

第 1 章 通用微机实验系统集成环境

1.1 实验系统

1.1.1 实验系统简介

在学习微机原理与接口技术的过程中，既要掌握正确的分析与设计方法，也要通过硬件实验环节来逐步掌握和提高。硬件实验和课程设计对硬件有很高的依赖性，必须有相应的实验平台。目前国内广泛应用的微机实验平台主要有西安唐都的微机教学实验系统、清华大学的 TPC 实验系统和复旦大学的启东实验系统。尽管这几种实验平台各具特点，但实验原理基本相同。

本书选用了由西安唐都科教仪器公司开发的 TD-PITE 32 位微机原理与接口技术教学实验系统，该系统为教学实验提供了完善的微机原理的软件实验调试平台和微机接口技术的硬件实验开发平台。主要特点如下：

(1) 采用开放式结构，模块化设计，支持开放实验。实验台上除了固定电路外还设有扩展实验区，可以自己设计实验电路，在扩展实验区上插上所选芯片并连线即可完成实验。

(2) 提供了单次脉冲、键盘扫描及数码管显示、开关输入及发光管显示、电子发声器、点阵 LED 显示、图形 LCD 显示、步进电机、直流电机及温度控制单元电路，全面支持“微机接口技术”及“微机控制及应用”的各项实验内容。

(3) 支持 Windows 2000、Windows XP 和 Windows 7 操作系统，支持 32 位和 64 位操作系统。采用 TD-PITE 集成开发实验软件，可以方便地对程序进行编辑、编译、链接和调试。

(4) 集成开发实验软件全面支持 80x86 汇编语言及 C 语言程序设计，可实现汇编语言与 C 语言混编。

1.1.2 实验系统构成

TD-PITE32 位微机接口实验系统是一套 80x86 微机原理与接口技术教学实验系统，其实验系统硬件结构如图 1-1 所示，实验台各个模块的布局如图 1-2 所示。

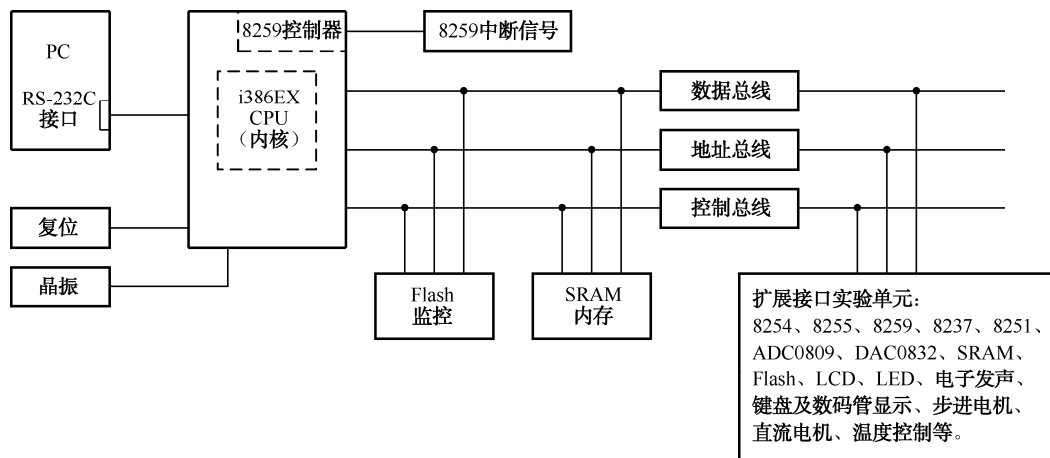


图 1-1 TD-PITE 32 位微机接口实验系统硬件结构图

电源	扩展实验区			i386EXCPU
时钟源				
温控单元	系统总线单元			
转换单元	8237DMA单元	A/D转换单元	点阵显示单元	
D/A转换单元	SRAM单元	8251单元	8254单元	单次脉冲单元
电子发声及步进电机单元	8255单元	开关及LED显示单元		键盘及数码管显示单元
驱动电路 直流电机单元				

图 1-2 TD-PITE 实验台布局图

1.2 实验系统硬件环境

1.2.1 80x86 微机系统单元

1. 系统总线

本书使用的 TD-PITE 微机实验系统采用组合式结构，即 i386EX 系统板加实验接口平台的形式。将 i386EX 系统板安装在实验接口平台上便构成 80x86 微机原理及接口技术教学实验系统，系统总线以排针和锥孔两种形式引出，做实验时，通过杜邦线或锥孔线与实验单元相连可完成相应的实验。80x86 微机系统引出信号线说明见表 1-1。

表 1-1 80x86 微机系统信号线说明

信号线	说明	信号线	说明
XD0~XD15	系统数据线(输入/输出)	MIR6、MIR7	主 8259 中断请求信号(输入)
XA1~XA20	系统地址线(输出)	SIR1	从 8259 中断请求信号(输入)
BHE#、BLE#	字节使能信号(输出)	MWR#、MRD#	存储器读、写信号(输出)
ADS#	地址状态信号(输出)	IOW#、IOR#	I/O 读、写信号(输出)
MY0、MY1	存储器待扩展信号(输出)	RST	复位信号(正输出)
IOY0~IOY3	I/O 接口待扩展信号(输出)	RST#	复位信号(负输出)
HOLD	总线保持请求(输入)	CLK	1MHz 时钟输出
HLDA	总线保持应答(输出)	CLK 12MHz	12MHz 时钟输出

注：#号表示该信号低电平有效。

2. 系统中的 8259 单元

由于 Intel 386EX 芯片内部集成有两片 8259A，且总线未开放 INTA 信号线，所以 8259 实验是使用 i386EX 的内部资源。主片 8259 将中断请求信号 IR6 和 IR7 开放，从片 8259 将中断请求信号 IR1 开放，以供实验使用。从片 8259 的 INT 与主片 8259 的 IR2 相连，完成两片 8259 的级联。关于这部分的内容详见 8259 中断实验部分。

1.2.2 接口实验单元

在本实验系统中，每个接口实验单元的电源与地均已连接好，文中电路图里的“圆圈”表示该信号通过排针引出，在实验中需要通过排线进行必要的连线来完成实验。

1. SRAM 实验单元

SRAM 实验单元由两片 62256 组成 32Kb×16 的存储器访问单元，数据宽度为 16 位，低字节与高字节的选择由 BLE、BHE 决定。如果只需要使用一片 32Kb×8 的存储器时，可以将 BLE 信号直接与 GND 相接。SRAM 实验单元电路如图 1-3 所示。

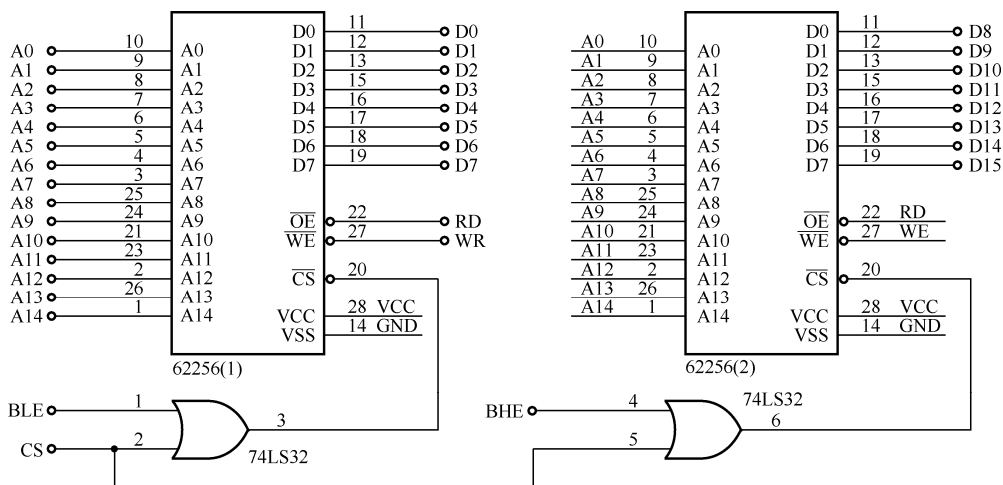


图 1-3 SRAM 实验单元电路

2. 8237DMA 实验单元

DMA 实验单元主要由一片 8237 和一片 74LS573 组成，如图 1-4 所示。

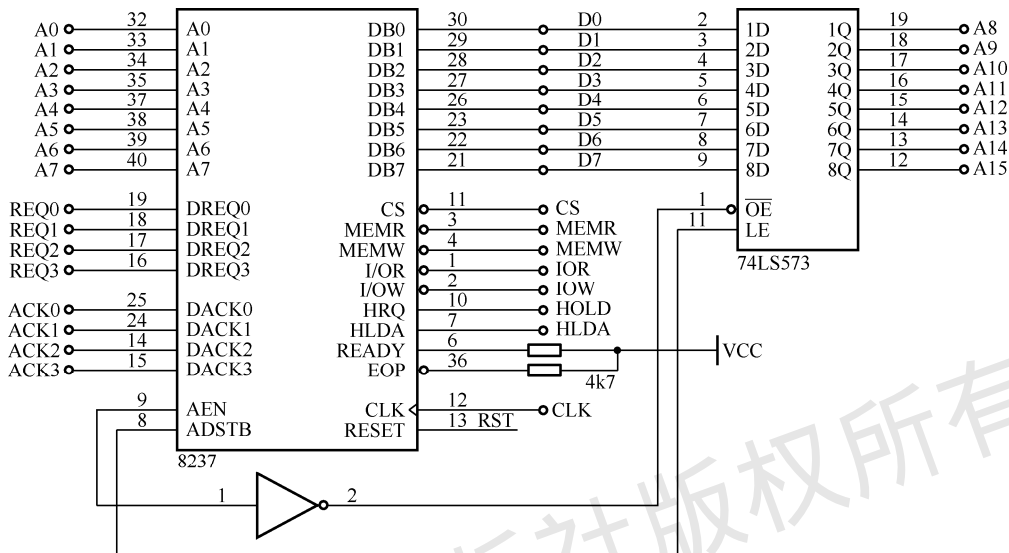


图 1-4 8237DMA 实验单元电路

3. 8254 定时器/计数器单元

8254 共有三个独立的定时器/计数器，其中 0 号和 1 号定时器/计数器开放出来可任意使用，2 号定时器/计数器用于为 8251 串行通信单元提供收发时钟，2 号定时器/计数器的输入为 1.8432MHz 时钟信号，输出连接到 8251 的 TxCLK 和 RxCLK 引脚上。定时器/计数器 0 的 GATE 信号连接好了上拉电阻，若不对 GATE 信号进行控制，可以在实验中不连接此信号。具体实验电路图如图 1-5 所示。

