

汇编语言与接口通讯实验指导书

实验一 系统认识实验

实验二 8255 并行接口实验

实验三 中断控制实验

实验四 8254 定时/计数器应用实验

实验五 键盘扫描及数码显示设计实验

实验六 点阵 LED 显示设计实验

实验七 电子发声设计实验

实验八 电子时钟实验

实验九 A/D 转换实验

实验十 D/A 转换实验

附录 1 Wmd86 联机软件使用说明

附录 2 系统编程信息

实验一 系统认识实验

实验目的

掌握TD-PITE 80X86 微机原理及接口技术教学实验系统的操作，熟悉Wmd86联机集成开发调试软件的操作环境。

实验设备

PC机一台，TD-PITE 实验装置一套。

实验内容

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

实验步骤

1. 运行Wmd86 软件，进入Wmd86 集成开发环境。
2. 根据程序设计使用语言的不同，通过在“设置”下拉列表来选择需要使用的语言，如图1-1-1所示。语言选择后，下次再启动软件，语言环境保持这次的修改不变。在这里，我们选择汇编语言。

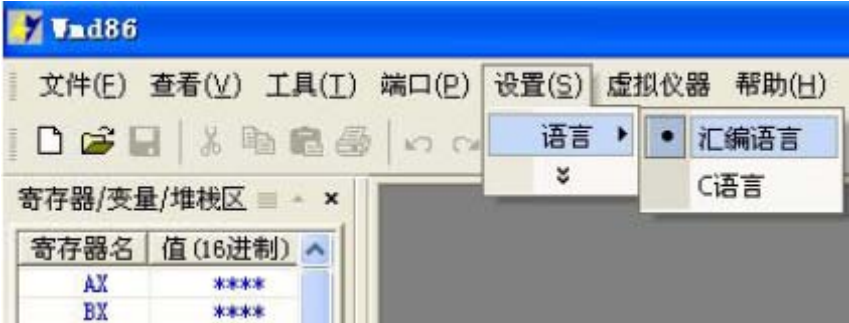


图1-1-1 语言环境选择界面

3. 语言选择后，点击新建或按Ctrl+N 组合键来新建一个文档，如图1-1-2所示。默认文件名为Wmd861。

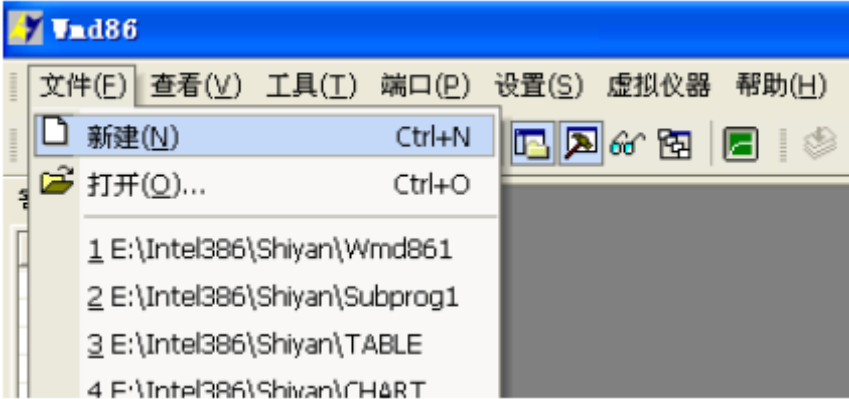


图1-1-2 新建文件界面

4. 编写实验程序，如图1-1-3所示，并保存，此时系统会提示输入新的文件名，输完后点击保存。



```

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
AA2:    JMP AA2
CODE    ENDS
END START

```

图1-1-3 程序编辑界面

5. 点击 ，编译文件，若程序编译无误，则输出如图1-1-4所示的输出信息，然后再点击  进行链接，链接无误输出如图1-1-5所示的输出信息。

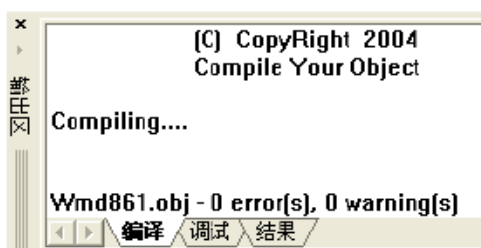




图1-1-4 编译输出信息界面



图1-1-5 链接输出信息界面

6. 连接PC与实验系统的通讯电缆，打开实验系统电源。
7. 编译、链接都正确并且上下位机通讯成功后，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”来检查通讯是否正常。点击  下载程序。  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。起始运行语句下会有一条绿色的背景。如图1-1-6所示。

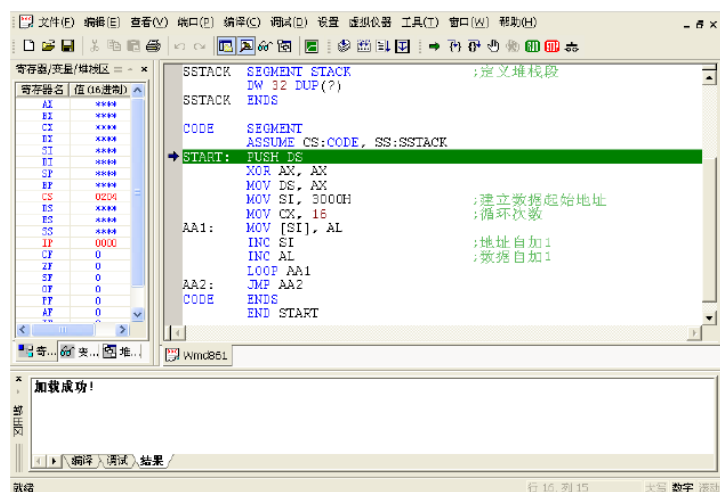




图1-1-6 加载成功输出显示界面

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1-1-7所示。存储器在初始状态时，默认数据为CC。



图1-1-7 内存地址单元数据显示

9. 点击按钮运行程序，然后再点击来停止程序运行，接下来观察程序运行结果，仍使用命令D0000:3000 来观察数据变化。如图1-1-8所示。

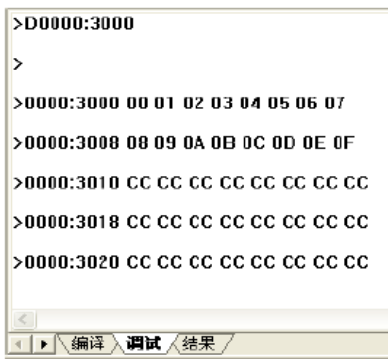



图1-1-8 运行程序后数据变化显示

10. 也可以通过在语句AA2: JMP AA2 处设置断点，断点显示如图1-1-9 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1-1-10 所示，输入11 后，按“空格”键，可以接着输入第二个数，如22，结束输入按“回车”键。

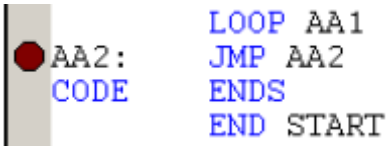


图1-1-9 断点设置显示

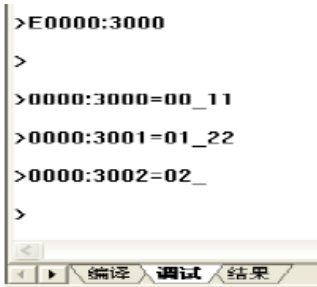


图1-1-10 修改内存单元数据显示界面

实验例程文件名为Wmd861.asm。

1.1.5 操作练习

编写程序，将内存 3500H 单元开始的 8 个数据复制到 3600H 单元开始的数据区中。通过调试验证程序功能，使用 E 命令修改 3500H 单元开始的数据，运行程序后使用 D 命令查看 3600H 单元开始的数据。

实验二 8255 并行接口实验

一、实验目的

- 1. 学习并掌握8255 的工作方式及其应用；
- 2. 掌握8255 典型应用电路的接法。

二、实验设备

PC机一台，TD-PITE 实验装置一套。

三、实验内容

- 1. 基本输入输出实验。编写程序，使8255 的A口为输入，B口为输出，完成拨动开关到数据灯的数据传输。要求只要开关拨动，数据灯的显示就发生相应改变。
- 2. 流水灯显示实验。编写程序，使8255 的A口和B口均为输出，数据灯D7~D0由左向右，每次仅亮一个灯，循环显示，D15~D8与D7~D0 正相反，由右向左，每次仅点亮一个灯，循环显示。

四、实验原理

并行接口是以数据的字节为单位与I/O 设备或被控制对象之间传递信息。CPU和接口之间的数据传送总是并行的，即可以同时传递8 位、16 位或32 位等。8255可编程外围接口芯片是Intel公司生产的通用并行I/O 接口芯片，它具有A、B、C三个并行接口，用+5V单电源供电，能在以下三种方式下工作：方式0--基本输入/输出方式、方式1--选通输入/输出方式、方式2--双向选通工作方式。8255 的内部结构及引脚如图2-5-1 所示，8255工作方式控制字和C口按位置位/复位控制字格式如图2-5-2所示。

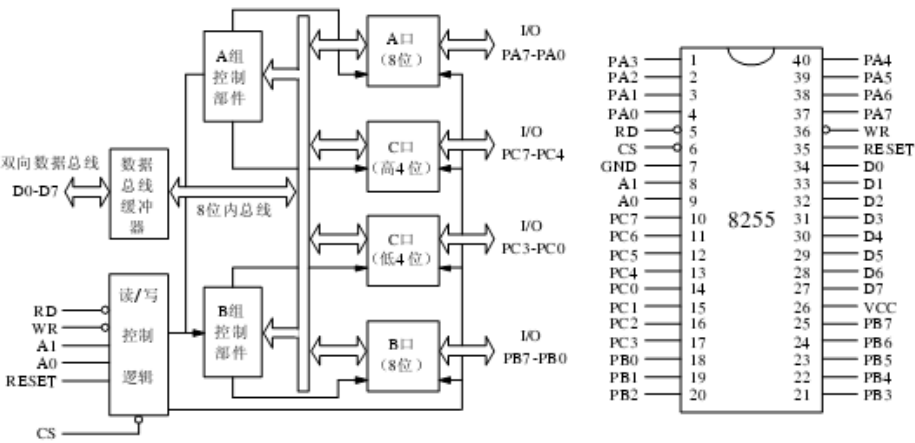


图2-5-1 8255内部结构及外部引脚图

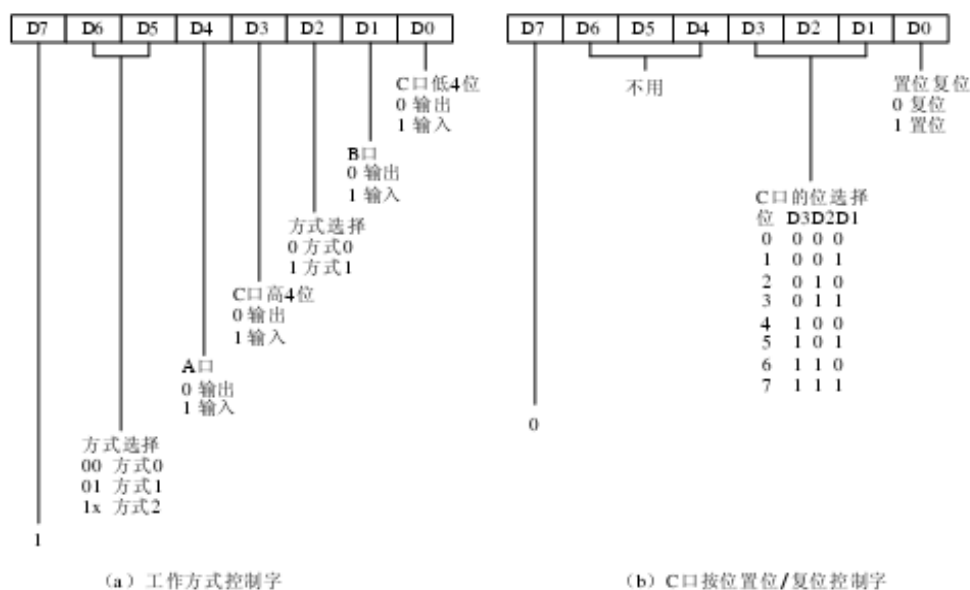


图2-5-2 8255控制字格式

五、 实验步骤

1. 基本输入输出实验

本实验使8255 端口A工作在方式0 并作为输入口，端口B工作在方式0 并作为输出口。用一组开关信号接入端口A，端口B 输出线接至一组数据灯上，然后通过8255 芯片编程来实现输入输出功能。具体实验步骤如下述：

- (1) 实验接线图如图2-5-3所示，按图连接实验线路图；
- (2) 编写实验程序，经编译、连接无误后装入系统；
- (3) 运行程序，改变拨动开关，同时观察LED 显示，验证程序功能。

