

# TPC-USB

## 通用微机接口实验系统

## 学生用实验指导书

### (汇编+VC)

刘晓坤 郑文锋 冯一兵 金丽霞 编写

清华大学计算机系  
清华大学科教仪器厂  
2006年11月

## 目 录

第一章 TPC-USB实验系统介绍.....	4
1.1 概述 .....	4
1.2 TPC-USB实验系统构成及特点 .....	4
第二章 TPC-USB实验系统硬件环境.....	5
2.1 USB模块介绍 .....	5
2.1.1 USB模块结构 .....	5
2.1.2 USB模块功能 .....	5
2.1.3 USB模块的对外接口 .....	5
2.1.4 USB模块跳线说明 .....	6
2.1.5 USB模块的安装 .....	7
2.1.6 USB模块连接测试 .....	11
2.2 扩展实验台结构及主要电路 .....	12
2.2.1 扩展实验台结构图 .....	12
2.2.2 实验台上包括的主要电路: .....	12
2.2.3 用户扩展实验区 .....	16
2.2.4 实验台跳线开关 .....	16
2.2.5 20芯双排插座 .....	17
2.2.6 直流稳压电源 .....	17
第三章 TPC-USB集成软件开发环境.....	18
3.1 TPC-USB集成开发环境软件包 .....	18
3.2 集成开发环境软件的安装 .....	18
3.2.1 用户程序的编辑和编译 .....	20
3.2.2 编译源程序 .....	22
3.2.3 用户程序的调试和运行 .....	22
3.2.4 常用调试命令 .....	25
3.2.5 实验项目的查看和演示 .....	27
3.2.6 实验项目的添加和删除 .....	28

3.2.7 集成开发环境帮助菜单 .....	29
第四章 汇编实验部分.....	32
实验一 I/O地址译码 .....	32
实验二 简单并行接口 .....	33
实验三 可编程定时器 / 计数器 (8253) .....	35
实验四 可编程并行接口 (一) (8255方式0) .....	39
实验五 七段数码管 .....	42
实验六 继电器控制 .....	43
实验七 竞赛抢答器 .....	45
实验八 交通灯控制实验 .....	47
实验九 中断 .....	48
实验十 可编程并行接口 (二) (8255方式1) .....	50
实验十一 数/模转换器 .....	52
实验十二 模/数转换器 .....	53
实验十三 串行通讯 .....	56
实验十四 DMA传送 .....	57
实验十五 集成电路测试 .....	61
实验十六 电子琴 .....	62
实验十七 8250串行通讯实验 .....	64
实验十八 步进电机控制实验 .....	66
实验十九 小直流电机转速控制实验 .....	68
实验二十 键盘显示控制器实验 .....	70
实验二十一 存储器读写实验 .....	75
实验二十二 双色点阵发光二极管显示实验 .....	76
第五章 VC++实验部分.....	81
1、基本输入输出-基本输入输出函数简介 .....	81
实验一 I/O地址译码 .....	82
实验二 简单并行接口 .....	84

实验三	可编程定时器 / 计数器 (8253)	85
实验四	可编程并行接口 (一) (8255方式0)	87
实验五	七段数码管	88
实验六	继电器控制	90
实验七	竞赛抢答器	91
实验八	交通灯控制实验	93
2、中断-中断函数简介		95
实验九	中断	95
实验十	可编程并行接口 (二) (8255方式1)	98
实验十一	数/模转换器	99
实验十二	模/数转换器	101
实验十三	串行通讯	103
3、DMA及RAM操作函数简介		105
实验十四	DMA传送	106
实验十五	集成电路测试	109
实验十六	电子琴	110
实验十七	8250串行通讯实验	112
实验十八	步进电机控制实验	114
实验十九	小直流电机转速控制实验	116
实验二十	键盘显示控制器实验	117
实验二十一	存储器读写实验	124
实验二十二	双色点阵发光二极管显示实验	125
附录一、随机光盘实验程序名称表		129
附录二: TPC-USB通用微机接口实验系统硬件实验指导 (汇编程序)		131
附录三: TPC-USB通用微机接口实验系统硬件实验指导 (C语言程序)		135

## 第一章 TPC-USB实验系统介绍

### 1.1 概述

在各种计算机外围接口不断推陈出新的今天，USB接口已经成为个人计算机最重要的接口方式之一，USB接口设备的应用也以惊人的速度发展，几乎新型的PC都100%支持USB技术。了解和掌握USB的应用及开发是计算机类、电子类、物理类本科生、大专生的新课题。

TPC-USB微机接口实验系统正是在这种背景下推出的。该设备在原TPC-2003A微机接口实验系统上配置了USB接口模块，直接与主机(PC)的USB接口连接，形成了一套完整的USB接口的微机接口实验系统。该系统适应当前高等院校所开设的《微机原理及其应用》和《微机接口技术》这两门课的实验，同时也提供了最新接口USB的实验，使学生在校学习期间不仅有机会接触常规接口，同时有机会接触新型的接口，为学生们今后从事微机开发应用打下基础。

### 1.2 TPC-USB实验系统构成及特点

该系统由一块USB总线接口模块、一个扩展实验台及软件集成实验环境组成。USB总线接口模块通过USB总线电缆与PC机相连，模块与实验台之间由一条50芯扁平电缆连接。其主要特点如下：

1. USB总线接口使用ISP1581 USB2.0高速接口芯片，完全符合USB2.0规范。提供了高速USB下的通信能力，即插即用。

2. 满足《微机原理与接口技术》课程教学实验要求。实验台接口集成电路包括：可编程定时器/计数器（8253）、可编程并行接口（8255）、数/模转换器（DAC0832）、模/数转换器（ADC0809）等。外围电路包括：逻辑电平开关、LED显示、七段数码管显示、8X8双色发光二极管点阵及驱动电路、直流电机步进电机及驱动电路、电机测速用光藕电路、数字测温传感器及接口电路、继电器及驱动电路、喇叭及驱动电路。8279键盘显示控制电路。

3. 在USB接口模块上扩展有DMA控制器8237及存储器，可以完成微机DMA传送以及USB的DMA传送等实验。

4. 开放式结构，模块化设计支持开放实验。实验台上除固定电路外还设有用户扩展实验区。有五个通用集成电路插座，每个插座引脚都有对应的“自锁紧”插孔，利用这些插孔可以搭试更多的自己设计的实验，方便的进行课程设计。

5. 功能强大的软件集成开发环境，支持Win2000;WinXP 等操作系统。可以方便的对程序进行编辑、编译、链接和调试，可以查看实验原理图，实验接线，实验程序并进行实验演示。可以增加和删除实验项目。

6. 实验程序可以使8086汇编和C语言编程实验。可以对汇编程序和C语言程序进行调试。

7. 系统还提供：字符、图形液晶显示实验模块；红外收发实验模块；无线通信实验模块；键盘显示实验模块等多种扩展实验模块（自选）。

8. 实验台自备电源，具有电源短路保护确保系统安全。

9. 使用USB接口与PC机相连，省却了打开主机箱安装接口卡的麻烦。

## 第二章 TPC-USB实验系统硬件环境

### 2.1 USB模块介绍

#### 2.1.1 USB模块结构

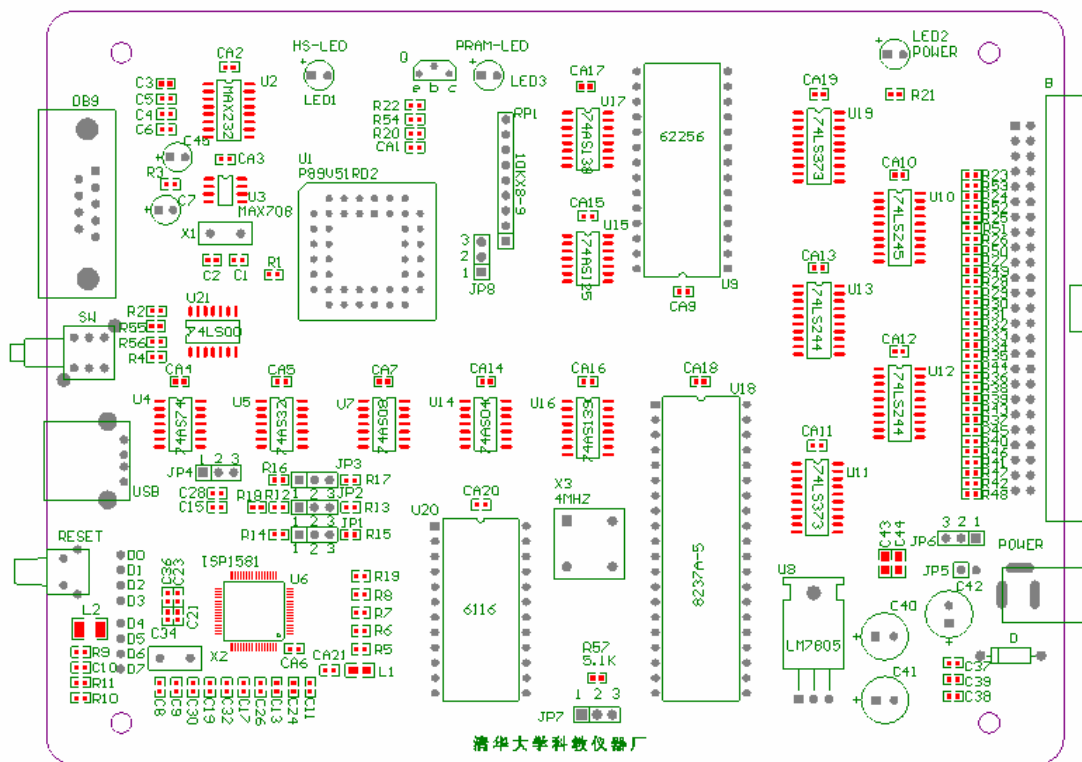


图2-1 USB模块结构图

#### 2.1.2 USB模块功能

1. 实验系统中的USB模块使用PHILIPS的ISP1581 USB2.0高速接口芯片，符合USB2.0接口规范，提供了高速USB下的通信能力。
2. 支持ISP下载，通过模块上的RS-232接口，可以对模块内部的MCU进行在线编程，对软件进行修改或在线升级。也可以通过RS-232接口下载实验程序到USB模块，进行实验。
3. 模块内扩展有DMA控制器8237及存储器，可以完成微机DMA传送和USB的DMA传送实验。
4. 该模块提供一个50线扁平电缆，通过该电缆将模块产生的仿ISA总线信号连到实验台上。

#### 2.1.3 USB模块的对外接口

1. 在该模块的右侧提供四个对外接口：

- ① 9芯通用RS-232接口，需要时可连到主机的COM1或COM2，对内部的MCU在线编程，对软件升级或修改。
- ② USB接口，连接到主机，实验时用于信息和数据的通信。
- ③ 清零按钮（RESET），用于对模块内部电路的初始化。

④实验方式转换按钮（SW），有些实验需要将实验程序下载到USB模块运行，需要时按一下该按钮以转换实验方式。（一般情况下，用户不要按此按钮，如果需要，在实验说明中会指出）。

2. 在模块的左侧提供二个对外接口：

①50线扁平电缆接口，为实验台提供仿ISA总线信号。信号安排与实验台上50芯信号插座信号一一对应。（见表十四 50芯总线插座信号）

②外接电源插孔，外接7~9V直流电源。平时USB模块与实验台相连时，使用实验台提供的电源，当USB模块单独使用或调试时，使用外接电源。

#### 2.1.4 USB模块跳线说明

在USB模块内，用一部分跳线选择ISP1581和其它芯片的工作模式，跳线的连接说明如下：

JP1: MODE1 ISP1581 ALE/A0 功能选择。

2—3短接 低电平 选择ALE功能（地址锁存使能）

1—2短接 高电平 选择A0功能（地址数据指示）

（USB模块出厂时选择2—3短接）

JP2: M0/DA1 选择ISP1581在通用处理器模式下的读写功能。

2—3短接 低电平 选择Motorola 类型的微处理器

1—2短接 高电平 选择8051 类型的微处理器

（USB模块出厂时选择1—2短接）

JP3: BUS/DA0 选择ISP1581 总线模式

2—3短接 低电平 选择断开总线模式，AD[7:0]多路复用

1—2短接 高电平 选择通用处理器模式，AD[7:0]8位地址线

（USB模块出厂时选择2—3短接）

JP4: ISP1581 片选信号选择

2—3短接 ISP1581 片选信号由MCU 产生

1—2短接 ISP1581 片选信号由地址译码产生

（USB模块出厂时选择1—2短接）

JP5: 无须用户设置

JP6: USB模块电源选择

2—3短接 选择外接电源

1—2短接 使用实验台电源

（USB模块出厂时选择1—2短接）

JP7: DMA控制器时钟选择

2—3短接 选择振荡器产生时钟

1—2短接 选择由MCU 产生时钟

（USB模块出厂时选择2—3短接）

JP8: MCU 编程方式选择

2—3短接 MCU 处于编程方式

1—2短接 MCU 处于正常工作方式

（USB模块出厂时选择1—2短接）

### 2.1.5 USB模块的安装

安装步骤如下：

1. 关上实验台电源。
2. 50线扁平电缆一端接USB模块的50芯插座，另一端接实验台50线插座。
3. USB电缆的一端接模块的USB口，另一端接主机USB口。
4. 打开实验台电源。
5. 系统将自行检测到模块的接入，选择用户光盘上的USB驱动程序完成驱动的安装。

安装驱动过程如下：

USB电缆接入主机，连接USB模块并加载电源后，系统将自行检测到模块的接入，提示用户发现新硬件并要求安装设备驱动：

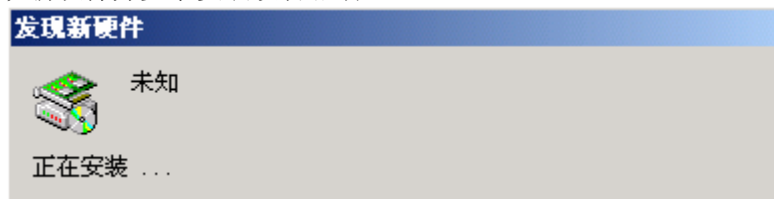


图2-2 系统发现新硬件

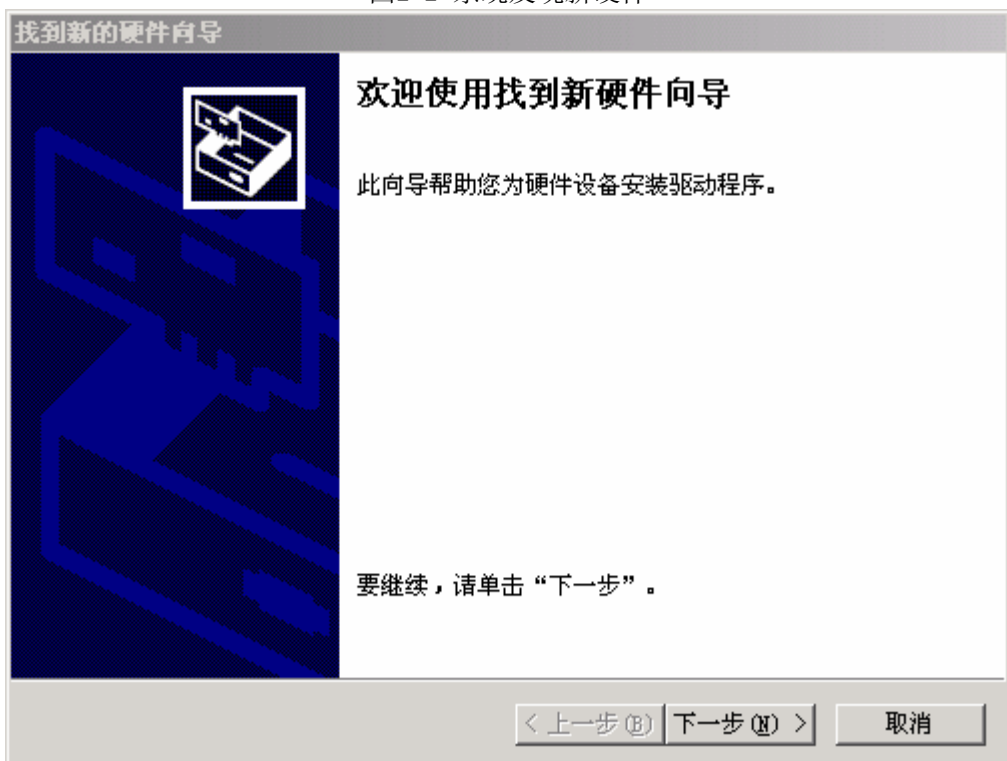


图2-3 提示找到新硬件

找到新硬件，需为此硬件指定设备驱动程序：



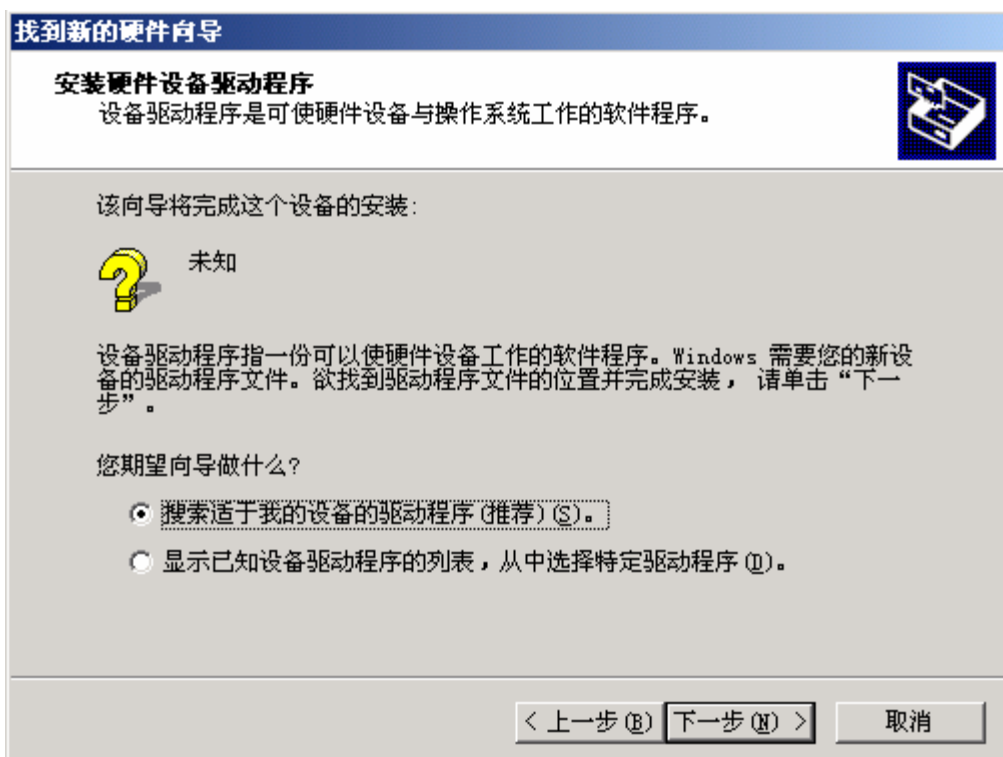


图2-4 提示按装驱动

选择驱动所在位置：(CD-ROM中“USB接口卡驱动”目录下或指定驱动所在位置)

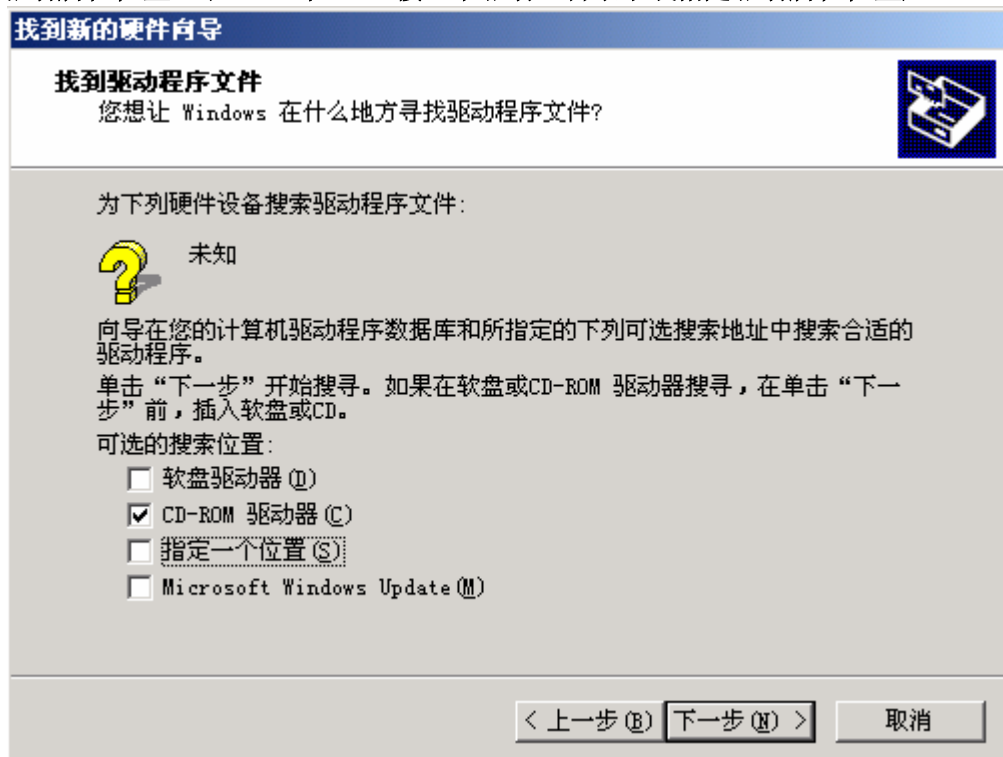


图2-5 指定驱动所在位置

浏览驱动所在位置并选定驱动安装信息文件TPCA.inf:

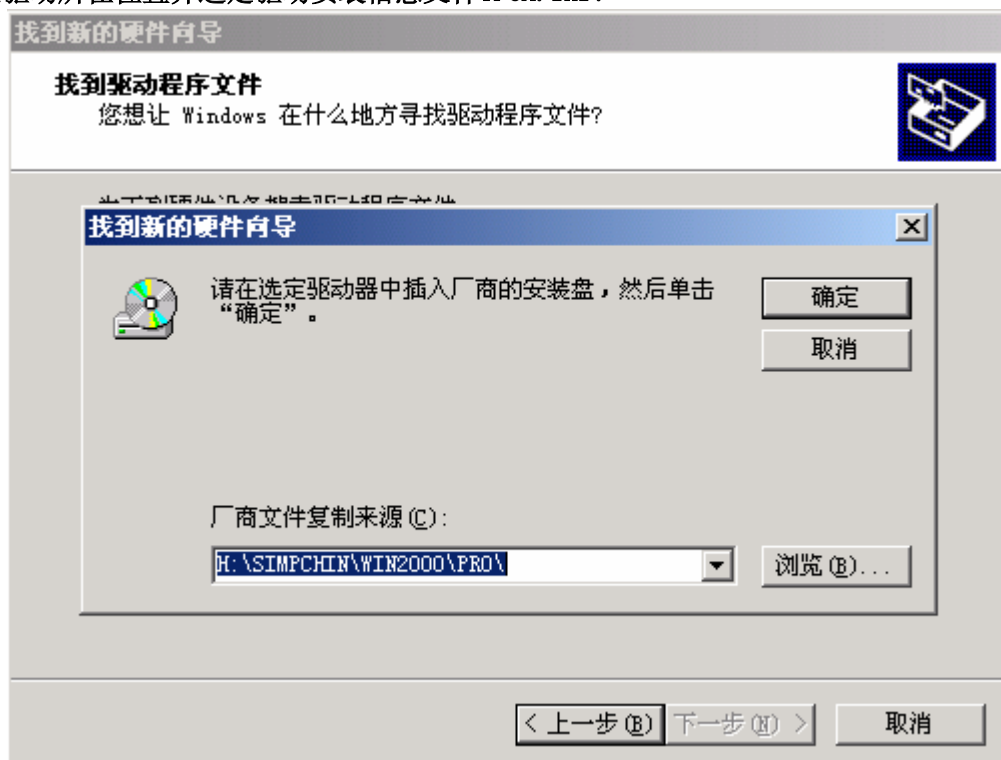


图2-6 浏览并找到驱动

选定TPCA.inf安装信息文件，并打开:



图2-7 找到驱动并选定

点击下一步，系统将自动为TPC设备安装其驱动:



图2-8 安装驱动

驱动安装完毕:

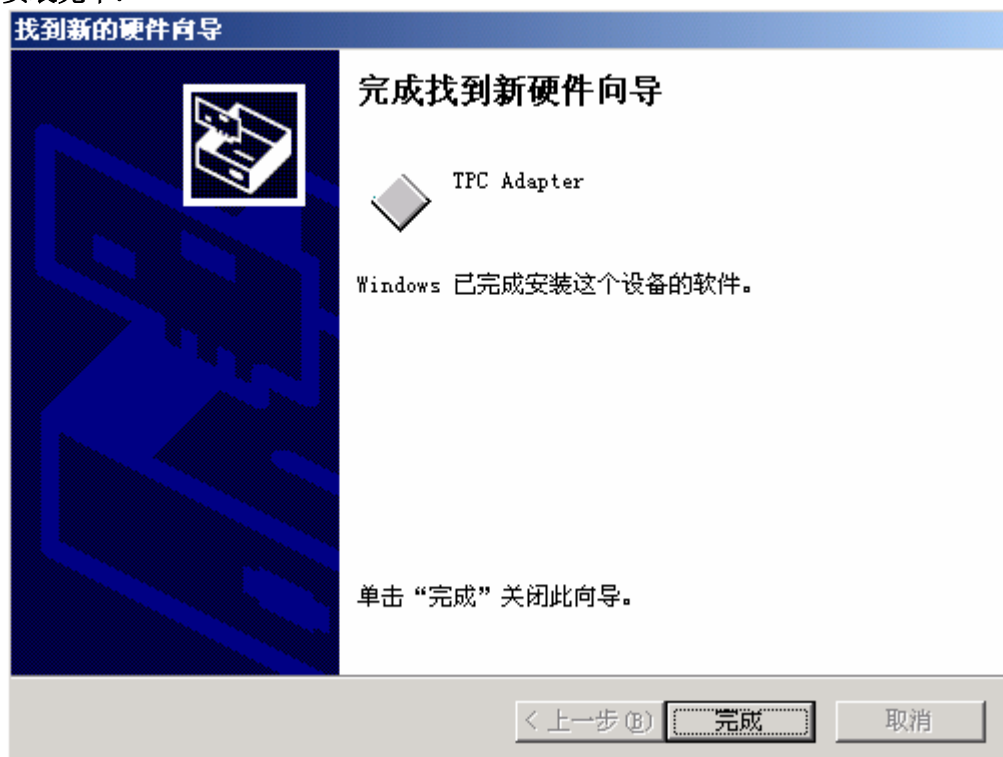


图2-9 完成安装

### 设备安装检测:

右键单击“我的电脑”，选择“属性”，选择硬件选项中的“设备管理器”，即可在通用串行总线控制器中找到已安装的TPC Adapter设备。至此安装完毕。

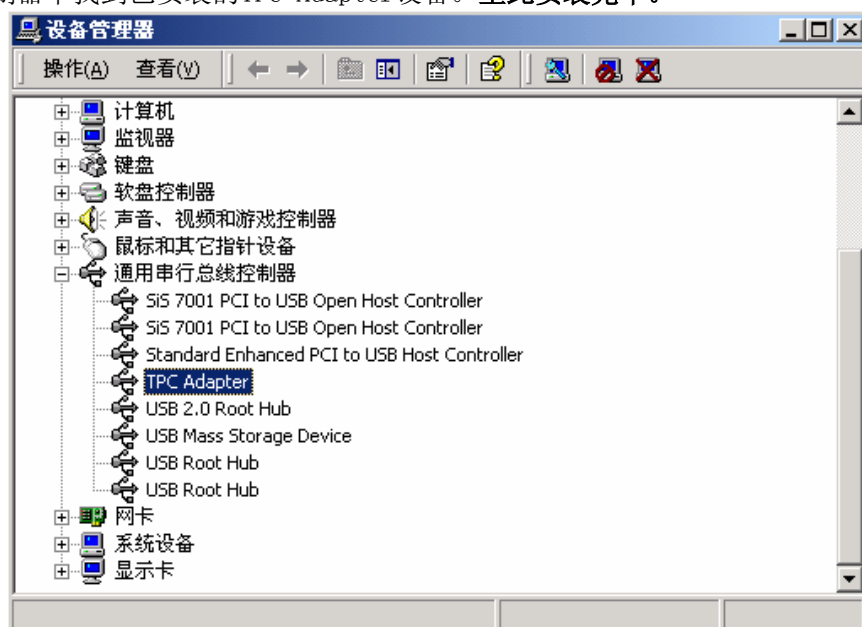


图2-10 查看安装

### 2.1.6 USB模块连接测试

驱动安装完成后，打开TPC-USB集成开发环境（集成开发环境的安装请参看3.2节‘集成开发环境的安装’），选定主菜单“选项”中的“硬件检测”，集成开发环境会检测到设备已连接，如果连接不正确，将会有错误提示。

