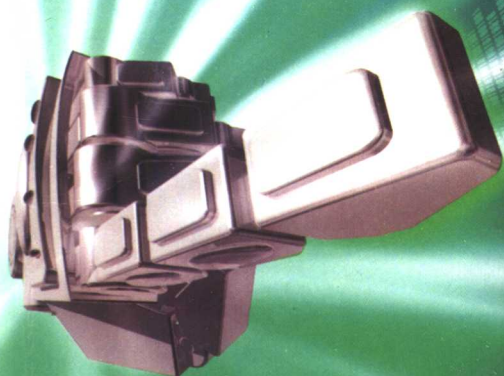




面向

21世纪

高级应用型人才

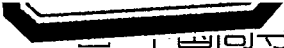
中国高等职业技术教育研究会推荐
高职高专系列规划教材

单片机应用实训教程

主 编 张永枫

副主编 王静霞 杨宏利

西安电子科技大学出版社
<http://www.xduph.com>

 中国职业技术教育研究会推荐

高职高专系列规划教材

单片机应用实训教程

主 编 张永枫

副主编 王静霞 杨宏丽

西安电子科技大学出版社

内 容 简 介

本书是中国高等职业技术教育研究会与西安电子科技大学出版社联合策划、组织编写的高职高专系列规划教材之一。

本书以实训项目为主线,突出应用和实践,将理论知识作为能力训练的支撑点,较好地把握了理论够用的“度”,适用于现代职业技能训练的教学过程。实训内容包括 MCS-51 单片机基础应用能力训练、综合应用能力训练、高级语言 C51 编程训练和应用较为广泛的 PIC 系列单片机应用训练。

本书可作为高职及应用型本科院校的电子、通信及相关专业的单片机应用实训教材,也可作为职业技能培训教材。

图书在版编目(CIP)数据

单片机应用实训教程 / 张永枫主编.

—西安:西安电子科技大学出版社,2005.2

(高职高专系列规划教材)

ISBN 7-5606-1485-X

I. 单… II. 张… III. 单片微型计算机, MCS-51—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(2004)第 142833 号

策 划 马乐惠

责任编辑 潘恩祥 马乐惠

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

http://www.xduph.com

E-mail:xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安交通大学印刷厂

版 次 2005年2月第1版 2005年2月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 18.875 插图 1

字 数 447千字

印 数 1~4000册

定 价 22.00元

ISBN 7-5606-1485-X/TP·0789(课)

XDUP 1756001-1

*** 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。

序

1999 年以来,随着高等教育大众化步伐的加快,高等职业教育呈现出快速发展的形势。党和国家高度重视高等职业教育的改革和发展,出台了一系列相关的法律、法规、文件等,规范、推动了高等职业教育健康有序的发展。同时,社会对高等职业技术教育的认识在不断加强,高等技术应用型人才及其培养的重要性也正在被越来越多的人所认同。目前,高等职业技术教育在学校数、招生数和毕业生数等方面均占据了高等教育的半壁江山,成为高等教育的重要组成部分,在我国社会主义现代化建设事业中发挥着极其重要的作用。

在高等职业教育大发展的同时,也有着许多亟待解决的问题。其中最主要的是按照高等职业教育培养目标的要求,培养一批具有“双师素质”的中青年骨干教师;编写出一批有特色的基础课和专业主干课教材;创建一批教学工作优秀学校、特色专业和实训基地。

为解决当前信息及机电类精品高职教材不足的问题,西安电子科技大学出版社与中国高等职业技术教育研究会分两轮联合策划、组织编写了“计算机、通信电子及机电类专业”系列高职高专教材共 100 余种。这些教材的选题是在全国范围内近 30 所高职高专院校中,对教学计划和课程设置进行充分调研的基础上策划产生的。教材的编写采取公开招标的形式,以吸收尽可能多的优秀作者参与投标和编写。在此基础上,召开系列教材专家编委会,评审教材编写大纲,并对中标大纲提出修改、完善意见,确定主编、主审人选。该系列教材着力把握高职高专“重在技术能力培养”的原则,结合目标定位,注重在新颖性、实用性、可读性三个方面能有所突破,体现高职教材的特点。第一轮教材共 36 种,已于 2001 年全部出齐,从使用情况看,比较适合高等职业院校的需要,普遍受到各学校的欢迎,一再重印,其中《互联网实用技术与网页制作》在短短两年多的时间里先后重印 6 次,并获教育部 2002 年普通高校优秀教材二等奖。第二轮教材预计在 2004 年全部出齐。

教材建设是高等职业院校基本建设的主要工作之一,是教学内容改革的重要基础。为此,有关高职院校都十分重视教材建设,组织教师积极参加教材编写,为高职教材从无到有,从有到优、到特而辛勤工作。但高职教材的建设起步时间不长,还需要做艰苦的工作,我们殷切地希望广大从事高等职业教育的教师,在教书育人的同时,组织起来,共同努力,编写出一批高职教材的精品,为推出一批有特色的、高质量的高职教材作出积极的贡献。