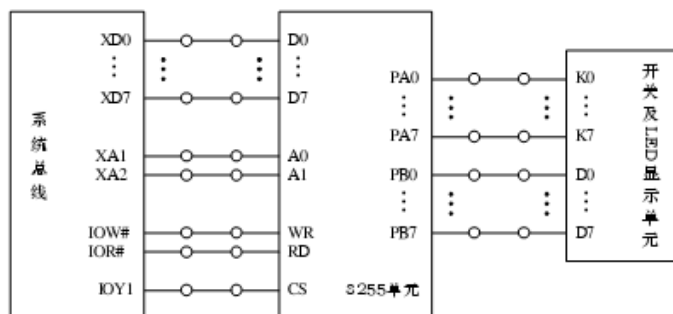


图2-5-3 8255基本输入输出实验接线图

2. 流水灯显示实验

使8255的A口和B口均为输出，数据灯D7~D0 由左向右，每次仅亮一个灯，循环显示，D15~D8 与D7~D0正相反，由右向左，每次仅点亮一个灯，循环显示。实验接线图如图2-5-5所示。实验步骤如下所述：

- (1) 按图2-5-5连接实验线路图；
- (2) 编写实验程序，经编译、链接无误后装入系统；
- (3) 运行程序，观察LED 灯的显示，验证程序功能；
- (4) 自己改变流水灯的方式，编写程序。

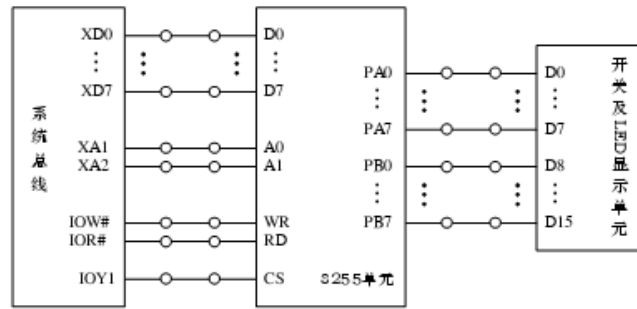


图2-5-5 8255流水灯实验接线图

注：文件另存时文件夹名中不能有空格

实验三 中断控制实验

一、实验目的

- 1. 掌握8259 中断控制器的工作原理；
- 2. 学习8259 的应用编程方法；
- 3. 掌握8259 级联方式的使用方法。

二、实验设备

PC机一台，TD-PITE 实验装置一套。

三、实验内容及步骤

1. 中断控制器8259 简介

在Intel 386EX 芯片中集成有中断控制单元（ICU），该单元包含有两个级联中断控制器，一个为主控制器，一个为从控制器。该中断控制单元就功能而言与工业上标准的82C59A 是一致的，操作方法也相同。从片的INT连接到主片的IR2 信号上构成两片8259 的级联。

在TD-PITE 实验系统中，将主控制器的IR6、IR7 以及从控制器的IR1 开放出来供实验使用，主片8259 的IR4 供系统串口使用。8259 的内部连接及外部管脚引出如图2-2-1所示。

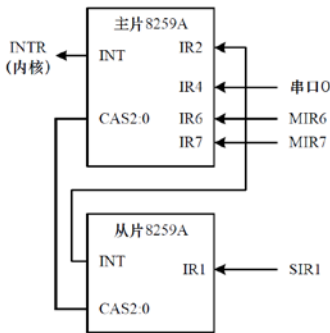


图2-2-1 8259内部连接及外部管脚引出图

表2-2-1 列出了中断控制单元的寄存器相关信息。

表2-2-1 ICU寄存器列表

寄存器	口地址	功能描述
ICW1 (主) ICW1 (从) (只写)	0020H 00A0H	初始化命令字 1: 决定中断请求信号为电平触发还是边沿触发。
ICW2 (主) ICW2 (从) (只写)	0021H 00A1H	初始化命令字 2: 包含了 8259 的基址中断向量号，基址中断向量是 IR0 的向量号，IR1的向量号，依此类推。
ICW3 (主) (只写)	0021H	初始化命令字 3: 用于识别从 8259设备连接到主控制器的 IR信号，内部的从 8259连的 IR2信号上。
ICW3 (从) (只写)	00A1H	初始化命令字 3: 表明内部从控制器级联到主片的 IR2信号上。
ICW4 (主) ICW4 (从) (只写)	0021H 00A1H	初始化命令字 4: 选择特殊全嵌套或全嵌套模式，使能中断自动结束方式。
OCW1 (主) OCW1 (从) (读/写)	0021H 00A1H	操作命令字 1: 中断屏蔽操作寄存器，可屏蔽相应的中断信号。
OCW2 (主) OCW2 (从) (只写)	0020H 00A0H	操作命令字 2: 改变中断优先级和发送中断结束命令。
OCW3 (主) OCW3 (从)	0020H 00A0H	操作命令字 3: 使能特殊屏蔽方式，设路中断查询方式，允许读出中断请求寄存器

(只写)		各寄存器。
IRR (主)	0020H	中断请求： 指出挂起的中断请求。
IRR (从)	00A0H	
(只读)		
ISR (主)	0020H	当前中断服务： 指出当前正在被服务的中断请求。
ISR (从)	00A0H	
(只读)		
POLL (主)	0020H	查询状态字： 表明连接到 8259 上的设备是否需要服务，如果有中断请求，该字 级最高的中断请求。
POLL (从)	0021H	
(只读)	00A0H	
	00A1H	

初始化命令字1 寄存器（ICW1）说明见图2-2-2所示。

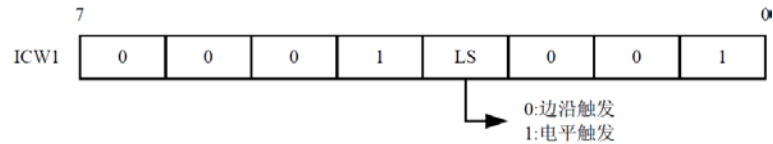


图2-2-2 初始化命令字1寄存器

初始化命令字2 寄存器（ICW2）说明见图2-2-3所示。

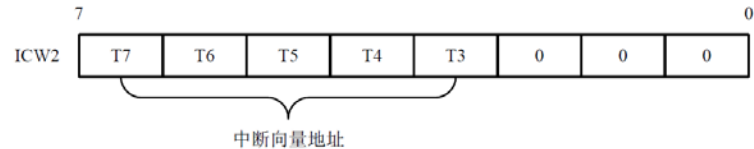
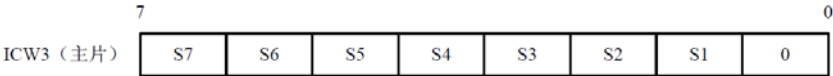


图2-2-3 初始化命令字2寄存器

初始化命令字3 寄存器（ICW3）说明，主片见图2-2-4，从片见图2-2-5。



- S7—S3: 0表示无8259被连接到主片上，1表示有8259被连接到主片上
- S2: 0表示内部从片没被使用，1表示内部从片被级联到主片的IR2上
- S1: 0表示无8259通过IR1连接到主片上，1表示有8259通过IR1连接到主片上

图2-2-4 主片初始化命令字3寄存器

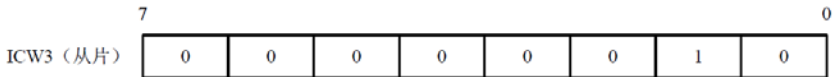
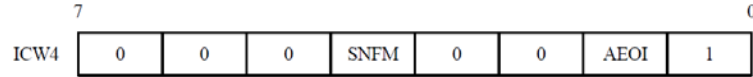


图2-2-5 从片初始化命令字3寄存器

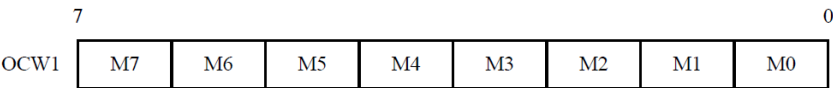
初始化命令字4 寄存器（ICW4）说明见图2-2-6。



- SNFM: 0=选择全嵌套模式，1=选择特殊全嵌套模式，仅主8259能够工作在特殊全嵌套模式
- AEOI: 0=禁止中断自动结束，1=使能中断自动结束，仅主8259能够工作中断自动结束方式

图2-2-6 初始化命令字4寄存器

操作命令字1 寄存器（OCW1）说明见图2-2-7。



- M7--M0: 0=对应 IR 信号上的中断请求得到允许，1=对应 IR 信号上的中断请求

被屏蔽

注意：设置屏蔽位不影响各自中断挂起位

图2-2-7 操作命令字1寄存器

操作命令字2 寄存器 (OCW2) 说明如图2-2-8所示。

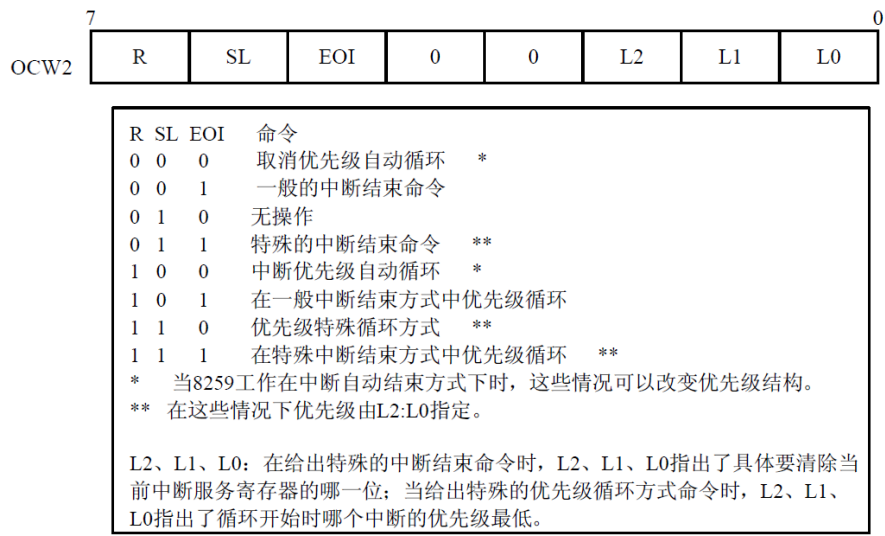


图2-2-8 操作命令字2寄存器

操作命令字3 寄存器 (OCW3) 说明如图2-2-9所示。

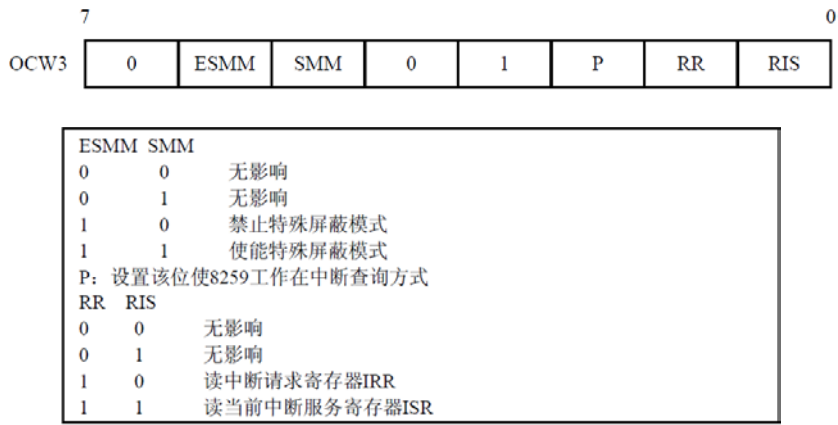
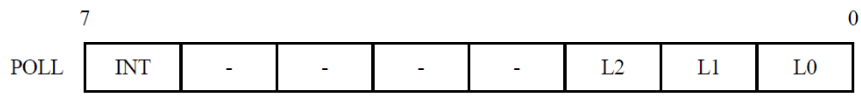


图2-2-9 操作命令字3寄存器

查询状态字 (POLL) 说明如图2-2-10 所示。



INT: 0=无请求

1=连接在8259上的设备请求服务

L2、L1、L0: 当INT为1时，这些位指出了需要服务的最高优先级的IR；当INT为0时这些位不确定。

图2-2-10 程序状态字寄存器

在对8259 进行编程时，首先必须进行初始化。一般先使用CLI 指令将所有的可屏蔽中断禁止，然后写入初始化命令字。8259有一个状态机控制对寄存器的访问，不正确的初始化顺序会造成异常初始化。在初始化主片8259 时，写入初始化命令字的顺序是：ICW1、ICW2、ICW3、然后是ICW4，初始化从片8259 的顺序与初始化主片8259 的顺序是相同的。

系统启动时，主片8259 已被初始化，且4 号中断源 (IR4) 提供给与PC 联机的串口通信使用，其它中断源被屏蔽。中断矢量地址与中断号之间的关系如下表所示：

主片中断序号	0	1	2	3	4	5	6	7
功能调用	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
矢量地址	20H~23H	24H~27H	28H~2BH	2CH~2FH	30H~33H	34H~37H	38H~3BH	3CH~3FH

说明	未开放	未开放	未开放	未开放	串口	未开放	可用	可用
从片中断序号	0	1	2	3	4	5	6	7
功能调用	30H	31H	32H	33H	34H	35H	36H	37H
矢量地址	C0H~C3H	C4H~C7H	C8H~CBH	CCH~CFH	D0H~D3H	D4H~D7H	D8H~DBH	DCH~DFH
说明	未开放	可用	未开放	未开放	未开放	未开放	未开放	未开放