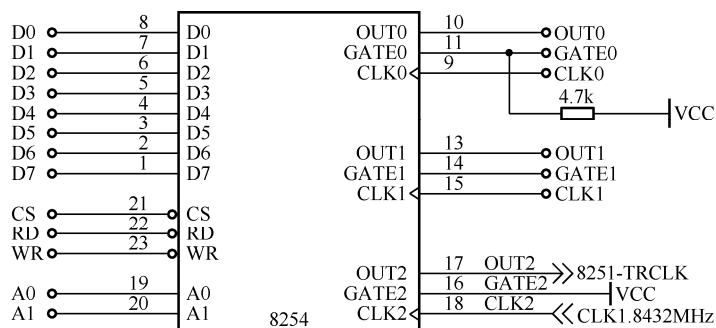


图 1-5 8254 定时器/计数器单元电路



4. 8255 并行接口单元

并行接口单元由一片 8255 组成，其复位信号已连接到系统复位上，如图 1-6 所示。

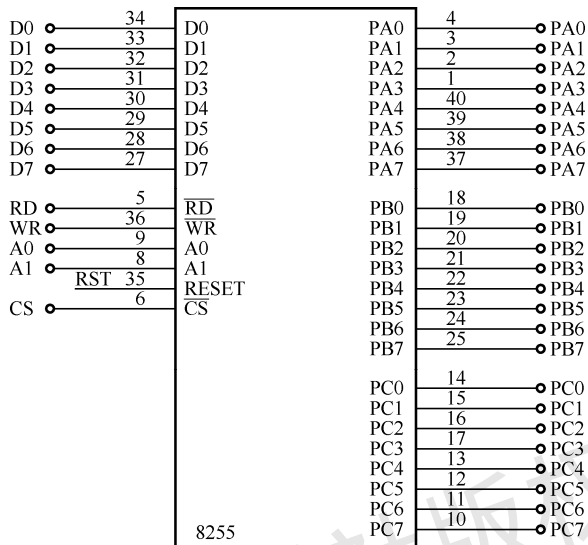


图 1-6 8255 并行接口实验单元电路

5. 8251 串行通信单元

如图 1-7 所示，串行通信控制器选用 8251，收发时钟来自于 8254 单元的定时器/计数器 2 的输出，控制器的复位信号已与系统连接好。

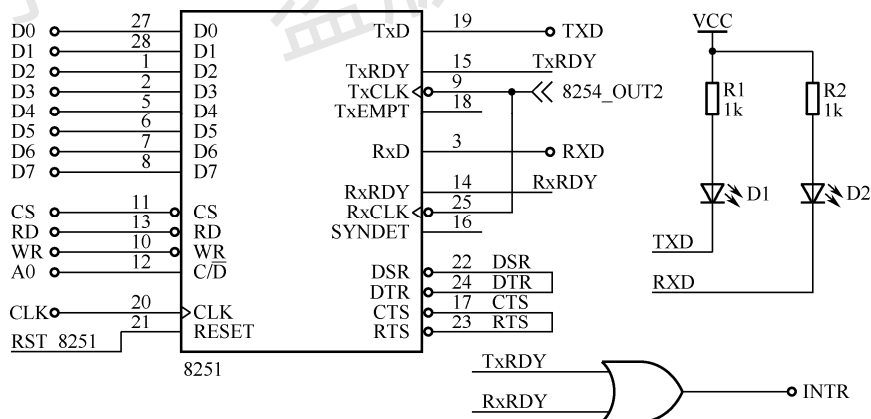


图 1-7 8251 串行通信实验单元电路

6. 模/数转换单元

模/数转换实验单元由 ADC0809 芯片及电位器电路组成，ADC0809 的 IN7 通道用于温度控制实验，增加一个 510Ω 的电阻与热敏电阻构成分压电路，如图 1-8 所示。

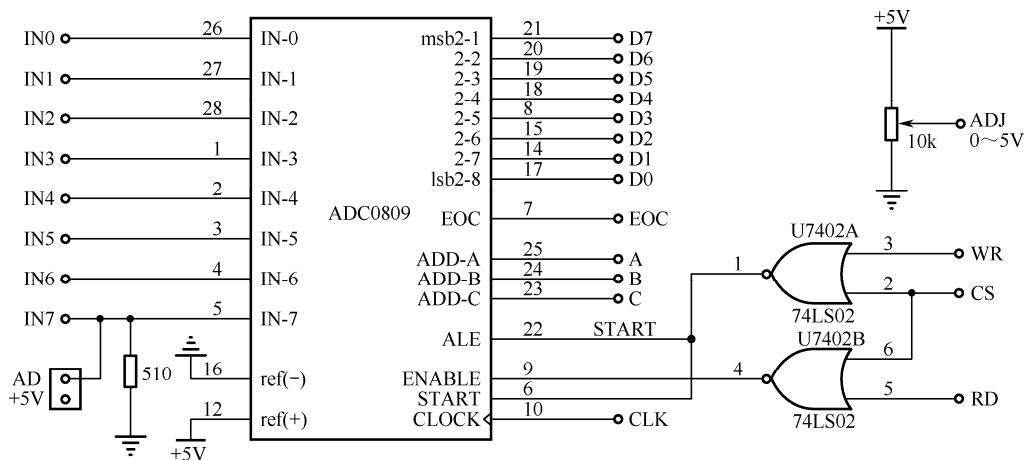


图 1-8 模/数转换实验单元电路

7. 数/模转换单元

D/A 转换实验单元由 DAC0832 与 LM324 构成，采用单缓冲方式连接。通过两级运算放大器组成电流转换为电压的转换电路，其电路如图 1-9 所示。

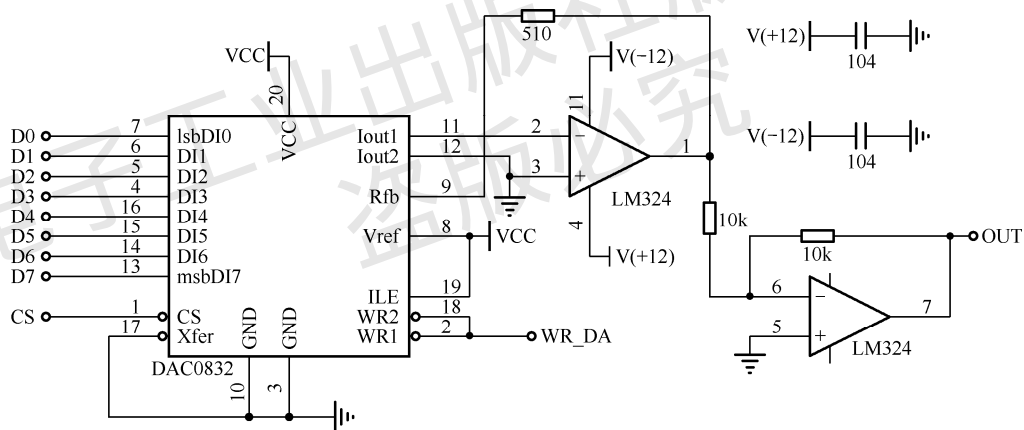


图 1-9 数/模转换实验单元电路

8. 键盘扫描及数码管显示单元

如图 1-10 所示，键盘扫描与数码管显示单元由 4 个共阴极数码管、4×4 键盘扫描阵列及显示驱动电路组成。

9. 点阵 LED 显示单元

点阵单元由 4 块 8×8LED 器件组成，74LS574 构成锁存电路，2803 构成驱动电路，如图 1-11 所示。ROW_x 表示某行，COL_x 表示某列，行为“0”，列为“1”，则对应行、列上的 LED 点亮。