中国高等职业技术教育研究会会长

李家亮

IT 类专业系列高职高专规划教材编审专家委员会名单

主 任：高 林 （北京联合大学副校长，教授）

副主任：温希东 （深圳职业技术学院电子通信工程系主任，教授）

李卓玲 （沈阳电力高等专科学校信息工程系主任，教授）

李荣才 （西安电子科技大学出版社总编辑，教授）

计 算 机 组：组长：李卓玲(兼) （成员按姓氏笔画排列）

丁桂芝 （天津职业大学计算机工程系主任，教授）

王海春 （成都航空职业技术学院电子工程系副教授）

文益民 （湖南工业职业技术学院信息工程系主任，副教授）

朱乃立 （洛阳大学电子工程系主任，教授）

李 虹 （南京工业职业技术学院电气工程系副教授）

陈 晴 （武汉职业技术学院计算机科学系主任，副教授）

范剑波 （宁波高等专科学校电子技术工程系副主任，副教授）

陶 霖 （上海第二工业大学计算机学院教授）

徐人凤 （深圳职业技术学院计算机应用工程系副主任，高工）

章海鸥 （金陵科技学院计算机系副教授）

鲍有文 （北京联合大学信息学院副院长，副教授）

电子通信组：组长：温希东(兼) （成员按姓氏笔画排列）

马晓明 （深圳职业技术学院电子通信工程系副主任，副教授）

于 冰 （宁波高等专科学校电子技术工程系副教授）

孙建京 （北京联合大学教务长，教授）

苏家健 （上海第二工业大学电子电气工程学院副院长，高工）

狄建雄 （南京工业职业技术学院电气工程系主任，副教授）

陈 方 （湖南工业职业技术学院电气工程系主任，副教授）

李建月 （洛阳大学电子工程系副主任，副教授）

李 川 （沈阳电力高等专科学校自动控制系副教授）

林训超 （成都航空职业技术学院电子工程系主任，副教授）

姚建永 （武汉职业技术学院电子信息系主任，副教授）

韩伟忠 （金陵科技学院龙蟠学院院长，高工）

项目总策划：梁家新

项目策划：马乐惠 云立实 马武装 马晓娟

电子教案：马武装

前 言

本书是中国高等职业技术教育研究会与西安电子科技大学出版社联合策划、组织编写的高职高专系列规划教材之一。深圳职业技术学院在组织编写了《数字电子技术》、《单片机应用技术》等教材的基础上,又承担了本书的编写工作。在本书的编写过程中汲取了多年来不同院校、专业的教学经验,力求在项目的选材与构建上从应用的角度出发,体现理论知识和实践能力并重,突出高职教育的特点。

单片机应用实训是一门实践性、综合性和应用性很强的专业基础训练课。作为单片机应用系统的开发应用人员,不仅要掌握单片机应用的基本原理和基本指令,还应具备较强的分析程序和编制程序的能力以及单片机应用系统的硬件设计、综合分析与调试能力。只有在应用项目的设计、分析与调试过程中不断地演练,才能在实践中逐步提高应用单片机的各种能力。本教材以提高使用者的分析、设计与调试能力为目的,通过各种典型实训项目进行强化训练,使读者从基本了解单片机的初学者逐步成为能熟练运用相关知识和技能,完成各种单片机应用系统的设计、调试与制作的高级专业人才。

本教材改变了以往以理论教学为核心,以实践教学支撑理论教学的理念。在符合教育教学认识规律的基础上,采用更适合高职教育特点的“以专业技术应用能力培养为核心,以实践教学为主线,以训练项目为先导,以理论教学为支撑”的教学新思路。在教材编写过程中避免只重学科体系,忽视能力培养,或只重操作实践,忽视理论知识的两种极端倾向,尤其是合理地把握了理论知识“必需、够用”的度。在整个教材体系中,突出了“以实践教学模块体系为主线,以应用性和综合性项目为先导,以理论知识模块为支撑”的编写思想。

全书分 MCS-51 单片机基础应用、综合应用、C51 应用和 PIC 单片机应用 4 个部分。每个部分都由相关的典型实训项目组成,从基础实训项目逐步过渡到综合实训项目,再扩展到用高级语言 C51 进行编程及 PIC 系列单片机的应用。各实训项目既有一定的独立性,又能前后衔接,利于使用者进一步扩展和发挥。

各实训项目是从应用的角度提出设计要求，再由此引入单片机应用系统的软硬件分析、设计与调试方法等内容。边分析、边设计、边调试，将辅助的理论知识 and 设备的操作、器件的使用等内容有机地融入到项目训练的过程之中。

本书的教学参考学时数在 54~80 学时之间，实际使用时可根据不同专业需要对实训项目进行取舍。本书内容取材合理，文字叙述清楚，可供三年及四年制高职、高专院校或应用型本科院校作为单片机应用实训或工种考核培训教材，也可作为选修课教材。

张永枫负责本书的总体策划，并编写了实训 1.1、1.2。杨宏丽、王静霞协助完成上述工作。杨宏丽编写实训 1.4~1.6，2.1，王静霞编写实训 3.1~3.6，宋荣编写实训 4.1~4.4，苏全编写实训 2.2、2.3，唐建东编写实训 1.3、1.7，毛丰江编写实训 1.8、2.4。