2. 8259 单中断实验

实验接线图如图2-2-11所示，单次脉冲输出与主片8259 的IR7 相连，每按动一次单次脉冲，产生一次外部中断，在显示屏上输出一个字符 “7” 。

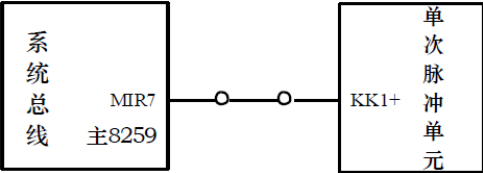


图2-2-11 8259单中断实验接线图

实验步骤

- (1) 按图2-2-11 连接实验线路；
- (2) 编写实验程序，经编译、链接无误后装入系统；
- (3) 运行程序，重复按单次脉冲开关KK1+，显示屏会显示字符 “7” 以说明响应了中断。

3. 8259 级联实验

实验接线图如图2-2-12 所示，KK1+连接到主片8259 的IR7 上，KK2+连接到从片8259 的IR1 上，当按一次KK1+时，显示屏上显示字符 “M7”，按一次KK2+时，显示字符 “S1”。编写程序。

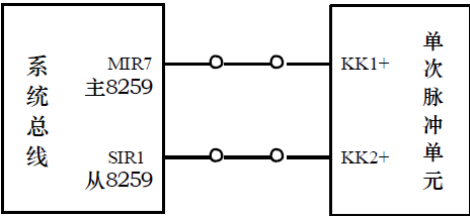


图2-2-12 8259级联实验

实验步骤

- (1) 按图2-2-12 连接实验线路；
- (2) 输入程序，编译、链接无误后装入系统；
- (3) 运行程序，按动KK1+或KK2+，观察实验结果，验证实验程序的正确性；
- (4) 若同时按下KK1+和KK2+，观察实验结果，解释实验现象。

实验四 8254 定时/计数器应用实验

一、实验目的

1. 掌握8254 的工作方式及应用编程；
2. 掌握8254 典型应用电路的接法。

二、实验设备

PC机一台，TD-PITE 实验装置一套，示波器一台。

三、实验内容

1. 计数应用实验。编写程序，应用8254 的计数功能，使用单次脉冲模拟计数，使每当按动‘KK1+’ 5 次后，产生一次计数中断，并在屏幕上显示一个字符‘M’。
2. 定时应用实验。编写程序，应用8254 的定时功能，产生一个1ms 的方波。

四、实验原理

8254 是Intel 公司生产的可编程间隔定时器。是8253 的改进型，比8253 具有更优良的性能。8254 具有以下基本功能：

- (1) 有3 个独立的16 位计数器；（CLK:输入脉冲信号，OUT: 输出信号，GATE: 输入门控信号）
- (2) 每个计数器可按二进制或十进制（BCD）计数；
- (3) 每个计数器可编程工作于6 种不同工作方式；
- (4) 8254 每个计数器允许的最高计数频率为10MHz（8253 为2MHz）；（计数频率更高）
- (5) 8254 有读回命令（8253 没有），除了可以读出当前计数单元的内容外，还可以读出状态寄存器的内容；

- (6) 计数脉冲可以是有规律的时钟信号，也可以是随机信号。计数初值公式为：

$$n = f_{CLKi} \div f_{OUTi}$$
 其中 f_{CLKi} 是输入时钟脉冲的频率， f_{OUTi} 是输出波形的频率。

图2-4-1 是8254 的内部结构框图和引脚图，它是由与CPU 的接口、内部控制电路和三个计数器组成。8254 的工作方式如下述：

- (1) 方式0: 计数到0 结束输出正跃变信号方式。
- (2) 方式1: 硬件可重触发单稳方式。
- (3) 方式2: 频率发生器方式。
- (4) 方式3: 方波发生器。
- (5) 方式4: 软件触发选通方式。
- (6) 方式5: 硬件触发选通方式。

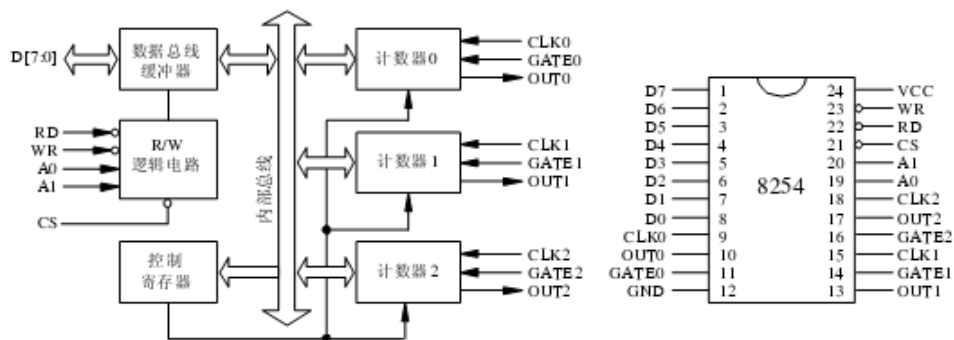


图2-4-1 8254的内部接口和引脚

8254 的控制字有两个：一个用来设置计数器的工作方式，称为方式控制字；另一个用来设置读回命令，称为读回控制字。这两个控制字共用一个地址，由标识位来区分。控制字格式如表2-4-1所示。

表2-4-1 8254的方式控制字格式

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		读/写格式选择		工作方式选择		计数码制选择	
00—计数器 0		00—锁存计数值		000—方式 0		0—二进制数	
01—计数器 1		01—读/写低 8 位		001—方式 1		1—十进制数	
10—计数器 2		10—读/写高 8 位		010—方式 2			
11—读出控制字标志		11—先读/写低 8 位再读/写高 8 位		011—方式 3			
				100—方式 4			
				101—方式 5			

表2-4-2 8254读出控制字格式

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0—锁存计数值	0—锁存状态信息	计数器选择（同方式控制字）		0	

表2-4-3 8254状态字格式

D7	D6	D5	D4	D3	D2	D1	D0
OUT 引脚现行状态 1—高电平 0—低电平		计数初值是否装入 1—无效计数 0—计数有效		计数器方式（同方式控制字）			

五、 实验步骤

1. 计数应用实验

编写程序，将8254 的计数器0 设置为方式3，计数值为十进制数4，用单次脉冲KK1+作为CLK0 时钟，OUT0 连接MIR7，每当KK1+按动5 次后产生中断请求，在屏幕上显示字符“M”。

实验步骤：

- (1) 实验接线如图2-4-2所示；
- (2) 编写实验程序，经编译、链接无误后装入系统；
- (3) 运行程序，按动KK1+产生单次脉冲，观察实验现象；
- (4) 改变计数值，验证8254 的计数功能。

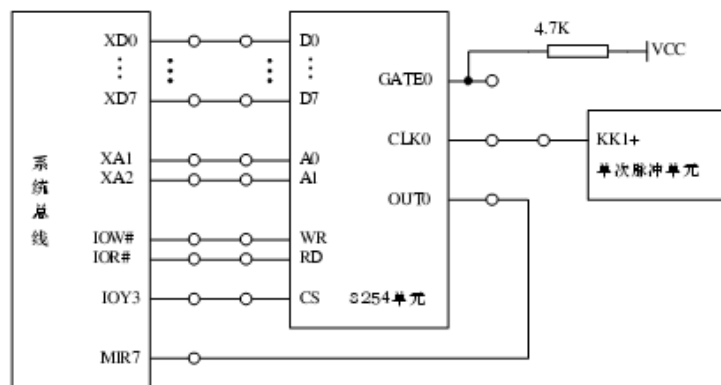


图2-4-2 8254计数应用实验接线图

注：8259A的IR0,系统计时每隔55ms产生1次中断

A8254 EQU 06C0H

B8254 EQU 06C2H

C8254 EQU 06C4H

CON8254 EQU 06C6H

初始化主片8259 端口地址为20H, 21H

MOV AL, 11H

OUT 20H, AL

;ICW1(芯片控制初始化命令字)见

P279接口书

MOV AL, 08H

OUT 21H, AL

;ICW2(中断类型寄存器中的内容)

MOV AL, 04H

OUT 21H, AL

;ICW3(标志主/从片的初始化命令

字)

MOV AL, 01H

OUT 21H, AL

;ICW4(方式控制字)

MOV AL, 6FH

;OCW1(操作命令字, 设定优先级特

殊循环方式)

OUT 21H, AL

2. 定时应用实验

编写程序, 将8254 的计数器0 和计数器1都设置为方式3, 用信号源1MHz 作为CLK0 时钟, OUT0 为波形输出1ms 方波, 再通过CLK1输入, OUT1输出1s方波。

实验步骤:

- (1) 接线图如图2-4-3所示;
- (2) 根据实验内容, 编写实验程序, 经编译、链接无误后装入系统;
- (3) 运行实验程序, 用示波器测试OUT0 输出, 验证程序功能。

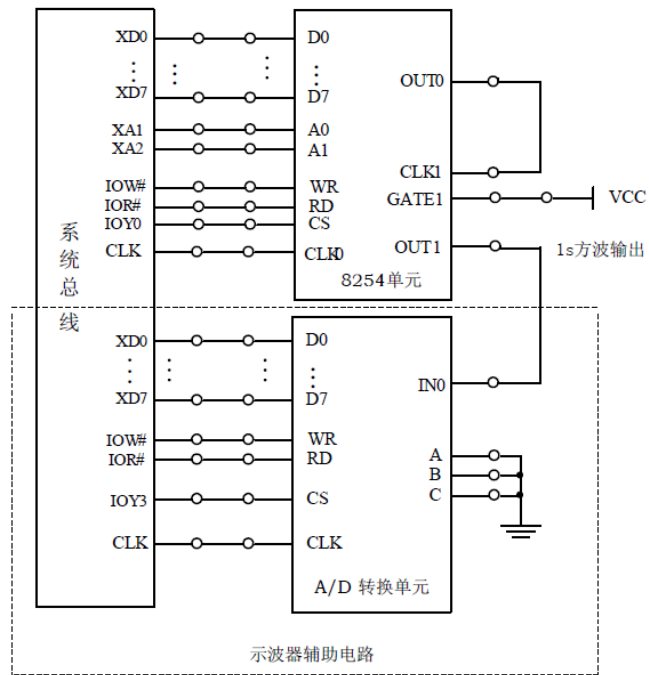


图2-4-4 8254定时应用实验接线图

实验五 键盘扫描及显示设计实验

一、实验目的

了解键盘扫描及数码显示的基本原理，熟悉8255 的编程。

二、实验设备

PC机一台，TD-PITE 实验装置一套。

三、实验内容

1、明白数码管的工作原理，手动控制实现相应数字，对共阴极LED，按a~dp从数据低位到高位连接时，写出0-F字符编码表。

2、利用8255单元控制一位数码管亮静态数字3或任何0-F中任何一位数字。(A口控制位选择, B口控制段显示)

3、利用8255单元控制六位数码管亮静态数字201465（A口控制位选择，B口控制段显示）

4、利用8255控制数码管显示动态输出00-FF。

5、将8255 单元与键盘及数码管显示单元连接，编写实验程序，扫描键盘输入，并将扫描结果送数码管显示。键盘采用4×4 键盘，每个数码管显示值可为0~F 共16 个数。实验具体内容如下：将键盘进行编号，记作0~F，当按下其中一个按键时，将该按键对应的编号在一个数码管上显示出来，当再按下一个按键时，便将这个按键的编号在下一个数码管上显示出来，数码管上可以显示最近4次按下的按键编号。键盘及数码管显示单元电路图如图2-9-1所示，实验参考接线图如图2-9-2所示。

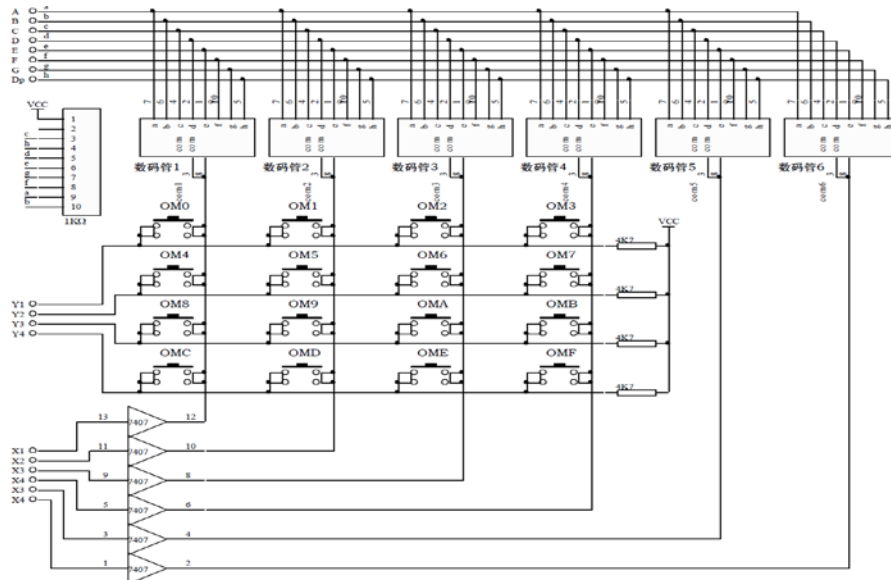


图2-9-1 键盘及数码管显示单元电路图

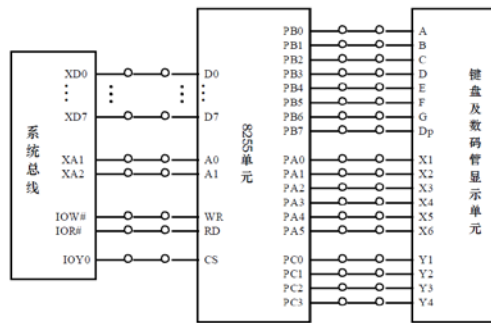


图2-9-2 8255键盘扫描及数码管显示实验线路图

四、实验步骤

1. 按图2-9-2连接线路图；
2. 编写实验程序，检查无误后编译、连接并装入系统；
3. 运行程序，按下按键，观察数码管的显示，验证程序功能。

实验六 点阵LED显示设计实验

一、实验目的

1. 了解LED 点阵的基本结构；
2. 学习LED 点阵扫描显示程序的设计方法。

二、实验设备

PC机一台，TD-PITE 实验装置一套。

三、实验内容

一、LED的点亮显示：

选中行（送行数据），送列1反数据，送列2反数据，延时。换行，重复以上步骤。

二、行列数据：PA口送数据，PC口送选通，锁存数据，撤销选通。

8行显示结束，行数据要清零。

三、字符显示：

1、汉字字模：16*16，从上到下，从左到右，左高位，每字32字节。

2、单个字显示：循环一，注意修改字模数据指针。

3、多字循环显示：同2，每个字重复显示多次，再下一个字，否则变字太快。每字指针改变32。

4、多字滚动显示：同2，显示1屏，重复多次，否则滚屏太快，然后滚动1行，指针先返回32，再加2。

5、多重的循环和调用，注意CX的保护，及参数传递和现场保护

6、延时：

DELLY: PUSH CX

MOV CX, 0100H (~0300H)