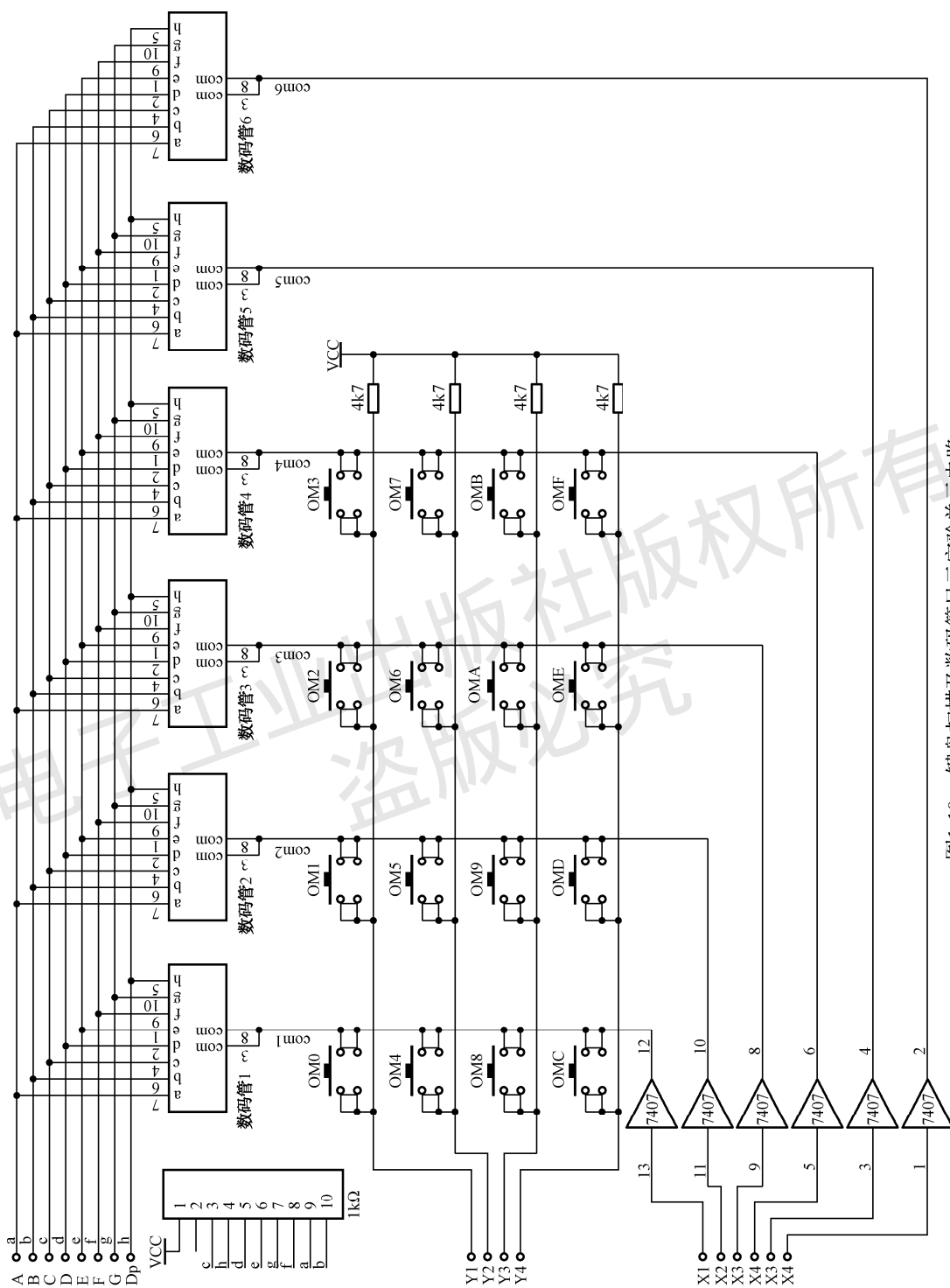


图1-10 键盘扫描及数码管显示实验单元电路

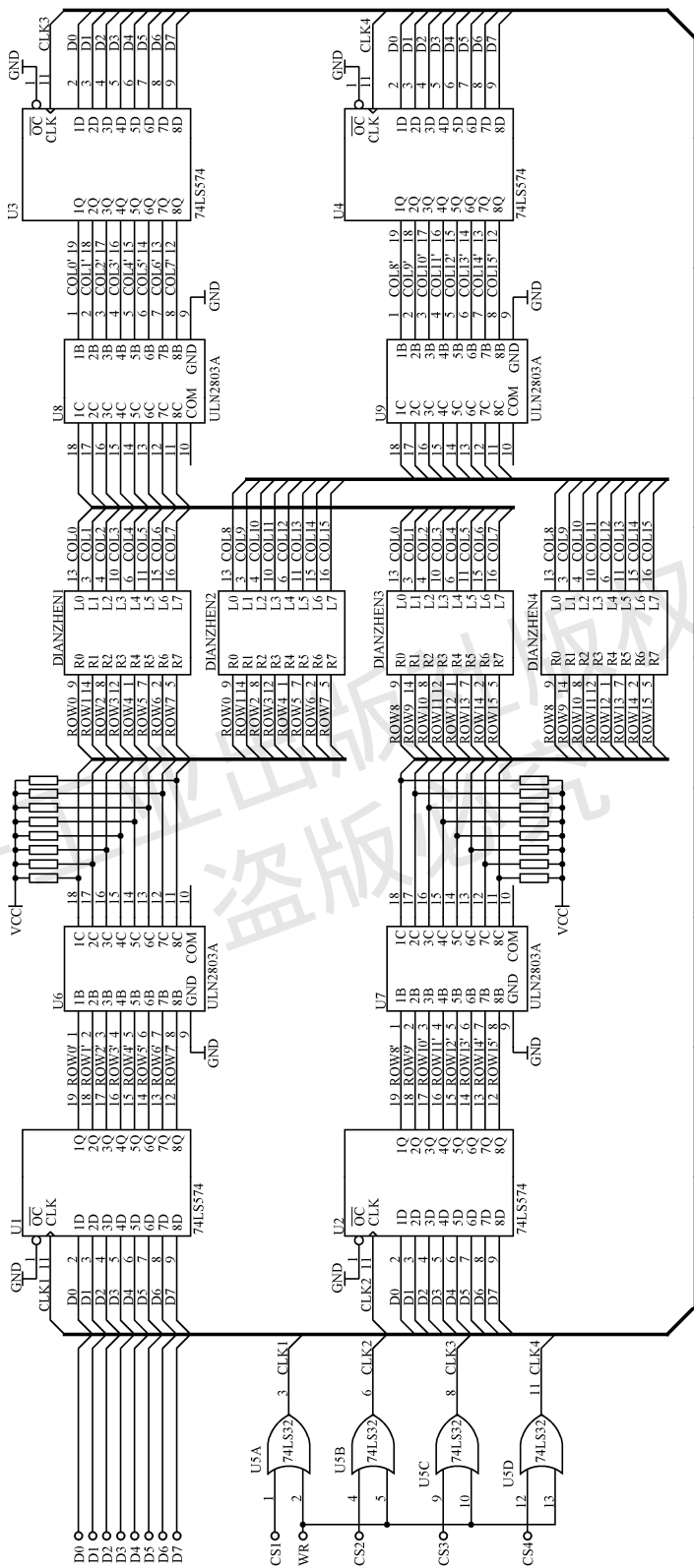


图1-11 点阵LED显示单元电路

10. 图形液晶 LCD 模块

液晶 MSC-G12864-5W 为 128×64 图形点阵液晶，LCD 类型为 STN，内置控制器，配置有 LED 背光。实验平台中的 LCD 为外接扩展件，在实验平台上留有 LCD 的扩展接口，做实验时，通过连接电缆将 LCD 与实验平台中的 LCD 接口相连，即可进行 LCD 的实验。平台中的多圈电位器可以调节液晶的对比度。LCD 接口电路如图 1-12 所示。

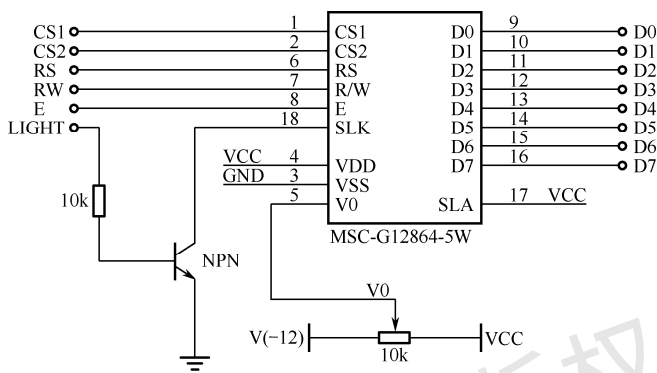


图 1-12 LCD 接口电路

11. 步进电机与直流电机单元

步进电机为四相八拍电机，如图 1-13 所示。直流电机单元由 DC 12V 直流电机及霍尔器件组成，如图 1-14 所示。UNL2803 为驱动接口芯片，由该芯片组成驱动电路，输入端 N 经过一个反相器连接到 2803 的输入端，其他四路 A、B、C、D 不经过反相器直接与 2803 相连。



图 1-13 步进电机单元电路

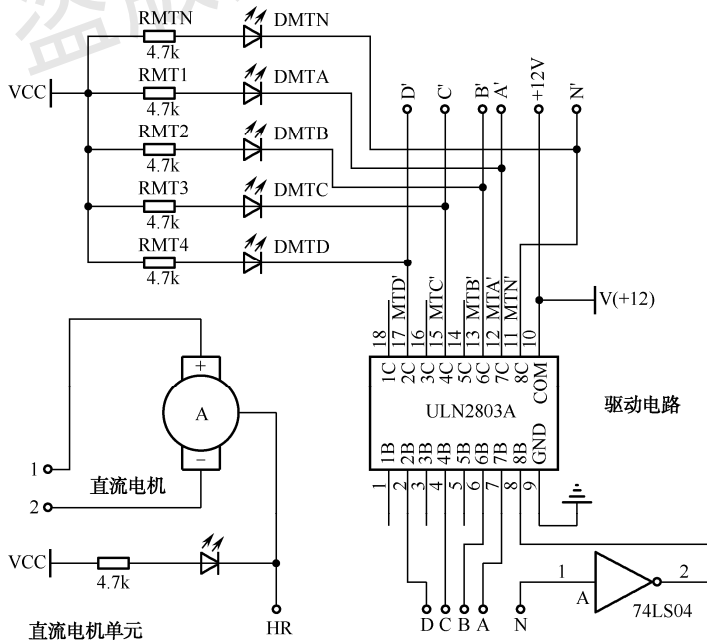


图 1-14 直流电机实验单元与驱动电路

12. 电子发声单元与温度控制单元

电子发声单元由放大电路与扬声器组成，如图 1-15 所示。温度控制单元主要由 7805、热敏电阻及大功率电阻组成，如图 1-16 所示，A 和 B 为热敏电阻的两端。