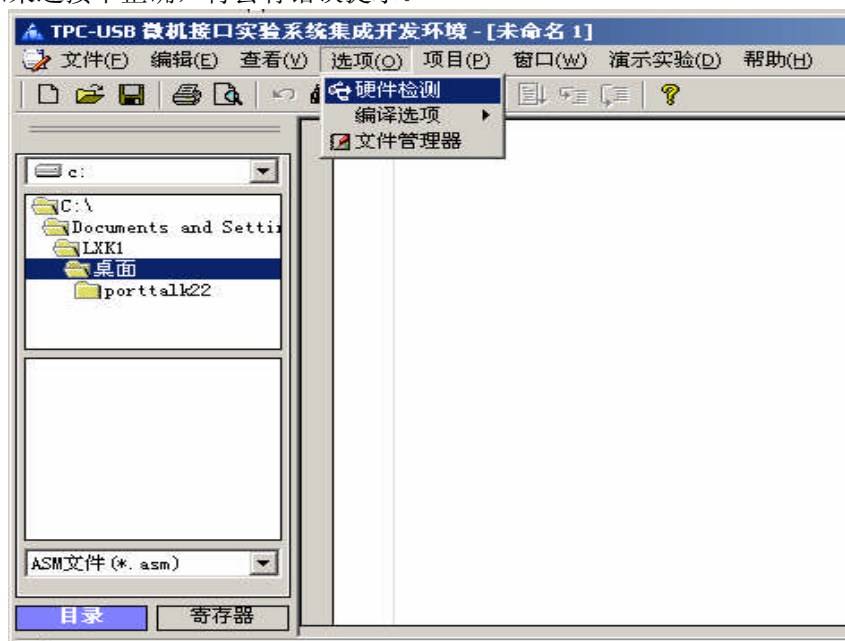


图2-11 硬件检测菜单



图2-12 正确连接提示



图2-13 硬件未连接

## 2.2 扩展实验台结构及主要电路

### 2.2.1 扩展实验台结构图

如图：图2-14 扩展实验台结构

### 2.2.2 实验台上包括的主要电路：

#### 1、50芯总线信号插座及总线信号插孔

1	+5V	11	E245	21	A7	31	A1	41	ALE
2	D7	12	IOR	22	A6	32	GND	42	T/C
3	D6	13	IOW	23	A5	33	A0	43	A16
4	D5	14	AEN	24	+12V	34	GND	44	A17
5	D4	15	DACK	25	A4	35	MEMW	45	A15
6	D3	16	DRQ1	26	GND	36	MEMR	46	A14
7	D2	17	IRQ	27	A3	37	CLK	47	A13
8	D1	18	+5V	28	-12V	38	RST	48	A12
9	D0	19	A9	29	A2	39	A19	49	A10
10	+5V	20	A8	30	GND	40	A18	50	A11

50芯总线信号插座在实验台左上方，总线插座信号安排如上表。各总线信号采用“自锁紧”插孔在标有“总线”的区域引出，有数据线D0-D7、地址线A19-A0、I/O读写信号IOR IOW、存储器读写信号 MEMR MEMW、中断请求 IRQ、DMA申请DRQ、DMA回答DACK、AEN 等。

#### 2、微机接口I/O地址译码电路

实验台上I/O地址选用280H—2BFH 64个，分8组输出：Y0-Y7，其地址分别为 280H—287H；288H—28FH；290H-297H；298H-29FH；2A0H-2A7H；2A8H-2AFH；2B0H-2B7H；2B8H-2BFH，8根输出线在实验台“I/O地址”处分别由自锁紧插孔引出。见图2-15

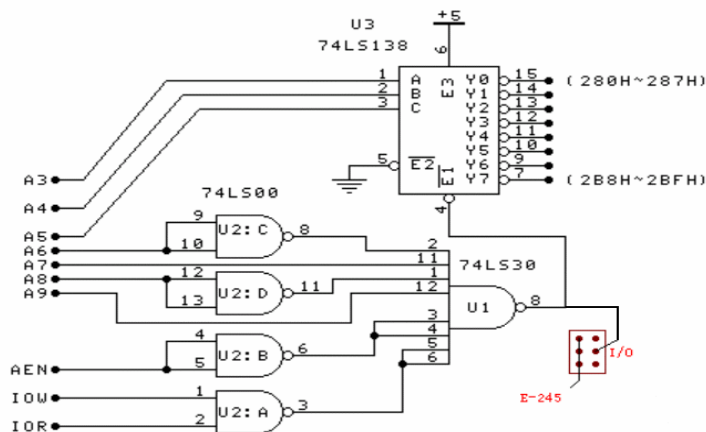


图2-15 I/O地址译码电路

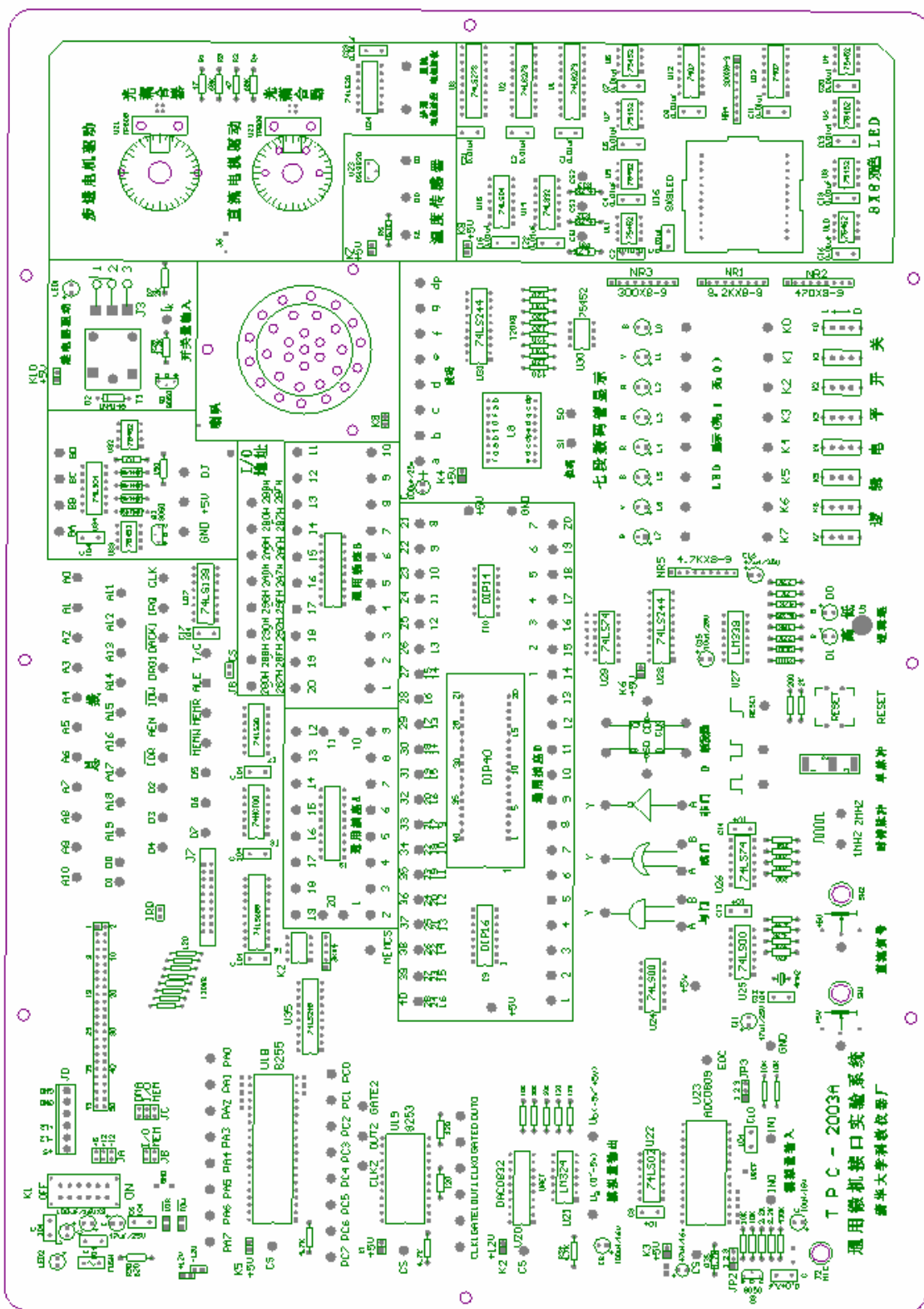


图2-14 扩展实验台结构

### 3、时钟电路

如图2-16所示，输出1MHZ、2MHZ两种信号，供定时器/计数器、A/D转换器、串行接口实验使用。

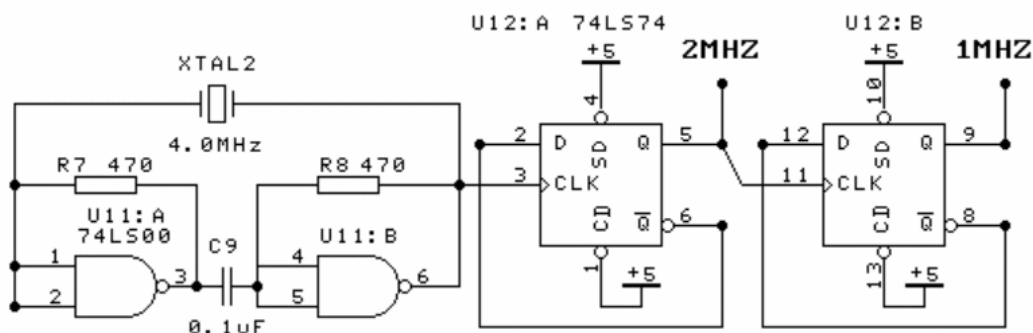


图2-16 时钟电路

### 4、逻辑电平开关电路

如图2-17所示，实验台右方有8个开关K0-K7，开关拨到“1”位置时开关断开，输出高电平。拨到“0”位置时开关接通输出低电平。电路中串接了保护电阻，接口电路不直接同+5V、GND相连，有效的防止因误操作损坏集成电路现象。

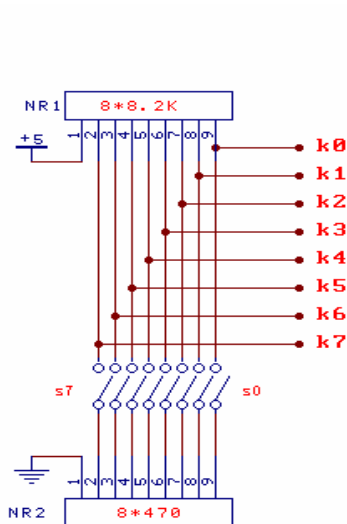


图2-17 逻辑电平开关电路

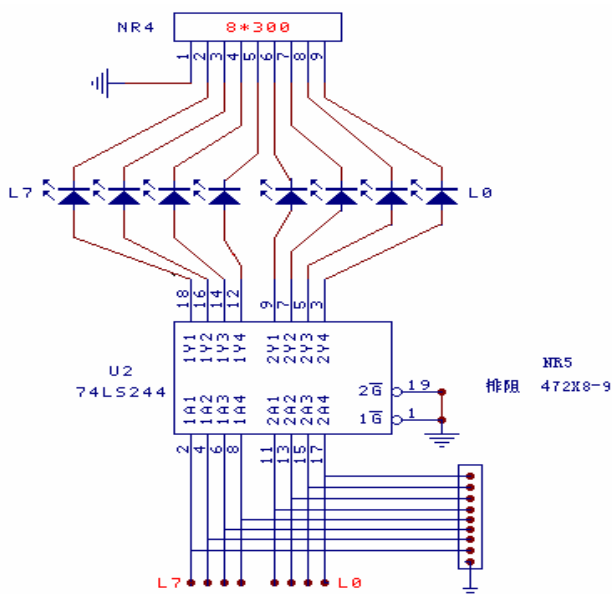


图2-18 发光二极管及驱动电路

### 5、LED显示电路

如图2-18所示，实验台上设有8个发光二极管及相关驱动电路（输入端L7~L0），当输入信号为“1”时发光，为“0”时灭。

### 6、七段数码管显示电路

实验台设有两个共阴极数码管及驱动电路，电路图如图2-19。段码输入端：a、b、c、d、e、f、g、dp，位码输入端：S0、S1。

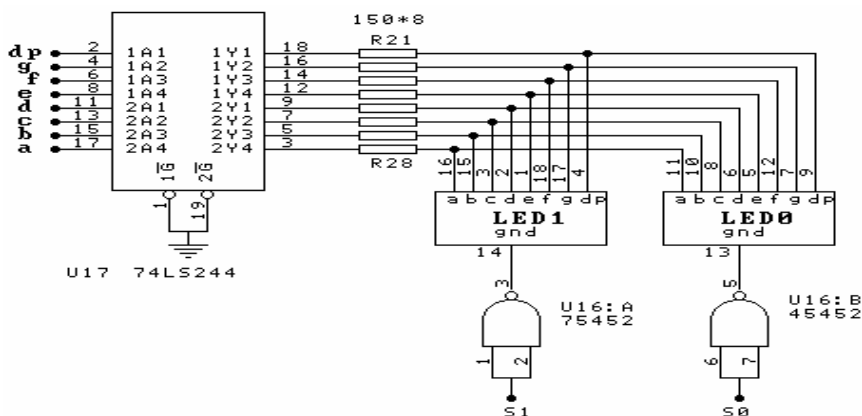


图2-19 数码管显示电路

## 7、单脉冲电路

如图2-20所示，采用RS触发器产生，实验者每按一次开关即可以从两个插座上分别输出一个正脉冲及负脉冲，供“中断”、“DMA”、“定时器/计数器”等实验使用。

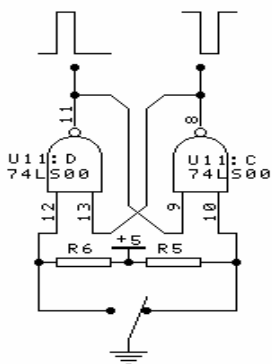


图2-20 单脉冲电路图

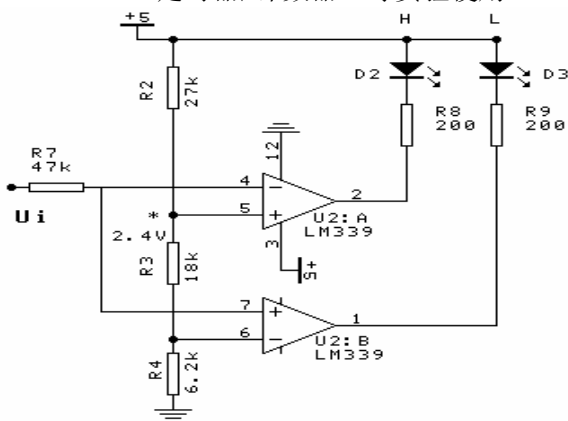


图22-21 逻辑笔电路

## 8、逻辑笔

如图2-21所示，当输入端Ui接高电平时红灯(H)亮，接低电平时绿灯(L)亮。

## 9、继电器及驱动电路

图2-22为直流继电器及相应驱动电路，当其开关量输入端“Ik”输入数字量“1”时，继电器动作，常开触点闭合红色发光二极管点亮。输入“0”时继电器常开触点断开发光二极灭。

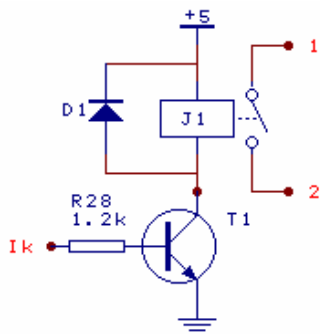


图2-22 继电器及驱动电路图

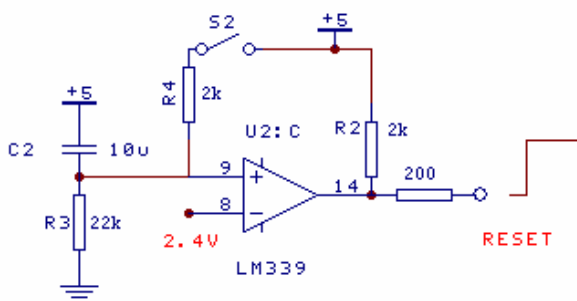


图2-23 复位电路



## 10、复位电路

图2-23为复位电路,实验台上有一复位电路,能在上电时,或按下复位开关RESET后,产生一个高电平的复位信号。

## 11、步进电机驱动电路

图2-24为步进电机的驱动电路,实验台上使用的步进电机驱动方式为二相励磁方式,BA、BB、BC、BD分别为四个线圈的驱动输入端,输入高电平时,相应线圈通电。DJ端为直流电机控制输入端。

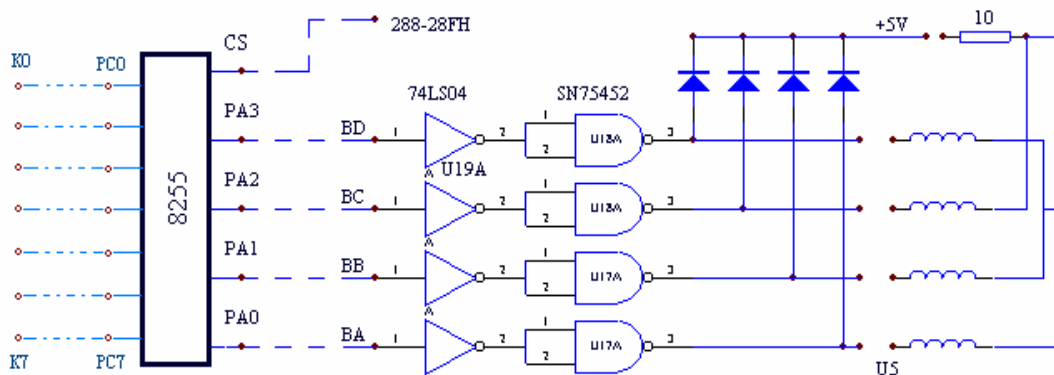


图2-24 步进电机驱动电路

## 12、接口集成电路

实验台上有微机原理及接口实验最常用接口电路芯片,包括:可编程定时器/计数器(8253)、可编程并行接口(8255)、数/模转换器(DAC0832)、模/数转换器(ADC0809),这里芯片与CPU相连的引线除去片选(CS)信号外都已连好,与外界连接的关键引脚在芯片周围用“自锁紧”插座引出,供实验使用。

## 13、逻辑门电路

实验台上设有几个逻辑门电路,包括“与门”、“或门”、“非门”、“触发器”供实验时选择使用。

### 2.2.3 用户扩展实验区

实验台上设有5个通用数字集成电路插座,其中“通用插座A”“通用插座B”为20芯,“通用插座D”为40芯活动插座以方便插拔器件。其余为14芯。插座的每个引脚都用自锁紧插孔引出。实验指导书中所列出的部分实验(简单并行接口、串行通信、集成电路测试等)电路就是利用这些插座搭试的。利用这些插座可以进行数字电路实验,也可以设计开发新的接口实验或让学生做课程设计、毕业设计等项目。

### 2.2.4 实验台跳线开关

为了方便实验,实验台上设有跳线开关,分以下几种:

1. 实验类型选择开关JB、JC:这两个跳线开关在在实验台的左上角,50线总线插座的左下方。在TPC-USB实验系统中不起作用,用户无须设置(出T均默认接在I/O)。

2. 模拟量输入选择开关JP2、JP3:在实验台ADC0809的左上角,分别用于模/数转换模拟量的输入极性选择,JP2的1、2两点短路时ADC0809的IN2可输入双极性电压( $-5V \sim +5V$ ),2、3两点短路时输入单极性电压( $0 \sim +5V$ )。JP3用于选择IN1的输入极性,选择方法与JP19

相同。

3. +5V电源插针：为减轻+5 V电源负载和各主要芯片的安全，在各主要实验电路附近都有相应的电源连接插针（标记为+5V），当实验需要该部分电路时，用短路子短接插针即可接通+5V电源. 对用不到的电路可将短路片拔掉确保芯片安全。

2.2.5 20芯双排插座

实验台上有一个20芯双排插座J7，用于外接附加的键盘显示实验板和其它用户开发的实验板。J7各引脚信号安排如下：

19	17	15	13	11	9	7	5	3	1
D0	D1	D2	D3	D4	D5	D6	D7	IRQ	CS
RES	+5V	+5V	IOR	IOW	A0	A1	CLK	GND	GND
20	18	16	14	12	10	8	6	4	2

在J7的附近有两个短路插针标有“CS”和“IRQ”。当“CS”的两点短接后，译码器的280H—287H连接到J7的CS端。当你扩展板上的实验需要中断信号时将“IRQ”的两端短接，不需要时应将其断开。

2.2.6 直流稳压电源

实验箱自备电源，安装在实验大板的下面，交流电源插座固定在实验箱的后测板上，交流电源开关在实验箱的右侧，交流电源开关自带指示灯，当开关打开时指示灯亮。在实验板右上角有一个直流电源开关，交流电源打开后再把直流开关拨到“ON”的位置，直流+5V +12V - 12V就加到实验电路上。

主要技术指标：    输入电压    AC 175—265V  
                         输出电压/电流+5V/2.5A    +12V/0.5A    -12V/0.5A  
                         输出功率    25W

## 第三章 TPC-USB集成软件开发环境

### 3.1 TPC-USB集成开发环境软件包

TPC-USB集成开发环境是TPC-USB实验系统所配套的软件。它提供了用户程序的编辑和编译，调试和运行，实验项目的查看和演示，实验项目的添加等功能，方便了学生和老师实验程序的编制和调试。本软件基于windows2000/XP/2003环境，界面简洁美观，功能齐全。集成开发环境主界面如图3-1：



图3-1 软件主界面

### 3.2 集成开发环境软件的安装

TPC-USB集成开发环境集编辑、编译、调试、实验演示等，功能齐全，使用简便，方便教师查看实验原理图及程序，利于教学。

安装步骤如下：

1. 从随机所带光盘目录中找到SETUP.EXE文件，鼠标双击该文件会出现以下界面，图3.2。（为确保安全，建议先将光盘文件拷贝到硬盘某文件夹中，在硬盘上运行SETUP.EXE文件）。



图3-2

2. 输入用户名、公司名和序列号后，点击“下一步”，会出现图3-3安装选择画面。

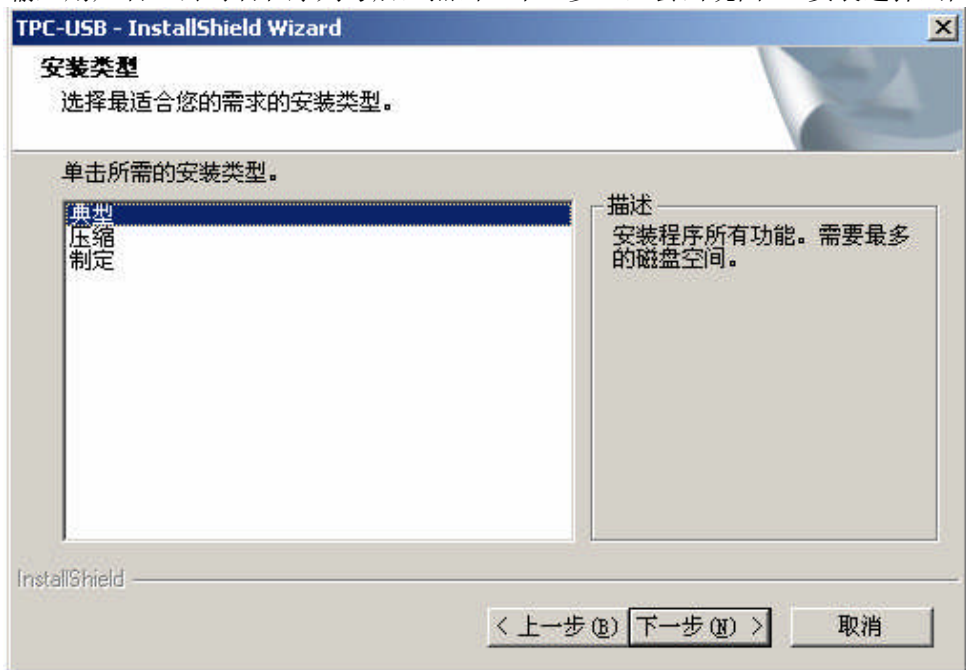


图3-3 安装选择

其中，“典型”安装是指安装主程序、实验演示程序和帮助文件。“压缩”是指只安装主程序和帮助文件，不安装实验演示程序。“制定”是在“主程序”、“实验演示”、“帮助文件”三个文件中选择你需要的安装。建议在教师实验机中选“典型”安装，在学生实验机中选“压缩”安装。

选择好以后，点击“下一步”，程序将自动将软件安装到你的机器上。

### 3.2.1 用户程序的编辑和编译

TPC-USB集成开发环境软件支持汇编程序(.asm文件)类型的程序开发。除了一般的编辑功能外，本软件还支持语法高亮显示，语法错误提示等功能，大大提高了程序的可读性。用户编辑好程序并保存后，即可方便地进行编译。

#### 1. 新建一个源程序

在当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“新建”，或是在工具栏中单击“新建”快捷按钮，会出现源程序编辑窗口，建议用“另存为”为文件取名保存后，就新建一个“.asm”文件。

#### 2. 打开一个源程序

当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“打开”，或是在工具栏中单击“打开”，会弹出“打开”文件选择窗口，“打开”窗口如图3-4所示：



图3-4 打开一个源程序

在窗口中“文件类型”下拉菜单中选择“ASM文档 (\*.asm)”一项，程序即显示当前目录下所有的asm文档，单击要选择的文件，选中的文件名会显示在“文件名”中，单击“打开”则打开当前选中的文档显示在文档显示区域。点击“取消”则取消新建源文件操作。

#### 3. 编辑源程序

本软件提供了基本的编辑功能，并实现了实时的语法高亮，各项操作说明如下：

##### 撤消

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“撤消”，或是在工具栏中单击“撤消”，即可撤消上一步剪切或粘贴操作。

##### 剪切

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“剪切”，或是在工具栏中单击“剪切”，即可将文档显示区域中选中的内容剪切到剪贴板。

### 复制

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“复制”，或是在工具栏中单击“复制”，即可将文档显示区域中选中的内容复制到剪贴板。

### 粘贴

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“粘贴”，或是在工具栏中单击“粘贴”，即可将剪贴板中当前内容粘贴到文档显示区域光标所在处。

### 全选

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“全选”，即可将文档区域中所有内容选中。

### 查找

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“查找”，弹出查找对话框如图3-5所示：

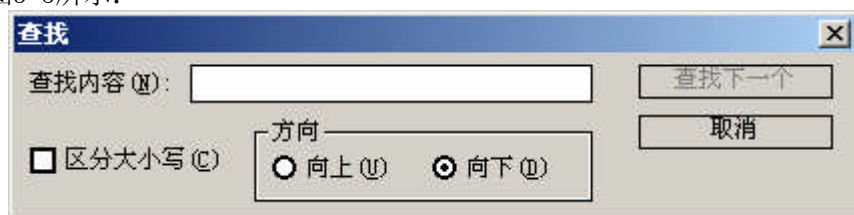


图3-5 查找

在查找内容一栏中输入需要查找的内容，可选择“区分大小写”的查找方式，单击“查找下一个”程序则在文档显示区域中搜索与查找内容匹配的字符串，找到第一个后则高亮显示，用户点击查找下一个则继续搜索下一个匹配字符串，点击“取消”退出查找操作。