DL1: NOP

LOOP DL1

POP CX

RET

7、整理画出流程图，写出程序，再去实验调试运行。

编写程序，控制点阵向上卷动显示“自己名字”。

实验系统中的 16×16 LED 点阵由四块 8×8 LED 点阵组成，如图2-11-1所示，8×8 点阵内部结构图如图2-11-2所示。由图2-11-2可知，当行为“0”，列为“1”，则对应行、列上的LED 点亮。图2-11-3为点阵外部引脚图。汉字显示如图2-11-4所示。

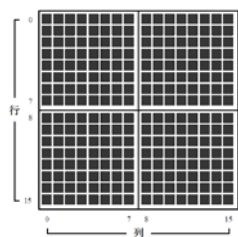


图2-11-1 16×16 点阵示意图

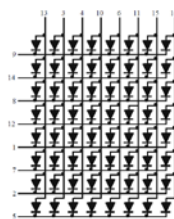


图2-11-2 点阵内部结构图

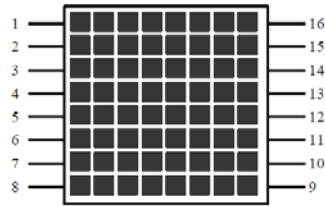


图2-11-3 点阵外部引脚图

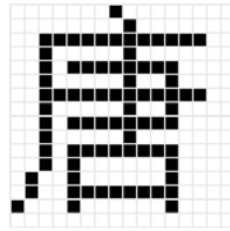


图2-11-4 显示示例

点阵实验单元电路图如图2-11-5所示。由于 2803 输出反向，所以行为 1，列为 0 时对于点的 LED 点亮。

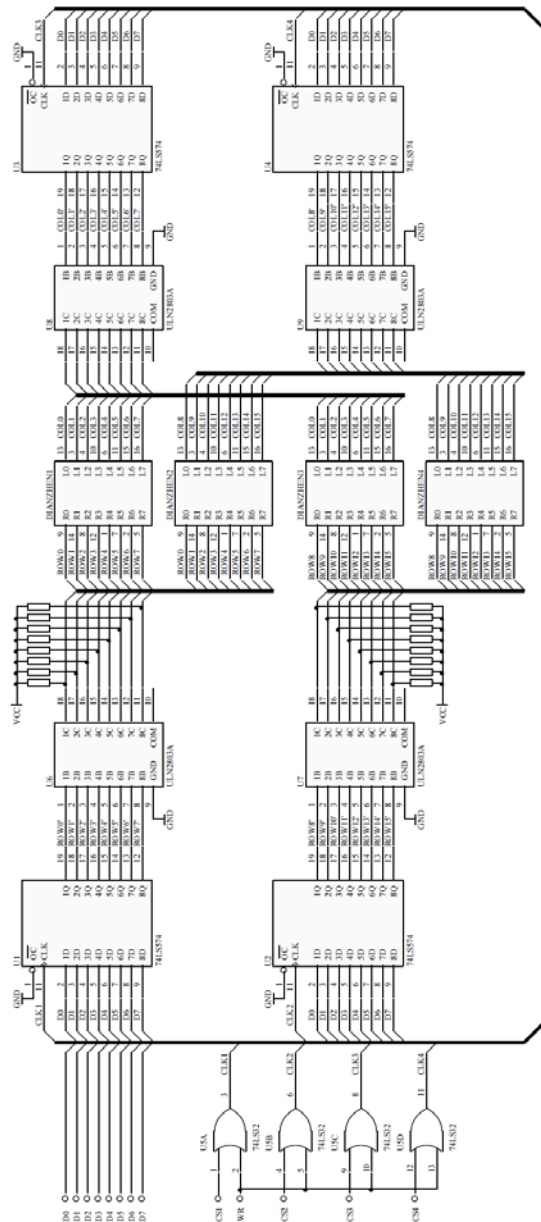


图2-11-5 点阵实验单元电路图

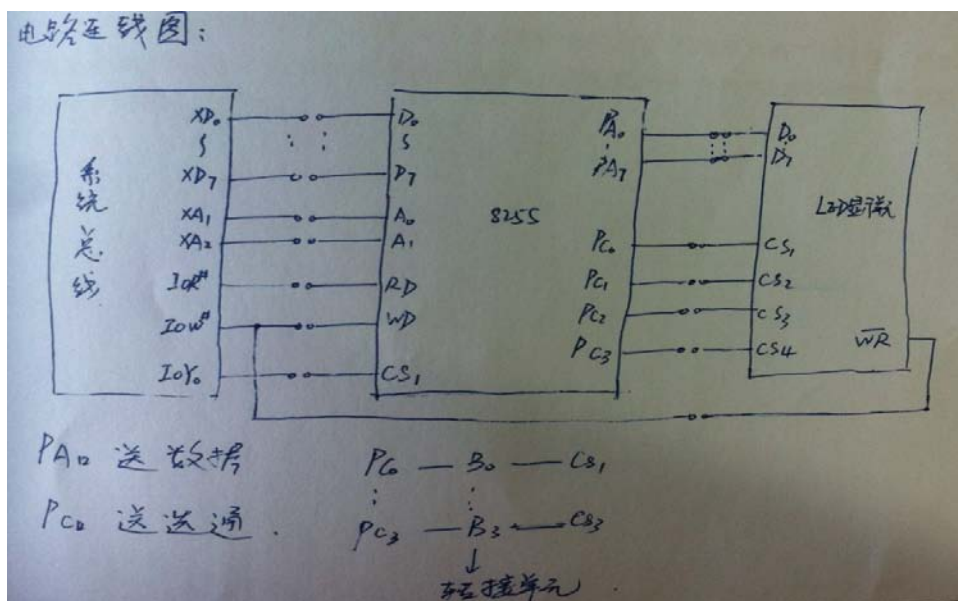


图2-11-6 点阵显示实验接线图

四、实验步骤

分析：点阵1、2共享行数据，3、4共享行数据，点阵1、3共享列数据，2、4共享列数据。行列数据都由一组数据端输入，通过CS1-CS4进行选通，所以需分时送且选通信号CS1-CS4与PC0-PC3相连，所以只需控制PC0-PC3即可

CS3 CS4

CS1->	1	2
CS2->	3	4

1. 按图2-11-6连接实验电路图；
2. 编写实验程序，检查无误后，编译、链接并装入系统；
3. 运行实验程序，观察点阵的显示，验证程序功能；
4. 自己可以设计实验，使点阵显示不同的符号。

使用点阵显示符号时，必须首先得到显示符号的编码，这可以根据需要通过不同的工具获得。在本例子中，我们首先得到了显示汉字的字库文件，然后将该字库文件修改后包含到主文件中。参考下节所述。

附：字符提取方法



1. 将 HZDotReader文件夹拷贝到硬盘上，然后双击文件 HZDotReader.exe 运行程序；
2. 在“设置”下拉菜单中选择“取模字体”选项，设置需要显示汉字的字体；

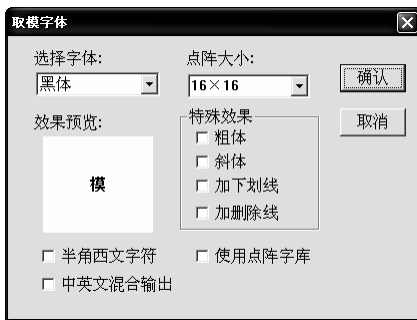


图2-11-7 取模字体对话框

3. 在“设置”下拉菜单中选择“取模方式”选项，在本系统中选择如图所示，即以横向 8 个连续点构成一个字节，最左边的点为字节的最低位，即 BIT0，最右边的点为 BIT7。16×16 汉字按每行 2 字节，共 16 行取字模，每个汉字共 32 字节，点阵四个角取字顺序为左上角→ 右上角→左下角→右下角；（与数据连线相关）

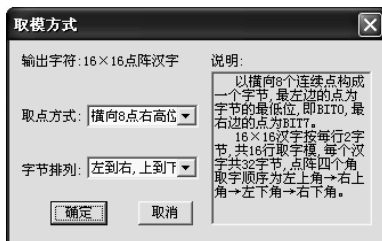


图 2-11-8 取模方式对话框

4.在“设置”下拉菜单中选择“输出设置”选项，以设置输出格式，可以为汇编格式或C语言格式，根据实验程序语言而定，如图2-11-9所示；

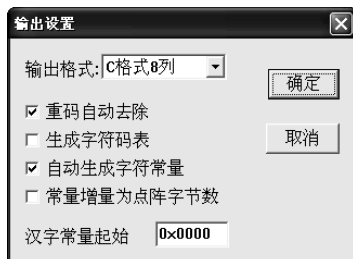


图 2-11-9 输出设置对话框

5. 点击 **字** 按钮，弹出字符输入对话框，输入“西安唐都科教仪器公司!”，如图2-11-10所示，然后点击输入按钮；

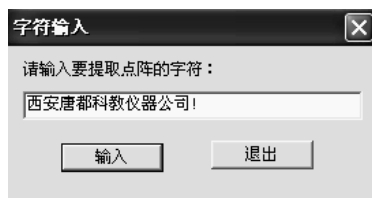


图2-11-10 字符输入对话框

6. 字符输入后，可得到输入字符的点阵编码以及对对应汉字的显示，如图2-11-11所示。此时可以对点阵进行编辑，方法是右键点击某一汉字，此时该汉字的编码反蓝，然后点击 编辑 下拉菜单中的‘编辑点阵’选项来编辑该汉字，如图 2-11-12所示。鼠标左键为点亮某点，鼠标 右键为取消某点。若无需编辑，则进行保存，软件会将此点阵文件保存为 dot 格式；

7. 使用 Word 软件打开保存的文件，然后将字库复制到自己的程序中使用。

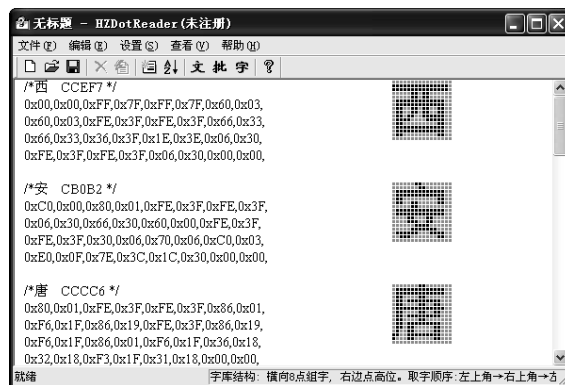


图2-11-11 字模生成窗口

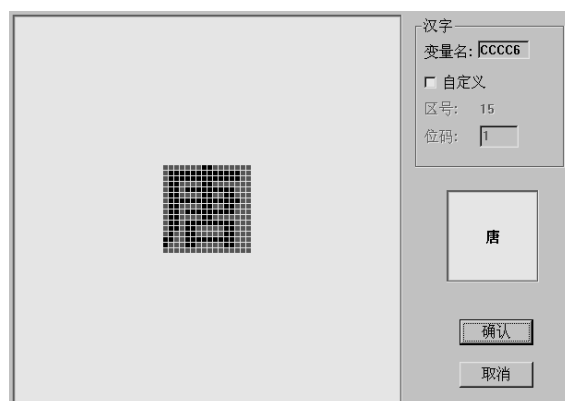


图2-11-12 点阵编辑窗

实验七 电子发声设计实验

一、实验目的

学习用8254 定时/计数器使蜂鸣器发声的编程方法。

二、实验设备

PC微机一台、TD-PITE 实验箱一台。

三、实验内容

根据实验提供的音乐频率表和时间表，编写程序控制8254，使其输出连接到扬声器上能发出相应的乐曲。

四、实验说明及步骤

一个音符对应一个频率，将对应一个音符频率的方波通到扬声器上，就可以发出这个音符的声音。将一段乐曲的音符对应频率的方波依次送到扬声器，就可以演奏出这段乐曲。利用8254的方式3——“方波发生器”，将相应一种频率的计数初值写入计数器，就可产生对应频率的方波。计数初值的计算如下：

$$\text{计数初值} = \text{输入时钟} \div \text{输出频率}$$

例如输入时钟采用1MHz，要得到800Hz 的频率，计数初值即为 $1000000 \div 800$ 。音符与频率对照关系如表2-10-1所示。对于每一个音符的演奏时间，可以通过软件延时来处理。首先确定单位延时时间程序（根据CPU 的频率不同而有所变化）。然后确定每个音符演奏需要几个单位时间，将这个值送入DL中，调用DALLY 子程序即可。