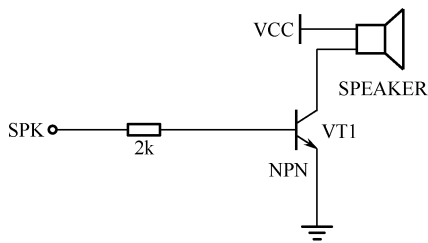


图 1-15 电子发声单元电路

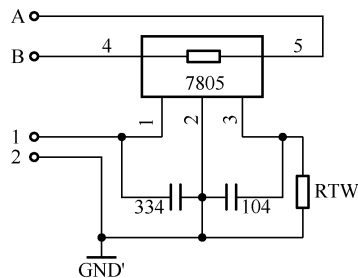


图 1-16 温度控制单元电路

13. 单次脉冲单元

该单元提供两组消抖动单次脉冲，分别为 KK1-、KK1+、KK2-、KK2+，如图 1-17 所示。“-”表示按下按键为低电平，“+”表示按下按键为高电平。

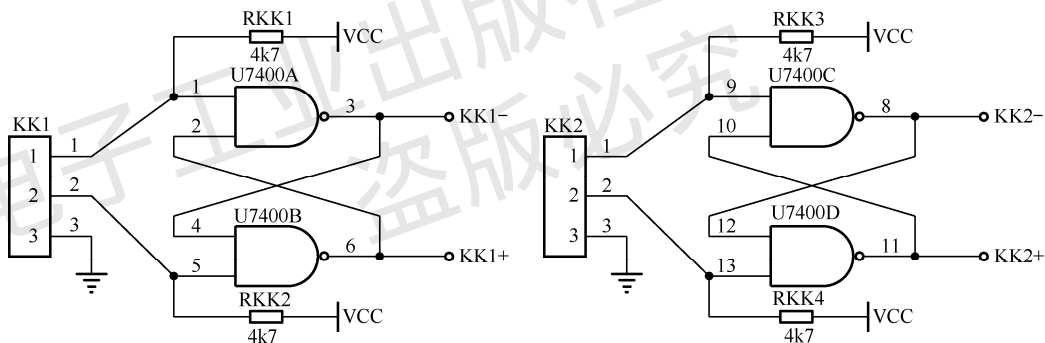


图 1-17 单次脉冲单元电路

14. 逻辑开关及 LED 显示单元

逻辑开关及 LED 显示单元由 16 组开关及 16 个 LED 组成，16 组开关未经过消抖动，16 个 LED 灯显示逻辑电平高低，为正逻辑，输入高电平 LED 点亮，如图 1-18 所示。

15. 转换单元

转换单元提供了排线和圆锥孔相互转接及扩展的单元。

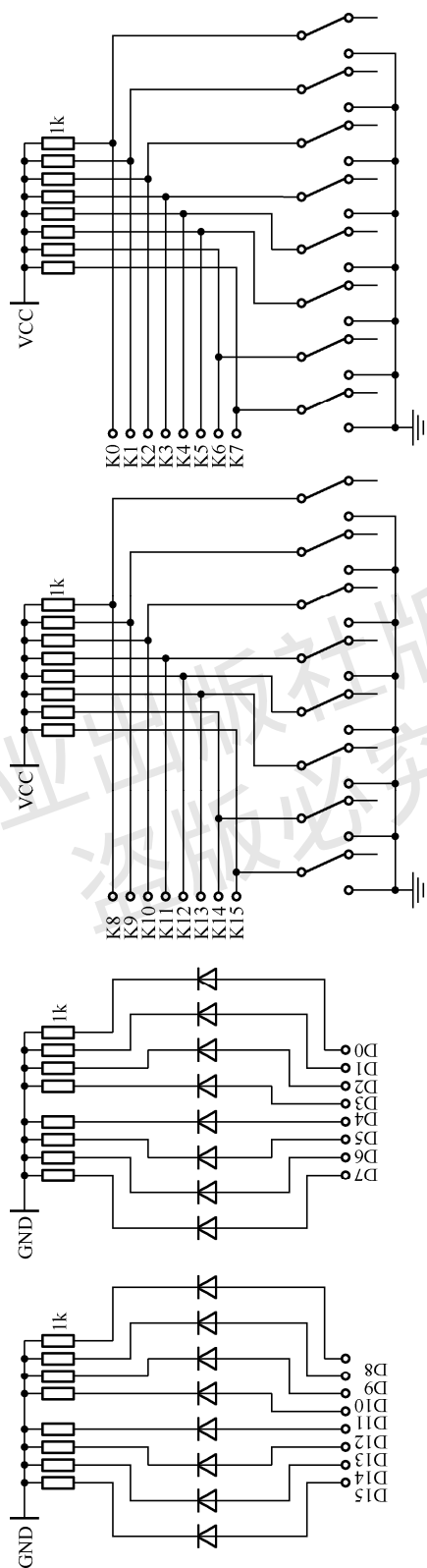


图1-18 逻辑开关及LED显示单元电路

16. 扩展单元

扩展单元提供 2 组 40 线通用集成电路扩展单元和一个扩展模块总线单元。扩展模块总线插座的信号定义如图 1-19 所示。

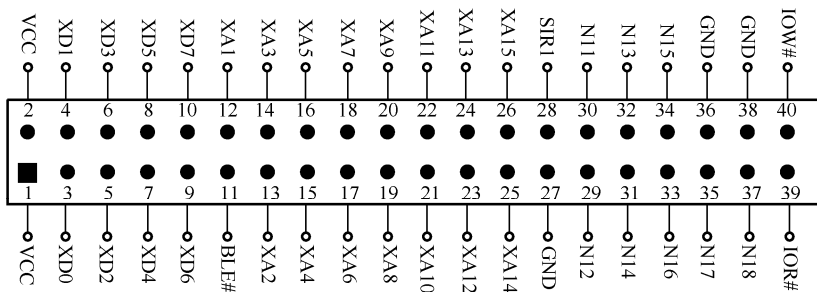


图 1-19 扩展模块总线 EX_BUS 单元引脚示意图

其中的 N11、N12、N13、N14、N15、N16、N17、N18 这 8 个引脚已连接到对应于它下方 40 线通用集成电路扩展单元的 11、12、13、14、15、16、17、18 脚上，可由用户根据需要来定义。其余的引脚于 x86 系统总线单元各引脚相对应。

17. 系统总线单元

系统总线单元各引脚见表 1-1。其中 CPU 选择开关分为 51 和 386 两个挡位。开关打到 386 挡，表明 i386EX CPU 与设备箱体上的通信串口（或 USB 口）相连，可与 PC 通信；开关打到 51 挡，表明选配的 TD-51 单片机开发板与设备箱体上的通信串口（或 USB 口）相连，可与 PC 通信。

18. 386 CPU 单元

386 CPU 单元右下角有 JDBG 短路设置：短路设置在 DBG 时，CPU 与 PC 联机进行调试及运行系统 SRAM 中的程序；短路设置在 RUN 时，CPU 与 PC 断开，若已经将设计的实验程序固化到系统的 Flash 存储器中的话，则系统此时复位后即可直接运行使用者的实验程序。

19. 时钟源单元

时钟源单元提供 3 个时钟供实验用，分别是 1.8432MHz、184.32MHz 和 18.432MHz。

1.2.3 程序固化及脱机运行

1. 程序固化

TD-PITE 实验系统可以将实验程序固化到系统存储器中，以实现脱机运行。将实验程序编译、链接无误后进行加载，加载完成便可以程序固化。单击“调试”菜单中的“固



化程序”便可以将程序固化到系统存储器中，如图 1-20 所示。



图 1-20 “调试”菜单

2. 脱机运行

如果系统存储器中已经固化有实验程序，则可以实现脱机运行程序。实验箱右侧有一个短路块 JDBG，若将短路块短接在 DBG 端，可以与 PC 端 Wmd86 软件联机进行调试；若将短路块短接在 RUN 端，可以实现程序的脱机运行。

如果将 i386EX 系统板嵌入其他应用中，为实现脱机运行程序，可以将信号 P3.6/PWRDOWN 连接到 GND。见附录 C 中的图 C.1，JP2 的 20 引脚就是信号 P3.6/PWRDOWN。

1.3 实验系统软件开发环境

1.3.1 软件系统概述

微机系统联机软件 Wmd86 是为 TD-PITE 微机原理及接口技术教学实验系统配套的集成开发调试软件，该软件具有汇编语言和 C 语言源语言级调试跟踪界面，比起传统的 DEBUG 调试，操作更简单，视觉效果更直接。如果用户习惯于 DEBUG 调试，也可以单击输出区的调试标签对源程序进行 DEBUG 调试。