### 替换

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“替换”，弹出替换对话框如图3-6所示：

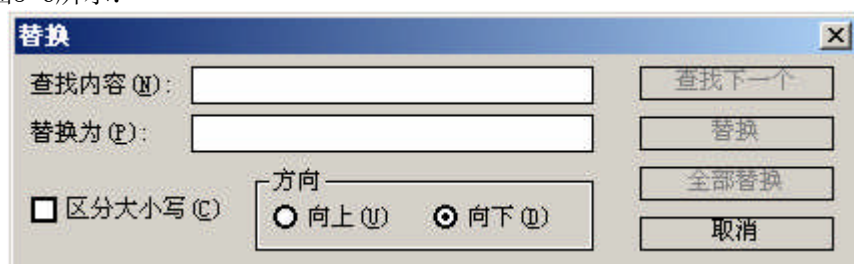


图3-6 查找下一个

在查找内容一栏中输入需要查找的内容，可选择“全字匹配”与“区分大小写”的查找方式，在替换为一栏中输入需要替换的内容，单击“查找下一个”程序则在文档显示区域中搜索与查找内容匹配的字符串，找到第一个后则高亮显示，用户可单击“替换”将匹配的字符串替换，也可单击“全部替换”将当前文档显示区域中所有与查找内容匹配的字符串全部替换。单击“查找下一个”则继续搜索下一个匹配字符串。也可单击“取消”退出查找操作。

#### 4. 保存源程序

当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“保存”，如果是无标题文档，用户需在提示下输入文档的名称及选择保存的路径，单击确定后保存；否则程序自动保存当前文档显示区域中显示的文档。或者选择菜单栏中的“文件”菜单，菜单下拉后选择“另存为”，并在提示下输入文档的名称及选择保存的路径，单击确定后保存。

### 3.2.2 编译源程序

#### 1. 编译调试窗口

在当前运行环境下，选择菜单栏中的“查看”菜单，单击编译调试窗口选项或是单击工具栏中“输出窗口”按钮则可对输出栏的进行显示。若当前环境显示编译调试窗口，则单击查看输出窗口选项即可隐藏该窗口，编译调试输出窗口选项即消失；若当前隐藏编译调试窗口，则单击输出窗口选项即可显示该窗口，编译调试窗口将显示。

#### 2. ASM编译

##### 汇编+链接

在当前运行环境下，选择菜单栏中的“项目”菜单，选择汇编+链接选项则程序对当前ASM源文件进行汇编与链接，编译调试窗口中输出汇编与链接的结果，若程序汇编或链接有错，则详细报告错误信息。双击输出错误，集成开发环境会自动将错误所在行代码高亮显示。

##### 开始+执行

在当前运行环境下，选择菜单栏中的“项目”菜单，选择开始+执行选项则程序对当前ASM源文件执行，程序自动运行。

### 3.2.3 用户程序的调试和运行

#### 1. ASM程序的调试

##### 寄存器窗口

在当前运行环境下，选择工作区的“寄存器”菜单，寄存器窗口即可显示。寄存器窗口中显示主要的寄存器名称及其在当前程序中的对应值，若值为红色，即表示当前寄存器的值。调试时，单步执行，寄存器会随每次单步运行改变其输出值，同样以红色显示。

##### 开始调试

在“选项”菜单中，“编译选项”选择“调试”，然后进行进行程序的编译和链接，编译和链接成功之后，调试工具将会显示，也可以在“项目”中选择“开始/结束调试”。即可开始进行程序的调试。编译选项选择如图3-7：

在ASM程序正常链接之后，选择菜单栏中的“开始/结束调试”菜单，选择开始调试选项，则对源程序进行反汇编，进入ASM的调试状态，并在寄存器窗口中显示主要的寄存器的当前值。

##### 设置/清除断点

在ASM的调试状态下，对程序代码所在某一行前的灰色列条单击鼠标，即对此行前设置了断点，如果清除断点，只需再在此行前的灰色列条上的断点单击鼠标，此断点标记将被清除。黄色箭头所指的行为当前单步执行到的所在行。设置/清除断点如图3-8所示：



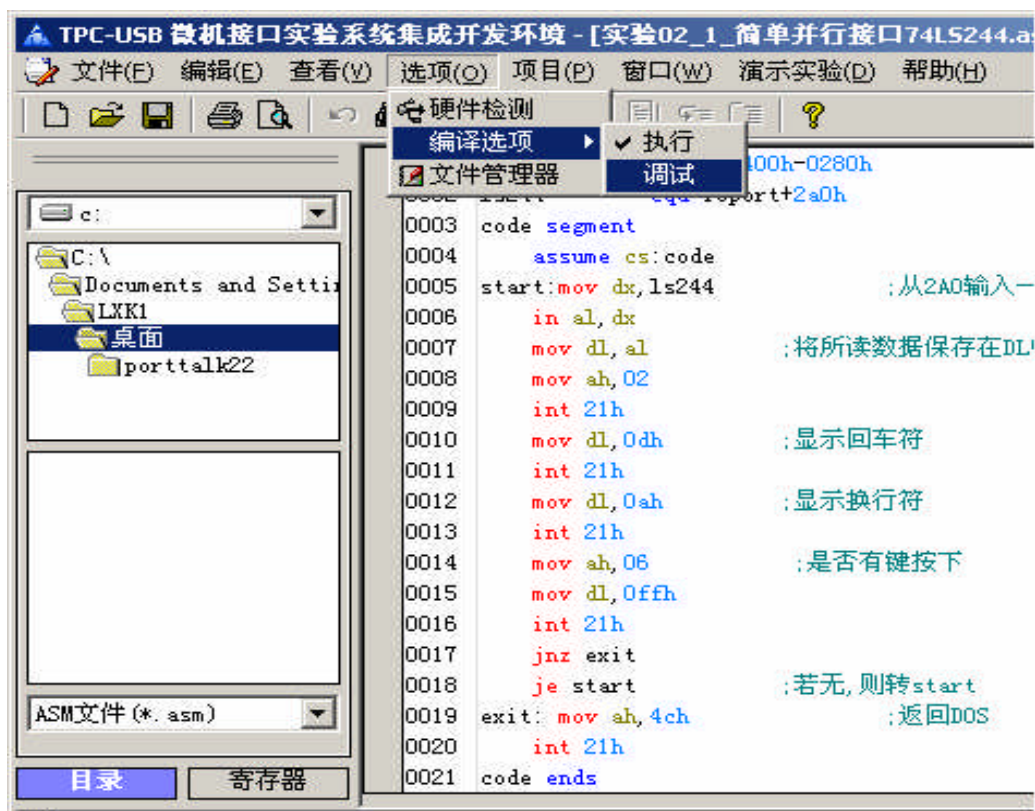


图3-7 编译选项的选择

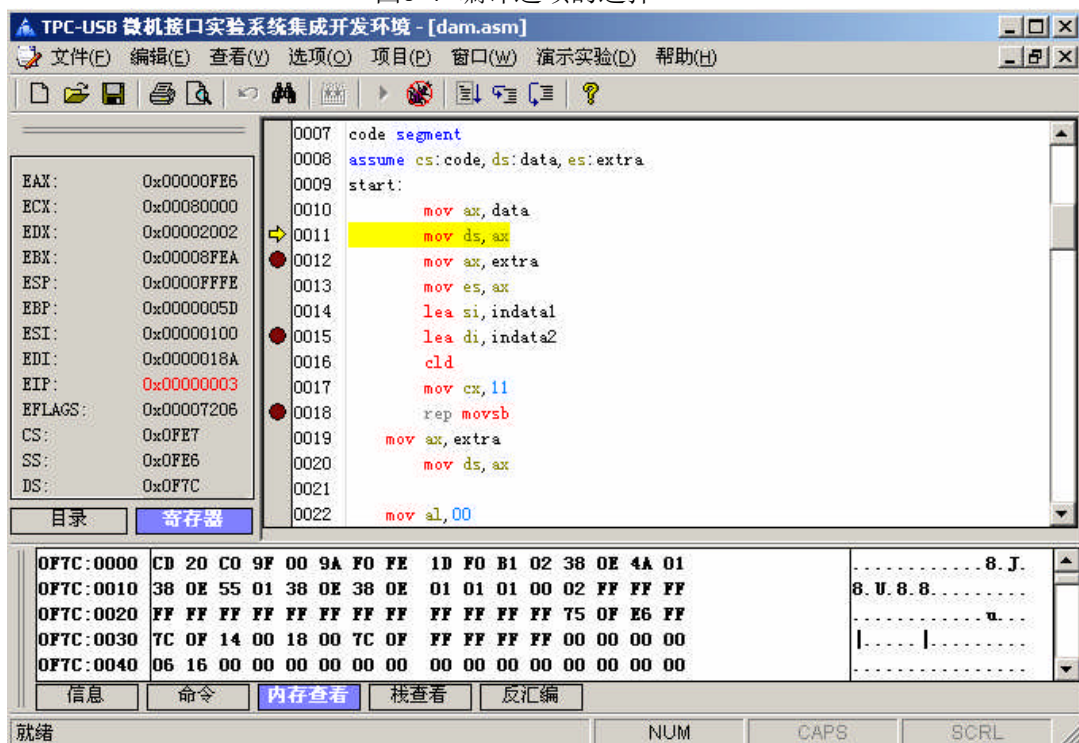


图3-8 设置/清除断点



## 连续运行

在ASM的调试状态下，选择“项目”菜单栏中的“连续运行”菜单或F5，则程序连续运行，直至碰到断点或程序运行结束。

## 单步

在ASM的调试状态下，选择“项目”菜单栏中的“单步执行”菜单或F11，则程序往后运行一条语句。

## 退出调试

在ASM的调试状态下，选择“项目”菜单栏中的“开始/结束调试”菜单，程序则退出ASM的调试状态。

## 命令调试

集成开发环境可以进行命令的调试，如图3-9：

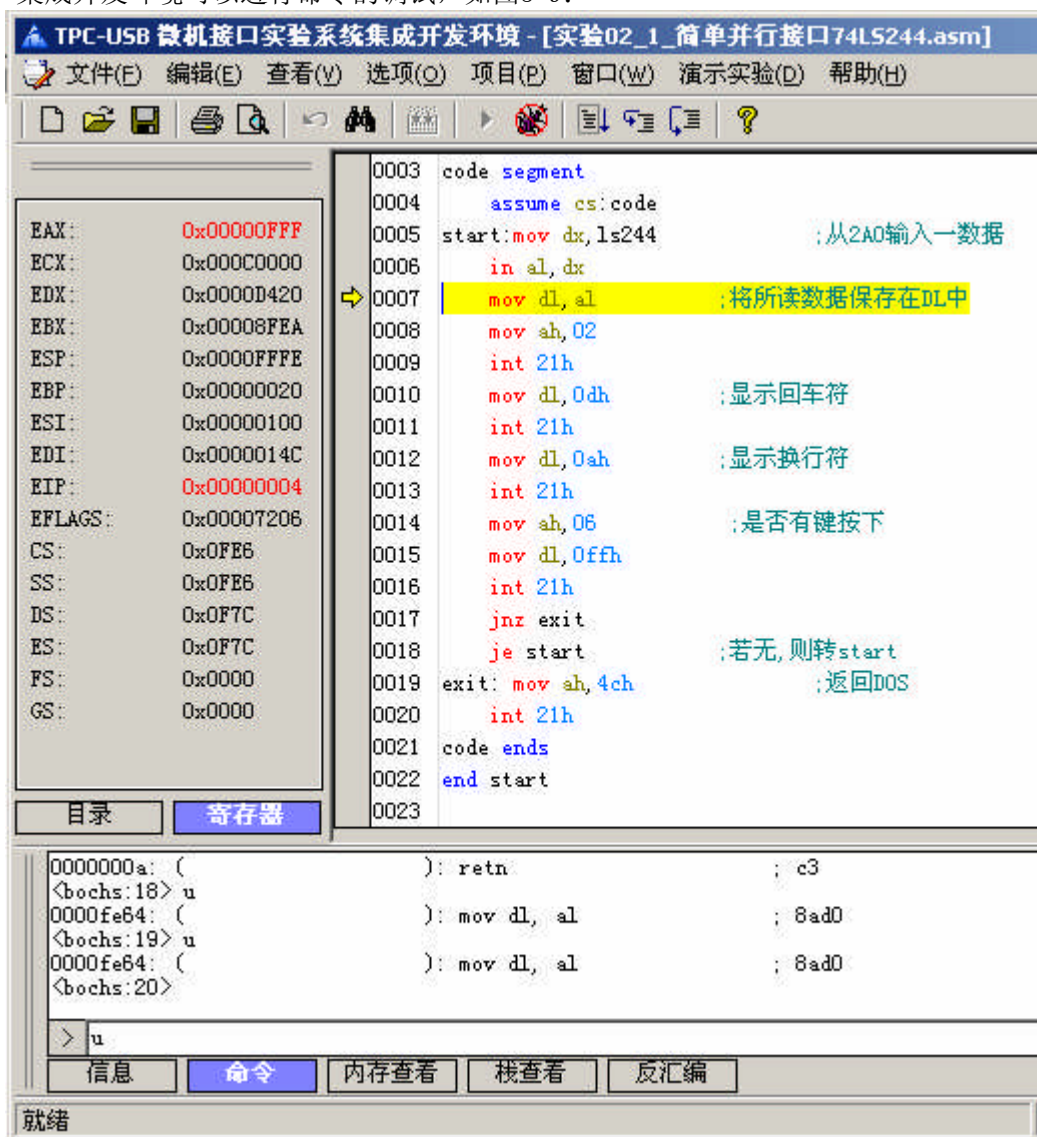


图3-9 命令调试

调试时，输出窗口可以输出编译信息，命令信息，内存查看信息，栈查看信息等。如图3-10：

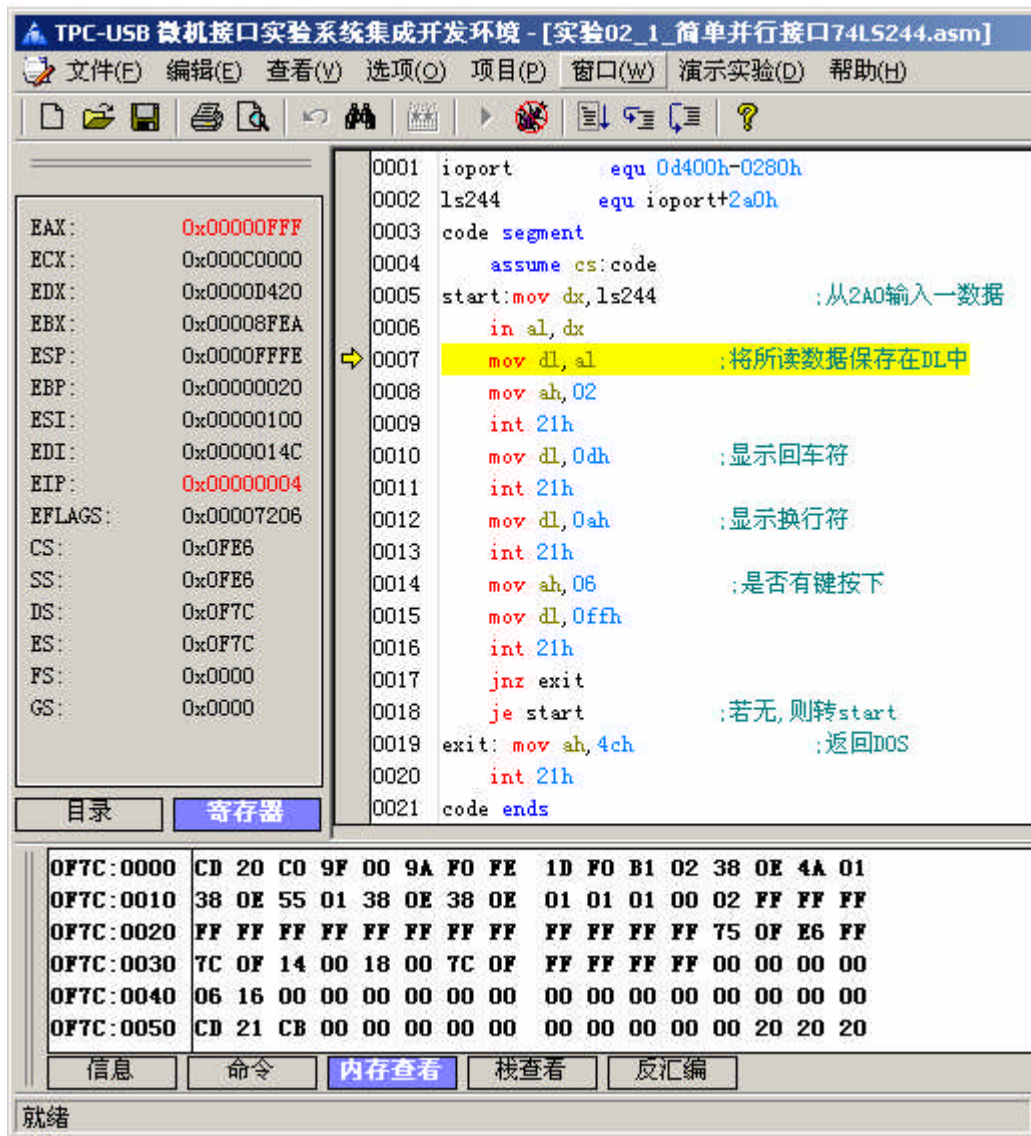


图3-10 内存查看输出窗口

### 3.2.4 常用调试命令

调试指令与debug稍有区别，具体调试命令如下：

bochs提供了强大的命令行调试功能，本集成开发环境在其之上包装了一个简便易用的图形界面。如果这个界面不能满足您的要求，还可以使用命令栏直接输入调试命令与bochs交互。所有调试命令bochs都提供了简要的用法说明，输入“help”（不带引号）可查看可用的命令，help 'cmd'（带引号）可查看命令cmd相关的帮助。

下面是一些常用的命令说明及示例：

#### 1. 反汇编 (u)

用法: u [/count] start end

反汇编给定的线性地址，可选参数'count'是反汇编指令的条数

例: u 反汇编当前 cs:ip 所指向的指令

u /10 从当前 cs:ip 所指向的指令起, 反汇编10条指令

u /12 0xfeff 反汇编线性地址 0xfeff 处开始的12条指令

## 2. 查看内存 (x)

用法: x /nuf addr

查看线性地址'addr'处的内存内容

nuf 由需要显示的值个数和格式标识[xduot cbhw m]组成, 未指明用何种格式的情况下将使用上一次的格式。

x: 十六进制

c: 字符

d: 十进制

b: 字节

u: 无符号

h: 半字

o: 八进制

w: 字 (四字节)

t: 二进制

m: 使用memory dump模式

例: x /10wx 0x234 以十六进制输出位于线性地址 0x234 处的 10 个双字

x /10bc 0x234 以字符形式输出位于线性地址 0x234 处的 10 个字节

x /h 0x234 以十六进制输出线性地址 0x234 处的 1 个字

## 3. 查看寄存器 (info reg)

用法: info reg

查看CPU整数寄存器的内容

## 4. 修改寄存器 (r)

用法: r reg = expression

reg 为通用寄存器

expression 为算术表达式

例: r eax = 0x12345678 对 eax 赋值 0x12345678

r ax = 0x1234 对 ax 赋值 0x1234

r al = 0x12 + 1 对 al 赋值 0x13

## 5. 下断点 (lb)

用法: lb addr

下线性地址断点

例: lb 0xfeff 在 0xfeff 下线性地址断点, 0f00:eff 所处线性地址就是 0xfeff

## 6. 查看断点情况 (info b)

用法: info b

## 7. 删断点 (del n)

用法: del n

删除第 n 号断点

例: del 2 删除 2 号断点, 断点编号可通过前一个命令查看

## 8. 连续运行 (c)

用法: c

在未遇到断点或是 watchpoint 时将连续运行

## 9. 单步 (n 和 s)

用法: n

执行当前指令, 并停在紧接着的下一条指令。如果当前指令是 call、ret, 则相当于 Step Over。

s [count]

执行 count 条指令

## 10. 退出 (q)

用法: q

## 2. C语言程序的调试

大多数实验所用的程序需要用到配套的Visual Studio生成的静态链接库(.lib)或动态链接库(.dll)文件, 因此本软件采用了Visual C++的调试系统。由于版权问题, 本软件没有提供Visual C++的编译和调试器, 需要用户自己安装。

### 3.2.5 实验项目的查看和演示

本软件提供了实验项目的查看和演示功能, 包括实验说明、实验原理图、实验流程图、ASM程序, 并可以运行实验程序, 使用户能方便快捷地了解感兴趣的实验。示例如图3-11:

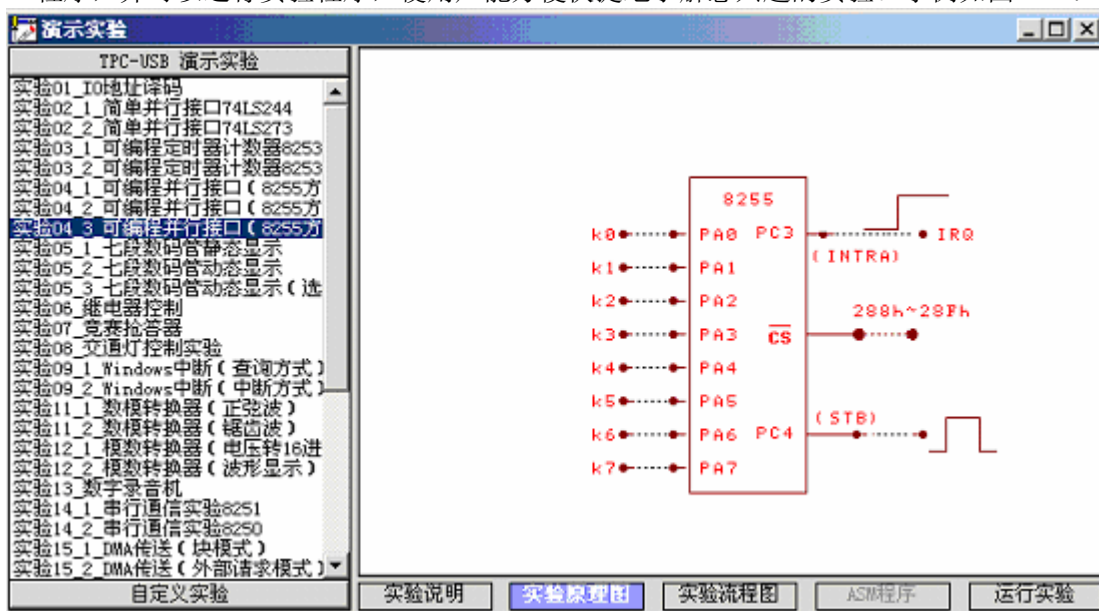


图3-11实验项目的查看和演示

各实验有几个子项, 包括实验说明、实验原理图、实验流程图、ASM程序和运行实验。单击对应子项, 即可查看对应的项目。

实验说明

双击实验说明子项, 即可弹出对应实验的实验说明。

实验原理图

双击实验原理图子项, 即可弹出对应实验的实验原理图。

实验流程图

双击实验流程图子项, 即可弹出对应实验的实验流程图。

ASM程序

双击ASM程序子项，即可弹出对应实验的ASM程序源文件。

运行实验

双击运行实验子项，即可执行对应实验的可执行程序。

### 3.2.6 实验项目的添加和删除

除预定义的26个常用实验外，本软件还支持自定义实验，方便用户扩展实验内容。用户可以自行添加实验，被添加的实验将作为“自定义实验”的子类，之后便能在演示实验中进行查看，查看方式和预定义实验相同。

#### 1. 添加实验

在当前运行环境下，选择菜单栏“演示实验”菜单选项，在下层目录中选择自定义实验选项，则出现TPC-USB自定义实验对话框，如图3-12所示：

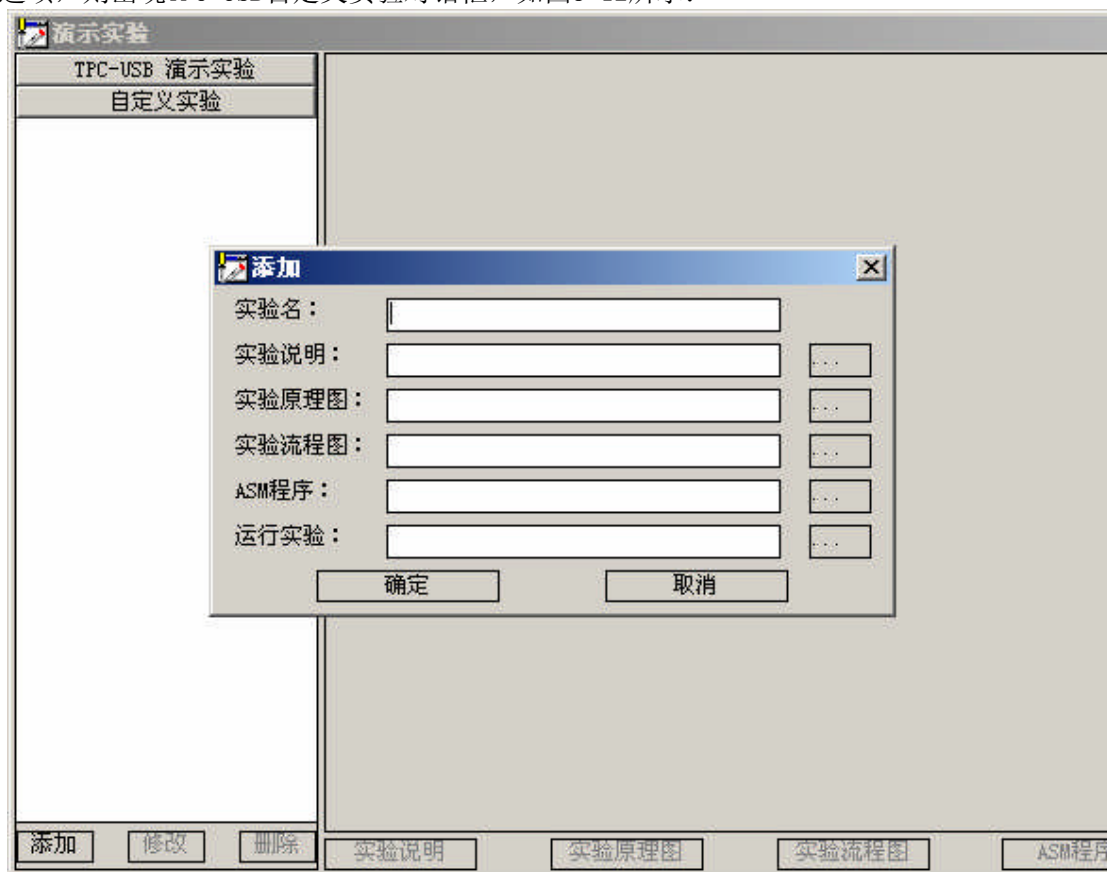


图3-12 自定义实验

用户可以对自定义实验进行添加和删除操作。点击添加实验按钮，则弹出添加实验对话框，如图3-13：

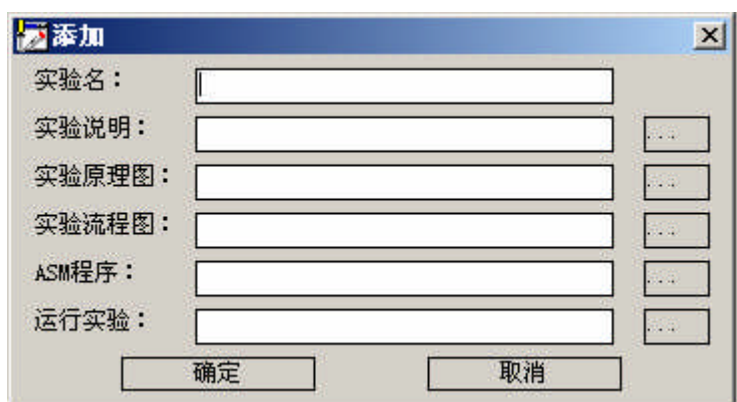


图3-13 添加实验

用户可以直接输入目标文件地址或是通过右侧的浏览按钮来选择文件，需要注意的是，添加实验时实验名称和可执行程序是必不可少的。

## 2. 删除实验

自定义实验是可以删除的。在当前运行环境下，选择菜单栏中的“演示实验”菜单，选择自定义实验选项，在自定义实验对话框中选定待删除的实验，点击删除实验按钮，则弹出确认对话框，确认后选定的实验将被删除，否则取消删除操作。如图3-14所示：