深圳职业技术学院韩秀清副教授、郑州经济管理学院王修武副教授和深圳职业技术学院工业中心电子技术实验室的老师们为本书的编写提供了很大帮助，在此表示衷心感谢。

由于编者水平有限，书中错误在所难免，热忱欢迎使用者提出批评和建议。

编 者

2004 年 12 月

于深圳职业技术学院

目 录

第 1 章 MCS-51 单片机基础应用实训	1
实训 1.1 单片机开发系统及使用	1
实训 1.2 数字钟应用电路	16
实训 1.3 LCD 显示器电路	31
实训 1.4 中断与定时/计数器的应用	44
实训 1.5 A/D 转换接口	53
实训 1.6 串行通信接口	60
实训 1.7 语音电路的应用	71
实训 1.8 实时时钟/日历芯片的应用	85
第 2 章 MCS-51 单片机综合应用实训	97
实训 2.1 单片机多功能实训电路的设计与测试	97
实训 2.2 单片机温度控制器的设计与调试	107
实训 2.3 单片机步进电机控制器的设计与调试	125
实训 2.4 接触式 IC 卡读写器的设计与调试	135
第 3 章 C51 应用实训	159
实训 3.1 C51 程序开发环境	159
实训 3.2 定时器/计数器 C51 程序设计	165
实训 3.3 数码显示和矩阵式键盘 C51 程序设计	174
实训 3.4 字符型 LCD 显示模块 C51 程序设计	187
实训 3.5 单片机串行通信 C51 程序设计	198
实训 3.6 数字钟 C51 程序的实现	208
第 4 章 PIC 单片机应用实训	217
实训 4.1 简易 LED 闪烁彩灯	218
实训 4.2 162 点阵字符液晶显示器的应用	234
实训 4.3 PIC 单片机 A/D 转换器的应用	258
实训 4.4 单线数字温度传感器的应用	267
附录	278
附录 1 PIC16C7X 系列单片机指令详解	278
附录 2 PIC16C7X 系列单片机程序设计基础	287
参考文献	294

第1章 MCS-51 单片机基础应用实训

实训 1.1 单片机开发系统及使用

1. 实训目的

- (1) 了解单片机开发系统的基本功能，掌握单片机开发系统的使用方法。
- (2) 通过对典型程序的调试操作训练，掌握运用开发系统快速有效地进行调试的基本方法。

2. 实训设备与器件

实训设备：单片机集成开发环境、综合实训板(综合实训板电路图参见书末附图)。

3. 实训步骤及要求

- (1) 熟悉单片机仿真开发系统的使用方法，掌握其基本功能与操作过程。
- (2) 用汇编语言编辑和输入源程序。
- (3) 对源程序进行汇编和纠错。
- (4) 调试程序。

4. 实训内容

对于不同的单片机开发系统，调试软件和调试环境也有所不同，例如，MICE-51 型单片机开发系统是在 DOS 环境下通过 MBUG 调试软件进行录入、编辑、汇编及调试，Insight 系列的 Me-52A 型是在 Windows 环境下通过 MedWin 集成开发软件完成各项编程与调试任务的，它们的基本功能大致相同。以 MedWin 集成开发软件的使用为例，介绍开发系统的使用步骤和调试方法。

1) 开发环境使用步骤

- (1) 将开发系统和目标板连接好，并接上电源。
- (2) 启动 MedWin 中文版，初次启动出现图 1.1.1(a)所示窗口，再次启动出现图 1.1.1(b)所示窗口。单击图 1.1.1(b)中的“取消”或“模拟仿真”进入 MedWin 集成开发环境，出现图 1.1.2 所示界面。

(3) 设置汇编(或编译)环境。第一次在 MedWin 中使用汇编语言汇编(C51 编译)环境需进行“编译/汇编/连接配置”(以后使用不需再配置)。单击“设置”菜单项，如图 1.1.3 所示，选择“设置向导”，弹出如图 1.1.4(a)所示的“编译/汇编/连接配置”窗口。

单击“下一步”按钮，弹出如图 1.1.4(b)所示的窗口，在该窗口中设置系统头文件路径和系统库文件路径。选择源程序扩展名为 ASM(或 C)，若采用汇编语言编制源程序，应选择 ASM，然后单击“完成”按钮即可。