Wmd86 集成开发软件具有如下特点：

- (1) 支持汇编语言和 C 语言两种编译环境。
- (2) 提供 16 位寄存器和 32 位寄存器状态切换。
- (3) 高度可视化的源语言级调试跟踪界面。
- (4) 实时监视寄存器，能够即时对寄存器中的值进行修改。
- (5) 可以选择要监视的全局变量，进行实时监视和即时修改。
- (6) 可以选择是否要实时监视堆栈寄存器。
- (7) 集成了一个专用图形显示的虚拟仪器。

1.3.2 软件使用说明

1. 软件主界面

集成开发软件的主界面主要分为三部分：程序编辑区、寄存器/变量/堆栈区和信息输出区，软件主界面如图 1-21 所示。



图 1-21 Wmd86 软件主界面

(1) 程序编辑区

程序编辑区域位于主界面右上部，用户可在程序编辑区用“新建”命令创建一个新文档，或用“打开”命令打开一个已存在的文档，在文档中用户可编辑程序。在程序编辑区可以同时打开多个文档，单击文档标签可激活任一文档。编译、链接、加载以及调试命令只针对当前活动文档。

(2) 寄存器/变量/堆栈区

寄存器/变量/堆栈区位于主界面左上部，包括三个部分：寄存器区、变量区和堆栈区。寄存器区和变量区用于实时监视寄存器和变量，也能够修改寄存器和变量的值，堆栈区主要用于实时监视堆栈寄存器。

(3) 信息输出区

信息输出区位于主界面底部，输出区包含三个部分：编译区、调试区和结果区。编译区显示编译和链接的结果，如果编译时有错误或警告，双击错误或警告信息，错误标识符



会指示到相应的有错误或警告的行。调试区主要用于 DEBUG 调试。结果区主要用于显示程序加载结果、程序运行输出结果和复位结果。结果区中有“加载中，请稍候…”的字样表示 Wmd86 联机软件正在把可执行文件加载到下位机中，用户此时应等待直到加载完成；“加载成功”表示加载完成且成功；“加载失败”表示加载完成且失败，此时应重新加载。程序中用“INT 10H”输出的结果也显示在此区。软复位或硬件复位成功后，结果区显示“复位成功！”。如果复位不成功则不显示此句。

2. 菜单功能

(1) “文件”菜单项

文件菜单如图 1-22 所示。

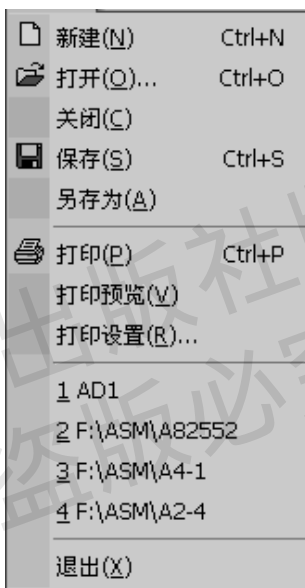


图 1-22 “文件”菜单

① 新建 (N)：用此命令在 Wmd86 软件中建立一个新文档。

② 打开 (O)：用此命令在窗口中打开一个现存的文档。您可同时打开多个文档，单击某文档的标签可激活此文档。您可用窗口菜单在多个打开的文档中切换。

③ 关闭 (C)：用此命令来关闭当前活动文档。Wmd86 会建议您在关闭文档之前保存对您的文档所做的改动。如果您没有保存而关闭了一个文档，您将会失去自从您最后一次保存以来所做的所有改动。在关闭一无标题的文档之前，Wmd86 会显示另存为对话框，建议您命名和保存文档。

④ 保存 (S)：用此命令将当前活动文档保存到它的当前的文件名和目录下。当您第一次保存文档时，Wmd86 显示“另存为”对话框以便您命名您的文档。

⑤ 另存为 (A)：用此命令来保存并命名活动文档。Wmd86 会显示“另存为”对话框以便您命名您的文档。

⑥ 打印 (P)：用此命令来打印一个文档。

⑦ 打印预览 (V)：用此命令按要打印的格式显示活动文档。当您选择此命令时，主窗口就会被一个打印预览窗口所取代。这个窗口可以按它们被打印时的格式显示一页或两页。

⑧ 打印设置 (R)：用此命令来选择连接的打印机及其设置。

⑨ 最近浏览文件：此列表显示最近打开过的文件，最多显示四个最近打开的文件。

⑩ 退出 (X)：用此命令来退出 Wmd86 集成开发软件。软件会提示您保存尚未保存的改动。

(2) “编辑”菜单项

“编辑”菜单如图 1-23 所示。



图 1-23 “编辑”菜单

① 撤消：可用此命令来撤消上一步操作。如果无法撤消上一步操作，菜单上的撤消命令会变灰。

② 重复：可用此命令来恢复撤消的编辑操作。如果无法恢复撤消的编辑操作，菜单上的重复命令会变灰。

③ 剪切 (T)：用此命令将当前被选取的数据从文档中删除并放置于剪贴板上。如当前没有数据被选取时，此命令则不可用。把数据剪切到剪贴板上将取代原先存放在那里的内容。

④ 复制 (C)：用此命令将被选取的数据复制到剪贴板上。如当前无数据被选取时，此命令则不可用。把数据复制到剪贴板上将取代以前存在那里的内容。

⑤ 粘贴 (P)：用此命令将剪贴板上内容的一个副本插入到插入点处。如剪贴板是空的，此命令则不可用。

⑥ 查找：单击此命令将弹出“查找”对话框，用于查找指定字符串。

⑦ 替换：单击此命令将弹出“替换”对话框，找到某一字符串，并用指定字符串替换之。

(3) “查看”菜单项

“查看”菜单如图 1-24 所示。

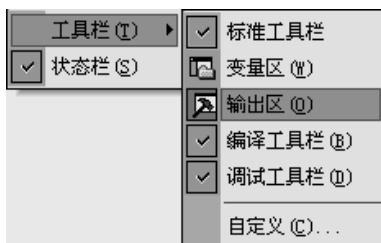


图 1-24 “查看”菜单

- ① 工具栏 (T): 显示或隐藏工具栏
- ② 状态栏 (S): 显示或隐藏状态栏
- ③ 工具栏如下。

- 标准工具栏: 用此命令可显示和隐藏标准工具栏。标准工具栏包括了 Wmd86 中一些最普通命令的按钮, 如文件打开。在工具栏被显示时, 一个打钩记号出现在该菜单项目的旁边。
- 变量区 (W): 用此命令可显示和隐藏寄存器/变量/堆栈区。
- 输出区 (O): 用此命令可显示和隐藏输出区。
- 编译工具栏 (B): 用此命令可显示和隐藏编译工具栏。
- 调试工具栏 (D): 用此命令可显示和隐藏调试工具栏。
- 自定义 (C): 见自定义功能。

(4) “端口”菜单项

“端口”菜单如图 1-25 所示。

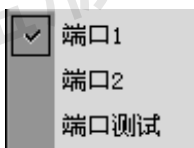


图 1-25 “端口”菜单

① 端口 1: 此命令用来选择串口 1 进行联机通信, 该命令会对串口 1 进行初始化操作, 并进行联机测试, 报告测试结果。

② 端口 2: 此命令用来选择串口 2 进行联机通信, 该命令会对串口 2 进行初始化操作, 并进行联机测试, 报告测试结果。

③ 端口测试: 此命令用来对当前选择的串口进行联机通信测试, 并报告测试结果。

(5) “编译”菜单项

“编译”菜单如图 1-26 所示。

① 编译 (C): 编译当前活动文档中的源程序, 在源文件目录下生成目标文件。如果有错误或警告生成, 则在输出区显示错误或警告信息, 双击错误或警告信息, 可定位到有错误或警告的行, 修改有错误或警告的行后应重新“编译”。如果编译没有错误生成 (即使有警告生成), 会激活“链接”菜单项和工具栏中的“链接”按钮, 以便进行链接。编译时

自动保存源文件中所做的修改。

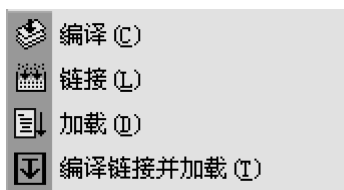


图 1-26 “编译”菜单

② 链接 (L)：链接编译生成的目标文件，在源文件目录下生成可执行文件。如果有错误或警告生成，则在输出区显示错误或警告信息，查看错误或警告信息修改源程序，修改后应重新“编译”和“链接”。如果链接没有错误生成（即使有警告生成），会激活“加载”菜单项和工具栏中的“加载”按钮，以便进行加载。

③ 加载 (D)：把链接生成的可执行文件加载到下位机。在加载过程中输出区有“加载中，请稍候...”的字样，用户此时应该等待直到加载完成。加载完成后，如果加载成功，输出区显示“加载成功！”，并激活“调试”菜单中的菜单项和调试工具栏中的按钮，此时 CS 和 IP 指向程序的开始执行行并在此行设置执行标记。如果加载失败，输出区显示“加载失败！”，此时“调试”菜单中的菜单项和调试工具栏中的按钮都不能使用，应重新进行“加载”。