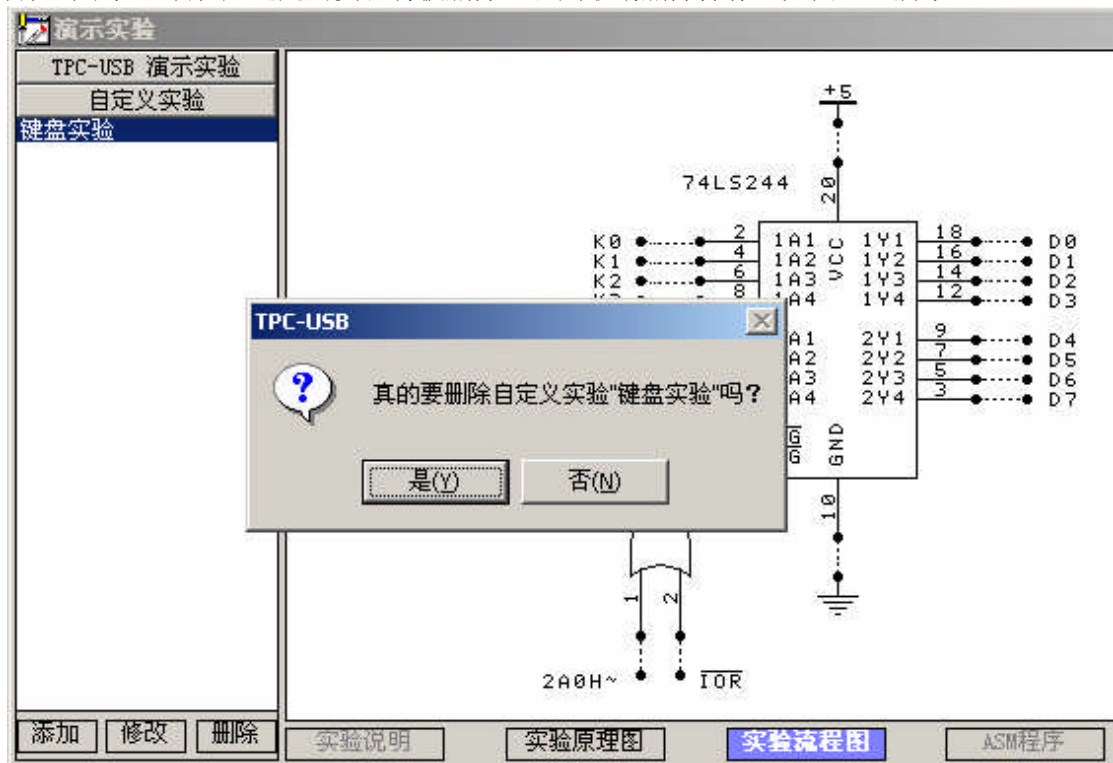


图3-14 实验删除

## 3.2.7 集成开发环境帮助菜单

集成开发环境帮助菜单方便教师和学生软件使用，芯片查寻，常用命令查询等。分别如图3-14，图3-15，图1-16：



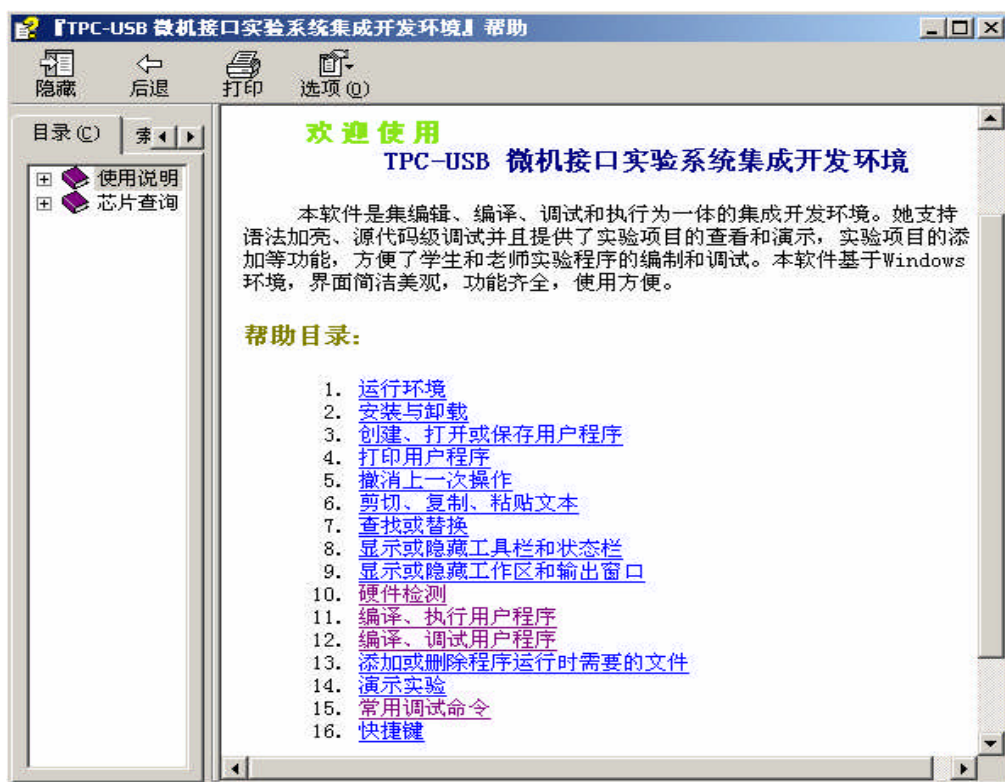


图3-14 集成开发环境帮助

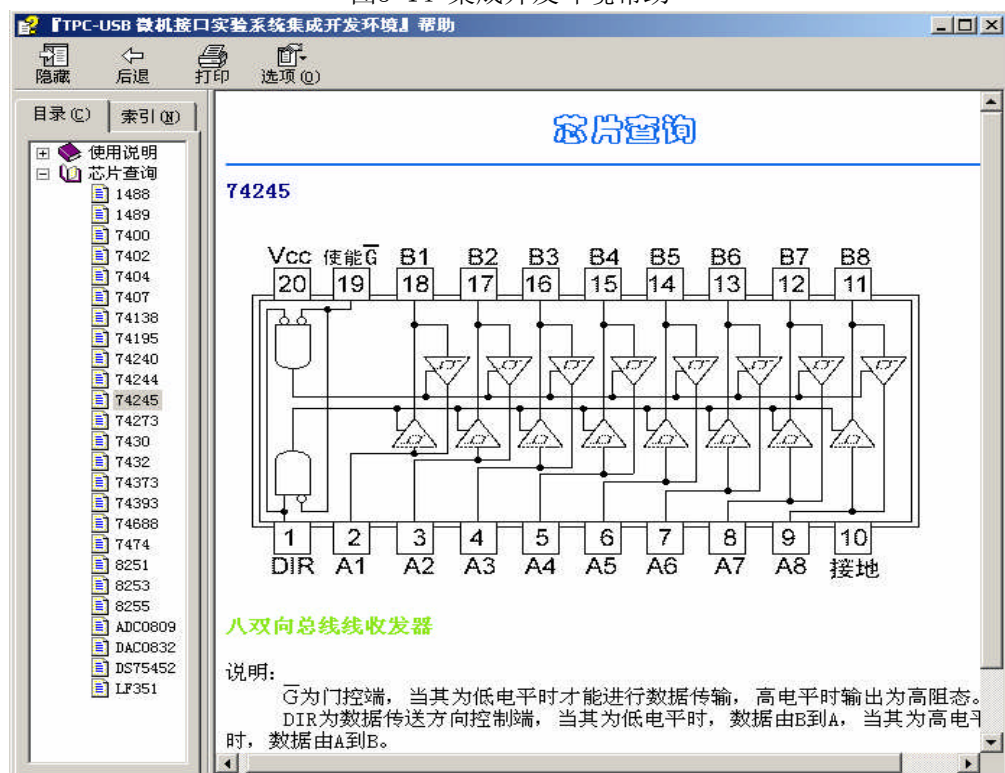


图3-15 常用芯片查询

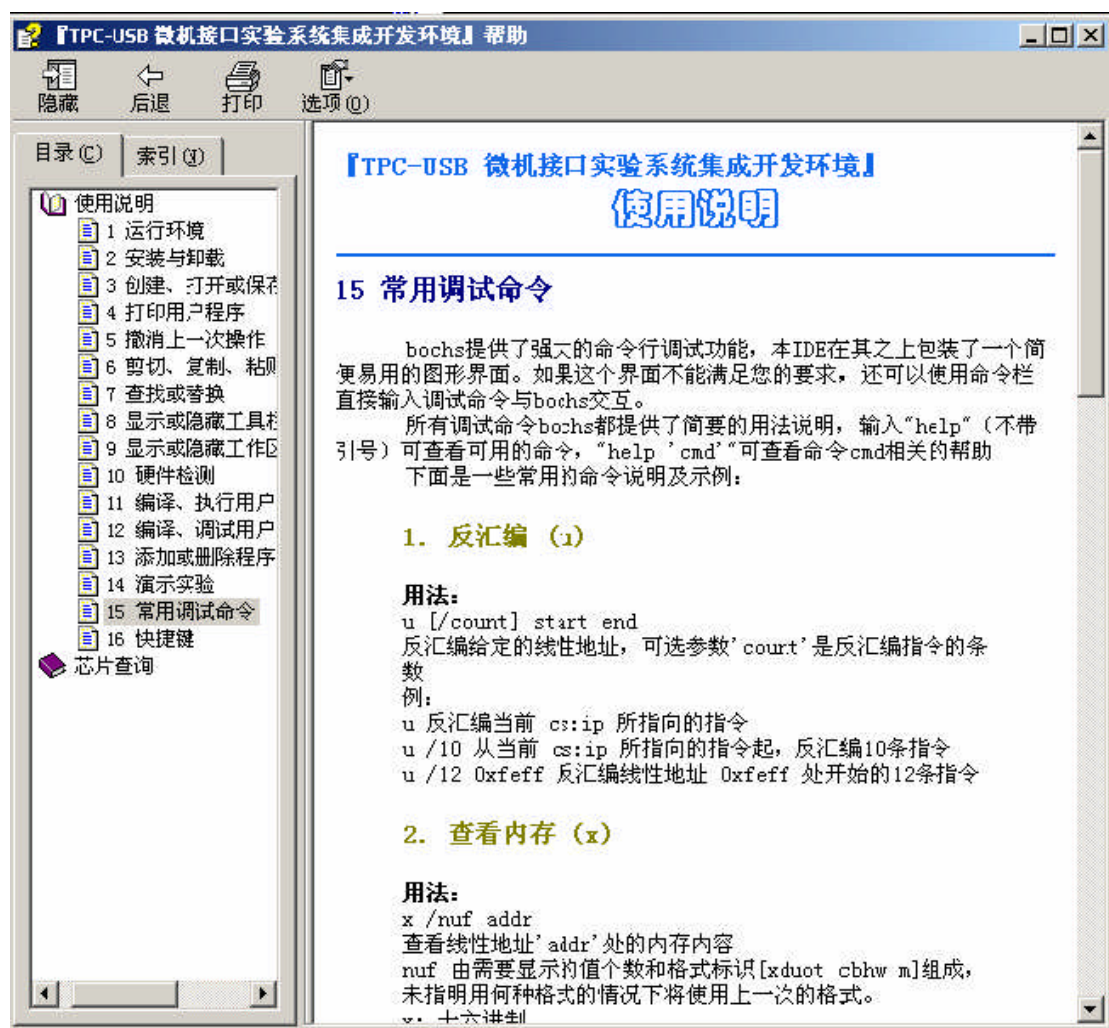


图3-16 常用调试命令



## 第四章 汇编实验部分

几点约定：

- 1、实验电路介绍中凡不加“利用通用插座”说明的均为实验台上已固定电路。
- 2、实验电路连线在图中均用虚线表示，实线为已连好电路。
- 3、所有实验程序中均加有延时子程序，请老师根据实验PC机情况，适当加、减延时程序的延时时间即可。

### 实验一 I/O地址译码

#### 一、实验目的

掌握I/O地址译码电路的工作原理。

#### 二、实验原理和内容

实验电路如图1-1所示，其中74LS74为D触发器，可直接使用实验台上数字电路实验区的D触发器，74LS138为地址译码器。译码输出端Y0~Y7在实验台上“I/O地址”输出端引出，每个输出端包含8个地址，Y0：280H~287H，Y1：288H~28FH，…… 当CPU执行I/O指令且地址在280H~2BFH范围内，译码器选中，必有一根译码线输出负脉冲。

例如：执行下面两条指令

MOV DX, 2A0H

OUT DX, AL (或IN AL, DX)

Y4输出一个负脉冲，执行下面两条指令

MOV DX, 2A8H

OUT DX, AL (或IN AL, DX)

Y5输出一个负脉冲。

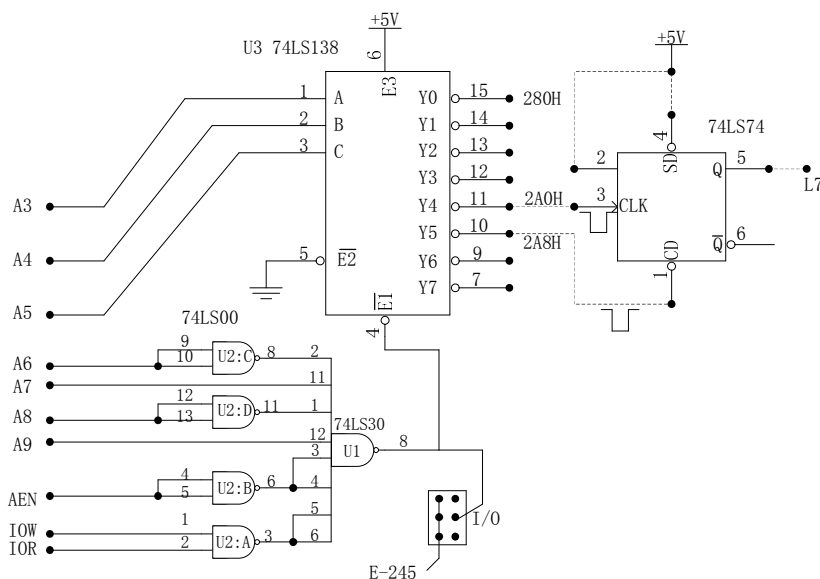


图1-1

利用这个负脉冲控制L7闪烁发光（亮、灭、亮、灭、……），时间间隔通过软件延时实现。

### 三、编程提示

1、实验电路中D触发器CLK端输入脉冲时，上升沿使Q端输出高电平L7发光，CD端加低电平L7灭。

## 实验二 简单并行接口

### 一、实验目的

掌握简单并行接口的工作原理及使用方法。

### 二、实验内容

1、按下面图2-1简单并行输出接口电路图连接线路(74LS273插通用插座，74LS32用实验台上的“或门”)。74LS273为八D触发器，8个D输入端分别接数据总线D0~D7，8个Q输出端接LED显示电路L0~L7。

2、编程从键盘输入一个字符或数字，将其ASC II 码通过这个输出接口输出，根据8个发光二极管发光情况验证正确性。

3、按下面图2-2简单并行输入接口电路图连接电路(74LS244插通用插座，74LS32用实验台上的“或门”)。74LS244为八缓冲器，8个数据输入端分别接逻辑电平开关输出K0~K7，8个数据输出端分别接数据总线D0~D7。

4、用逻辑电平开关预置某个字母的ASC II 码，编程输入这个ASC II 码，并将其对应字母在屏幕上显示出来。

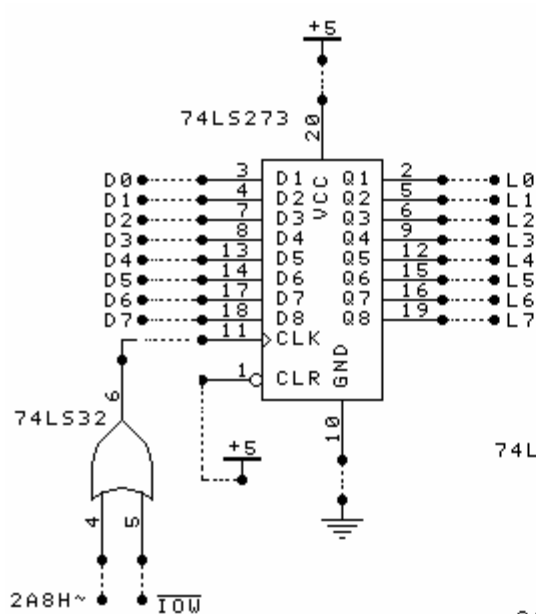


图2-1

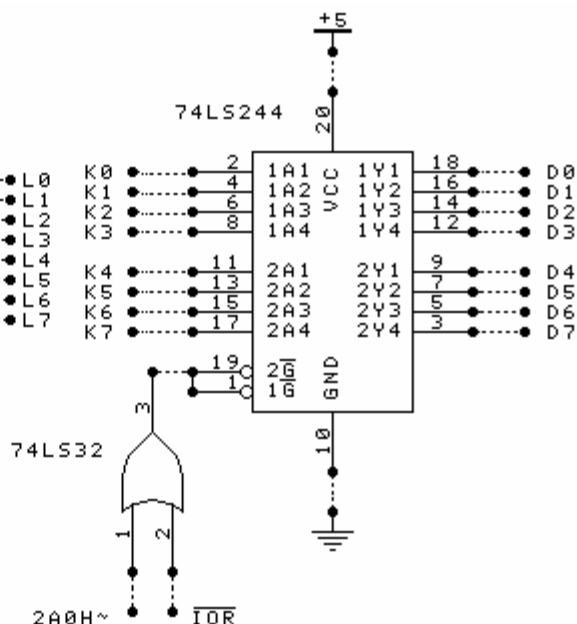


图2-2

### 三、编程提示

1、上述并行输出接口的地址为2A8H，并行输入接口的地址为2A0H，通过上述并行接口电路输出数据需要3条指令：

```
MOV    AL, 数据
MOV    DX, 2A8H
OUT    DX, AL
```

通过上述并行接口输入数据需要2条指令：

```
MOV    DX, 2ADH
IN     AL, DX
```

### 2、参考流程图

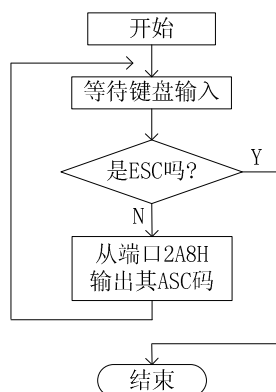


图2-3 参考程序1

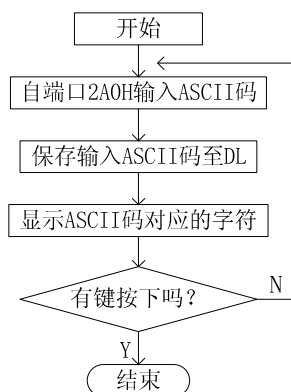


图2-4 参考程序2

### 实验三 可编程定时器 / 计数器 (8253)

#### 一、实验目的

掌握8253的基本工作原理和编程方法，用示波器观察不同方式下的波形。

#### 二、实验内容

- 1、按图3-1虚线连接电路，将计数器0设置为方式0，计数器初值为N( $N \leq 0FH$ )，用手动逐个输入单脉冲，编程使计数值在屏幕上显示，并同时用逻辑笔观察OUT0电平变化(当输入N+1个脉冲后OUT0变高电平)。

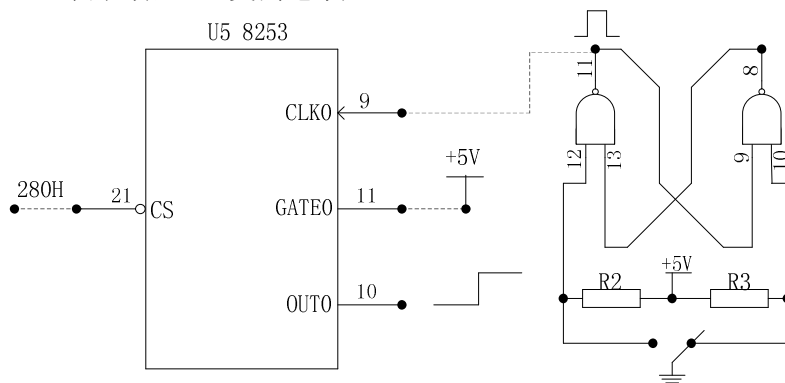


图3-1

- 2、按图3-2连接电路，将计数器0、计数器1分别设置为方式3，计数初值设为1000，用逻辑笔观察OUT1输出电平的变化(频率1HZ)。

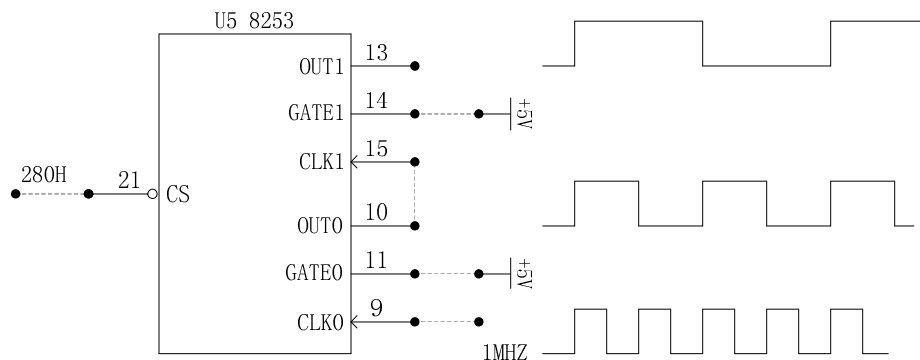


图3-2

- 3、按图3-3连接电路，将计数器0设置为方式3(方波)，计数器设置为方式2(分频)。实现计数器0的输出为方波，计数器1的输出是计数器0输出的分频波形。人机交互界面设计：实现在显示屏上提示输入计数器0(方波)的参数和计数器1(分频信号)的参数。如下所示：

```
counter1:____  
counter2:____  
continue?(y/n)____
```

实现用键盘直接输入修改程序中方波参数和分频信号参数，以改变方波的宽度，分频信号的周期和分频数，不需重新修改源代码。用示波器观察计数器0和计数器1的输出波形及其关系，并在纸上画出CLK0、OUT0、OUT1的波形。

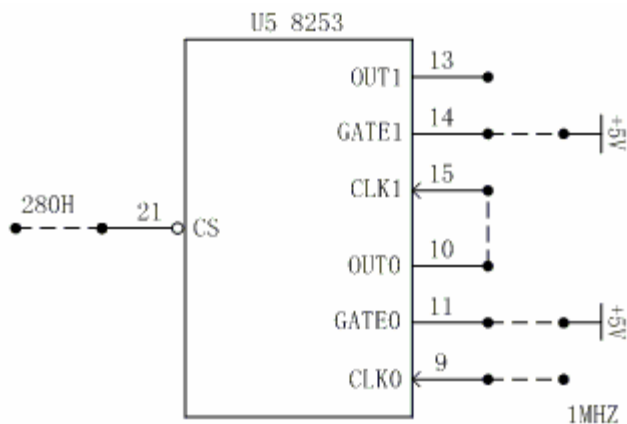


图3-3

### 三、编程提示

- |               |      |
|---------------|------|
| 1、8253控制寄存器地址 | 283H |
| 计数器0地址        | 280H |
| 计数器1地址        | 281H |
| CLK0连接时钟      | 1MHZ |

2、参考流程图(见图3-4、图3-5、图3-6)：

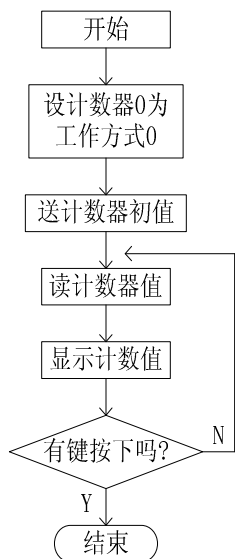


图3-4

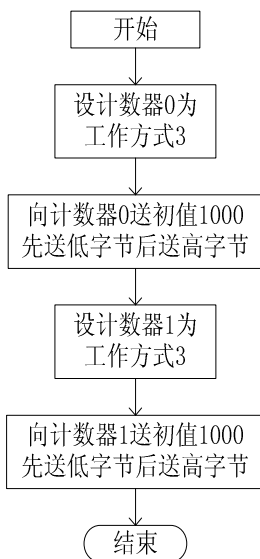


图3-5

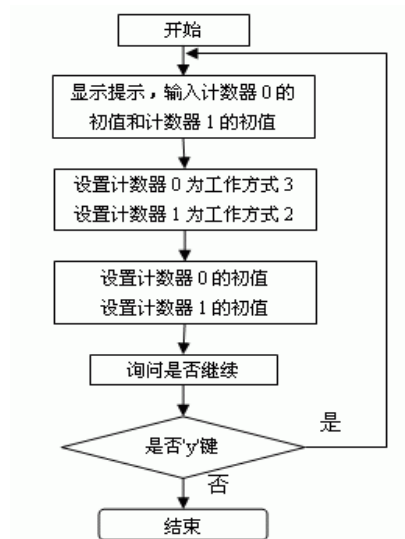


图3-6

## 实验四 可编程并行接口（一）（8255方式0）

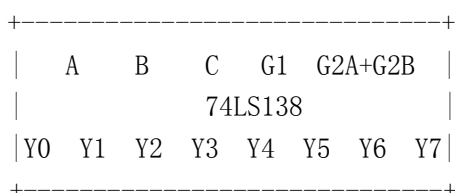
### 一、实验目的

- 1、通过实验，掌握8255工作于方式0以及设置A口为输出口，C口为输入口的方法。
- 2、掌握8255三个数据端口与被测IC芯片的硬件连接方法。
- 3、通过实验掌握用8255并行口模拟集成电路测试仪，对集成电路进行逻辑测试的方法。

### 二、实验内容

- 1、实验电路如图4-1，8255C口接逻辑电平开关K0~K7，A口接LED显示电路L0~L7。
- 2、编程从8255C口输入数据，再从A口输出。
- 3、按图4-2连接硬件电路，以测试3-8译码器74LS138为例。

人机交互界面设计：按照集成电路74LS138芯片的引脚图显示在屏幕上，如下所示：



Test Again ?(Y/N)', '\$'