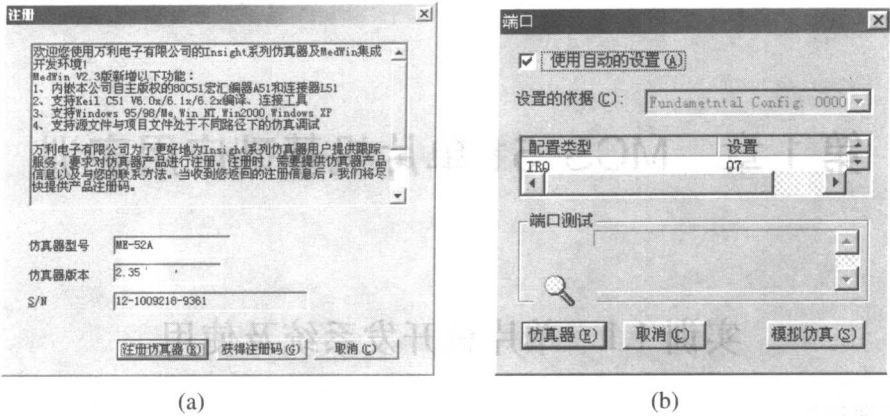


图 1.1.1 进入窗口

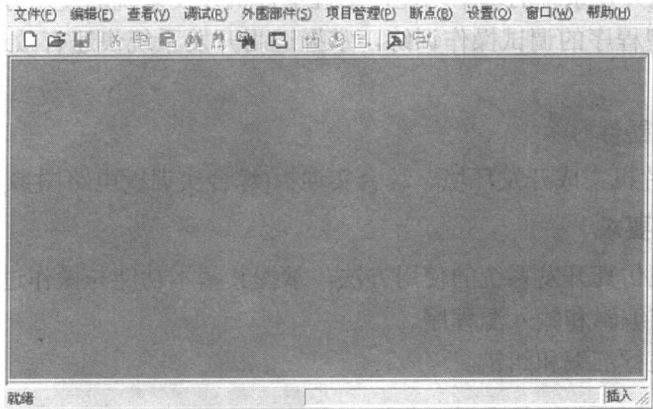


图 1.1.2 集成开发环境界面

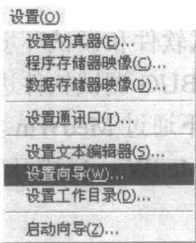


图 1.1.3 设置菜单项

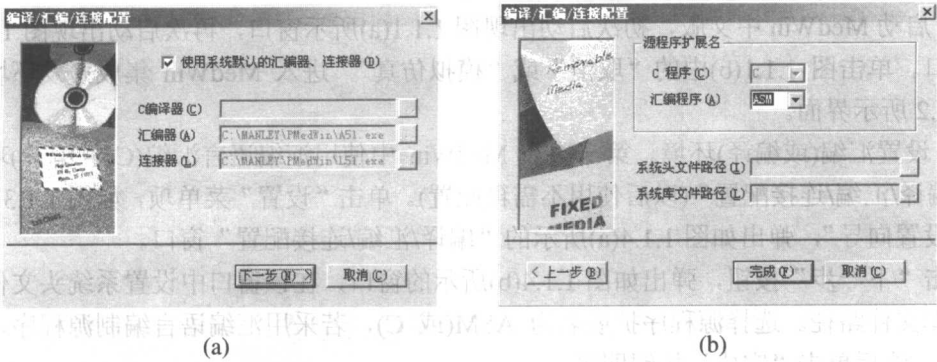


图 1.1.4 编译、汇编、连接配置窗口

(4) 新建 NEW(或打开 Open)文件。在图 1.1.2 中单击“文件”选项，出现图 1.1.5 所示菜单，选择“新建”(或“打开”)文件，出现图 1.1.6 新建文件界面，选择文件存放路径，输入文件名，单击打开按钮即可。

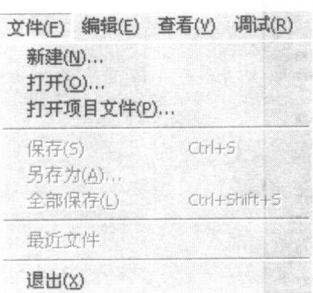


图 1.1.5 文件处理菜单项

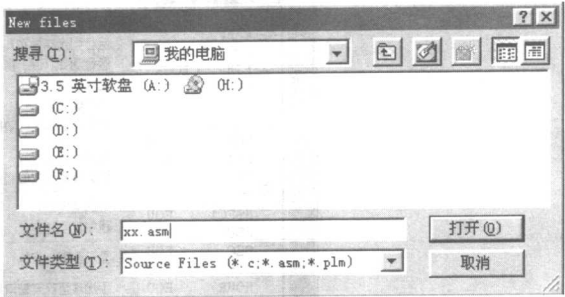


图 1.1.6 新建文件界面

可使用开发系统提供的编辑器编辑扩展名为.ASM的源程序(或在 Windows 和 DOS 环境下编辑的源程序)，如*.ASM。编制源程序时，可在每条指令的后面加必要的文字注释，但注释前须用分号间隔。若用 C 语言编制源程序时，则文件名应为*.C。

(5) 对源程序进行汇编(或编译)。源程序编好后，在图 1.1.2 中单击“项目管理”，如图 1.1.7 所示。选择“编译/汇编”菜单项(或 Ctrl+F7)对当前的源程序进行“编译/汇编”。若采用汇编语言编制源文件，则将对当前文件进行汇编。若采用 C 语言编制源文件，将对当前文件进行编译。