④ 编译链接并加载 (T)：依次执行编译、链接和加载。

(6) “设置”菜单

“设置”菜单如图 1-27 所示。

① 语言：设置语言环境。

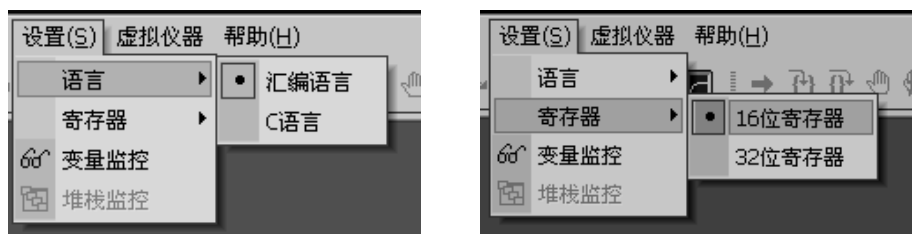
- 汇编语言：设置编译环境为汇编语言环境。此时可编辑、编译和链接 IBM-PC 汇编语言源程序。

- C 语言：设置编译环境为 C 语言环境。此时可编辑、编译和链接 C 语言源程序。由于监控目前不支持浮点运算，故 C 语言程序中不应该出现浮点运算，如果 C 语言程序中出现浮点运算，链接时会出现错误。

② 寄存器：设置寄存器格式。

- 16 位寄存器：设置成 16 位寄存器，可观察到 16 位寄存器的变化。

- 32 位寄存器：设置成 32 位寄存器，可观察到 32 位寄存器的变化。



(a) 语言环境设置

(b) 寄存器格式设置

图 1-27 “设置”菜单



③ 变量监控：加载成功后才可用“变量监控”按钮，如图 1-28 所示。



图 1-28 “变量监控”对话框

左边的列表框为程序中的全局变量，系统只能监视全局变量，右边的列表框为系统正在监视的全局变量。选择需要监视的变量，单击“加入监视”按钮，以实现对该变量的监视。如果想从变量区去掉某一正在监视的变量，选中变量后单击“停止监视”按钮将其从列表中删除。

在汇编语言源文件中，数据段定义的变量并不是全局变量，因此数据段定义的变量并不出现在图 1-28 所示对话框的左边列表，要想监视这些变量，必须使它们成为全局变量。使一个变量成为全局变量的方法是用关键字 PUBLIC 在源程序的最前面声明之，格式是：PUBLIC symbol[,...]，范例如下：

```
PUBLIC  mus_time
PUBLIC  mus_freq

DATA1  SEGMENT
mus_time  DB  01h
mus_freq  DW  1234H
DATA1  ENDS
```

数据段 DATA1 中的数据 mus_time、mus_freq 经过 PUBLIC 声明后成为全局变量，编译、链接、加载完成后，可对这两个变量进行监控。

在 C 语言源文件中，函数内部定义的变量不是全局变量，函数外面定义的变量才是全局变量，因此系统只能监视函数外面定义的变量。要想监视某一变量，应该把他定义在函数的外面。

④ 堆栈监控。

“堆栈监控”对话框如图 1-29 所示，选择“不监控堆栈”单选项，确定后不监视堆栈

寄存器，选择“监控堆栈”单选项，确定后监视堆栈寄存器。默认选项为“不监控堆栈”。

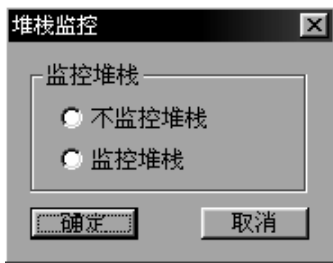


图 1-29 堆栈监控对话框

(7) “调试”菜单项

“调试”菜单如图 1-30 所示。



图 1-30 “调试”菜单

① 设置断点/删除断点 (B)：当前光标所在的行为当前行，如果当前行无断点则在当前行设置断点，如果当前行有断点则删除当前行的断点。设置断点后的行如图 1-31 所示。并不是源程序的所有行都可以设置断点，如伪操作行和空行不能设置断点。源程序设置的断点数不能超过 8 个。

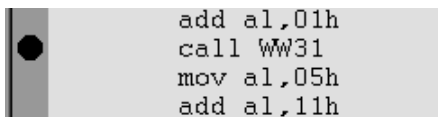


图 1-31 断点设置

② 清除所有断点 (D)：清除源程序中设置的所有断点，只有当设置的断点数大于零时，该菜单才激活。

③ 设置起点 (J)：当前光标所在的行为当前行，此命令把当前行设置为程序的起点，即程序从此行开始运行，寄存器区的 CS 和 IP 的值刷新后指向此行。设置程序起点的行如图 1-32 所示。并不是源程序的所有行都可以设置起点，如伪操作行和空行不能设置起点。

CODE	SEGMENT
	ASSUME CS:CODE,DS:DAT
→ START:	mov ax,DAT
	mov ds,ax
	call far ptr ww1
	add al,01h
	add al,01h

图 1-32 起点设置

④ 单步 (I): 单击此命令使程序执行一条语句, 如果是函数则进入函数内部, 执行后刷新所有的变量和寄存器的值。

并不是所有的语句行都适用单步, 如系统调用语句不应该使用单步, 而应该用跳过命令跳过该语句行。

⑤ 跳过 (O): 单击此命令使程序执行一个函数, 执行后刷新所有变量和寄存器的值。只有在当前执行行为函数调用或系统调用时, 才用此命令。如果函数内部有断点, 单击“跳过”指令后, 程序会停在函数内部有断点的行。

⑥ 运行到断点/运行: 从当前执行行开始向后运行, 如果没有断点, 则运行直到程序结束。如果有断点, 则运行到断点后停止, 运行到断点后如果再次单击此菜单, 则从当前断点位置继续执行, 直到再次遇到断点或程序结束。

⑦ 停止: 发送此命令使程序停止运行, 程序停止后刷新所有寄存器和变量。

⑧ 固化程序: 将实验程序固化到系统存储器 Flash 中, 以实现程序的脱机运行。此命令只有在程序正常加载后执行。

(8) “虚拟仪器”菜单项

具体各项说明详见 1.3.2 节专用图形显示介绍部分。

(9) “窗口”菜单项

如图 1-33 所示, 窗口菜单提供了以下命令, 这些命令使您能在应用程序窗口中安排多个文档的多个视图。

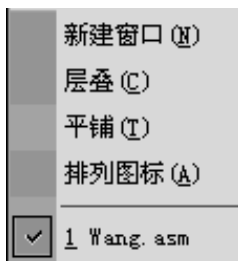


图 1-33 “窗口”菜单

① 新建窗口 (N): 用此命令来打开一个具有与活动的窗口相同内容的新窗口。您可同时打开数个文档窗口以显示文档的不同部分或视图。如果您对一个窗口的内容做了改动, 所有其他包含同一文档的窗口也会反映出这些改动。当您打开一个新的窗口, 这个新窗口就成了活动的窗口并显示于所有其他打开窗口之上。

② 层叠 (C): 用此命令按相互重叠形式来安排多个打开的窗口。

③ 平铺 (T): 用此命令按互不重叠形式来安排多个打开的窗口。

④ 排列图标 (A): 用此命令在主窗口的底部安排被最小化的窗口的图标。如果在主窗口的底部有一个打开的窗口, 则有可能会看不见某些或全部图标, 因为它们在这个文档窗口的下面。

⑤ 窗口 1, 2, ...: Wmd86 在窗口菜单的底部显示出当前打开的文档窗口的清单。有一个打钩记号出现在活动的窗口的文档名前。从该清单中挑选一个文档可使其窗口成为活动窗口。

(10) “帮助”菜单项

“帮助”菜单提供以下的命令, 为您提供使用这个应用程序的帮助。

① 关于 (A) Wmd86: 用此命令来显示所使用的 Wmd86 软件版本的版权信息和版本号。

② 帮助主题 (H): 用此命令来显示软件帮助信息。可以查看关于使用 Wmd86 的一些指令以及各种不同类型参考资料。

3. 工具栏功能介绍

(1) 标准工具栏

标准工具栏按钮图标如下。



① 新建文档: 用此按钮在 Wmd86 中建立一个新文档。

② 打开文档: 用此按钮在一个新的窗口中打开一个现存的文档。

③ 保存: 用此按钮将当前活动文档保存到其当前的文件名和目录下。

④ 剪切: 用此按钮将当前被选取的数据从文档中删除并放置于剪贴板上。

⑤ 复制: 用此按钮将被选取的数据复制到剪贴板上。

⑥ 粘贴: 用此按钮将剪贴板上内容的一个副本插入到插入点处。

⑦ 打印: 用此按钮来打印一个文档。

⑧ 撤销: 用此按钮来撤销上一步编辑操作。

⑨ 恢复: 用此按钮来恢复撤销的编辑操作。

⑩ 变量区: 用此按钮可显示和隐藏变量和寄存器区。

⑪ 输出信息区: 用此按钮可显示和隐藏输出区。

⑫ 选择要监视的全局变量: 加载成功后才可用此按钮。单击此按钮, 可进行全局变量监视。

⑬ 选择是否要监视堆栈: 单击此按钮将弹出堆栈监控对话框。


(2) 编译工具栏


编译工具栏按钮图标如下。



① 编译: 编译当前活动文档中的源程序, 在源文件目录下生成目标文件。

② 链接: 链接编译生成的目标文件, 在源文件目录下生成可执行文件。

③  加载：把链接生成的可执行文件加载到上位机。


④  编译链接并加载：依次执行编译、链接和加载。


(3) 调试工具栏


调试工具栏按钮图标如下。





①  设置起点：当前光标所在的行为当前行，此命令把当前行设置为程序的起点，即程序从此行开始运行，寄存器区的 CS 和 IP 的值刷新后指向此行。


②  单步：单击此命令使程序执行一条语句。

③  跳过：单击此命令使程序执行一个函数，执行后刷新所有变量和寄存器的值。

④  设置断点/删除断点：为光标所在行设置断点或删除当前行的已有断点。源程序设置的断点数不能超过 8 个。

⑤  清除所有断点：清除源程序中设置的所有断点。

⑥  运行到断点/运行：从当前执行行开始向后运行，如果没有断点，则运行直到程序结束。如果有断点，则运行到断点后停止，运行到断点后再次单击此按钮，则程序从当前断点位置继续执行，直到再次遇到断点或程序结束。