原理图为图4-3，流程图为4-4：

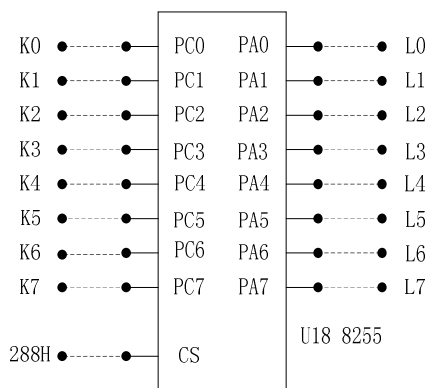


图4-1

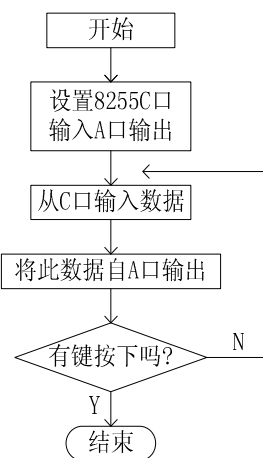


图4-2

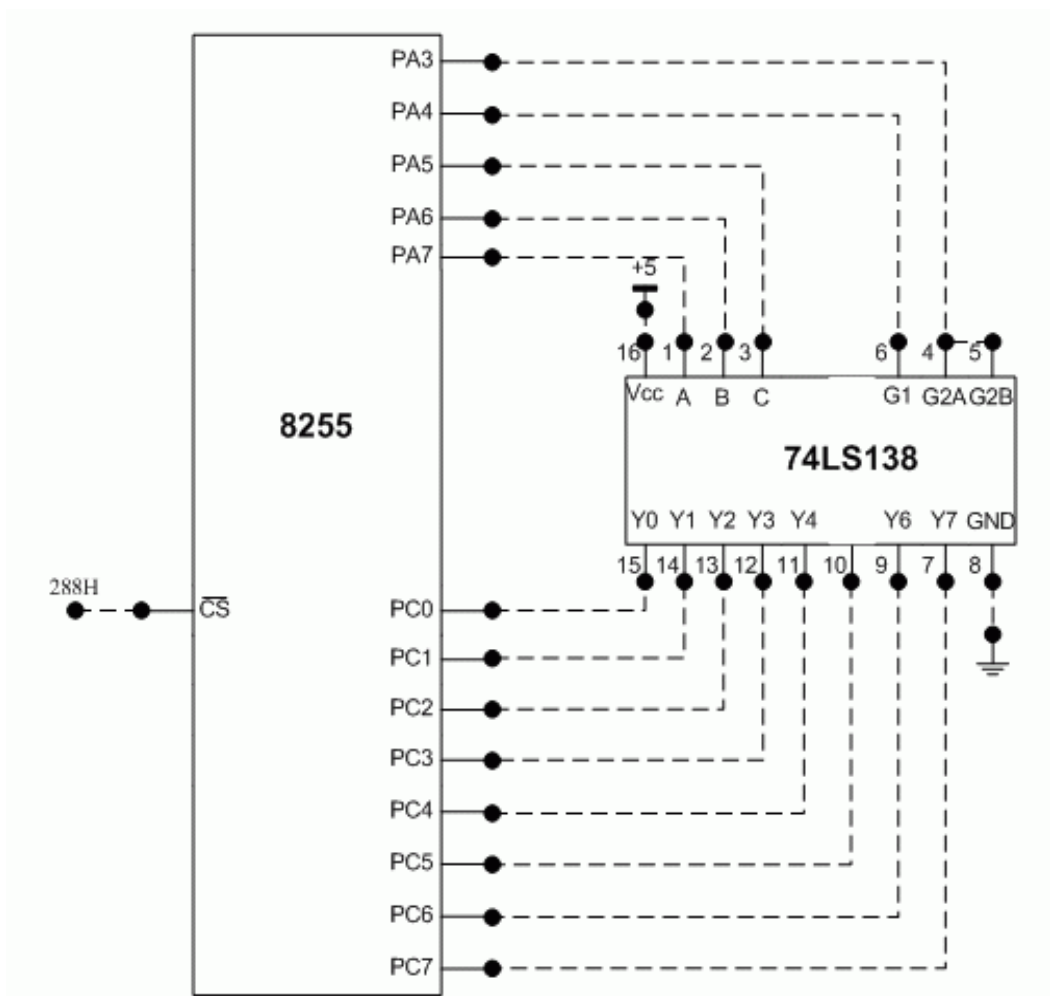


图4-3

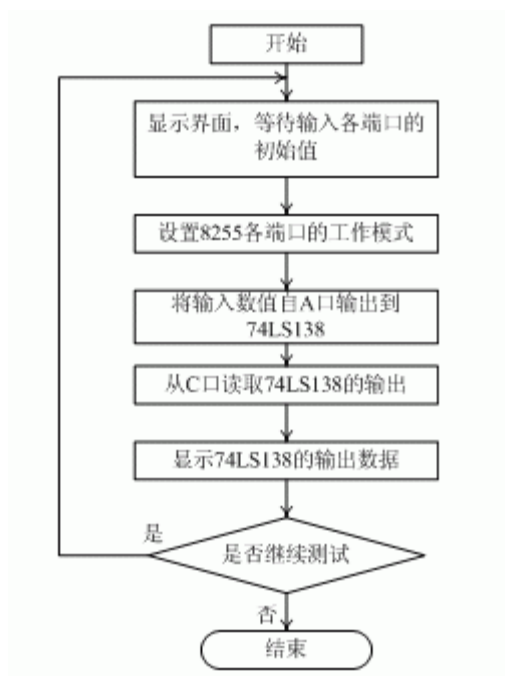


图4-4

### 三、编程提示

- 1、8255控制寄存器端口地址           28BH  
     A口的地址                       288H  
     C口的地址                       28AH
- 2、参考流程图(见图4-2、图4-4):



## 实验五 七段数码管

### 一、实验目的

掌握数码管显示数字的原理

### 二、实验内容

- 1、静态显示:按图5-1连接好电路,将8255的A口PA0~PA6分别与七段数码管的段码驱动输入端a~g相连,位码驱动输入端S1接+5V(选中),S0、dp接地(关闭)。编程从键盘输入一位十进制数字(0~9),在七段数码管上显示出来。

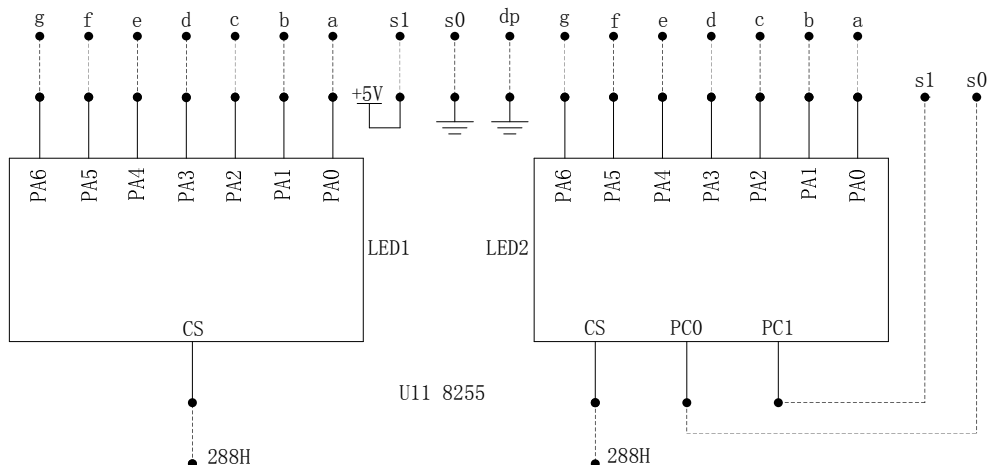


图5-1

图5-2

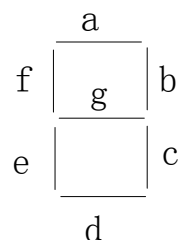
- 2、动态显示:按图5-2连接好电路,七段数码管段码连接不变,位码驱动输入端S1, S0接8255 C口的PC1, PC0。编程在两个数码管上显示“56”。
- 3、动态显示(选作):使用图5-2的电路,编程在两个数码管上循环显示“00-99”。

### 三、编程提示

1、实验台上的七段数码管为共阴型,段码采用同相驱动,输入端加高电平,选中的数码管亮,位码加反相驱动器,位码输入端高电平选中。

2、七段数码管的字型代码表如下表:

显示字形	g	e	f	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh



### 3、参考流程图(见图5-3、图5-4)

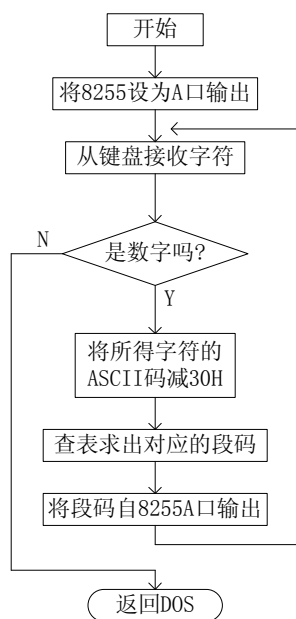


图5-3

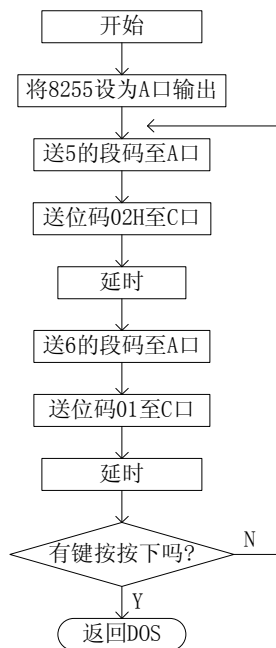


图5-4

## 实验六 继电器控制

### 一、实验目的

- 1、了解微机控制直流继电器的一般方法。
- 2、进一步熟悉使用8255、8253。

### 二、实验内容

实验电路如图6-1，按虚线连接电路：CLK0接1MHZ，GATE0，GATE1，接+5V，OUT0接CLK1，OUT1接PA0，PC0接继电器驱动电路的开关输入端Ik。编程使用8253定时，让继电器周而复始的闭合5秒钟(指示灯灯亮)，断开5秒钟(指示灯灯灭)。

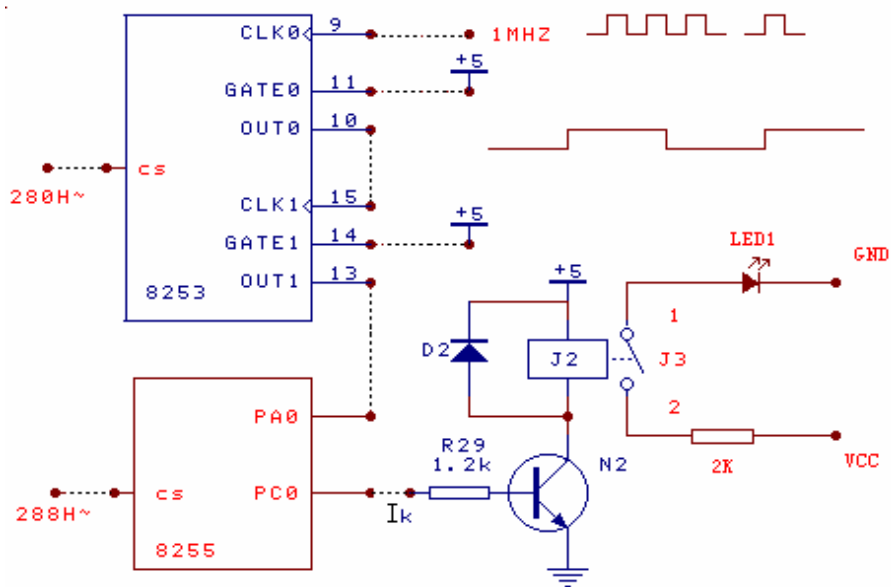


图6-1

### 三、编程提示

1、将8253计数器0设置为方式3、计数器1设置为方式0并联使用，CLK0接1MHZ时钟，设置两个计数器的初值(乘积为5000000)启动计数器工作后，经过5秒钟OUT1输出高电平。通过8255A口查询OUT1的输出电平，用C口PC0输出开关量控制继电器动作。

2、继电器开关量输入端输入“1”时，继电器常开触点闭合，电路接通，指示灯发亮，输入“0”时断开，指示灯熄灭。

3、参考流程图(见图6-2)：

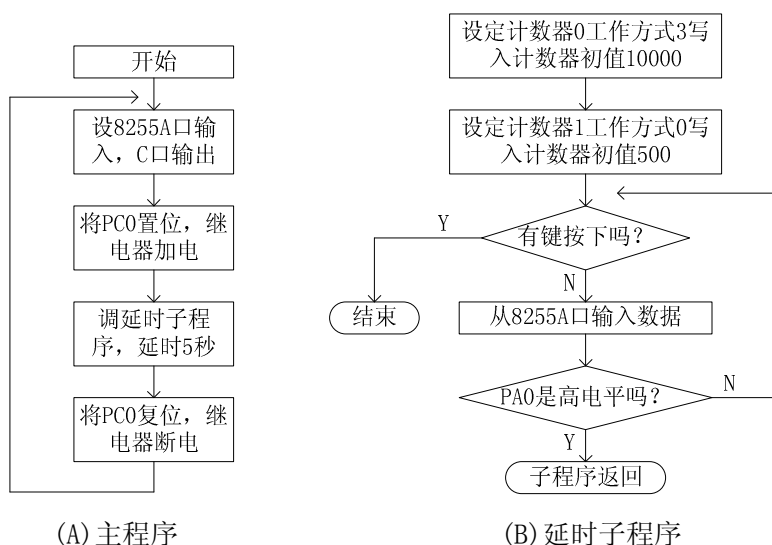


图6-2

## 实验七 竞赛抢答器

## 一、实验目的

- 1、了解微机化竞赛抢答器的基本原理。
- 2、进一步学习使用并行接口。

## 二、实验内容

图7-1为竞赛抢答器(模拟)的原理图,逻辑开关K0~K7代表竞赛抢答按钮0~7号,当某个逻辑电平开关置“1”时,相当某组抢答按钮按下。在七段数码管上将其组号(0~7)显示出来,并使微机扬声器响一下。

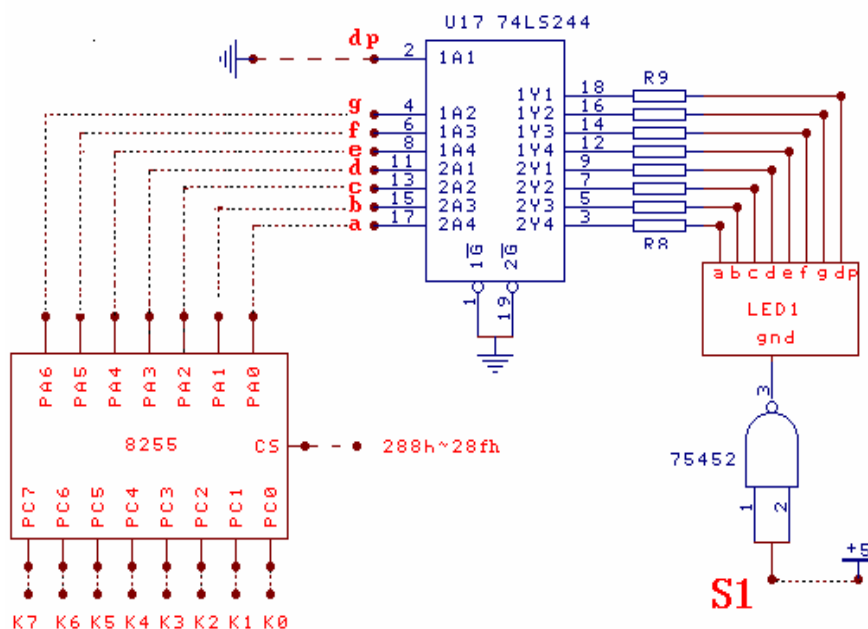


图7-1

### 三、编程提示

设置8255为C口输入、A口输出,读取C口数据,若为0表示无人抢答,若不为0则有人抢答。根据读取数据可判断其组号。从键盘上按空格键开始下一轮抢答,按其它键程序退出。

#### 四、参考流程图 (见图7-2)

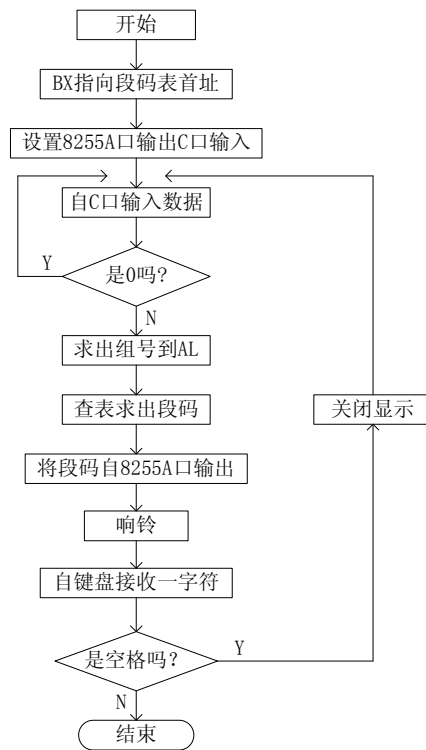


图7-2

## 实验八 交通灯控制实验

### 一、实验目的

通过并行接口8255实现十字路口交通灯的模拟控制,进一步掌握对并行口的使用。

### 二、实验内容

如图8-1, L7、L6、L5作为南北路口的交通灯与PC7、PC6、PC5相连, L2、L1、L0作为东西路口的交通灯与PC2、PC1、PC0相连。编程使六个灯按交通灯变化规律亮灭。

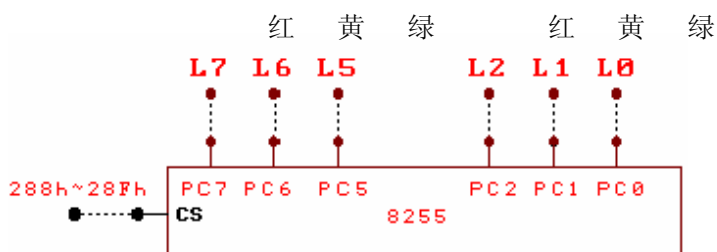


图8-1

### 三、编程提示: 十字路口交通灯的变化规律要求:

- (1) 南北路口的绿灯、东西路口的红灯同时亮30秒左右。
- (2) 南北路口的黄灯闪烁若干次, 同时东西路口的红灯继续亮。
- (3) 南北路口的红灯、东西路口的绿灯同时亮30秒左右。
- (4) 南北路口的红灯继续亮、同时东西路口的黄灯亮闪烁若干次。
- (5) 转(1)重复。

### 四、参考流程图

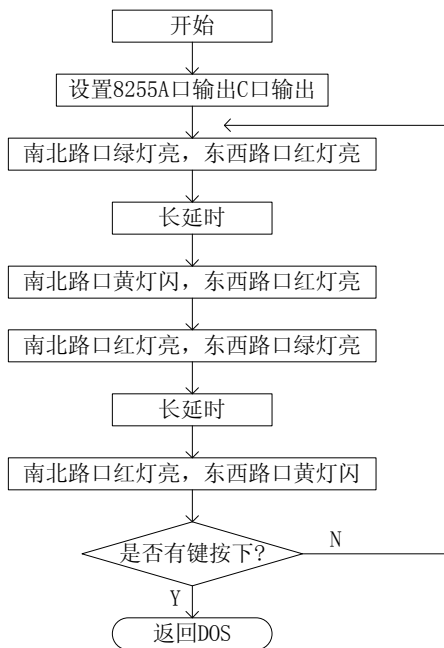


图8-2

## 实验九 中断

### 一、实验目的

- 1、掌握PC机中断处理系统的基本原理。
- 2、学会编写中断服务程序。

### 二、实验原理与内容

#### 1、实验原理

PC机用户可使用的硬件中断只有可屏蔽中断，由8259中断控制器管理。中断控制器用于接收外部的中断请求信号，经过优先级判别等处理后向CPU发出可屏蔽中断请求。IBMPC、PC/XT机内有一片8259中断控制器对外可以提供8个中断源：

中断源	中断类型号	中断功能
IRQ0	08H	时钟
IRQ1	09H	键盘
IRQ2	0AH	保留
IRQ3	0BH	串行口2
IRQ4	0CH	串行口1
IRQ5	0DH	硬盘
IRQ6	0EH	软盘
IRQ7	0FH	并行打印机

8个中断源的中断请求信号线IRQ0～IRQ7在主机62线ISA总线插座中可以引出，系统已设定中断请求信号为“边沿触发”，普通结束方式。对于PC/AT及286以上微机内又扩展了一片8259中断控制，IRQ2用于两片8259之间级连，对外可以提供16个中断源：

中断源	中断类型号	中断功能
IRQ8	070H	实时时钟
IRQ9	071H	用户中断
IRQ10	072H	保留
IRQ11	073H	保留
IRQ12	074H	保留
IRQ13	075H	协处理器
IRQ14	076H	硬盘
IRQ15	077H	保留

TPC-USB实验板上，固定的接到了3号中断IRQ3上，即进行中断实验时，所用中断类型号为0BH。

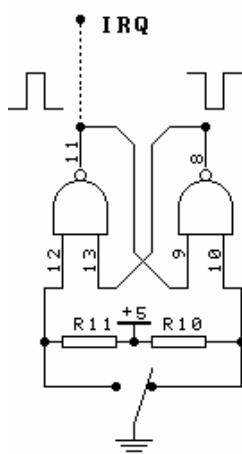


图9-1

## 2、实验内容

实验电路如图9-1，直接用手动产单脉冲作为中断请求信号（只需连接一根导线）。要求每按一次开关产生一次中断，在屏幕上显示一次“TPCA Interrupt!”，中断10次后程序退出。

## 三、参考流程图

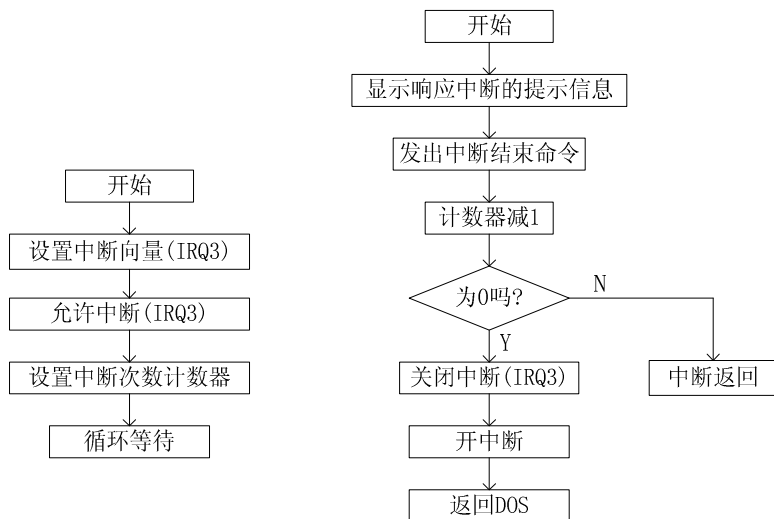


图9-2



## 实验十 可编程并行接口（二）（8255方式1）

### 一、实验目的

- 1、掌握8255工作方式1时的使用及编程。
- 2、进一步掌握中断处理程序的编写。

### 二、实验内容

- 1、按图10-1，8255方式1的输出电路连好线路。
- 2、编程:每按一次单脉冲按钮产生一个正脉冲使8255产生一次中断请求，让CPU进行一次中断服务:依次输出01H、02H，04H，08H，10H，20H，40H，80H使L0~L7依次发光，中断8次结束。
- 3、按图10-2，8255方式1输入电路，连好线路。

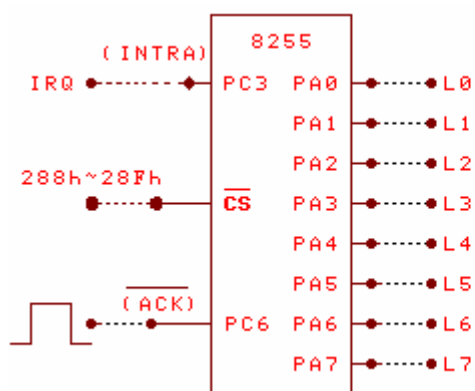


图10-1 输出电路

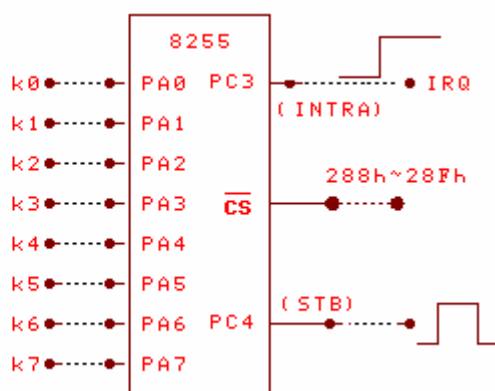


图10-2 输入电路

- 4、编程:每按一次单脉冲按钮产生一个正脉冲使8255产生一次中断请求，让CPU进行一次中断服务:读取逻辑电平开关预置的ASCII码，在屏幕上显示其对应的字符，中断8次结束。

### 三、参考流程图:（如图10-3、图10-4）

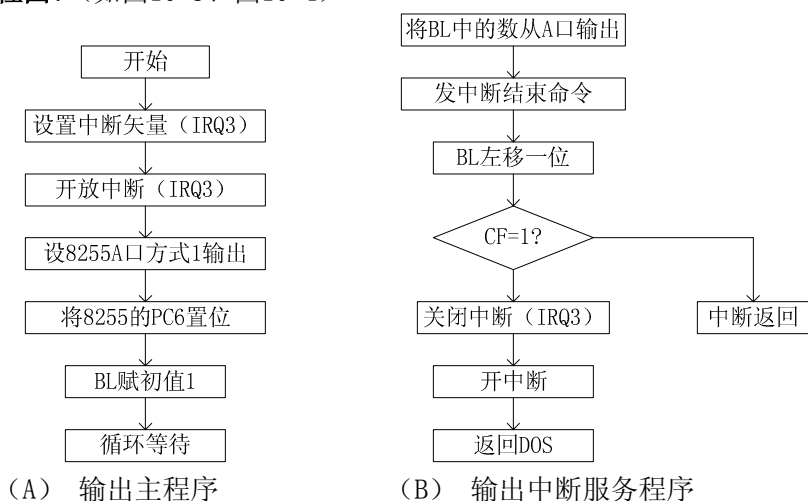
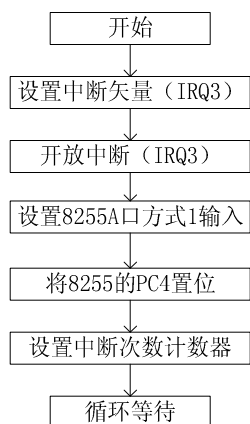
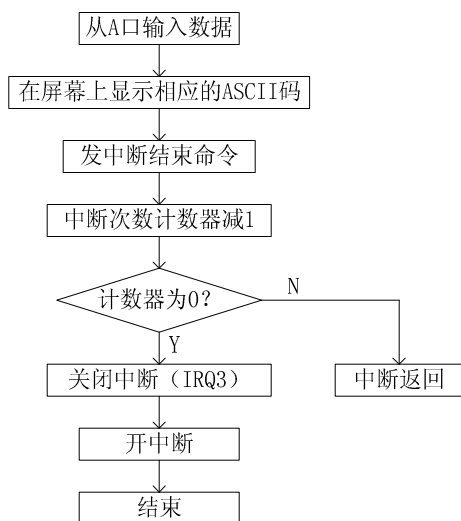


图10-3



(A) 输出主程序



(B) 输出中断服务程序

图10-4

## 实验十一 数/模转换器

### 一、实验目的

了解数/模转换器的基本原理，掌握DAC0832芯片的使用方法。

### 二、实验内容

1、实验电路原理如图11-1，DAC0832采用单缓冲方式，具有单双极性输入端(图中的U<sub>a</sub>、U<sub>b</sub>)，利用debug输出命令(Out 290 数据)输出数据给DAC0832，用万用表测量单极性输出端U<sub>a</sub>及双极性输出端U<sub>b</sub>的电压，验证数字与电压之间的线性关系。

2、编程产生以下波形(从U<sub>b</sub>输出，用示波器观察)