表2-10-1 音符与频率对照表 （单位：Hz）

音符 音调	1 ·	2 ·	3 ·	4 ·	5 ·	6 ·	7 ·
A	221	248	278	294	330	371	416
B	248	278	312	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	185	196	221	248	278
E	165	185	208	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	248	262	294	330	371
音符 音调	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	371	393	441	495	556
E	330	371	416	441	495	556	624
F	350	393	441	467	525	589	661
G	393	441	495	525	589	661	742
音符 音调	1 ·	2 ·	3 ·	4 ·	5 ·	6 ·	7 ·
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1248	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	742	786	882	990	1112
E	661	742	833	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

下面提供了乐曲《友谊地久天长》实验参考程序。程序中频率表是将曲谱中的音符对应的频率值依次记录下来（B 调、四分之二拍），时间表是将各个音符发音的相对时间记录下来（由曲谱中节拍得出）。

频率表和时间表是一一对应的，频率表的最后一项为0，作为重复的标志。根据频率表中的频率算出对应的计数初值，然后依次写入8254的计数器。将时间表中相对时间值带入延时程序来得到音符演奏时间。实验参考程序流程如图2-10-1所示。

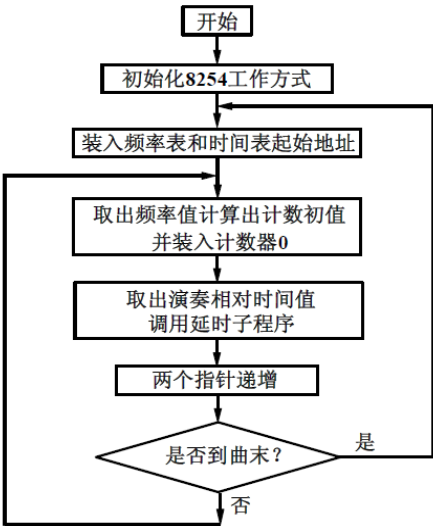


图2-10-1 实验参考流程图

电子发声电路图如图2-10-2所示。

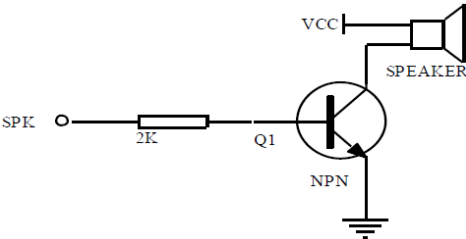


图2-10-2 电子发声单元电路图

实验步骤如下：

1. 参考图2-10-3所示连接实验线路；
2. 编写实验程序，经编译、连接无误后装入系统；
3. 运行程序，听扬声器发出的音乐是否正确。

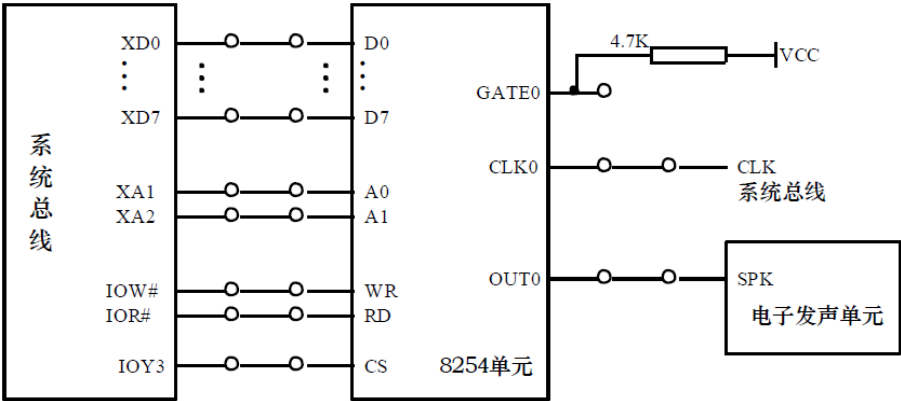


图2-10-3 8254电子发声实验接线图

实验八 电子时钟实验

一、实验目的

了解数码管显示的基本原理，熟悉8255、8254 的编程。

二、实验设备

PC机一台，TD-PITE 实验装置一套。

三、实验内容及设计要求

- 1、总体内容：设计一电子时钟，能在数码管上显示时间并计时。
- 2、接口设计：根据题目和所用的接口电路芯片设计出完整的接口电路,并在实验系统上完成电路的连接和调试通过.
- 3、程序设计:要求画出程序框图,设计出全部程序并给出程序设计说明和程序注释。
- 4、前期完成的实验有：8259A中断控制器实验，8254定时计数器实验，8255并行接口实验。

四、概要设计

2.1设计内容

本次课程设计的内容为电子时钟的设计与实现，利用定时器从 0 开始进行计时，将计时的结果显示在数码管上。每隔 1 秒，秒钟计时一次，到 60 秒，分钟加 1。根据需要，需选择的芯片有 8254，8259，8255，系统时钟频率 1.041667MHz；电子时钟的显示格式 HH:MM:SS 由左到右分别为时、分、秒，最大记时 59: 59: 59 超过这个时间时分秒位都清零从 00: 00: 00 重新开始。

2.2设计环境与器材

(1) PC微机一台

用于对程序的编译测试等，同时还需要对实验设备进行控制，提供整个程序的运行平台，并且收集和释放硬件信号，实现程序功能。

(2) 微机原理实验箱一台

此设备必须能提供8254、8255、8259和数码管等必要芯片。并且能通过接受PC机传来的信息，显示出相应的功能。以支持电子时钟的实现。

(8) 导线若干条

用于电路和芯片之间的连接。

2.3电路原理与主要芯片功能

2.3.1 电路原理

首先利用程序硬性规定分、秒的起始时间为零。然后通过8254计时器分频，并将以分得的频率接通8259中断控制器，进而通过CPU响应可屏蔽中断达到按秒计时的效果。

具体做法是将四个七段管显示器接到两片可并行接口8255芯片上，以此来控制七段管显示器的位操作以及段操作。并为之在数据段开辟一处空间，专门存放将要显示的数位代码。

8254 计时器将由 PC 机给定的频率加以分频，得到的新频率加到 8259 中断控制器的中断请求输入端。主程序为此可屏蔽中断设置专门的中断向量码。CPU 获取此中断向量码后转向待需解决的程序进行操作。而这个程序便是整个主程序的“心脏”。通过它将使电子钟的显示时间以秒为进位，达到秒、分的增加。

2.3.2 8255 主要功能

8255 可编程外围接口芯片是 Intel 公司生产的通用并行 I/O 接口芯片，它具有 A、B、C 三个并行接口，并行接口是以数据的字节为单位与 I/O 设备或被控制对象之间传递信息。CPU 和接口之间的数据传送总是并行的，即可以同时传递 8 位、16 位、32 位等。用 +5V 单电源供电，能在以下三种方式下工作：方式 0--基本输入/出方式、方式 1--选通输入/出方式、方式 2--双向选通工作方式。8255 的内部结构及引脚如图 2.1 所示：

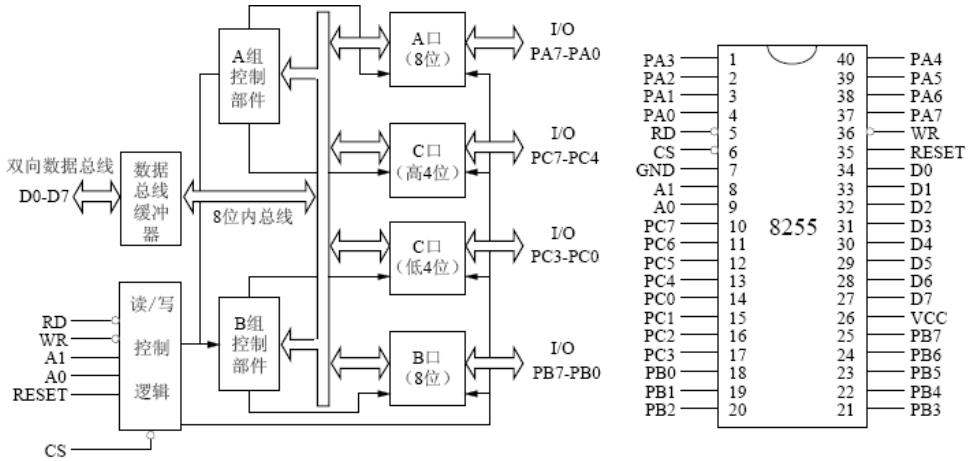


图 2.1 内部结构及引脚

8255 工作方式控制字和 C 口按位置位/复位控制字格式如图 2.2 所示：

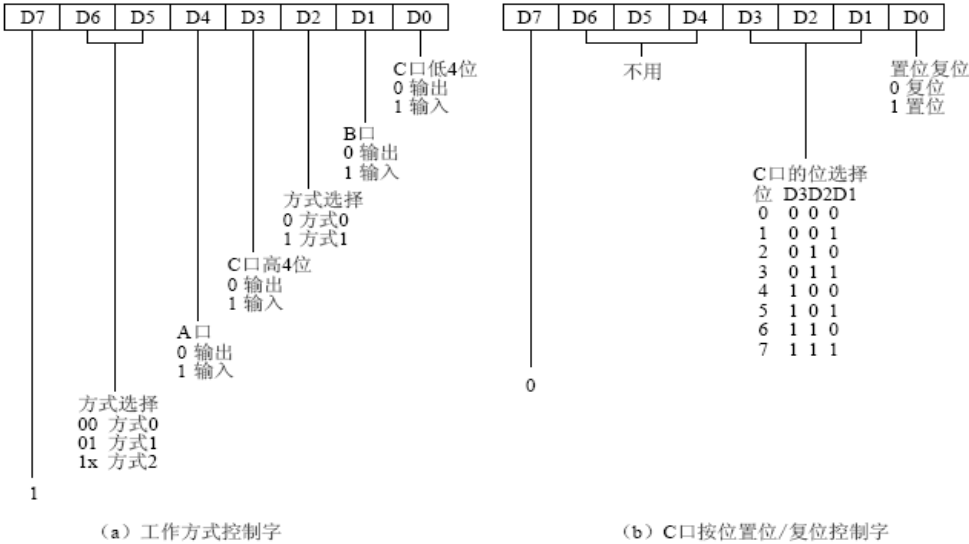


图 2.2 置位/复位控制字格式

2.3.3 8259 主要功能

中断控制器 8259 是 Intel 公司专为控制优先级中断而设计开发的芯片。它将中断源优先级 排队、辨别中断源以及提供中断矢量的电路集于一片中，因此无需附加任何电路，只需对 8259A 进行编程，就可以管理 8 级中断，并选择优先模式和中断请求方式，即中断结构可以由用户编程来设定。同时，在不需增加其他电路的情况下，通过多片 8259A 的级

连，能构成多达 64 级的矢量中断系统。它的管理功能包括：1) 记录各级中断源请求，2) 判别优先级，确定是否响应 和响应哪一级中断，3) 响应中断时，向 CPU 传送中断类型号。8259A 的内部结构和引脚如图 2.3 所示：

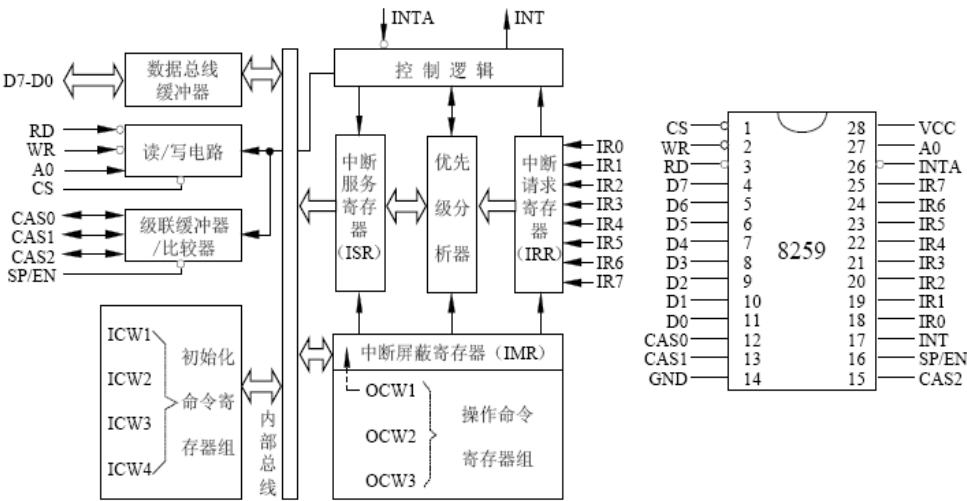


图 2.3 内部结构和引脚

8259 的命令共有 7 个，一类是初始化命令字，另一类是操作命令。8259A 的编程就是根据 应用需要将初始化命令字 ICW1-ICW4 和操作命令字 OCW1- OCW3 分别写入初始化命令寄存器 组和操作命令寄存器组。