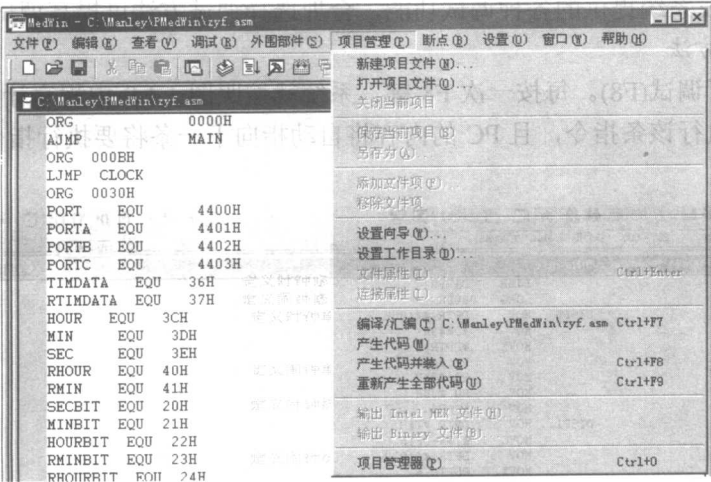


图 1.1.7 项目管理菜单

(6) 排除错误。文件经过“编译/汇编”后，在消息窗口将会出现纠错信息，该信息将提示错误出现的位置及错误的类型和数量等，使用者可根据该信息对源程序的错误进行纠正，纠正后再重新进行“编译/汇编”直至错误信息数量为“0”。

(7) 产生代码并装入仿真器。在图 1.1.7 所示的“项目管理”菜单栏中，选择“产生代码并装入”菜单项(Ctrl+F8)，将生成的文件代码装入(Load)单片机开发系统的仿真 RAM 中。

(8) 调试程序。产生代码并装入仿真器完成后，在图 1.1.2 中单击“调试”菜单，如图 1.1.8 所示。

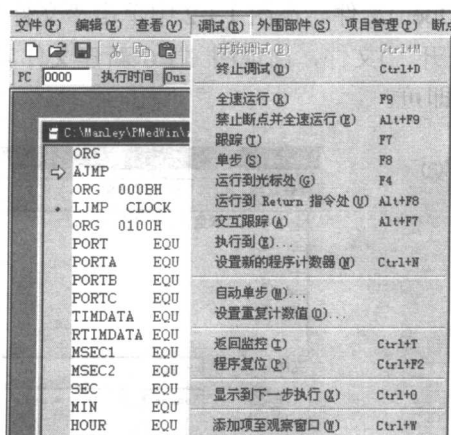


图 1.1.8 程序调试菜单

再根据调试的需要选择不同的调试方法对程序进行调试。在编译/汇编源程序时, 汇编(或编译)系统只能提示源程序的逻辑、符号等方面的错误信息, 而对程序运行的结果是否正确、运行的过程是否符合编程者的设计要求等将无法作出正确判别。因此, 设计者必须运用开发系统所提供的各种调试功能, 快速有效地排查程序存在的各种问题, 直至程序完全符合设计要求为止。

(9) 输出代码文件。程序调试完毕, 可在图 1.1.7 中选择“产生代码”, 生成相应的目标文件, 以便将目标程序写入芯片。

熟练掌握开发系统提供的各种调试功能, 合理选择调试方法可提高调试程序的效率。

2) 常用调试方法

(1) 单步运行调试(F8)。每按一次 F8 键, 系统就按照图 1.1.9 中程序计数器 PC 所指示的地址(箭头处)执行该条指令, 且 PC 的内容将自动指向下一条将要执行指令的地址, 箭头也向下移动一次。

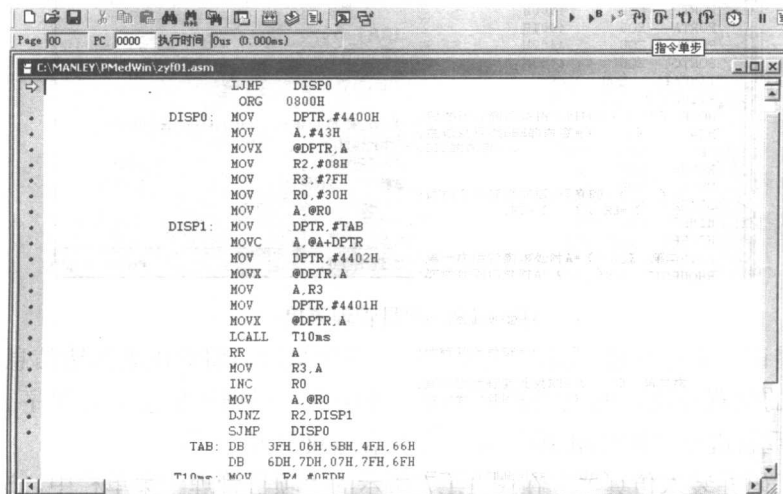


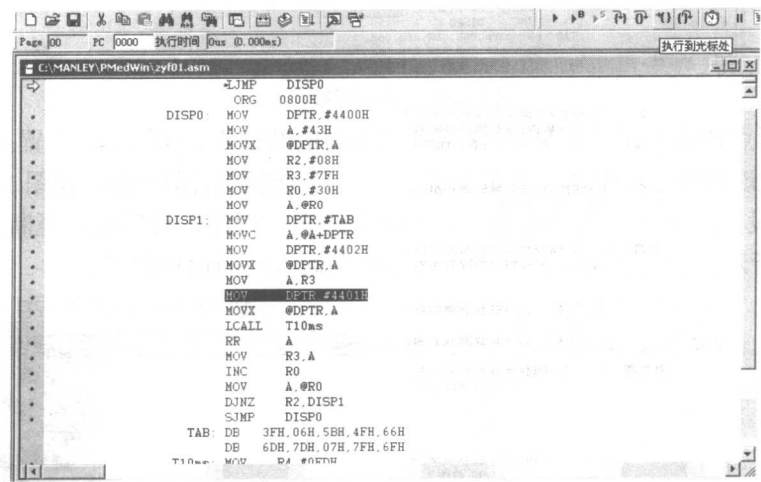
图 1.1.9 单步运行

若单步运行的是调用子程序指令(LCALL XX、ACALL XX), 它将把被调用子程序内部的所有指令全部执行完毕, PC 的内容将自动指向该调用指令的下一条指令处。所以采用单