- (1) 锯齿波
- (2) 正弦波

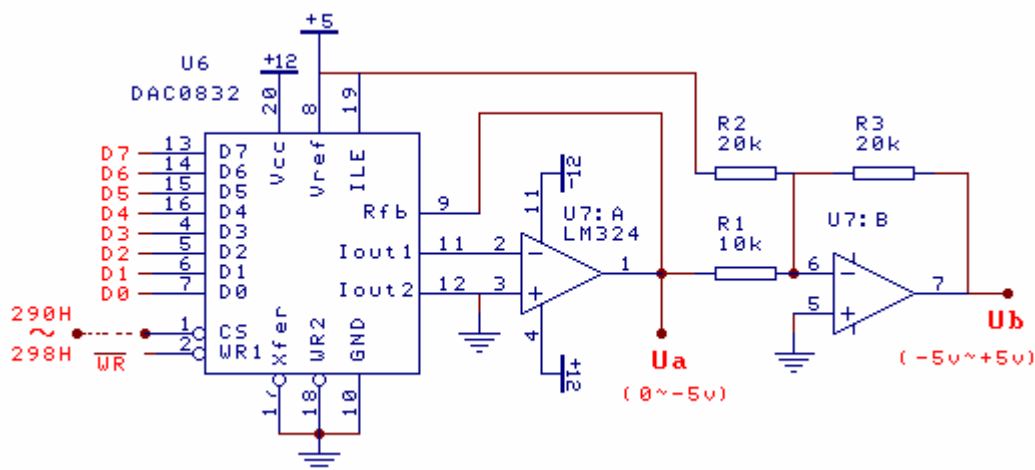


图11-1

### 三、编程提示

1、8位D/A转换器DAC0832的口地址为290H，输入数据与输出电压的关系为：

$$U_a = -\frac{U_{REF}}{256} \times N$$

$$U_b = -\frac{2U_{REF}}{256} \times N - 5$$

(U<sub>REF</sub>表示参考电压,N表示数数据)，这里的参考电压为P C机的+ 5 V 电源。

2、产生锯齿波只须将输出到DAC0832的数据由0循环递增。产生正弦波可根据正弦函数建一个下弦数字量表，取值范围为一个周期，表中数据个数在16个以上。

四、参考流程图：

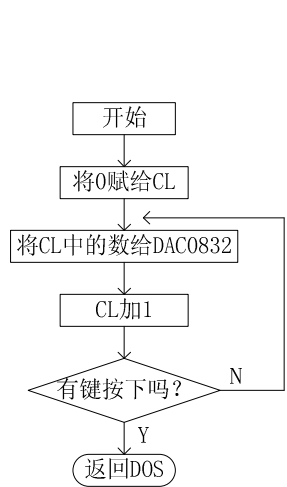


图11-2 锯齿波

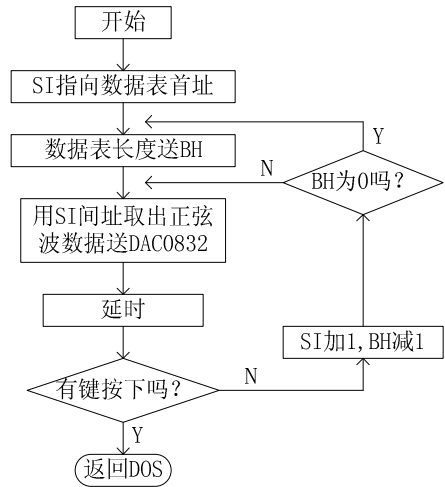


图11-3 正弦波

实验十二 模/数转换器

一、实验目的

了解模/数转换的基本原理，掌握ADC0809的使用方法。

二、实验内容

1、实验电路原理图如图12-1。通过实验台左下角电位器RW1输出0~5V 直流电压送入ADC0809通道0(IN0), 利用debug的输出命令启动 A / D 转换器，输入命令读取转换结果，验证输入电压与转换后数字的关系。

启动IN0开始转换：0 0298 0

读取转换结果： I 0298

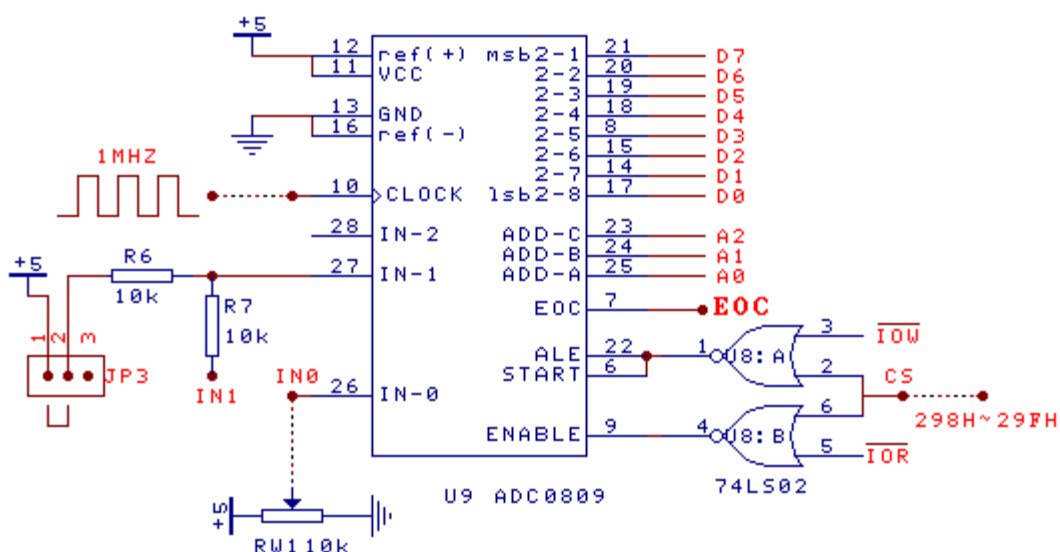


图12-1 模数转换电路

2、编程采集IN0输入的电压,在屏幕上显示出转换后的数据(用16进制数)。

3、将JP3的1、2短接,使IN2处于双极性工作方式,并给IN1输入一个低频交流信号(幅度为±5V),编程采集这个信号数据并在屏幕上显示波形。

### 三、实验提示

1、ADC0809的IN0口地址为298H, IN1口地址为299H。

2、IN0单极性输入电压与转换后数字的关系为:

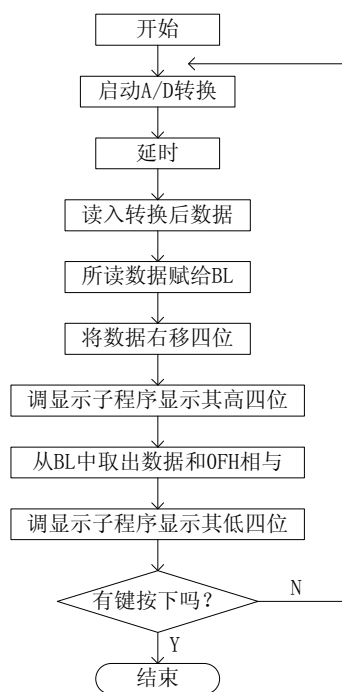
$$N = \frac{U_i}{U_{REF}/256}$$

其中 $U_i$ 为输入电压,  $U_{REF}$ 为参考电压, 这里的参考电压为PC机的+5V电源。

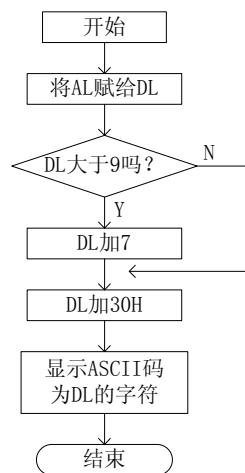
3、一次A/D转换的程序可以为

```
MOV     DX, 口地址
OUT     DX, AL      ; 启动转换
        ; 延时
IN      AL, DX      ; 读取转换结果放在AL中
```

### 四、参考流程图 (见图12-2、图12-3)



(A) 主程序



(B) 显示子程序

图12-2

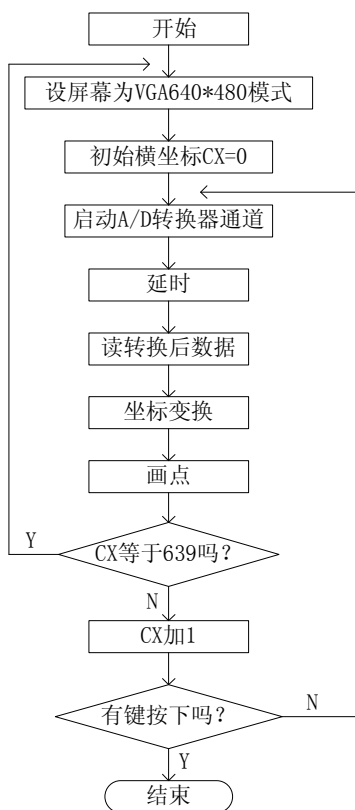


图12-3

## 实验十三 串行通讯

### 一、实验目的

- 1、了解串行通讯的基本原理。
- 2、掌握串行接口芯片8251的工作原理和编程方法。

### 二、实验内容

- 1、按图13-1连接好电路, (8251插通用插座) 其中8253计数器用于产生8251的发送和接收时钟, TXD和RXD连在一起。
- 2、编程: 从键盘输入一个字符, 将其ASCII码加 1 后发送出去, 再接收回来在屏幕上显示, 实现自发自收。

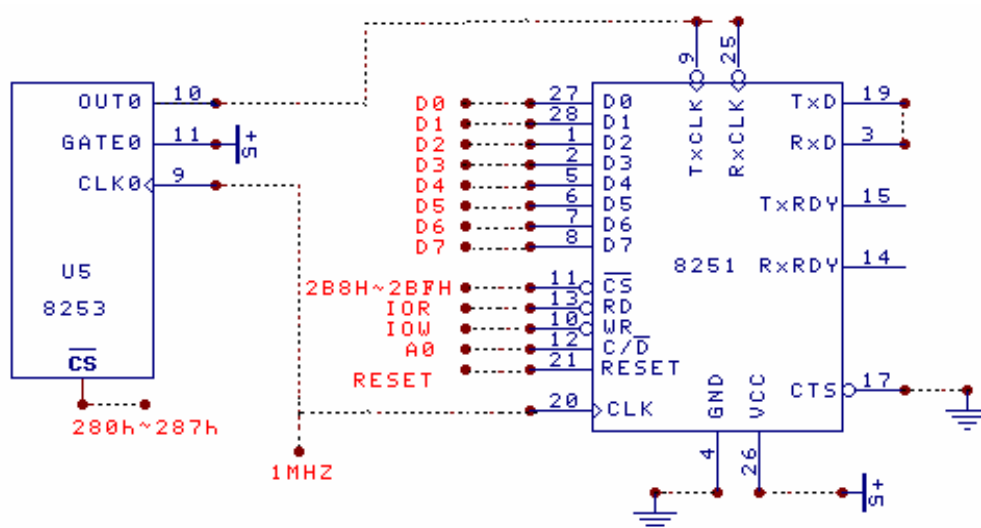


图13-1 串行通讯电路

### 三、实验提示

- 1、图示电路8251的控制口地址为2B9H, 数据口地址为2B8H。
- 2、8253计数器的计数初值=时钟频率/(波特率×波特率因子), 这里的时钟频率接1MHz, 波特率若选1200, 波特率因子若选16, 则计数器初值为52。
- 3、收发采用查询方式。

### 四、参考流程图 (见图13-2)

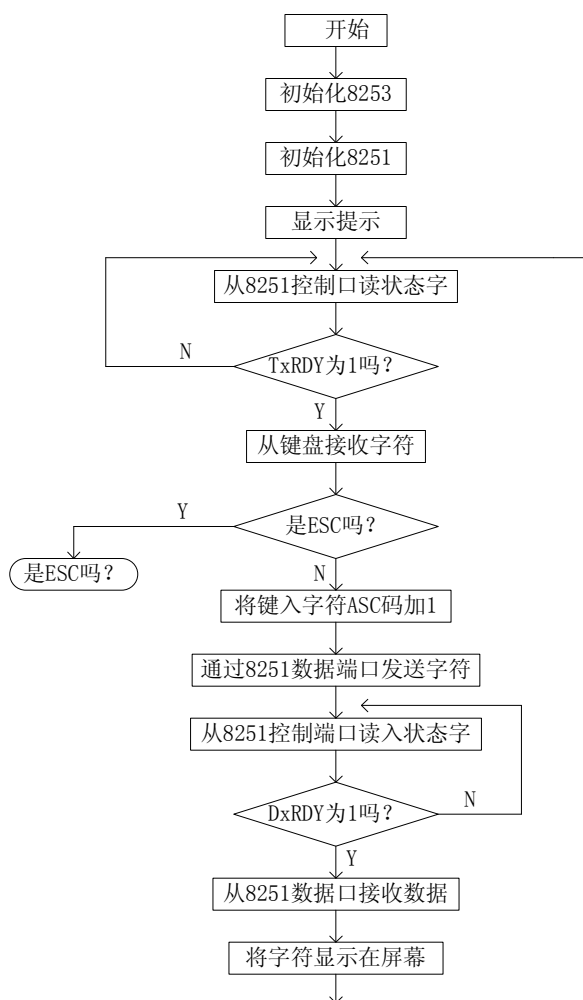


图13-2

## 实验十四 DMA传送

### 一、实验目的

- 1、掌握PC机工作环境下进行DMA方式数据传送(Block MODE和Demand Mode) (块传送、外部请求传送)方法。
- 2、掌握DMA的编程方法。

### 二、实验内容

- 1、按照图14-1将实验箱通用插座D区6116连接好。编程将USB的6116缓冲区D4000H, 偏移量为0的一块数据循环写入字符A~Z, 用Block MODE DMA方式传送到实验箱的6116的缓冲区D6000H上, 并察看送出的数据是否正确。



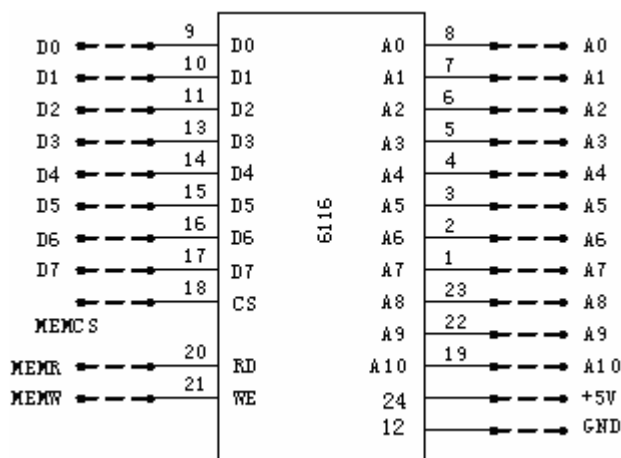


图14-1

2、用通用插座按图14-2连接好电路（74LS74利用实验台上的D触发器）。编程将主机内存缓冲区D4000H, 偏移量为0的10个数据，使用Demand Mode DMA方式从内存向外设传送。

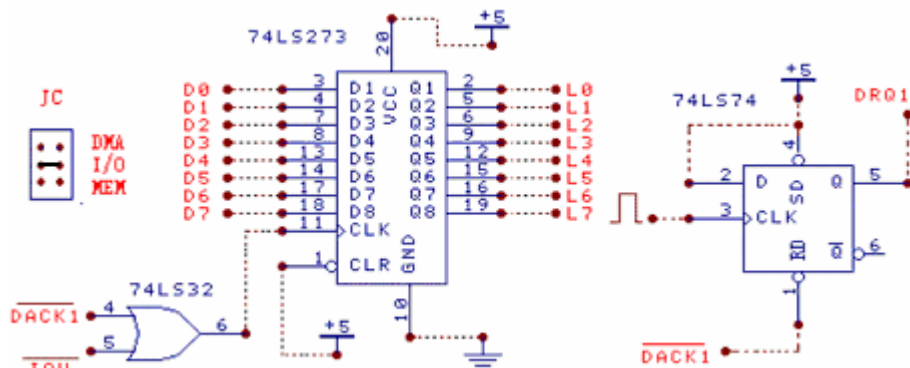


图14-2

3、用通用插座按图14-3连接好电路（74LS74利用实验台上的D触发器）。编程在主机内存缓冲区D4000H, 偏移量为0的位置开辟数据缓冲区，使用Demand Mode DMA方式从外设向内存传送8个数据并存入数据缓冲区，编程不断显示缓冲区的数据。

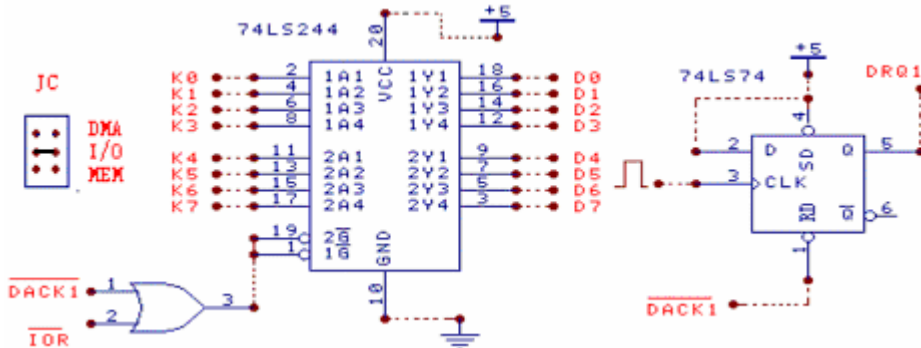


图14-3

### 三、实验提示

- 1、DMA 请求是由单脉冲输入到 D 触发器，由触发器的 Q 端向 DRQ1 发出的。CPU 响应后发出  $\overline{DACK1}$ ，将触发器 Q 置成低电平以撤消请求。
- 2、汇编程序中，为避免与系统 8237 有冲突，TPC-USB 模块上的 8237 端口范围为 10H-1F，即按通常模式进行 DMA 编程时，对 8237 所有端口均加 10H。

#### 四、参考流程图（见图14-4、图14-5、图14-6）

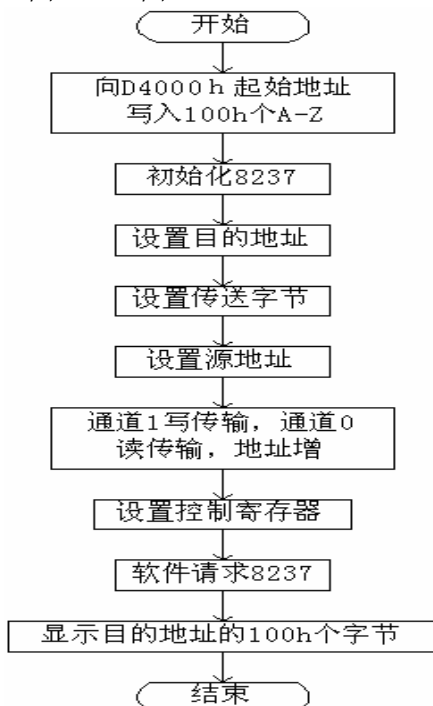


图14-4

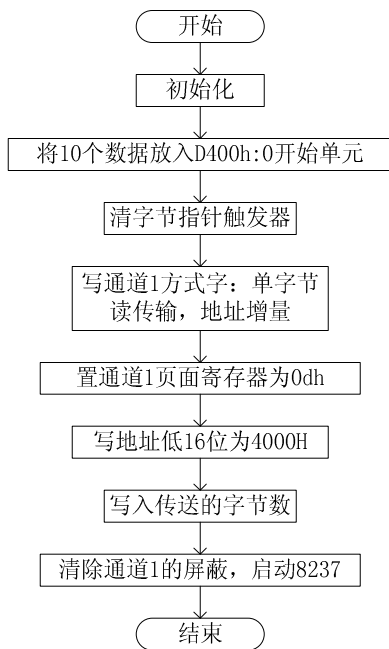


图14-5

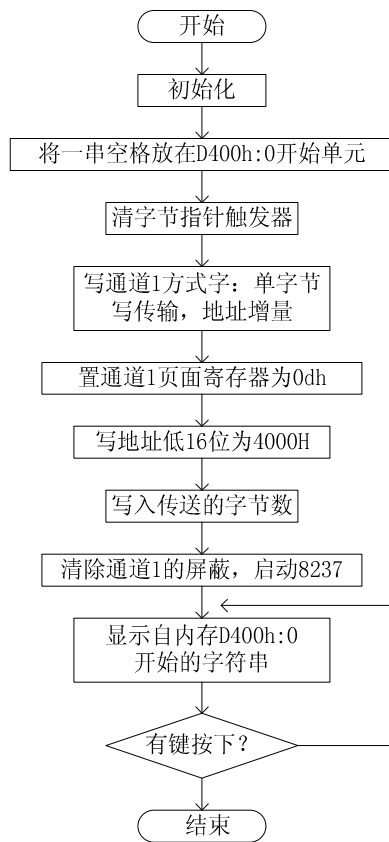


图14-6

## 实验十五 集成电路测试

### 一、实验目的

通过对四与非门(74LS00)集成电路芯片的测试,了解测试一般数字集成电路方法,进一步熟悉可编程并行接口8255 的使用。

### 二、实验内容

1、按图15-1连接电路,其中74LS00插在通用插座上。

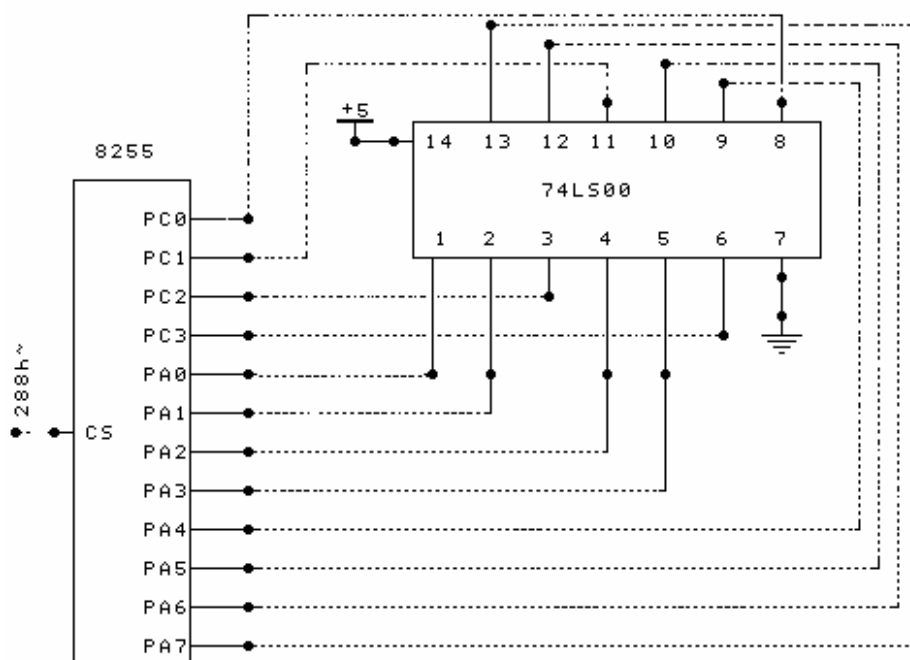


图15-1 集成电路测试电路图

### 2、编程提示:

将8255设置在方式0下工作,使A口输出,C口输入,先后通过A口分别给每一个门电路送入四个测试信号(00、01、10、11),相应从C口读出每一个门电路的输出结果,与正常值(1、1、1、0)进行比较,若相等,则此芯片好,否则为坏。

### 三、参考流程图 (见图15-2)

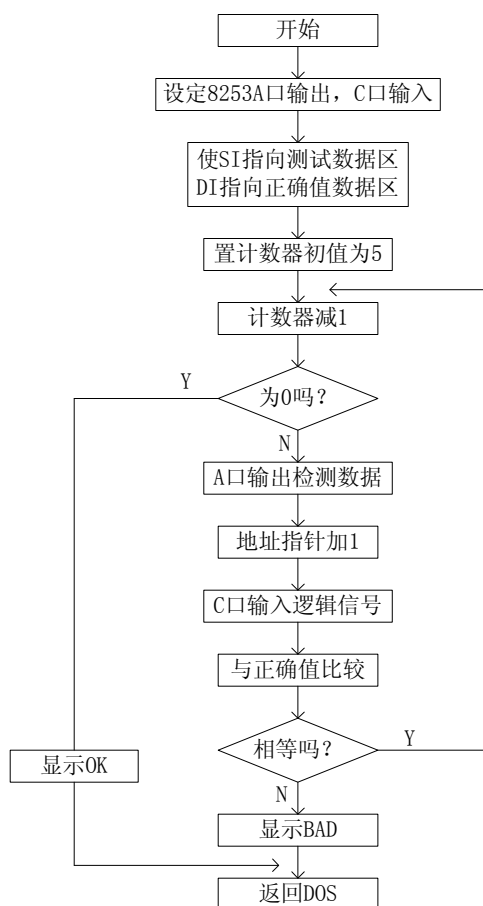


图15-2 集成电路测试流程图

## 实验十六 电子琴

### 一、实验目的

- 1、通过8253产生不同的频率信号，使PC机成为简易电子琴。
- 2、了解利用8255和8253产生音乐的基本方法。

### 二、实验内容

实验电路如图16-1，8253的CLK0接1MHZ时钟，GATE0接8255的PA1，OUT0和8255的PA0接到与门的两个输入端，K8跳线连接喇叭，编程使计算机的数字键1、2、3、4、5、6、7作为电子琴按键，按下即发出相应的音阶。

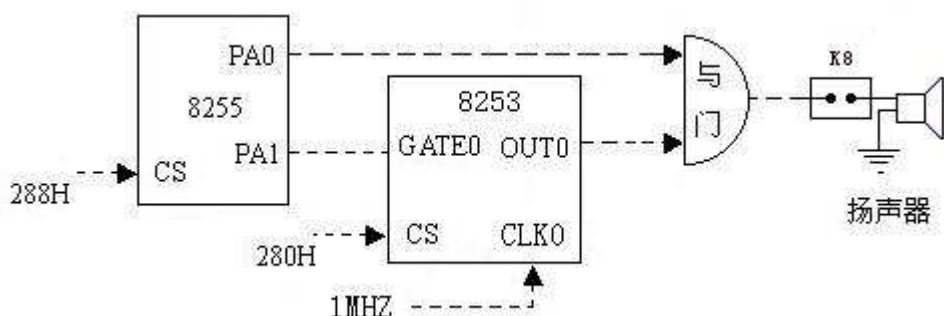


图16-1 电子琴电路

### 三、编程提示：

1、利用8255的PA0口来施加控制信号给与门，用来控制扬声器的开关状态。再利用设置不同的计数值，使8253产生不同频率的波形，使扬声器产生不同频率的音调，达到类似与音阶的高低音变换。对于音乐，每个音阶都有确定的频率。

各音阶标称频率值：

音 阶	1	2	3	4	5	6	7	1 <sub>·</sub>
低频率(单位:Hz)	262	294	330	347	392	440	494	524
高频率(单位:Hz)	524	588	660	698	784	880	988	1048

### 四、参考流程图（见图16-2）

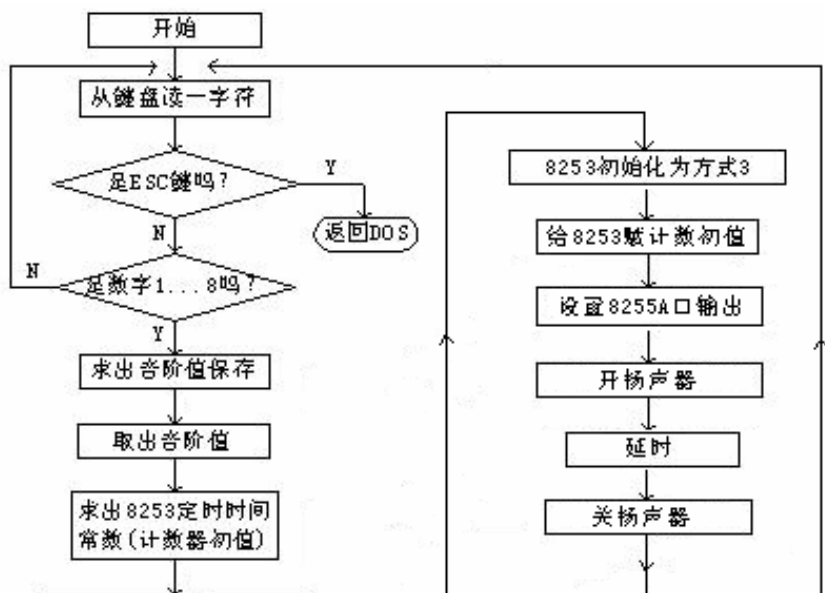


图16-2 主程序

## 实验十七 8250串行通讯实验

### 一、实验目的

- 1、进一步了解串行通信的基本原理。
- 2、掌握串行接口芯片8250的工作原理和编程方法。

### 二、实验内容

- 1、按图17-1连接线路，图中8250芯片插在通用插座上。
- 2、编程:从键盘输入一个字符，将其ASCII码加1后发送出去，再接收回来在屏幕上加1后的字符显示出来，实现自发自收。