OCW1-OCW3 各命令字格 式如图 2.4 所示，其中 OCW1 用于设置中断屏蔽操作字，OCW2 用于设置优先级循环方式和 中断结束方式的操作命令字，OCW3 用于设置和撤销特殊屏蔽方式、设置中断查询方式以及设 置对 8259 内部寄存器的读出命令。



图 2.4 OCW1-OCW3 各命令字格

2. 3. 4 8254主要功能

8254 是 Intel 公司生产的可编程间隔定时器。是 8253 的改进型，比 8253 具有更优良的性能。8254 具有以下基本功能：

- 有 3 个独立的 16 位计数器；
- 每个计数器可按二进制或十进制（BCD）计数； 每个计数器可编程工作于 6 种不同

工作方式；每个计数器允许的最高计数频率为 10MHz；

●8254 有读回命令，除了可以读出当前计数单元的内容外，还可以读出状态寄存器的内容。

●计数脉冲可以是有规律的时钟信号，也可以是随机信号。计数初值公式为 $n=f_{CLKi} \div f_{OUTi}$ ，其中 f_{CLKi} 是输入时钟脉冲的频率， f_{OUTi} 是输出波形的频率。

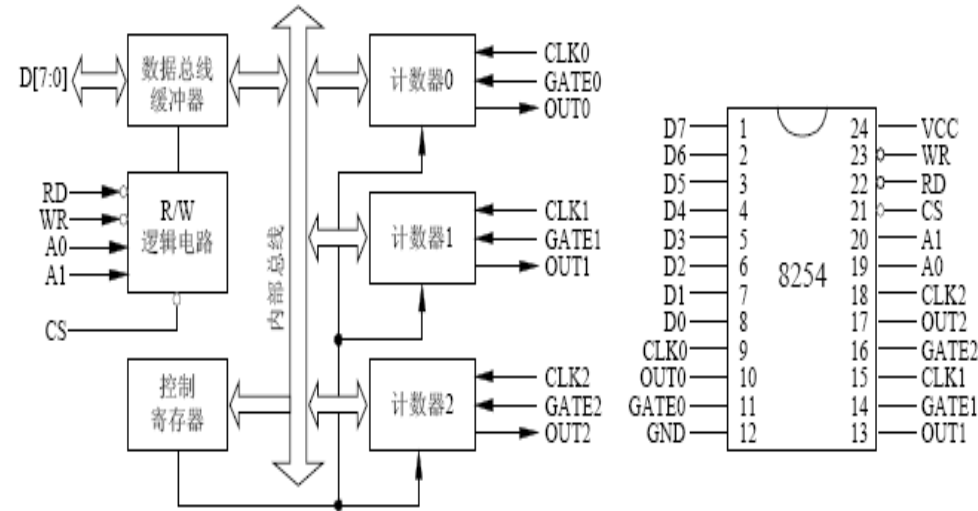


图 2.5 内部结构框图和引脚图

上图是 8254 的内部结构框图和引脚图，它是由与 CPU 的接口、内部控制电路和三个计数器组成。8254 的工作方式如下述：

- (1) 方式 0：计数到 0 结束输出正跃变信号方式。
- (2) 方式 1：硬件可重触发单稳方式。
- (3) 方式 2：频率发生器方式。
- (4) 方式 3：方波发生器。
- (5) 方式 4：软件触发选通方式。
- (6) 方式 5：硬件触发选通方式。

8254 的控制字有两个：一个用来设置计数器的工作方式，称为方式控制字；另一个用来设置读回命令，称为读回控制字。这两个控制字共用一个地址，由标识位来区分。当读回控制字的 D4 位为 0 时，由该读回控制字 D1~D2 位指定的计数器的状态寄存器内容将被锁存到状态寄存器中。状态字格式如图 2.6 所示：

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		读/写格式选择		工作方式选择			计数码制选择
00—计数器 0		00—锁存计数值		000—方式 0			0—二进制数
01—计数器 1		01—读/写低 8 位		001—方式 1			1—十进制数
10—计数器 2		10—读/写高 8 位		010—方式 2			
11—读出控制 字标志		11—先读/写低 8 位 再读/写高 8 位		011—方式 3			
				100—方式 4			
				101—方式 5			

图 2.6 8254 状态字格式

五、 程序接线图及流程图

3.1 实验接线图

系统的连线图主要由五个部分构成。8259 中断控制器、8255 并口控制器、8254 定时计数器、键盘及数码管显示单元、开关及 LED 显示单元构成。如下图所示：

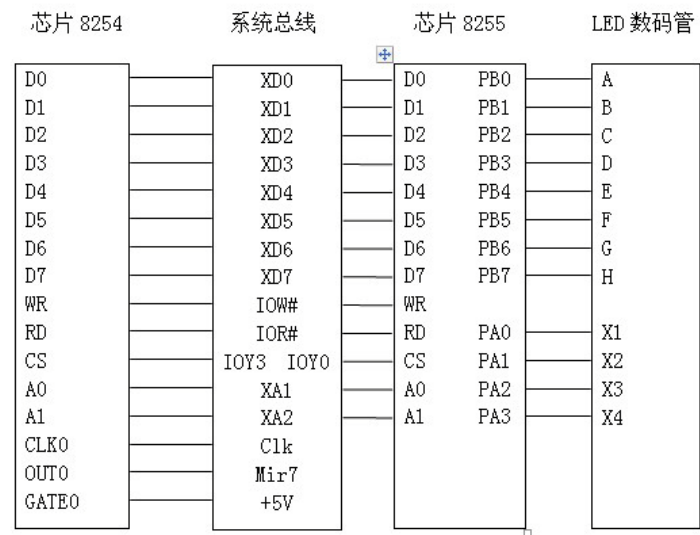


图 3.4 实验接线图

3.2 主程序流程图

主程序的流程图是对整个程序的逻辑的图象表示。首先是设置 8254、8255、8259 的各个端口地址。接着初始化各个芯片，开始计时，并通过 8259 中断控制时间的正确显示。在整个过程中都在检测是否有按键响应。有则停止，没有则返回继续执行。

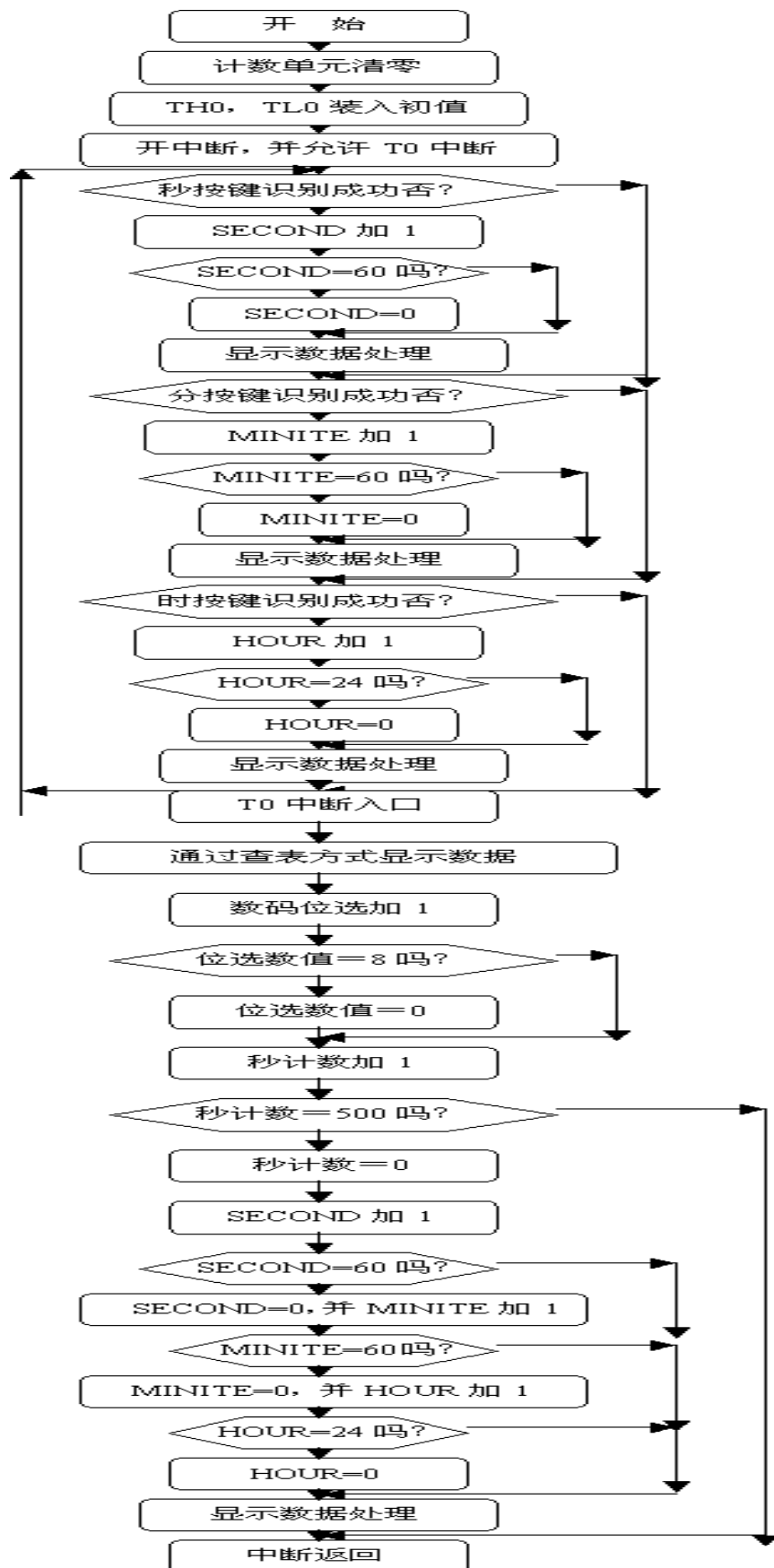


图 3.3 系统实现流程图

实验九 A/D转换实验

一、实验目的

1. 学习理解模/数信号转换的基本原理；
2. 掌握模/数转换芯片ADC0809的使用方法。

二、实验设备

PC机一台，TD-PITE 实验装置一套，万用表一个。

三、实验内容

编写实验程序，将ADC单元中提供的0V~5V信号源作为ADC0809的模拟输入量，进行A/D 转换，转换结果通过变量进行显示。

四、实验原理

ADC0809包括一个8 位的逐次逼近型的ADC部分，并提供一个8 通道的模拟多路开关和联合寻址逻辑。用它可直接输入8 个单端的模拟信号，分时进行A/D 转换，在多点巡回检测、过程控制等应用领域中使用非常广泛。ADC0809的主要技术指标为：

- 分辨率：8 位
- 单电源：+5V
- 总的不可调误差： $\pm 1\text{LSB}$
- 转换时间：取决于时钟频率
- 模拟输入范围：单极性 0~5V
- 时钟频率范围：10KHz~1280KHz

ADC0809的外部管脚如图2-8-1所示，地址信号与选中通道的关系如表2-7-1所示。

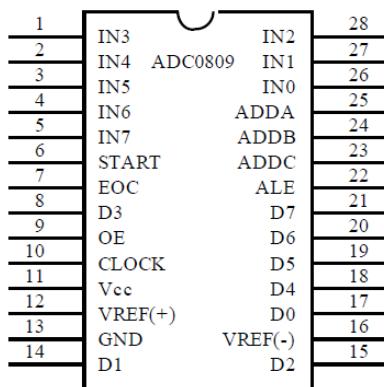


图2-7-1 ADC0809外部引脚图

表2-7-1 地址信号与选中通道的关系

地 址			选中通道
A	B	C	
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6

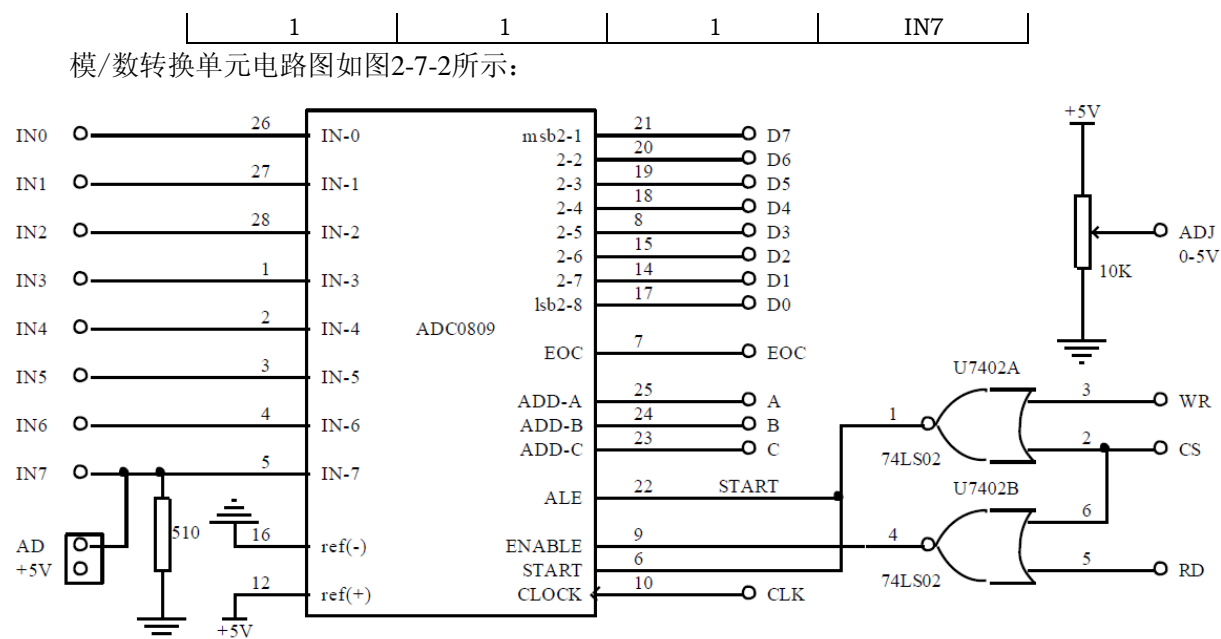


图2-7-2 模/数转换电路图

五、实验步骤

1. 按图2-7-3连接实验线路；
2. 编写实验程序，经编译、链接无误后装入系统；
3. 将变量VALUE 添加到变量监视窗口中；
4. 在JMP START语句行设置断点，使用万用表测量ADJ 端的电压值，计算对应的采样值，然后运行程序；
5. 程序运行到断点处停止运行，查看变量窗口中VALUE 的值，与计算的理论值进行比较，看是否一致（可能稍有误差，相差不大）；
6. 调节电位器，改变输入电压，比较VALUE 与计算值，反复验证程序功能。