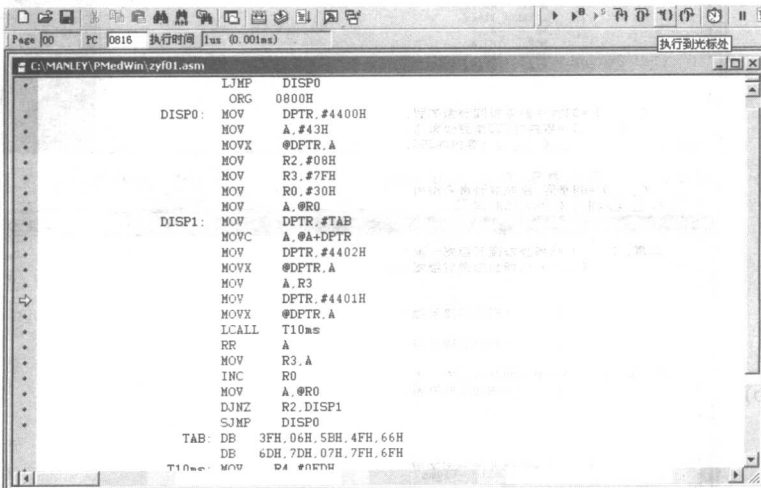
步运行能快速观察被调用子程序执行后的最终结果，但无法观察子程序内部各条指令的执行状况。

(2) 跟踪运行调试(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。但当执行调用指令(LCALL XX、ACALL XX)时，跟踪运行可以跟踪到子程序内部。所以跟踪运行调试可观察程序从主程序转入子程序、子程序内部各条指令的运行及子程序返回的运行过程。

(3) 全速运行至光标处调试(F4)。先将光标调到某条需要观察执行结果的指令处，如图 1.1.10(a)所示。再按 F4 键，程序将从当前 PC 所指示的位置全速运行到光标处，如图 1.1.10(b)所示。此方法可根据操作者的实际需要，快速观察程序运行至某处的执行结果，加快调试程序的速度。



(a)



(b)

图 1.1.10 执行至光标处

(4) 全速连续运行调试(F9)。当按 F9 键时，程序将从当前的 PC 处开始全速连续运行程序，如图 1.1.11 所示。可通过停止按钮终止程序的运行，全速连续运行调试便于观察程序

连续运行状态下相关显示及控制的动态过程。

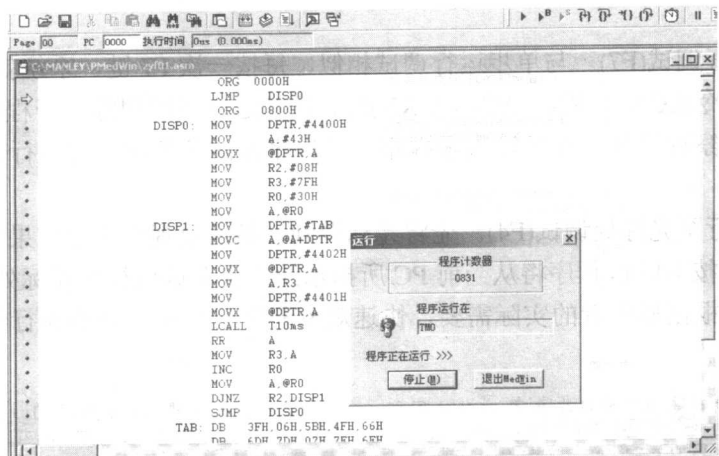


图 1.1.11 全速连续运行

(5) 设置断点调试(F2)。用鼠标单击某条指令前的圆标点, 或将光标设置在某条指令处, 再按 F2 键, 在该指令前将出现一个黄色标记符“!”(或红色标记线), 如图 1.1.12(a)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停止, 如图 1.1.12(b)所示。此方法可快速观察程序运行到断点处的运行结果。