⑦  停止：发送此命令使程序停止运行，程序停止后刷新所有寄存器和变量的值。

4. 专用图形显示

专用图形显示功能主要用于观察“直流电机闭环调速”跟踪曲线（实时显示给定转速、反馈转速的曲线响应关系）及“温度单元温度闭环控制”跟踪曲线（实时显示给定温度和反馈温度的曲线响应关系）。“直流电机闭环调速”实验响应曲线示例如图 1-34 所示。

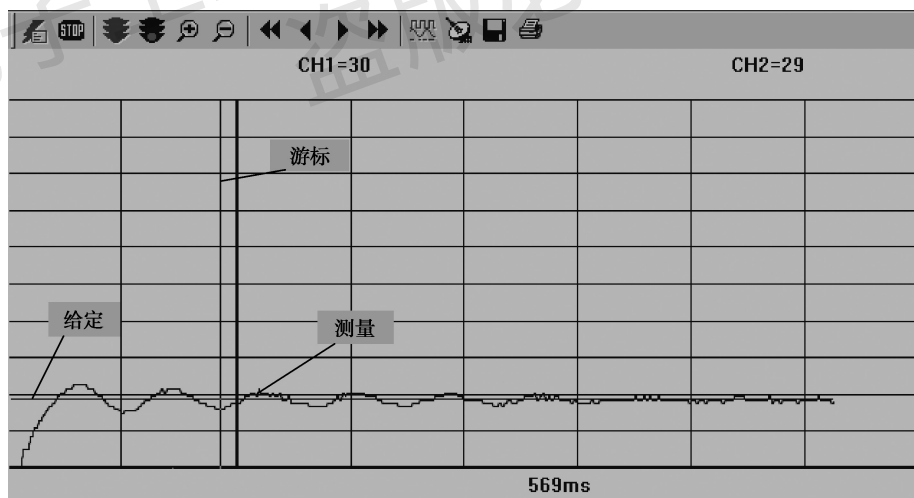


图 1-34 “直流电机闭环调速”实验响应曲线

(1) 显示说明


① CH1=30：要求电动机达到的转速值。


② CH2=29：运行状态下表示电机当前的转速值，暂停状态下表示指定时刻的电机的


转速值。


③ 569ms: 在运行态下不出现此值, 只有在暂停状态下才出现此值, 图形中有一条竖直线为游标, 只有在暂停状态下才出现。“569ms”表示游标所在位置的时刻与图形最左端时刻的差值。图形中的黄线为每个时刻要求电机达到的转速值, 图形中的绿线表示每个时刻电机的实际转速值。


(2) 工具栏功能简介


①  按钮: 运行加载在下位机中的程序。“CH1=”后显示的是当前时刻要求电机达到的转速值, “CH2=”后显示的是当前电机实际达到的转速值。


②  停止: 使下位机中运行的程序停止。

③  暂停: 在运行状态下使能, 使波形暂停显示并出现游标。


④  继续: 在暂停状态下使能, 使波形继续显示, 游标消失。


⑤  放大: 放大波形。

⑥  缩小: 缩小波形。

⑦  快速左移游标: 在暂停状态下, 使游标快速向左移动。“CH1=”后显示的是游标所在时刻要求电机达到的转速值, “CH2=”后显示的是游标所在时刻电机实际达到的转速值。“XX ms”表示游标所在位置的时刻与图形最左端时刻的差值。

⑧  左移游标: 在暂停状态下, 使游标向左缓慢移动。

⑨  快速右移游标: 在暂停状态下, 使游标向右缓慢移动。

⑩  右移游标: 在暂停状态下, 使游标快速向右移动。

⑪  记录波形: 单击此按钮, 出现如图 1-35 所示对话框。

选中“图一”单选按钮, 单击“确定”按钮, 系统会把当前时刻的波形保存到图一中, 共可保存三幅图。图形只是保存于数据缓冲中, 供图形比较时使用。



⑫  显示波形: 显示保存到图一、图二和图三中的波形, 此时可以对几幅图进行比较, 如图 1-35 所示。



图 1-35 “存储波形”对话框

⑬  保存波形: 以.bmp 格式保存当前屏幕上的波形到指定文件。

⑭  打印波形: 打印当前屏幕上的波形。

5. 右键菜单功能

如果用户在程序编辑区单击右键, 出现编辑菜单, 如果在非客户区单击右键, 出现工具栏菜单。分别介绍如下:



“编辑”菜单如图 1-36 所示，“编辑”菜单提供了“剪切”、“复制”、“粘贴”命令。

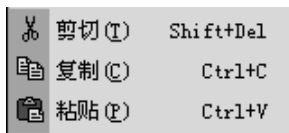


图 1-36 “编辑”菜单

“工具栏”菜单如图 1-37 所示，与“查看”菜单中的工具栏的下拉菜单内容相同，功能亦相同。



图 1-37 “工具栏”菜单

6. Debug 调试命令

Wmd86 集成开发软件输出区集成有 Debug 调试，单击“调试”标签进入 Debug 状态，会出现命令提示符“>”，主要命令叙述如下：

A 汇编命令

格式：A[段址:] [偏移量] ↵

A 段址:偏移量 ↵——从段址:偏移量构成的实际地址单元起填充汇编程序的目标代码。

A 偏移量 ↵——从默认的段址与给定的偏移量构成的实际地址单元起填充汇编程序目标代码。

A ↵——从默认段址:默认偏移量构成的实际地址单元起填充汇编程序的目标代码。

输入上述命令后，屏幕显示地址信息，即可输入源程序。若直接回车，则退出命令。汇编程序输入时，数据一律为十六进制数，且省略 H 后缀。[m]类操作一定要在[]之前标注 W（字）或 B（字节），如 MOV B[2010], AX, MOV W[2010], AX。

例：在“>”提示符下输入 A2000 ↵，此时默认的段址 CS 为 0000，规定偏移量 IP 为 2000，屏幕显示与操作示例见表 1-2。

表 1-2 小汇编操作示例

显示内容	输入内容
0000:2000	MOV AX, 1234 ↵
0000:2003	INC AX ↵

续表

显示内容	输入内容
0000:2004	DEC AX ↵
0000:2005	JMP 2000 ↵
0000:2007	↵

B 断点设置

在系统提示符下，输入 B ↵，系统提示[i]:，等待输入断点地址。输入断点地址后回车，系统继续提示[i+1]:。若直接按回车键，则结束该命令。系统允许设置最多 10 个断点，断点的清除只能是通过系统复位或重新上电来实现。B 命令示例见表 1-3。

表 1-3 B 命令示例

显示内容	输入内容
>	B ↵
[0]:	2009 ↵
[1]:	↵

D 显示内存数据命令

格式: D[[段址:]起始地址,[尾地址]] ↵

D 命令执行后屏幕上显示一段地址单元中的数据，在显示过程中，可用 Ctrl+S 键来暂停显示，用任意键继续；也可用 Ctrl+C 键终止数据显示，返回监控状态。

E 修改存储单元内容命令

格式: E[[段址:]偏移量] ↵

该命令执行后，则按字节显示或修改数据，可通过“空格”键进入下一高地址单元数据的修改，使用“-”键则进入下一低地址单元进行数据的修改，并可填入新的数据来修改地址单元的内容。若按回车键，则结束 E 命令。E 命令示例见表 1-4。

表 1-4 E 命令示例

显示内容	输入内容
>	E3500 ↵
0000:3500 00_	05 空格
0000:3501 01_	空格
0000:3502 02_	-
0000:3501 01_	↵

G 运行程序命令

格式: G=[段址:]偏移量 ↵

GB=[段址:]偏移量] ↵

其中，G 格式表示无断点连续运行程序，GB 格式表示带断点连续运行程序，连续运行



过程中，当遇到断点或按下 Ctrl+C 键时，终止程序运行。

M 数据块搬移

格式：M 源地址,尾地址 目标地址 ↵

R 检查和修改寄存器内容命令

格式：R ↵ 或 R 寄存器名 ↵

R ↵ 操作后，屏幕显示：CS=XXXX DS=XXXX IP=XXXX AX=XXXX F=XXXX

若需要显示并修改特定寄存器内容，则选择 R 寄存器名 ↵ 操作。如 RAX ↵，则显示：AX=XXXX，按回车键，结束该命令。若输入四位十六进制数并回车，则将该数填入寄存器 AX 中，并结束该命令。

T 单步运行指定的程序

格式：T=[段址:]偏移量] ↵

每次按照指定的地址或 IP/PC 指示的地址，单步执行一条指令后则显示运行后的 CPU 寄存器情况。

U 反汇编命令

格式：U[[段址:]起始地址[,尾地址]]

