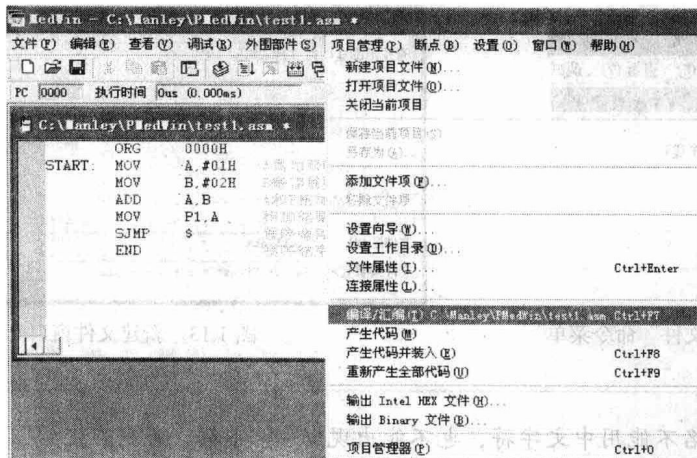


图 1.15 “项目管理”命令菜单

② 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”，如图 1.16 所示。

③ 产生代码并装入仿真器。在“项目管理”命令菜单中选择“产生代码并装入”，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 1.17 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，黄色箭头指示的是下一条将要执行的指令。

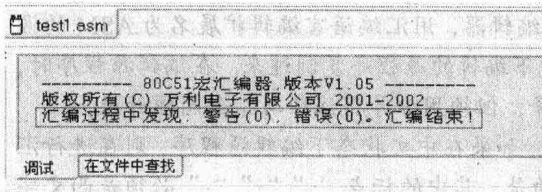


图 1.16 纠错消息窗口

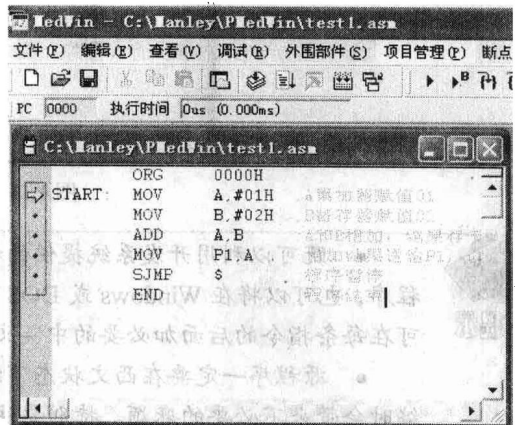


图 1.17 代码装入后的程序窗口

步骤 5——调试运行程序

运行程序有多种操作方法，包括全速连续运行（F9）、跟踪运行（F7）、单步运行（F8）、运行至光标处（F4）、断点运行等，可以使用快捷键，也可以在如图 1.18 所示的“调试”命令菜单中选择，还可以使用工具栏中的调试按钮，如图 1.19 所示。

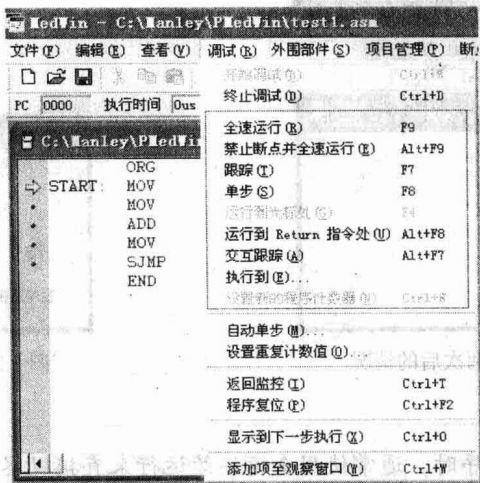


图 1.18 “调试” 命令菜单

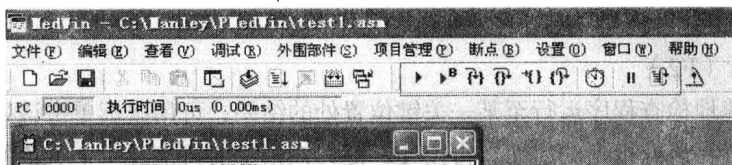


图 1.19 工具栏中的调试按钮

下面介绍上述快捷键对应的运行方式。

1. 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。黄色箭头始终指向下一条要执行的指令。如图 1.20 所示, 表明按了两次 F8 键, 黄色箭头指向第 3 条指令 ADD A,B, 表示这条指令将要被执行。

2. 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。但是, 当程序中有子程序时, 二者的运行就有所不同了。

3. 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, 程序将从当前位置全速运行到光标处。此方法可加快调试程序的速度。

4. 全速连续运行 (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, 程序将从当前位置处开始全速连续运行程序, 并出现如图 1.21 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。

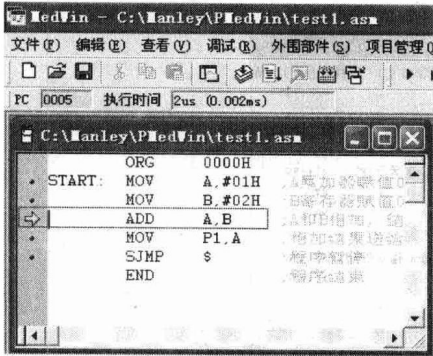


图 1.20 单步运行两次后的情况

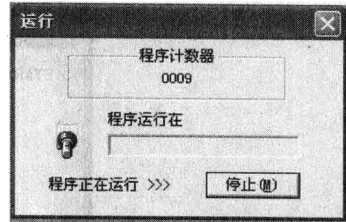


图 1.21 程序全速运行窗口



小提示

在调试应用程序时，通常使用全速连续运行来直接观察运行结果，如果一次就调试成功，当然是最理想的结果，但是实际情况往往不是这样，必须借助其他手段来逐步调试。

5. 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“!”，如图 1.22 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 1.23 所示。

与运行至光标处 (F4) 的方法相比，断点运行的方法对断点有记忆功能，当重复调试程序时，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。

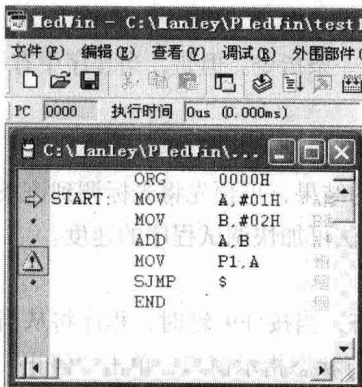


图 1.22 设置断点调试

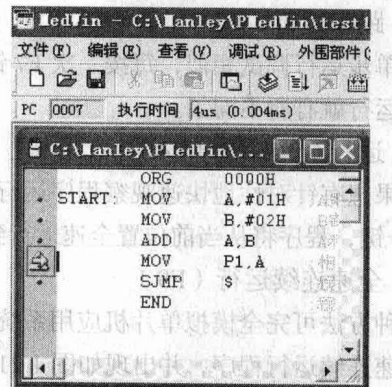


图 1.23 程序运行到断点处停止



小知识

在一个应用程序的调试过程中，可以综合应用各种调试手段，但并不是必须使用所有的调试手段。一般当调试子程序时，采用跟踪运行比较合适；当有中断程序时，更多使用断点运行的方法。