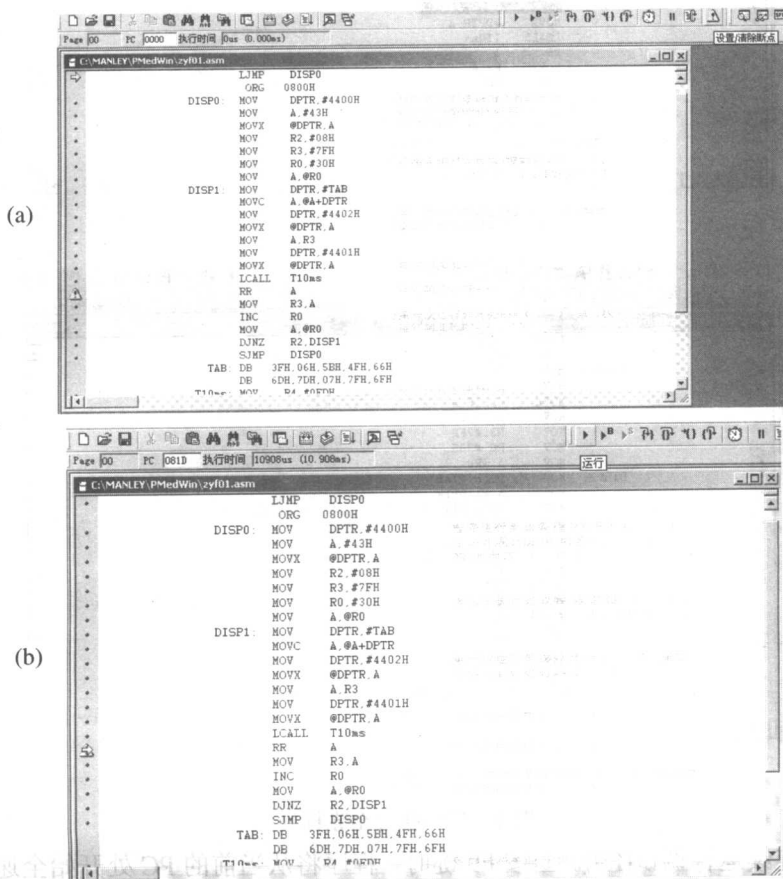


图 1.1.12 断点运行

与全速运行至光标处(F4)调试方法相比,设置断点后,当重复调试程序时,程序只要运行到此处都会停在该断点处,该方法特别适合于调试循环程序。可根据需要在程序的不同位置设置多个断点。取消断点可用鼠标单击断点标记或在断点处再按 F2 键。

(6) 自动单步运行调试。该方法可自动地单步运行逐条程序,且两条指令间的间隔时间可调,如图 1.1.13 所示。

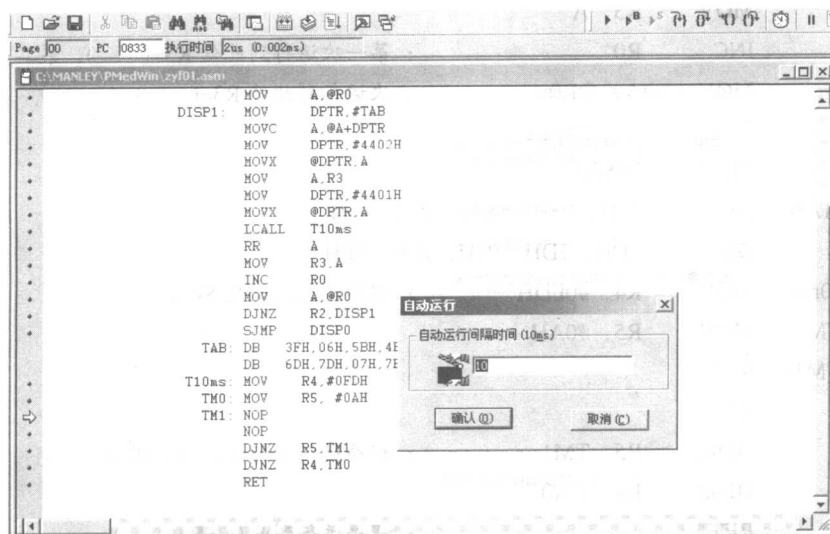


图 1.1.13 自动单步运行

(7) 设置程序计数器 PC 的内容(Ctrl+N)。单片机在复位时自动将 PC 的内容设定为 0000H,若要修改当前 PC 的内容,可将光标移到指定位置,再按 Ctrl+N 键,当前程序计数器 PC 的内容便被设置在此处。在调试程序时,有时需从某一地址处开始执行程序,可运用此方法修改程序起始地址 PC 的内容。

3) 程序调试实例 1: 查表显示程序的分析与调试

(1) 源程序。

```

ORG 0000H                ; 汇编完成后, PC 的初始值=(    )
LJMP    DISP0
ORG 0800H
DISP0:  MOV    DPTR, #4400H    ; 程序运行到该条指令时 PC=(    )
        MOV    A, #43H        ; 在该处观察 DPH 的内容=(    )
        MOVX   @DPTR, A       ; DPL 的内容=(    )
        MOV    R2, #08H
        MOV    R3, #7FH
        MOV    R0, #30H        ; 该指令执行完成后, 观察 R0=(    )、
        MOV    A, @R0          ; R2=(    )、R3=(    )
DISP1:  MOV    DPTR, #TAB
        MOVC   A, @A+DPTR
        MOV    DPTR, #4402H    ; 第一次运行到该处时 A=(    ), 第二
        MOVX   @DPTR, A       ; 次运行到该处时 A=(    )