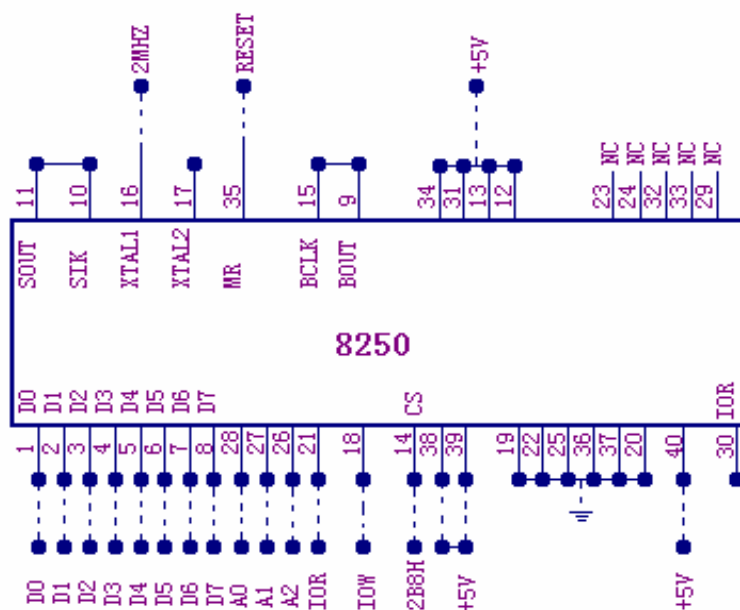


图17-1

### 三、实验提示

#### 1、8250介绍:

INC8250是一个可编程序异步通讯单元芯片，在微机系统中起串行数据的输入输出接口作用。此外，它还包含有可编程序波特率发生器，它可用1~65535的因子对输入时钟进行分频，以产生波特率十六倍的输入输出时钟。

2、8250时钟接2MHZ，若选波特率为9600，波特率因子为16，则因子寄存器中分频数为13。所以因子寄存器低字节送13，高字节为00H。

3、图中CS接02B8H~02BFH:

下表为各寄存器选择地址一览表。表中DLAB为线控制寄存器的最高位，也叫因子寄存器存取位。当DLAB为0时选接收数据缓冲器，发送数据寄存器和中断允许寄存器。当DLAB为1时选因子寄存器的低字节和高字节。

DLAB	A2	A1	A0	选中寄存器
0	0	0	0	接收缓冲器（读）发送保持寄存器（写）

0	0	0	1	中断允许寄存器
X	0	1	0	中断标志寄存器（仅用于读）
X	0	1	1	线控制寄存器
X	1	0	0	MODEM控制寄存器
X	1	0	1	线状态寄存器
X	1	1	0	MODEM状态寄存器
X	1	1	1	无
1	0	0	0	因子寄存器（低字节）
1	0	0	1	因子寄存器（高字节）

4、收发采用查询方式。

#### 四、参考流程

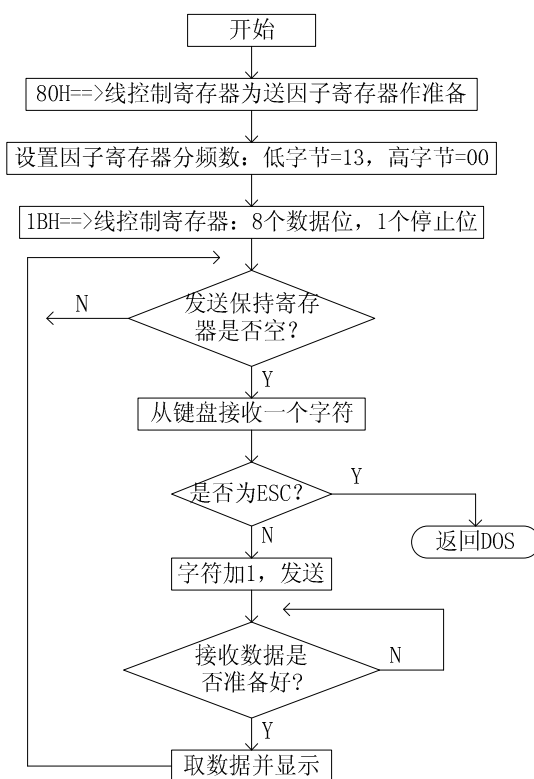


图17-2



## 实验十八 步进电机控制实验

### 一、实验目的

- 1、了解步进电机控制的基本原理。
- 2、掌握控制步进电机转动的编程方法。

### 二、实验内容

- 1、按图18-1连接线路，利用8255输出脉冲序列，开关K0~K6控制步进电机转速，K7控制步进电机转向。8255 CS接288H~28FH。PA0~PA3接BA~BD；PC0~PC7接K0~K7。
- 2、编程:当K0~K6中某一开关为“1”（向上拨）时步进电机启动。K7向上拨电机正转，向下拨电机反转。

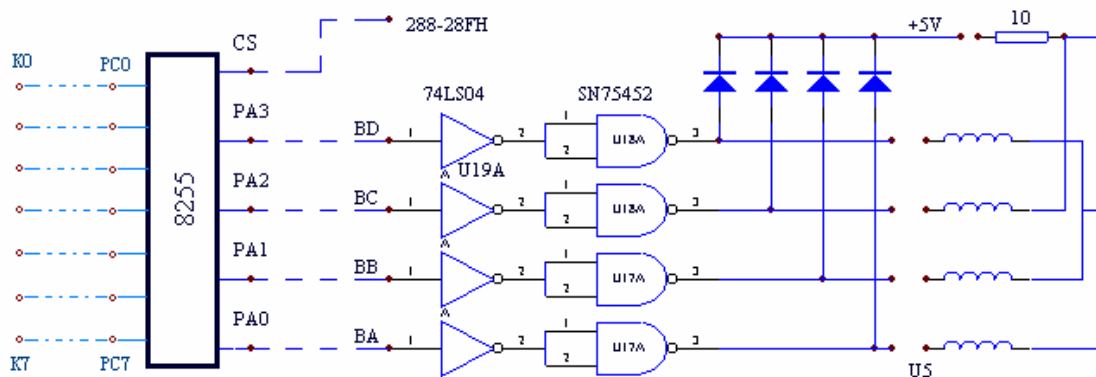


图18-1

### 三、实验说明

步进电机驱动原理是通过对每相线圈中的电流的顺序切换来使电机作步进式旋转。驱动电路由脉冲信号来控制，所以调节脉冲信号的频率便可改变步进电机的转速。

如图18-2所示：本实验使用的步进电机用直流+5V电压，每相电流为0.16A，电机线圈由四相组成，即： $\phi 1$  (BA)； $\phi 2$  (BB)； $\phi 3$  (BC)； $\phi 4$  (BD)

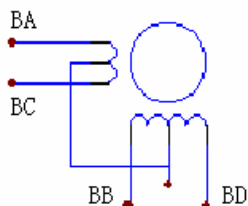
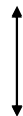


图18-2

驱动方式为二相激励方式，各线圈通电顺序如下表。

顺序 \ 相	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$
0	1	1	0	0
1	0	1	1	0

反时针方向回转



正时针方向回转

2	0	0	1	1
3	1	0	0	1

表中首先向 $\phi 1$ 线圈— $\phi 2$ 线圈输入驱动电流，接着 $\phi 2-\phi 3$ ， $\phi 3-\phi 4$ ， $\phi 4-\phi 1$ ，又返回到 $\phi 1-\phi 2$ ，按这种顺序切换，电机轴按顺时针方向旋转。

实验可通过不同长度延时来得到不同频率的步进电机输入脉冲，从而得到多种步进速度。

#### 四、参考流程图

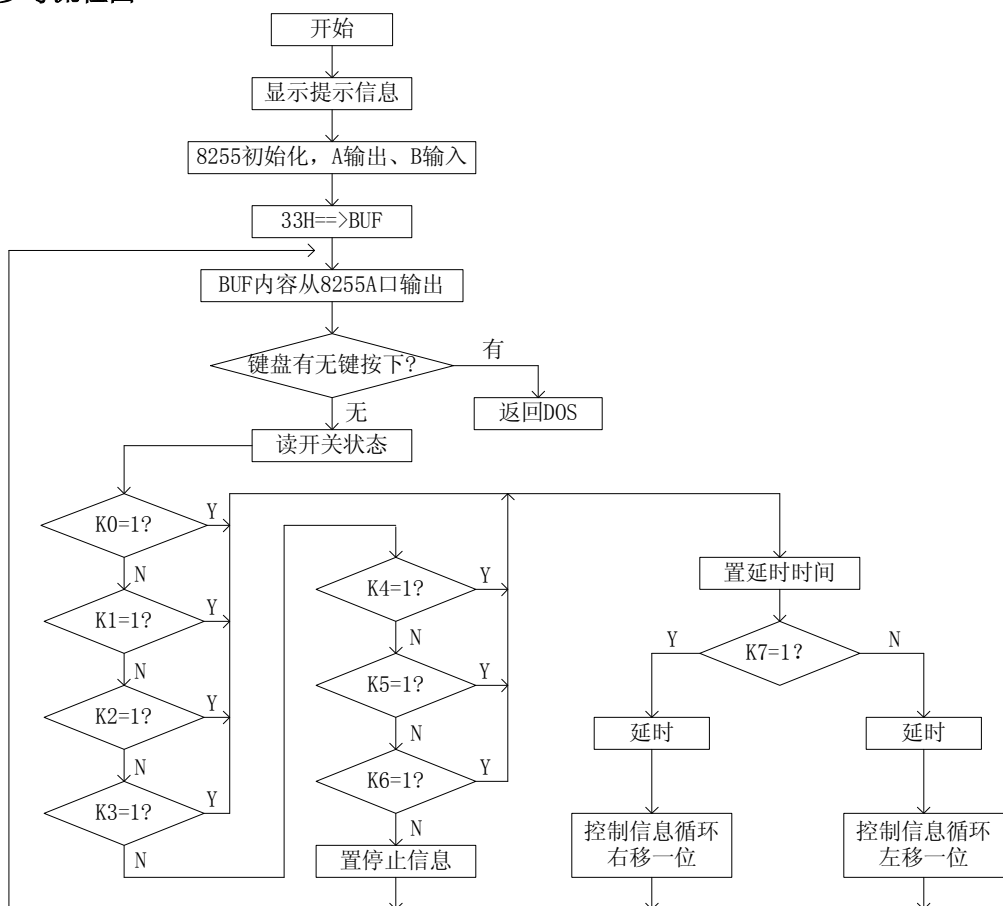


图18-3

## 实验十九 小直流电机转速控制实验

### 一、实验目的

- 1、进一步了解DAC0832的性能及编程方法。
- 2、了解直流电机控制的基本方法。

### 二、实验内容

- 1、按图19-1线路接线。DAC0832的CS接290H~297H，Ub接DJ插孔，LM324 CS接288H~28FH。
- 2、编程利用DAC0832输出一串脉冲，经放大后驱动小直流电机，利用开关K0~K5控制改变输出脉冲的电平及持续时间，达到使电机加速，减速之目的。

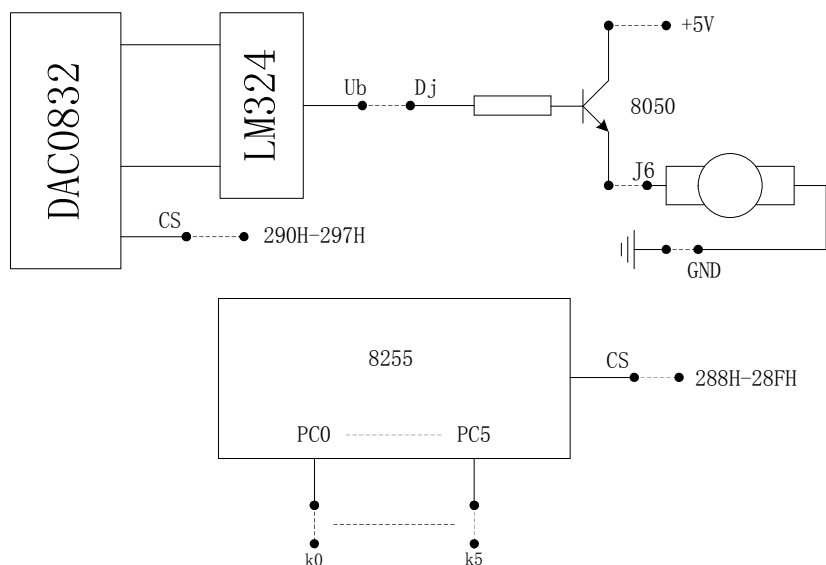


图19-1

### 三、实验原理简述

小直流电机的转速是由Ub输出脉冲的占空比来决定的，正向占空比越大转速越快，反之越慢。见下图19-2例：

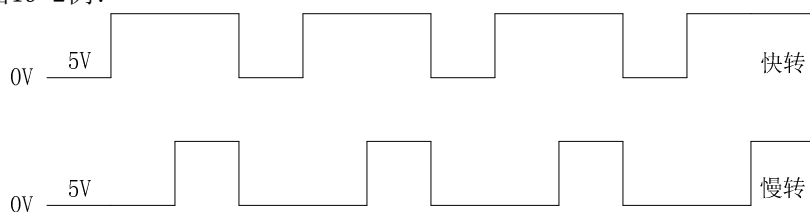


图19-2

在本实验中，模拟量输出Ub为双极性，当输入数字量小于80H时输出为负，输入等于80H时为0V，输入大于80H时输出为正。因而本实验中，DAC0832输入数字量只有2个(80H和FFH)，通过不同的延迟时间达到改变小电机转速的目的。

### 四、参考程序框图

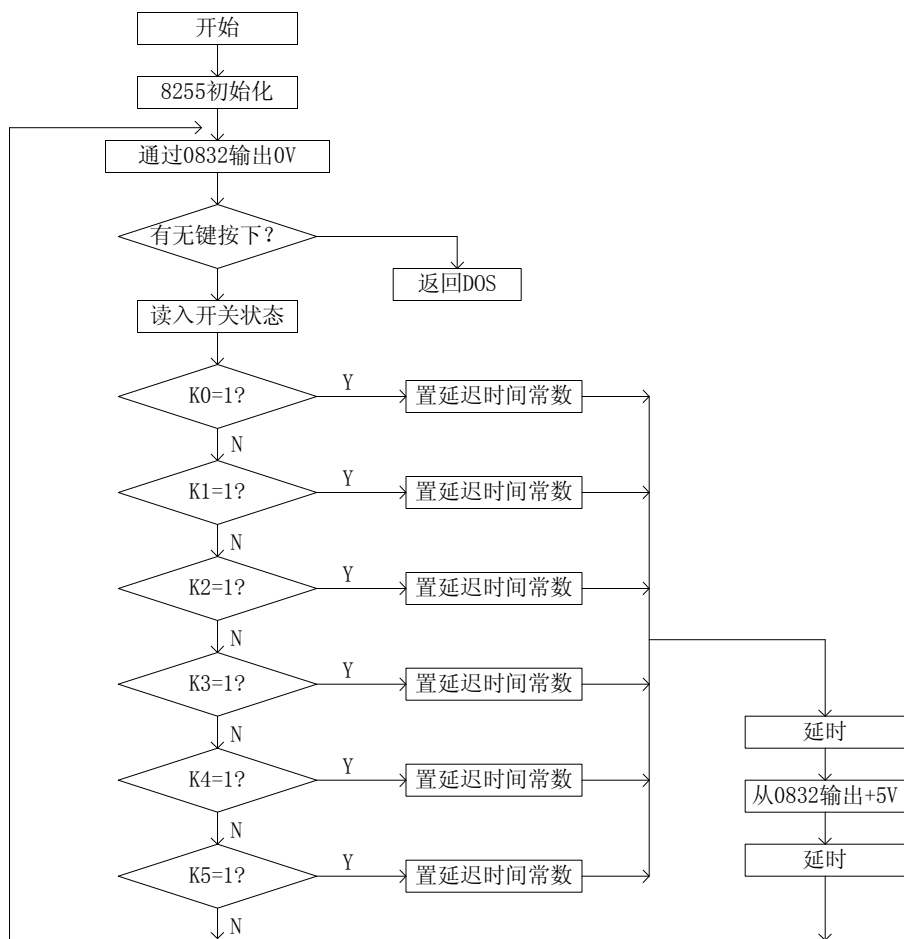


图19-3

实验二十 键盘显示控制器实验

一、实验目的

- 1、掌握8279键盘显示电路的基本功能及编程方法。
- 2、掌握一般键盘和显示电路的工作原理。
- 3、进一步掌握定时器的使用和中断处理程序的编程方法。

二、实验内容

1、本实验的实验电路如图20-1，它做在一块扩展电路板上，用一根20芯扁平电缆与实验台上扩展插头J7相连。

2、编程1:使得在小键盘上每按一个键，6位数码管上显示出相应字符，它们的对应关系如下：

小键盘		显示	小键盘		显示
0	—	0	C	—	C
1	—	1	D	—	d
2	—	2	E	—	E
3	—	3	F	—	F
4	—	4	G	—	q
5	—	5	M	—	□
6	—	6	P	—	p
7	—	7	W	—	U
8	—	8	X	—	
9	—	9	Y	—	4
A	—	□	R	—	返回
B	—	b			

3、编程2:中断编程

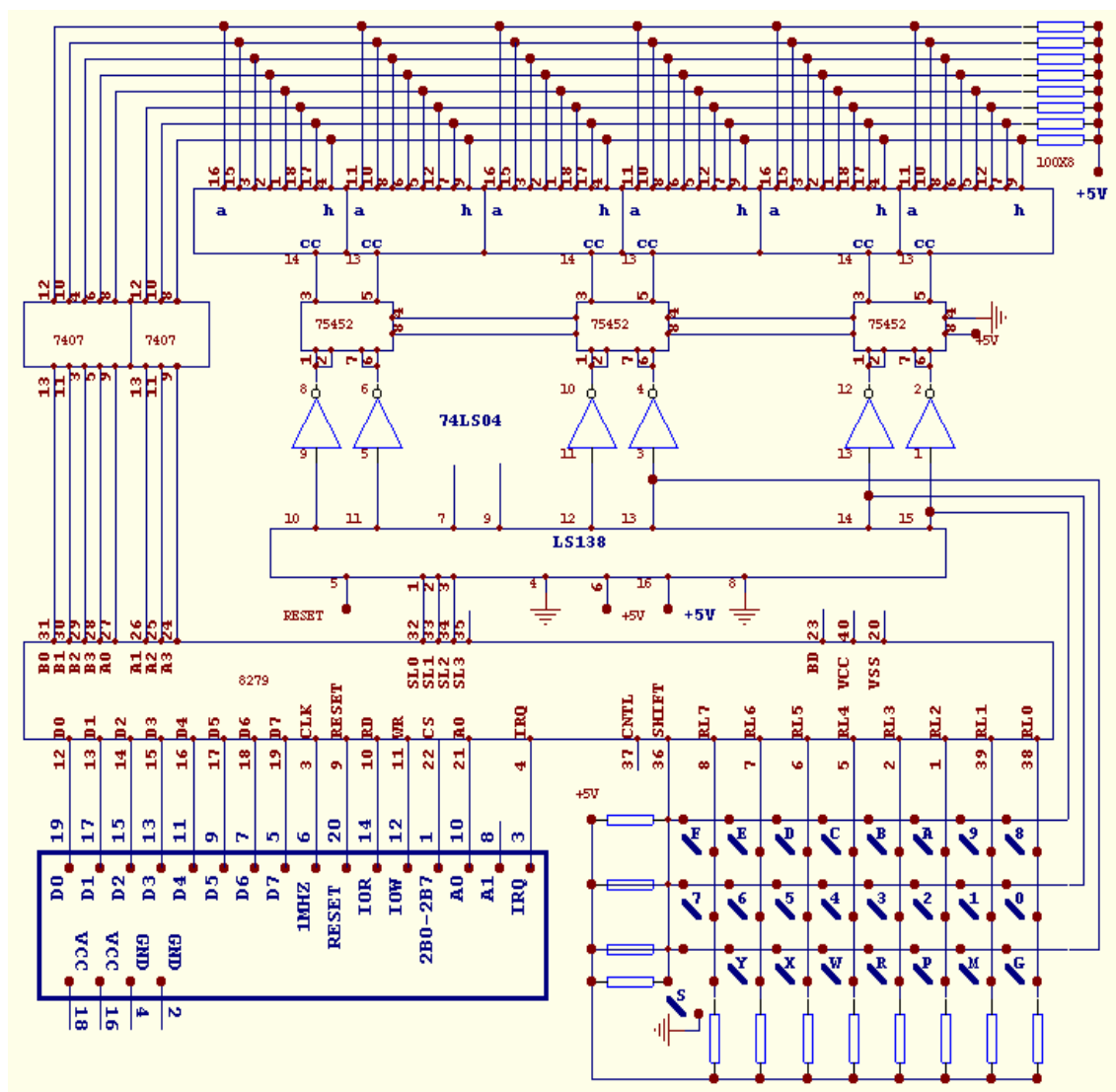
利用实验台上提供的定时器8253和扩展板上提供的8279以及键盘和数码显示电路，设计一个电子钟。由8253中断定时，小键盘控制电子钟的启停及初始值的预置。

电子钟显示格式如下：

XX. XX. XX. 由左向右分别为时、分、秒

要求具有如下功能：

- ①、C键:清除，显示全零。
- ②、G键:启动，电子钟计时。
- ③、D键:停止，电子钟停止计时。
- ④、P键:设置时、分、秒值。输入时依次为时、分、秒，同时应有判断输入错误的的能力，若输入有错，则显示:E————。此时敲P键可重新输入预置值。
- ⑤、E键:程序退出。



### 三、编程1 接线方法

#### 四、编程2 接线方法

- 2、实验台上8253 CLK0 接 1MHZ, GATE0 和 GATE1接+5V, OUT0 接 CLK1, OUT1 接 IRQ, CS接 280H~287H。

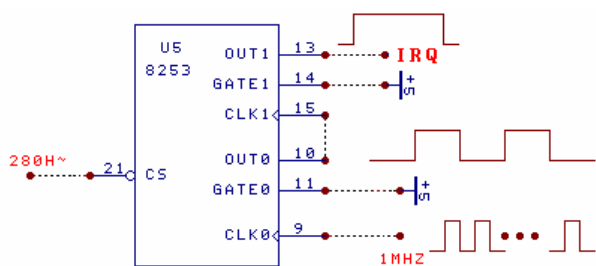
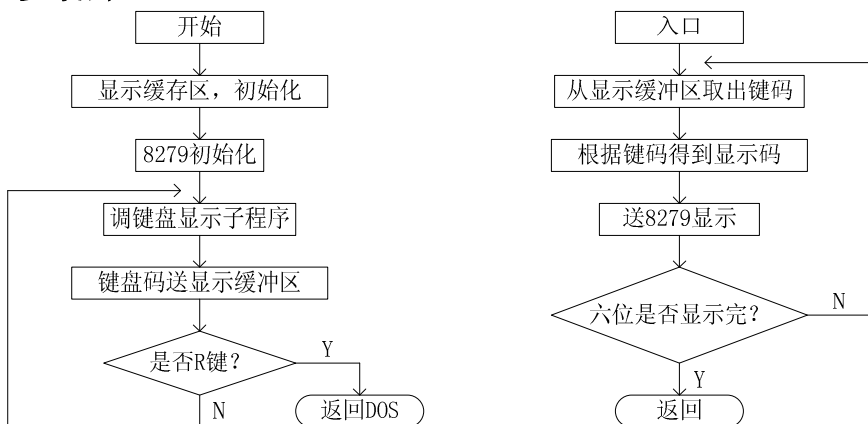


图20-2

## 五、编程1 参考流程



主程序流程图:

显示子程序流程图:DISP

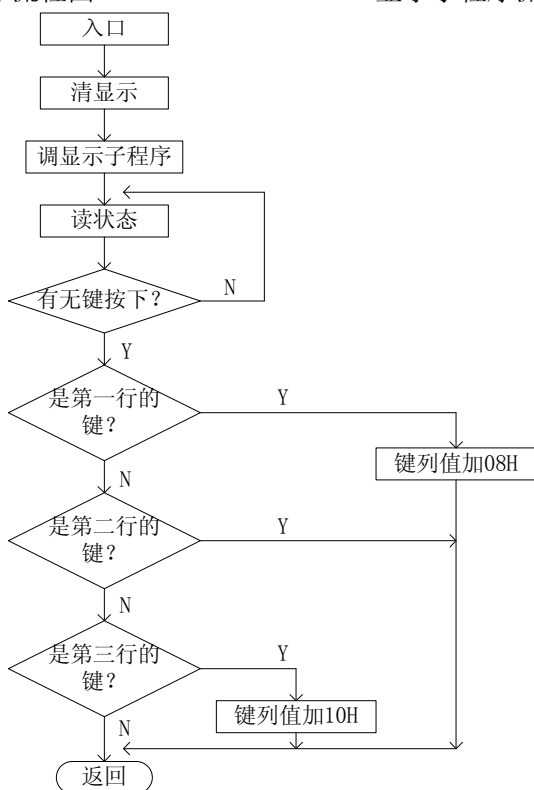
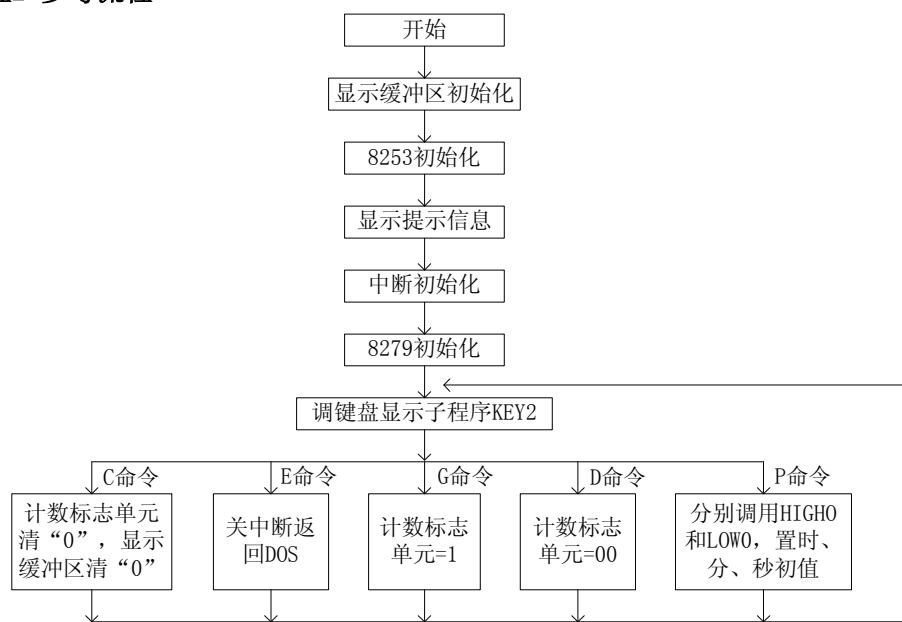
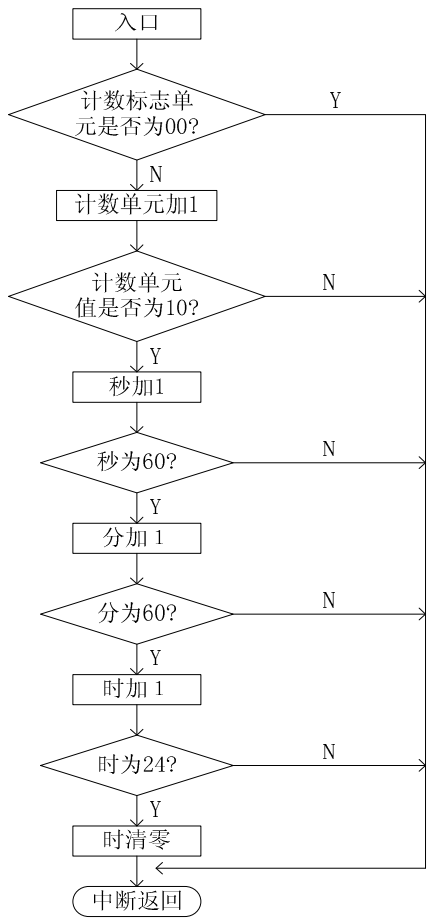