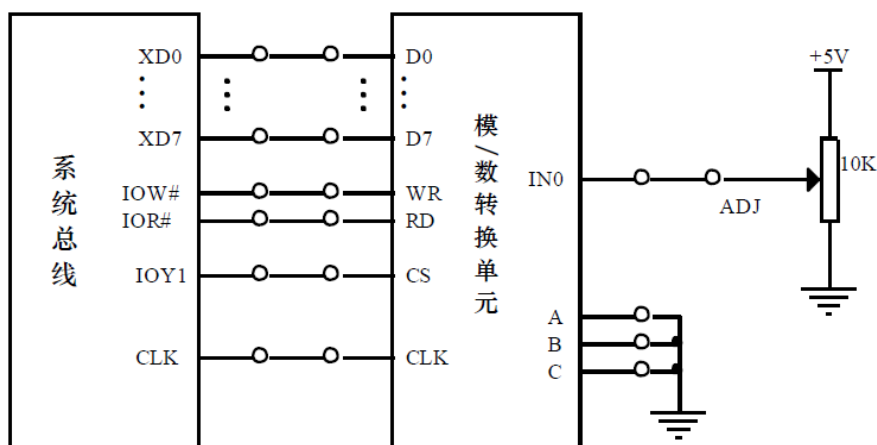


图2-7-3 AD转换实验接线图

实验十 D/A转换实验

一、实验目的

1. 学习数/模转换的基本原理；
2. 掌握DAC0832的使用方法。

二、实验设备

PC机一台，TD-PITE 实验装置一套，示波器一台。

三、实验内容

设计实验电路图实验线路并编写程序，实现D/A转换，要求产生锯齿波、脉冲波，并用示波器观察电压波形。

四、实验原理

D/A 转换器是一种将数字量转换成模拟量的器件，其特点是：接收、保持和转换的数字信息，不存在随温度、时间漂移的问题，其电路抗干扰性较好。大多数的D/A转换器接口设计主要围绕D/A集成芯片的使用及配置响应的外围电路。DAC0832是8 位芯片，采用CMOS 工艺和R-2RT 形电阻解码网络，转换结果为一对差动电流Iout1 和Iout2 输出，其主要性能参数如表2-8-1示，引脚如图2-8-1所示。

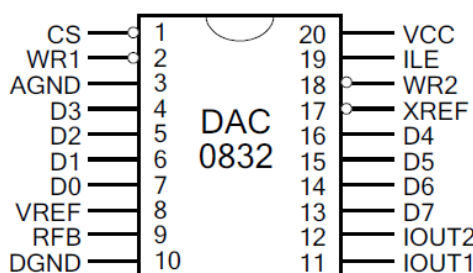


图2-8-1 DAC0832引脚图

表2-8-1 DAC0832性能参数

性能参数	参数值
分辨率	8 位
单电源	+5V~ +15V
参考电压	+10V~-10V
转换时间	1Us
满刻度误差	± 1LSB
数据输入电平	与 TTL 电平兼容

D/A 转换单元实验电路图如图2-8-2所示：

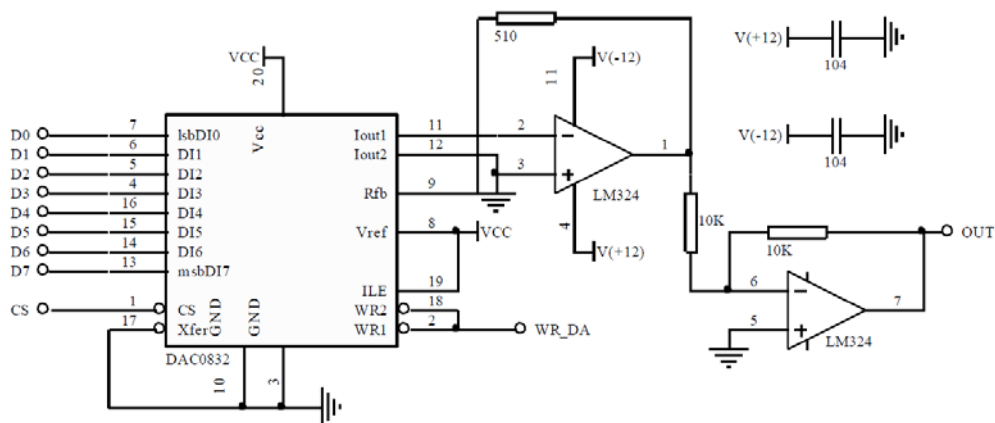






图2-8-2 D/A 实验单元电路图

五、 实验步骤

1. 实验接线图如图2-8-3所示，按图接线；
2. 编写实验程序，经编译、链接无误后装入系统；
3. 单击  按钮，运行实验程序，用示波器测量 DA 的输出，观察实验现象。
4. 用示波器观察波形的方法：单击虚拟仪器菜单中的  示波器 按钮或直接单击工具栏的  按钮，在新弹出的示波器界面上单击  按钮运行示波器，观测实验波形。
5. 自行编写实验程序，产生三角波形，使用示波器观察输出，验证程序功能。

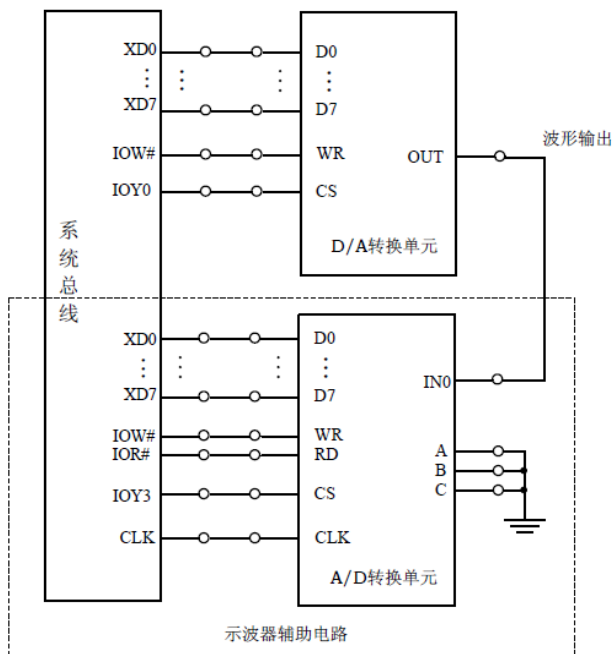


图2-8-3 D/A实验接线图

附录1 Wmd86 联机软件使用说明

附1.1 菜单功能

1. 文件菜单项

文件菜单如附图 1-1 所示。

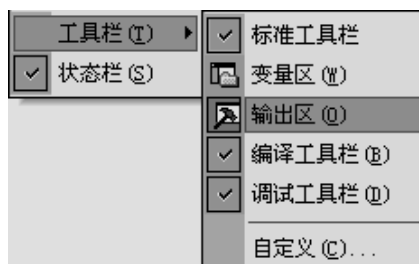
- (1) 新建 (N) : 用此命令在 Wmd86 中建立一个新文档。
- (2) 打开 (O) : 用此命令在窗口中打开一个现存的文档。
- (3) 关闭 (C) : 用此命令来关闭当前活动文档。
- (4) 保存 (S) : 用此命令将当前活动文档保存到它的当前文件名和目录下。当您第一次保存文档时, Wmd86 显示另存为 对话框以便您命名您的文档。
- (5) 另存为 (A) : 用此命令来保存并命名活动文档。
- (6) 打印 (P) : 用此命令来打印一个文档。
- (7) 打印预览 (V) : 用此命令来打印当前显示活动文档。
- (8) 打印设置 (R) : 用此命令来选择连接的打印机及其设置。
- (9) 最近浏览文件: 通过此列表, 直接打开最近打开过的文件。
- (10) 退出 (X) : 用此命令来结束 Wmd86 的运行阶段。Wmd86 会提示您保存尚未保存的改动。



附图 1-1 文件菜单

2. 查看菜单项

查看菜单如附图 1-2 所示。



附图 1-2 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

a、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, 一个打勾记号出现在该菜单项目的旁边。

b、变量区 (W) : 用此命令可显示和隐藏寄存器/变量/堆栈区。

c、输出区 (O) : 用此命令可显示和隐藏输出区。

d、编译工具栏 (B) : 用此命令可显示和隐藏编译工具栏。

e、调试工具栏 (D) : 用此命令可显示和隐藏调试工具栏。

f、自定义 (C) : 见自定义功能。

3. 端口菜单项

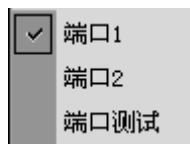
端口菜单如附图 1-3 所示。

(1) 端口1: 此命令用来选择串口1进行联机通讯, 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进行联机通讯, 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讯测试, 并报告测试结果

。



附图 1-3 端口菜单

4. 编译菜单项

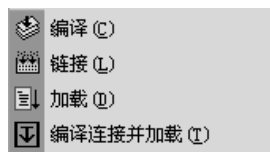
编译菜单如附图 1-4 所示。

(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。

(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。

(3) 加载 (D) : 把链接生成的可执行文件加载到下位机。加载成功, 输出区显示“加载成功!”。

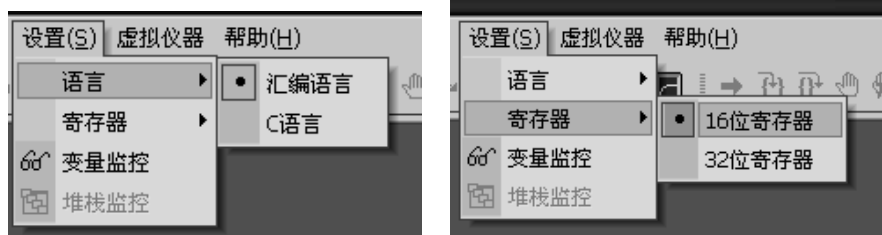
(4) 编译链接并加载 (T) : 依次执行编译、链接和加载。



附图 1-4 编译菜单

5. 设置菜单

设置菜单如附图 1-5 所示。



附图 1-5 设置菜单

(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。

(3) 变量监控：加载成功后才可用此按钮。系统只能监视全局变量。在汇编语言源文件中

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上图所示的对话框的左边列表，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。

(4) 堆栈监控：用于选择是否监控堆栈。

6. 调试菜单项

调试菜单如附图 1-6 所示。



附图 1-6 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。源程序设置的断点数不能超过 8 个。

(2) 清除所有断点 (D) 清除源程序中设置的所有断点。

(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。

(4) 单步 (T) 点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。

(5) 跳过 (O) 点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。

(6) 运行/运行到断点：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。

(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。固化程序之前，必须先将程序加到静态存储器中，然后才能进行固化程序的操作。

附1.2 工具栏功能介绍

1. 标准工具栏

标准工具栏共有十二个按钮，如下图所示。







- (1)  按钮：用此按钮在 Wmd86 中建立一个新文档。
- (2)  按钮：用此命令在一个新的窗口中打开一个现存的文档。
- (3)  按钮：用此命令将当前活动文档保存到其当前的文件名和目录下。
- (4)  按钮：用此命令将当前被选取的数据从文档中删除并放诸于剪贴板上。
- (5)  按钮：用此命令将被选取的数据复制到剪切板上。
- (6)  按钮：用此命令将剪贴板上内容的一个副本插入到插入点处。
- (7)  按钮：用此命令来打印一个文档。
- (8)  按钮：用此命令来撤消上一步编辑操作。
- (9)  按钮：用此命令来恢复撤消的编辑操作。
- (10)  按钮：用此按钮可显示和隐藏变量和寄存器区。
- (11)  按钮：用此按钮可显示和隐藏输出区。
- (12)  按钮：加载成功后才可用此按钮。点击此按钮，可进行全局变量监视。
- (13)  按钮：堆栈监控按钮，点击此按钮将弹出堆栈监控对话框。

2. 编译工具栏

编译工具栏共有五个按钮，其图如下：










- (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下位机。
- (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