```

```

MOV    A, R3
MOV    DPTR, #4401H
MOVX   @DPTR, A           ; 运行到此处 SP=(    )
LCALL  T10ms
RR     A                   ; 运行到此处 SP=(    )
MOV    R3, A
INC    R0                  ; 第一次运行到此处 R3=(    ), 第二
MOV    A, @R0              ; 次运行到此处 R3=(    )
DJNZ   R2, DISP1
SJMP   DISP0
TAB:   DB    3FH, 06H, 5BH, 4FH, 66H
       DB    6DH, 7DH, 07H, 7FH, 6FH
T10ms: MOV    R4, #0FDH     ; 程序运行到此处 SP=(    )
TM0:   MOV    R5, #0AH
TM1:   NOP
       NOP
       DJNZ   R5, TM1       ; 程序循环运行 3 次后, R5=(    )
       DJNZ   R4, TM0
       RET                  ; 程序运行到此处 SP=(    )
                               ; 程序连续运行后, 可观察的实验现象是(    )

```

(2) 调试步骤。

① 为程序加注释。在调试程序前, 必须首先读懂源程序, 弄清每条指令执行的结果及每段程序执行的结果与功能, 分析并完成程序中的填空内容。

② 录入实训程序, 设文件名为 PROGRAM01.ASM。

③ 对源程序进行汇编。

④ 根据提示进行纠错。

⑤ 产生代码并装入仿真器。

⑥ 确定程序调试前的初始参数, 将显示单元 30H~37H 中的内容修改为 01H~08H。

⑦ 通过调试确定填空内容的正确性。

(3) 调试过程。调试程序时, 应先读懂源程序, 确定出能反映程序正确与否的观测点及用什么调试方法和步骤能快速地检验观测点的参数及路径的变化, 制定出具体的调试步骤; 通过操作来验证已完成填空内容的正确性。在调试上述程序时可从以下几方面入手:

首先修改 RAM 单元中的内容。根据题目要求, 先将 30H~37H 单元中的初始值设置为 01H~08H; 再运行程序检查结果。如先用连续运行方式(F9)运行程序, 观察程序运行的结果是否能显示设定单元的内容。在实际调试程序时, 由于各种原因, 程序运行的结果往往不一定能做到一次就达到设计者的要求, 可能存在各种未知的原因而产生错误。因此, 必须灵活运用各种调试手段, 快速有效地找出故障点, 分析出产生故障的原因并排除所有错误, 直到满足设计要求为止。调试手段一般包括:

① 检查程序运行的路径是否正确。可采用单步(F8)运行的调试方法, 边运行边观察程序运行的路径与预先设计的运行路径是否一致。若不一致, 可根据运行过程中地址的变化

找出故障点, 分析产生故障的原因。

② 检查程序运行到某处的执行结果。如检查程序执行到某处后, 相关单元(内部或外部)、工作寄存器、特殊功能寄存器中内容的变化。由此可判断程序运行到某处时是否正确。为提高调试速度, 可将光标停留在预观察点, 再用 F4 键快速将程序运行至光标处。通过对运行结果的观察与分析, 可判断结果是否正确。若出现错误, 应及时分析产生错误的原因并加以修正。

③ 检查子程序调用的运行过程。先用单步运行(F8)的方法运行调用子程序指令, 然后观察程序能否运行到该调用指令的下一条指令处, 若能, 则说明子程序调用的运行过程是正确的; 再检查子程序的出口内容是否正确, 若两者都正确, 则调试完毕。若执行了调用指令后, 程序不能返回到该调用指令的下一条指令处, 出现系统提示忙或子程序出口结果不正确时, 则应重新用跟踪运行(F7)的调试方法运行调用指令, 以便跟踪运行到子程序的内部, 再通过单步或跟踪运行等方法逐条运行, 直至找到产生错误的原因并加以修正为止。

④ 检查循环程序的运行过程。若程序中有循环结构, 可先将光标或断点预置在循环程序的最后一条指令处, 然后用全速运行到光标处(F4)或连续运行(F9)到断点处的方法运行程序。若提示系统忙, 可能出现了死循环等错误, 此时, 应考虑用单步运行的方法检查程序循环运行的路径变化是否正确。为缩短调试时间, 可在调试循环程序前, 将循环初始值中的循环次数改小些, 通过观察运行路径和指令运行的结果, 找出循环程序内部出现的故障并加以修正。

在上述调试过程中, 主要是针对顺序结构、分支结构、循环结构和子程序结构的调试, 究竟采用什么样的调试方法更适宜, 应根据被调试程序的结构特点和程序运行结果的观测点来合理选择。

4) 程序调试实例 2: 中断显示程序分析与调试

(1) 源程序。

```

ORG    0000H
START: LJMP    MAIN
        ORG    0003H
        LJMP    GINTO          ; 程序运行至此处(SP)=(    )
        ORG    000BH
        LJMP    GT0
        ORG    0030H
MAIN:   MOV     P1, #0FFH      ; 程序运行至此处, 指示灯的状态为(    )
        MOV     SP, #30H      ; 在执行该指令前, (SP)=(    )
        MOV     20H, #10H
        MOV     TMOD, #02H    ; 定时器 T0 设定为(    )方式
        MOV     TL0, #00H
        MOV     TH0, #00H
        MOV     IP, #01H      ; 中断优先级的排列顺序为(    )
        MOV     IE, #83H      ; (    )中断源被允许中断
        MOV     TCON, #101H   ; 边沿触发方式

```