1.3.3 实验系统认识实验

1. 实验目的

掌握 TD-PITE 微机原理与接口技术教学实验系统的操作，熟悉 Wmd86 集成开发调试软件的操作环境，为后续软件实验和硬件实验奠定基础。

2. 实验设备

- (1) 微型计算机 1 台。
- (2) TD-PITE 微机接口实验系统 1 套。

3. 实验内容

编写汇编实验程序，将 00H~0FH 共 16 个数写入内存从 3000H 开始的连续 16 个存储单元中。

4. 实验步骤

(1) 打开 Wmd86 软件，进入 Wmd86 集成开发环境。

(2) 根据程序设计所选择语言的不同，通过在“设置”下拉列表来选择需要使用的语言和寄存器类型。这里的语言选项我们选择“汇编语言”，寄存器设置成“16 位寄存器”，如图 1-38、图 1-39 所示。设置语言环境后，下次再启动软件，语言环境将保持此次的设置不变。



图 1-38 语言环境设置界面

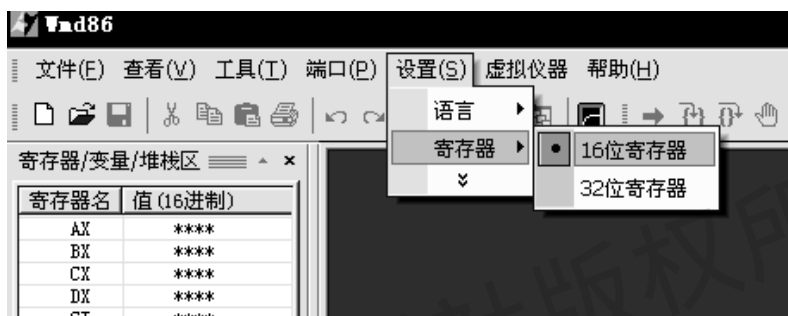


图 1-39 寄存器设置界面

(3) 选择好语言和寄存器后，单击“新建”命令或按 Ctrl+N 键来新建一个文档，如图 1-40 所示。系统默认的文件名为 Wmd861。

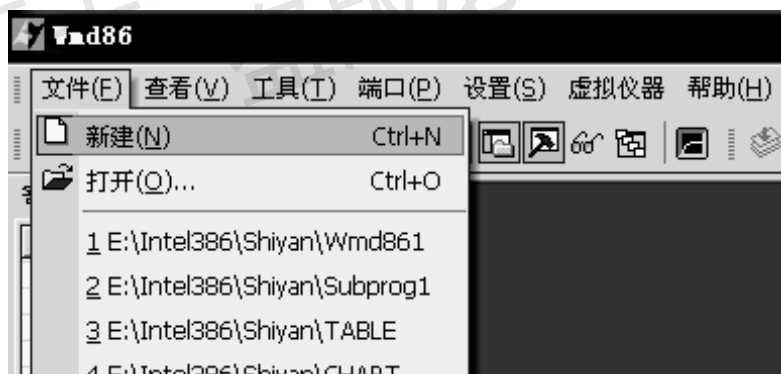


图 1-40 新建文件界面

(4) 在程序编辑区输入如下内容的实验程序，执行保存操作时系统会提示输入新的文件名，输完后单击“保存”按钮。

```

SSTACK  SEGMENT STACK                ;定义堆栈段
          DW 32 DUP(?)
SSTACK  ENDS
    
```



```

CODE    SEGMENT
        ASSUME CS:CODE, SS:SSTACK
START:  PUSH DS
        XOR AX, AX
        MOV DS, AX
        MOV SI, 3000H           ;建立数据起始地址
        MOV CX, 16             ;循环次数
AA1:    MOV [SI], AL
        INC SI                  ;地址自加 1
        INC AL                  ;数据自加 1
        LOOP AA1
        MOV AX, 4C00H
        INT 21H                 ;程序终止
CODE    ENDS
        END START

```

(5) 单击 按钮，编译文件，若程序编译无误，则可以继续单击 按钮进行链接，链接无误后方可以加载程序。编译、链接后输出区会显示如图 1-41 所示的输出信息。



图 1-41 输出区信息显示界面

(6) 连接 PC 与实验系统的通信电缆，打开实验系统电源。

(7) 编译、链接都正确并且上下位机通信成功后，就可以下载程序，联机调试了。可以通过端口列表中的“端口测试”来检查通信是否正常。单击 按钮下载程序。 按钮为编译、链接、下载组合按钮，通过该按钮可以将编译、链接、下载一次完成。下载成功后，在输出区的结果窗中会显示“加载成功！”，表示程序已正确下载。程序中起始运行语句下会有一条绿色的背景，如图 1-42 所示。

(8) 将输出区切换到调试窗口，使用 D0000:3000 命令查看内存 3000H 起始地址的数据，如图 1-43 所示。存储器在初始状态时，默认数据为 CC。

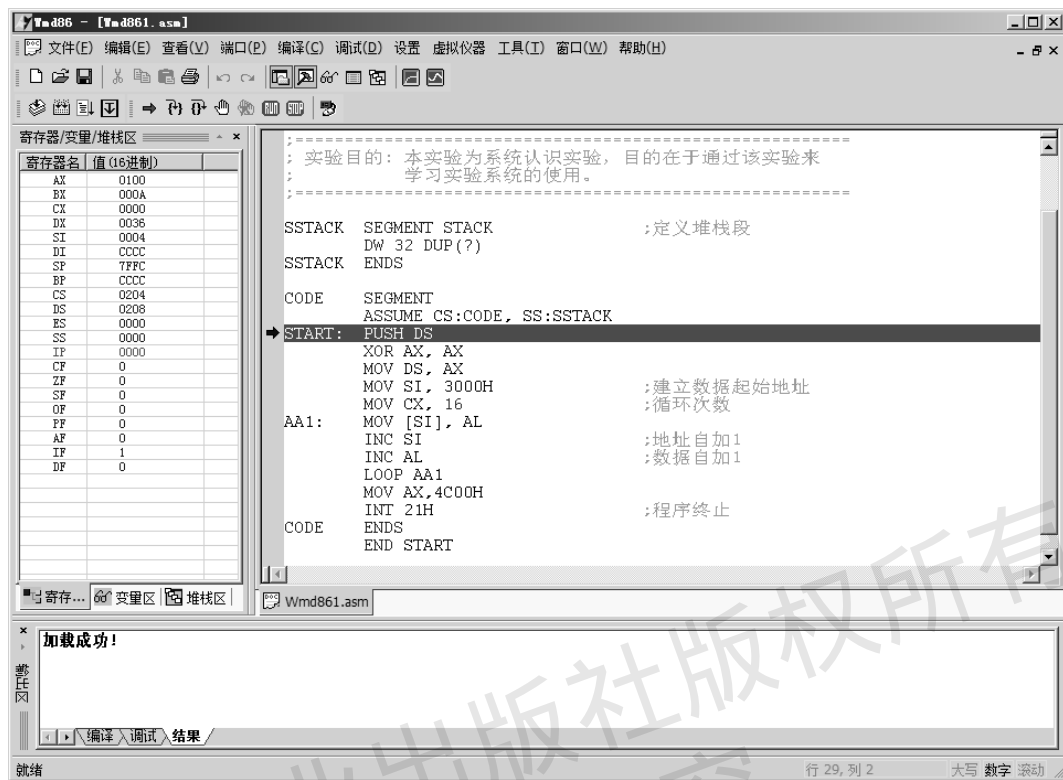

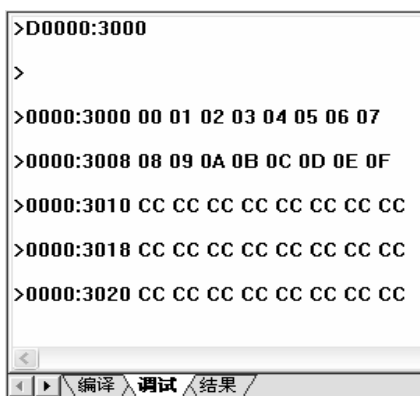


图 1-42 加载成功显示界面



图 1-43 内存地址单元数据显示

(9) 单击  按钮运行程序, 等待程序运行停止后, 通过 D0000:3000 命令来观察程序运行结果, 如图 1-44 所示。



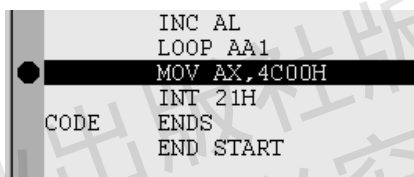
```

>D0000:3000
>
>0000:3000 00 01 02 03 04 05 06 07
>0000:3008 08 09 0A 0B 0C 0D 0E 0F
>0000:3010 CC CC CC CC CC CC CC CC
>0000:3018 CC CC CC CC CC CC CC CC
>0000:3020 CC CC CC CC CC CC CC CC

```

图 1-44 运行程序后数据变化显示

(10) 也可以通过设置断点，断点显示如图 1-45 所示，然后运行程序，当遇到断点时程序会停下来，然后观察数据。可以使用 E0000:3000 来改变该地址单元的数据，如图 1-46 所示，输入 11 后，按空格键，可以接着输入第二个数，如 22，按回车键结束输入。

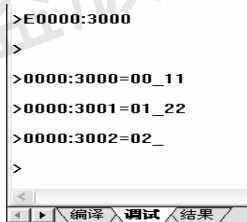


```

INC AL
LOOP AA1
MOV AX,4C00H
INT 21H
CODE ENDS
END START

```

图 1-45 断点设置显示



```

>E0000:3000
>
>0000:3000=00_11
>0000:3001=01_22
>0000:3002=02_
>

```

图 1-46 修改内存单元数据显示界面