调试工具栏共有八个按钮，其图如下：



- (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：点击此命令使程序执行一条语句。
- (3)  跳过：点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，运行到断点后再次点击此按钮，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。
- (8) 固化程序：程序加载到存储器后，可用固化程序功能将该程序烧写到 FLASH 存储器里面。

附1.3 专用图形显示

主要用于观察“直流电机闭环调速实验”及“温度单元或电烤箱闭环温度控制实验”的响应曲线，本界面可以观察系统的给定值、反馈值及控制量之间的响应曲线关系。

实验中给定值、反馈值都为单极性，屏幕最底端对应值为 00H，最顶端对应值为 FFH，对于时间刻度值由于采样周期不同存在以下关系：

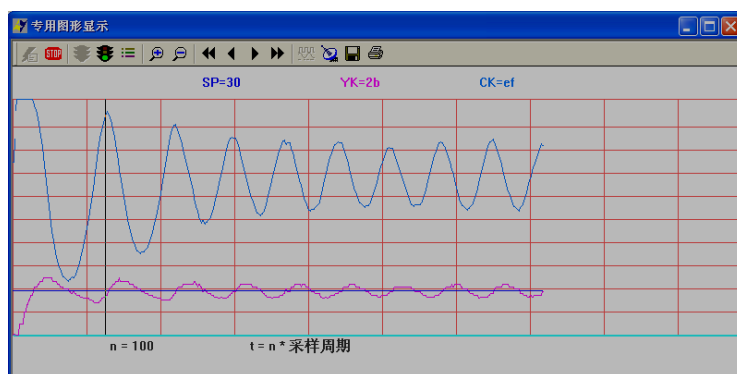
实际时间（秒）= $n(\text{实际刻度值}) \times \text{采样周期}$

控制量具有双极性，00H~7FH 为负值，80H~FFH 为正值。

直流电机闭环调速实验中，电机转速范围为 6 转/秒~48 转/秒。即：给定值 SPEC 范围约在 06H~30H 之间。示例程序中给定 SPEC = 30H 为 48 转/秒。TS = 14H，由于 8253 OUT2 接 IRQ6 中断为 1ms，故采样周期=14H×1ms = 0.02s。如实际刻度值 $n = 100$ ，则实际响应时间 = $0.02 \times 100 = 2\text{s}$ 。

温度闭环控制实验中，温度单元的 7805 控制范围的最佳温度范围为 50℃~70℃，不要过高。即给定值 SPEC 范围约在 14H(20℃)~46H(70℃)之间。示例程序中 SPEC = 30H 为 48℃。

TS = 64H，由于 8253 OUT2 接 IRQ6 中断为 10ms，故采样周期 = $64\text{H} \times 10\text{ms} = 1\text{s}$ ；如实际刻度值 $n = 100$ ，则实际响应时间(秒) = $1 \times 100 = 100\text{s}$ 。界面如下：





1) 显示说明


- (1) SP=30H: 要求电机达到的转速值 48 转/秒 (或要求达到的温度值 48℃)
- (2) YK=2bH: 运行状态下表示电机当前的转速值 43 转/秒 (或当前的温度值 43℃)
暂停 状态下表示指定时刻电机的转速值 (或指定时刻的温度值)
- (3) CK=efH: 运行状态下表示当前控制量的输出值, 暂停状态下表示指定时刻控制量的输出值。控制量 CK 在正数值时转化成 PWM 输出占空比为:

$$\frac{EFH-80H}{FFH-80H} \times 100\% = 87.4\%$$


2) 工具栏功能简介


- (1)  按钮: 启动并运行程序。运行加载在下位机中的程序。 , SP=' 后显示的值是当 前时刻系统的给定值, , YK=' 后显示 的值是当前系统的反馈值, , CK=' 后显示的值是当前时刻系统控制量的值。


- (2)  按钮: 停止程序运行。使下位机中运行的程序停止。


- (3)  按钮: 暂停程序运行。在运行状态下使波形暂停显示并出现游标。

- (4)  按钮: 退出暂停状态, 使波形继续显示, 游标消失。

- (5)  按钮: 显示选择按钮, 可选择性的选择要显示的波形。


- (6)  按钮: 放大波形。


- (7)  按钮: 缩小波形。


- (8)  按钮: 快速左移游标。在暂停状态下, 使游标快速向左移动。 , SP=' 后显示的值 是游标所在时刻系统的给定值, , YK=' 后显示的值是游标所在时刻系统的反馈值, , CK=' 后显示的值是游标所在时刻系统控制量的值。

- (9)  按钮: 左移游标。在暂停状态下, 使游标向左移动。 , SP=' 后显示的

值是游标 所在时刻系统的给定值，， YK=’ 后显示的值是游标所在时刻系统的反馈值，， CK=’ 后显示的值是游标所在时刻系统控制量的值。


(10)  按钮：右移游标。在暂停状态下，使游标向右移动。， SP=’ 后显示的值是游标 所在时刻系统的给定值，， YK=’ 后显示的值是游标所在时刻系统的反馈值，， CK=’ 后显示的值是游标所在时刻系统控制量的值。

(11)  按钮：快速右移游标。在暂停状态下，使游标快速向 右移动。， SP=’ 后显示的 值是游标所在时刻系统的给定值， YK=’ 后显示的值是游标所在时刻系统的反馈值， CK=’ 后显示的值是游标所在时刻系统控制量的值。

(12)  按钮：记录波形。点击此按钮，出现如下对话框：



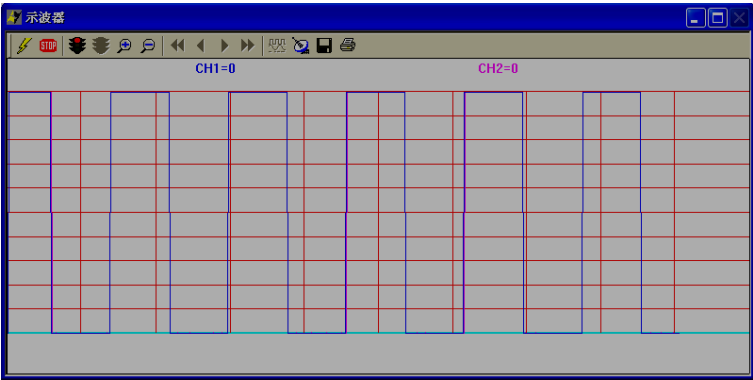
选中“图一”单选按钮，点击确定，系统会把当前时刻的波形保存到图一中，选中“图二”单选按钮，点击确定，系统会把当前时刻的波形保存到图二中，选中“图三”单选按钮，点击确定，系统会把当前时刻的波形保存到图三中。















(13)  按钮：显示保存到图一，图二和图三中的波形。

(14)  按钮：打印当前屏幕上的波形。

附 1.4 示波器

主要用于“8254 定时/计数器实验”、“D/A 转换实验”及“8251 串行接口实验”中波形的观察。



- (1)  按钮：启动示波器。 , CH1=' 、 , CH2=' 后分别显示游标当前位置的采样值。
- (2)  按钮：停止使下位机中运行的程序停止。
- (3)  按钮：在运行状态下使能，使波形暂停显示并出现游标。
- (4)  按钮：在暂停状态下使能，使波形继续显示，游标消失。
- (5)  按钮：放大波形。
- (6)  按钮：缩小波形。
- (7)  按钮：在暂停状态下，使游标快速向左移动。 , CH1=' 、 , CH2=' 后分别显示 游标当前位置的采样值。 , $T = xxx'$ 表示游标所在位置的时刻与图形最左端时刻的差值。
- (8)  按钮：在暂停状态下，使游标向左缓慢移动。
- (9)  按钮：在暂停状态下，使游标向右缓慢移动。
- (10)  按钮：在暂停状态下，使游标快速向右移动。
- (11)  按钮：点击此按钮，出现如右图所示对话框：
 选中“图一”单选按钮，点击确定，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)  按钮：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。
- (13)  按钮：以.bmp 格式保存当前屏幕上的波形到指定文件。
- (14)  按钮：打印当前屏幕上的波形。



附1.5 Debug调试命令

Wmd86 软件输出区集成有 Debug 调试, 点击调试标签, 进入 Debug 状态, 会出现命令提示符 “>”, 主要命令叙述如下:

A 进入小汇编

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“>”提示符下键入A2000↵, 此时默认的段址CS为0000, 规定偏移量IP为 2000, 屏幕显示与操作为:

附表 1-1 小汇编操作示例

显示内容	键入内容
0000:2000	MOV AX, 1234↵
0000:2003	INC AX↵
0000:2004	DEC AX↵
0000:2005	JMP 2000↵
0000:2007	↵

B 断点设置

在系统提示符下, 键入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]: 。若直接键入回车, 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。例:

附表 1-2 B 命令示例

显示内容	键入内容
>	B↵
[0]:	2009↵
[1]:	↵

D 显示一段地址单元中的数据

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 来暂停显示, 用任意键继续; 也可用 Ctrl+C 终止数据显示, 返回监控状态。

E 编辑指定地址单元中的数据

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。若输入回车, 则结束 E 命令。例:

附表 1-3 E 命令示例

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

G 运行程序

格式：G=[段址:]偏移量 ↵

G=[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 寄存器或片内 RAM 区显示与修改

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，键入回车键，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

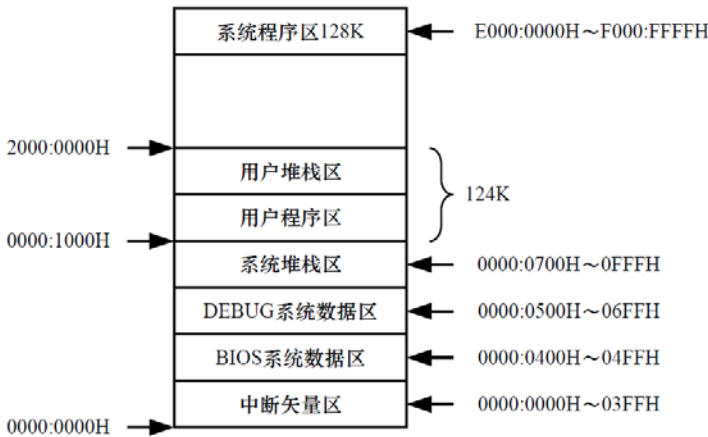
格式：U[[段址:]起始地址[, 尾地址]]

附录 2 系统编程信息

附 2.1 地址分配情况

1. 系统内存分配

系统内存分配情况如附图 2-1 所示。系统内存分为程序存储器和数据存储器，程序存储器为一片 128KB 的 FLASH ROM，数据存储器为一片 128KB 的 SRAM。



附图 2-1 系统内存分配

2. 系统编址

采用内存与 IO 独立编址形式，内存地址空间和外设地址空间是相对独立的。内存地址是连续的 1M 字节，从 00000H~FFFFFH。外设的地址范围从 0000H~FFFFH，总共 64K 字节。

(1) 存储器编制

存储器编址情况见下表。

附表 2-1 存储器编址

	信号线	编址空间
系统程序存储器		E0000H~FFFFFH
系统数据存储器		00000H~1FFFFH
扩展存储器	MY0	80000H~9FFFFH
	MY1	A0000H~BFFFFH

即 SRAM 空间：00000H~1FFFFH 共 128K

其中：00000H~00FFFFH 为 4K 系统区

01000H~1FFFFH 为 124K 用户使用区

FLASH 空间：0E0000H~0FFFFFFH 共 128K

其中：0E0000H~0EFFFFH 为 64K 供用户使用区

0F0000H~0FFFFFFH 为 64K 系统监控区

(2) 输入/输出接口编址

输入/输出接口编址见下表。

附表 2-2 输入/输出接口编址

	信号线	编址空间
主片 8259		20H、21H

从片 8259		A0H、A1H
扩展 I/O 接口	IOY0	0600H~063FH
	IOY1	0640H~067FH
	IOY2	0680H~06BFH
	IOY3	06C0H~06FFH

附 2.2 常用 BIOS 及 DOS 功能调用说明

附表 2-3 INT 03H 使用说明

入口：无
功能：程序终止

附表 2-4 INT 10H 使用说明

入口：AH=01H，AL=数据
功能：写 AL 中的数据到屏上

入口：AH=06H，DS: BX=字串首址，且字串尾用 00H 填充
功能：显示一字串，直到遇到 00H 为止

附表 2-5 INT 16H 使用说明

入口：AH=00H
功能：读键盘缓冲到 AL 中，读指针移动，ZF=1 无键值，ZF=0 有键值

入口：AH=01H
功能：检测键盘缓冲，并送到 AL 中，读指针不动，ZF=1 无键值，ZF=0 有键值

附表 2-6 INT 21H 使用说明

入口：AH=00H 或 AH=4CH
功能：程序终止

入口：AH=01H
功能：读键盘输入到 AL 中并回显

入口：AH=02H，DL=数据
功能：写 DL 中的数据到显示屏

入口：AH=08H
功能：读键盘输入到 AL 中无回显

入口：AH=09H，DS:DX=字符串首地址，字符串以 '\$' 结束
功能：显示字符串，直到遇到 '\$' 为止

入口：AH=0AH，DS:DX=缓冲区首地址，(DS:DX)=缓冲区最大字符数，(DS:DX+1)=实际输入字符数，(DS:DX+2)=输入字符串起始地址
功能：读键盘输入的字符串到 DS:DX 指定缓冲区中并以回车结束