图20-3 键盘显示子程序流程图:KEY2

六、编程2 参考流程

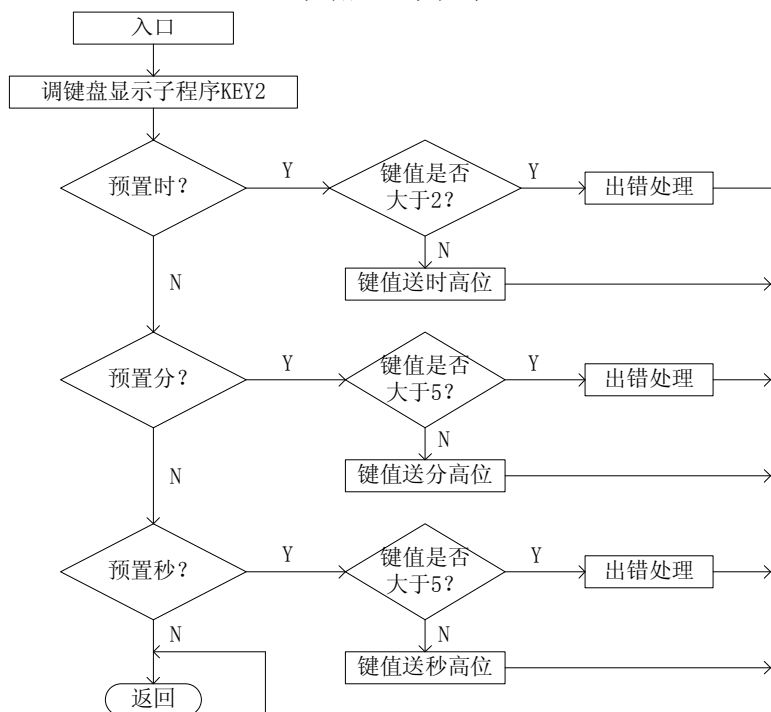


主程序流程图：





# 中断处理子程序:



## 预置时、分、秒高位子程序HIGH0:

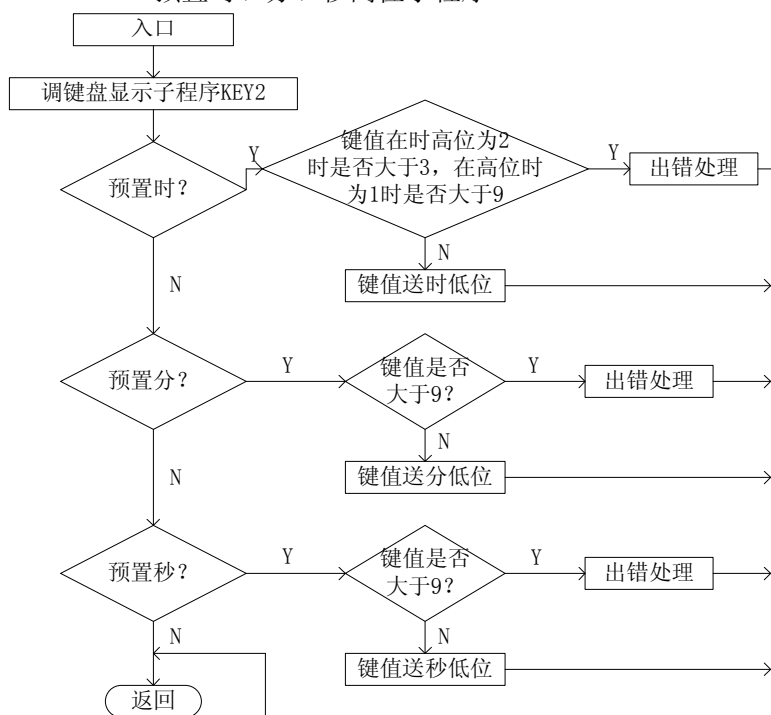


图20-4 预置时、分、秒低位子程序LOW0:

## 实验二十一 存储器读写实验

### 一、实验目的

- 1、熟悉6116静态RAM的使用方法，掌握PC机外存扩充的手段。
- 2、通过对硬件电路的分析，学习了解总线的工作时序。

### 二、实验内容

- 1、 硬件电路如下：

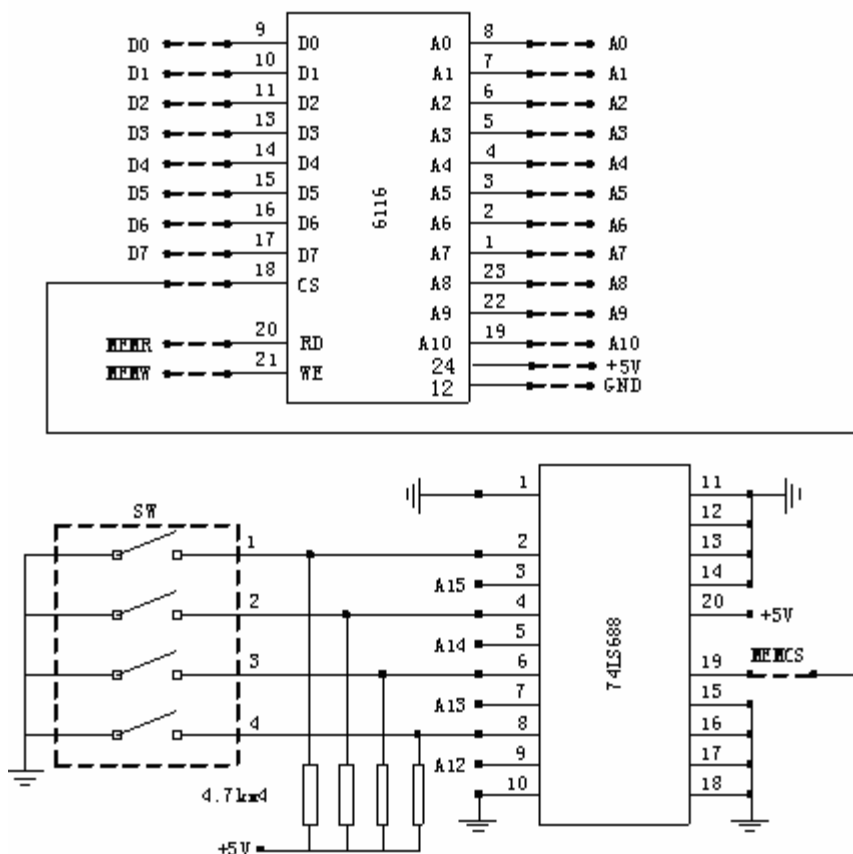


图21-1

2、编制程序，将字符A-Z循环写入扩展的6116RAM中，然后再将6116的内容读出来显示在主机屏幕上。

### 三、编程提示

- 1、注意:TPC-USB已为扩展的6116指定了段地址:0d000H。
- 2、TPC-USB模块外扩储器的地址范围为0D4000H-0D7fffH。
- 3、通过片选信号的产生方式，确定6116RAM在PC机系统中的地址范围。因为段地址已指定，所以其地址为CS=A15 and A14 and A13 and A12，实验台上设有地址选择微动开关K2，

拨动开关，可以选择4000-7fff的地址范围。编制程序，从0d6000H开始循环写入100h个A-Z。

开关状态如下：

1	2	3	4	地址
ON	OFF	ON	OFF	d4000h
ON	OFF	OFF	ON	d6000h

#### 四、程序框图

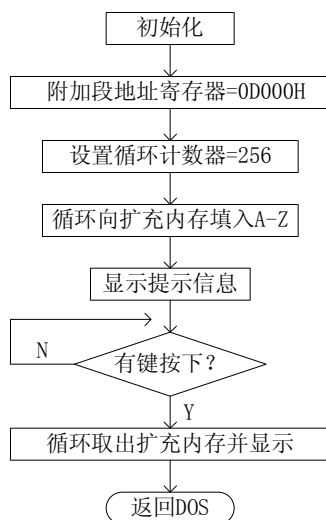


图21-2

## 实验二十二 双色点阵发光二极管显示实验

### 一、实验目的

- 1、了解双色点阵LED显示器的基本原理。
- 2、掌握PC机控制双色点阵LED显示程序的设计方法。

### 二、实验原理

点阵LED显示器是将许多LED类似矩阵一样排列在一起组成的显示器件，双色点阵LED是在每一个点阵的位置上有红绿或红黄或红白两种不同颜色的发光二极管。当微机输出的控制信号使得点阵中有些LED发光，有些不发光，即可显示出特定的信息，包括汉字、图形等。车站广场由微机控制的点阵LED大屏幕广告宣传牌随处可见。

实验仪上设有一个共阳极8×8点阵的红黄两色LED显示器，其点阵结构如图所示。该点阵对外引出24条线，其中8条行线，8条红色列线，8条黄色列线。若使某一种颜色、某一个LED发光，只要将与其相连的行线加高电平，列线加低电平即可。

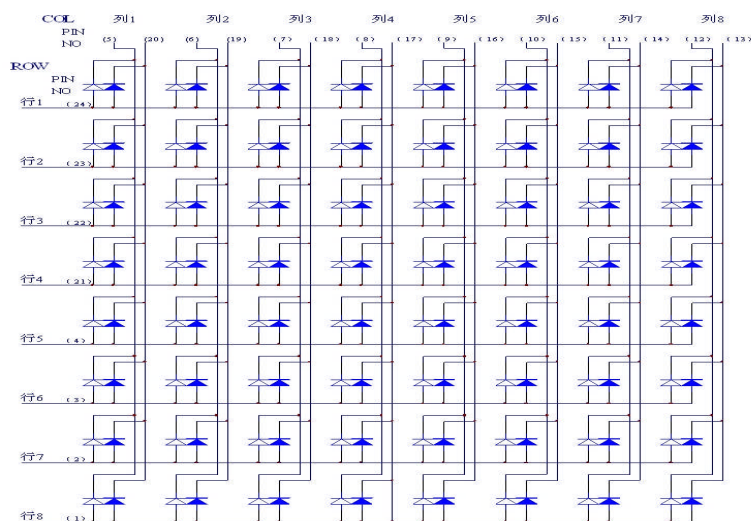


图22-1

例如欲显示汉字“年”，采用逐列循环发光。首先由“年”的点阵轮廓，确定点阵代码(如图所示)根据“年”的点阵代码，确定逐列循环发光的顺序如下：

- ① 行代码输出 44H； 红色列代码输 01H； 第一列2个红色LED发光。
- ② 行代码输出 54H； 红色列代码输 02H； 第二列3个红色LED发光。
- ③ 行代码输出 54H； 红色列代码输 04H； 第三列3个红色LED发光。
- ④ 行代码输出 7FH； 红色列代码输 08H； 第四列7个红色LED发光。
- ⑤ 行代码输出 54H； 红色列代码输 10H； 第五列3个红色LED发光。
- ⑥ 行代码输出 DCH； 红色列代码输 20H； 第六列5个红色LED发光。
- ⑦ 行代码输出 44H； 红色列代码输 40H； 第七列2个红色LED发光。
- ⑧ 行代码输出 24H； 红色列代码输 80H； 第八列2个红色LED发光。

在步骤①~⑧之间可插入几ms的延时，重复进行①~⑧即可在LED上稳定的显示出红色“年”字。若想显示黄色“年”，只需把红色列码改为黄色列码即可。

### 三、实验内容：

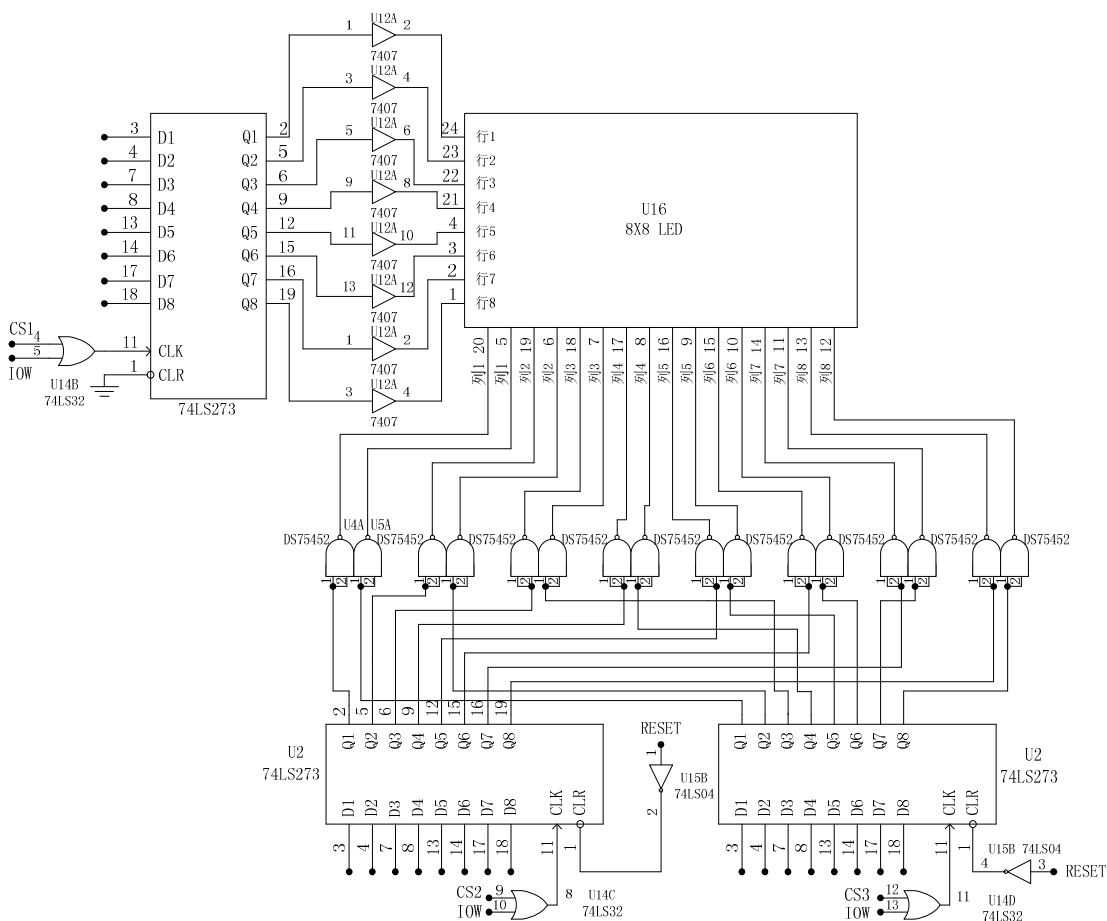


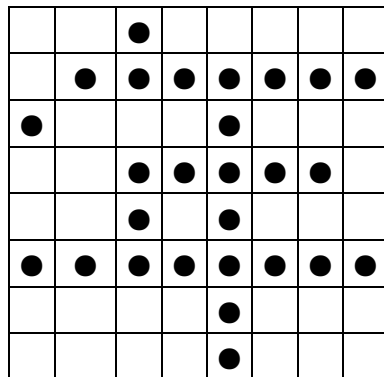
图22-2

1、实验仪上的点阵LED及驱动电路如下图所示，行代码、红色列代码、黄色列代码各用一片74LS273锁存。行代码输出的数据通过行驱动器7407加至点阵的8条行线上，红和黄列代码的输出数据通过驱动器DS75452反相后分别加至红和黄的列线上。行锁存器片选信号为CS1，红色列锁存器片选信号为CS2，黄色列锁存器片选信号为CS3。

2、接线方法:行片选信号 CS1 接 280H; 红列片选信号 CS2 接 288H; 黄列片选信号 CS3 接 290H。

3、编程重复使LED点阵红色逐列点亮，再黄色逐列点亮，再红色逐行点亮，黄色逐行点亮。

4、编程在LED上重复显示红色“年”和黄色“年”。



四、逐行、逐列显示参考流程图

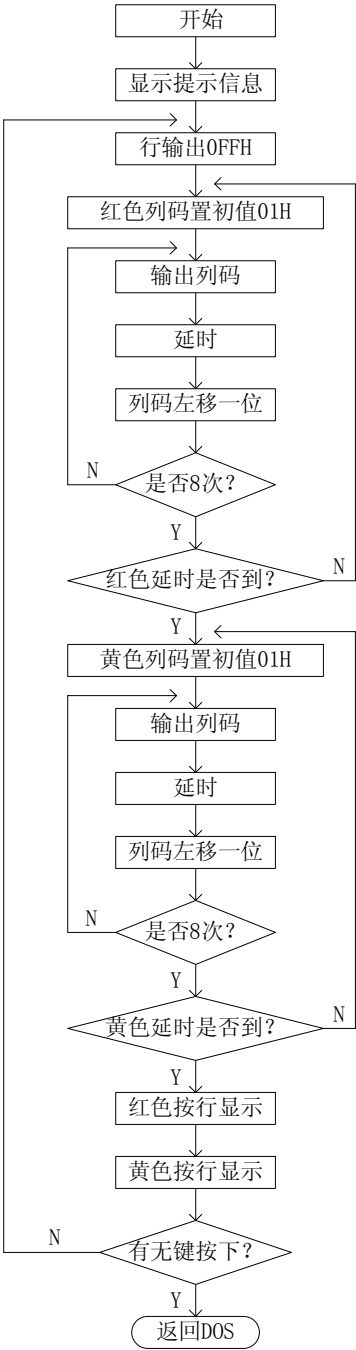


图22-3

五、显示“年”参考流程

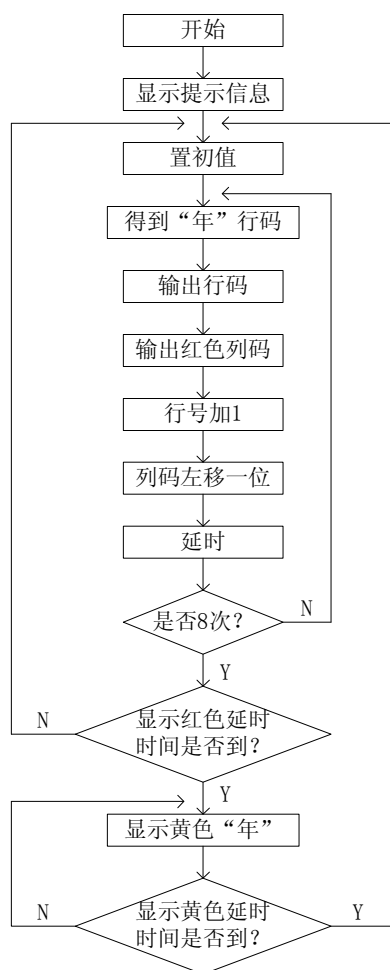


图22-4

## 第五章 VC++实验部分

### 1、基本输入输出-基本输入输出函数简介

1、Startup();

语法: BOOL Startup()

功能描述: 查询TPC-USB微机接口实验装置是否可用, 如果可用则打开。

参数: 无

返回值: 如果设备存在并且可用, 则返回True, 否则返回False

备注: 应用程序在对TPC-USB做任何操作之前必须调用该函数, 应用程序结束时必须使用Cleanup函数关闭该设备。

2、void Cleanup();

语法: void Cleanup()

功能描述: 关闭设备。

参数: 无

返回值: 无

备注: 应用程序结束时必须使用Cleanup函数关闭该设备。它和Startup成对使用。

3、PortReadByte;

语法: BOOL PortReadByte(DWORD address, BYTE \*pdata);

功能描述: 读TPC-USB某个的IO端口值。

参数:

address: 指明要读的IO端口地址。

pdata: 该函数执行完后, address所指明的端口值被填入该地址。

返回值: 如果读成功, 则返回True, 否则返回False。

备注: 应用程序使用该函数前必须先调用Startup函数。

例子:

```
BYTE    data;
```

```
DWORD   address = 0x283;
```

```
if (!Startup())
```

```
{
```

```
    //ERROR .. 出错处理
```

```
}
```

```
    if (!PortReadByte(address, &data))
```

```
{
```

```
    //ERROR .. 出错处理
```

```
}
```

```
    //SUCCESS .. 成功, 此时data里存放地址为address的IO端口的值
```

4、PortWriteByte;

语法: BOOL PortWriteByte(DWORD address, BYTE data);

功能描述: 将给定值写入TPC-USB所指明的IO端口。



参数:

address:指明要写的硬件I/O端口地址

data: 该函数执行完后, data将被写入address所指明的I/O端口

返回值:如果读成功, 则返回True, 否则返回False。

备注:应用程序使用该函数前必须先调用Startup。

例子:

```
BYTE    data;
DWORD   address = 0x283;
if (!Startup())
{
    //ERROR .. 出错处理
}
if (!PortReadByte(address, &data))
{
    //ERROR .. 出错处理
}
//SUCCESS .. 此时已经将值data写入address所指明的I/O端口
```

几点约定:

- 1、实验电路介绍中凡不加“利用通用插座”说明的均为实验台上已固定电路。
- 2、实验电路连线在图中均用虚线表示, 实线为已连好电路。
- 3、所以实验程序均加有延时子程序, 请老师根据实验PC机情况, 适当加减延时衡的延时时间即可。

## 实验一 I/O地址译码

### 一、实验目的

掌握I/O地址译码电路的工作原理。

### 二、实验原理和内容

实验电路如图1-1所示, 其中74LS74为D触发器, 可直接使用实验台上数字电路实验区的D触发器, 74LS138为地址译码器。译码输出端Y0~Y7在实验台上“I/O地址”输出端引出, 每个输出端包含8个地址, Y0: 280H~287H, Y1: 288H~28FH, …… 当CPU执行I/O指令且地址在280H~2BFH范围内, 译码器选中, 必有一根译码线输出负脉冲。

例如: 执行下面一条指令

```
PortWriteByte(0X2a0, 0X10);
```

Y4输出一个负脉冲, 执行下面一条指令

```
PortWriteByte(0X2a8, 0X10);
```

Y5输出一个负脉冲。

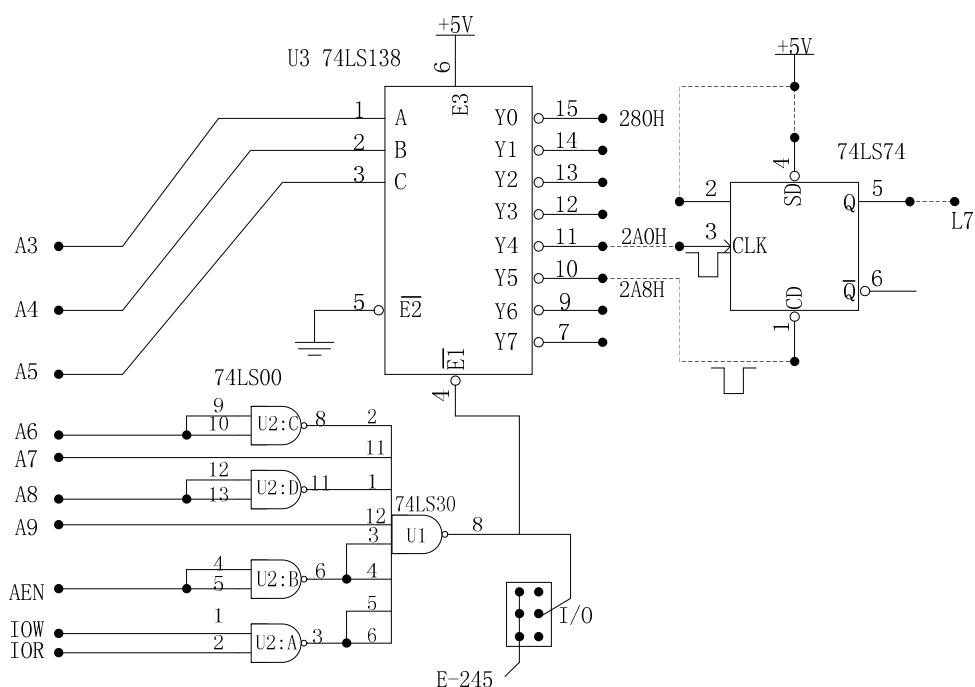


图1-1

利用这个负脉冲控制L7闪烁发光（亮、灭、亮、灭、……），时间间隔通过软件延时实现。

### 三、编程提示

1、实验电路中D触发器CLK端输入脉冲时，上升沿使Q端输出高电平L7发光，CD端加低电平L7灭。

2、参考程序：YMQ.CPP

```
#include<stdio.h>
#include<conio.h>
#include "..\\ApiEx.h"
#pragma comment(lib,"..\\ApiEx.lib")
void main()
{
    if(!Startup())
    {
        printf("ERROR: Open Device Error!\n");
        return;
    }
    printf("Press any key to exit!");
    while(!kbhit())
    {
        PortWriteByte(0x2a0, 0x10);
```

```

Sleep(1000);
PortWriteByte(0x2a8, 0x10);
Sleep(1000);
}
Cleanup();
}

```

## 实验二 简单并行接口

### 一、实验目的

掌握简单并行接口的工作原理及使用方法。

### 二、实验内容

1、按下面图2-1简单并行输出接口电路图连接线路(74LS273插通用插座, 74LS32用实验台上的“或门”)。74LS273为八D触发器, 8个D输入端分别接数据总线D0~D7, 8个Q输出端接LED显示电路L0~L7。

2、编程从键盘输入一个字符或数字, 将其ASC II 码通过这个输出接口输出, 根据8个发光二极管发光情况验证正确性。

3、按下面图2-2简单并行输入接口电路图连接电路(74LS244插通用插座, 74LS32用实验台上的“或门”)。74LS244为八缓冲器, 8个数据输入端分别接逻辑电平开关输出K0~K7, 8个数据输出端接数据总线D0~D7。

4、用逻辑电平开关预置某个字母的ASC II 码, 编程输入这个ASC II 码, 并将其对应字母在屏幕上显示出来。

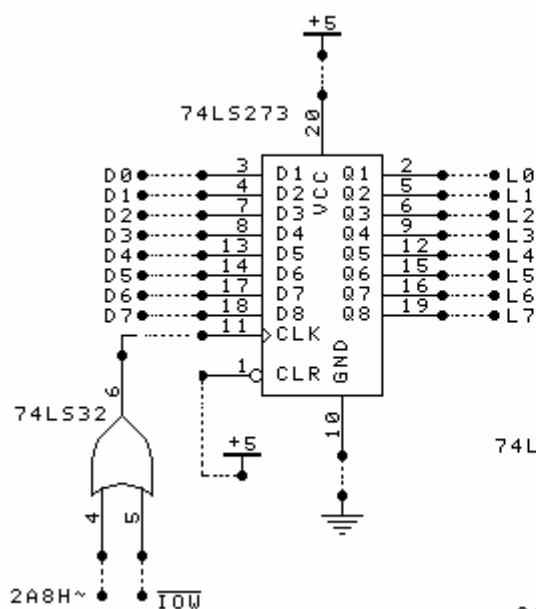


图2-1

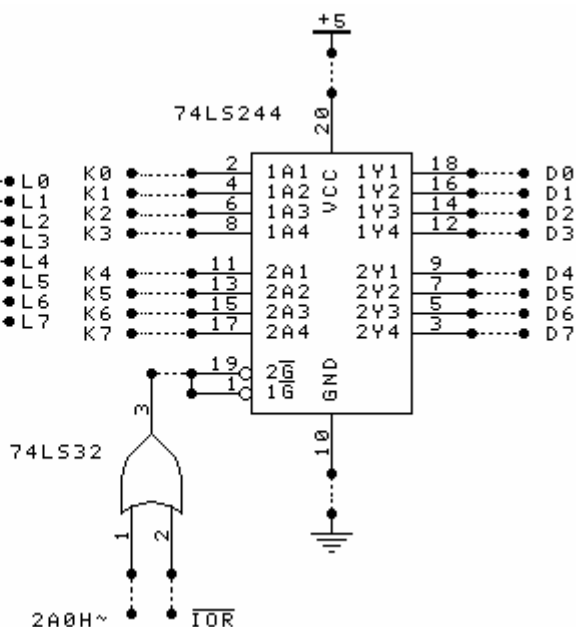


图2-2

### 三、编程提示

1、上述并行输出接口的地址为2A8H，并行输入接口的地址为2A0H，通过上述并行接口电路输出数据需要一条指令：

PortWriteByte(0X2a8, BYTE data);

通过上述并行接口输入数据需要一条指令：

PortWriteByte(0X2a0, & data);

#### 2、参考流程图

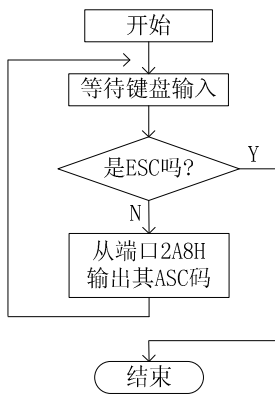


图2-3

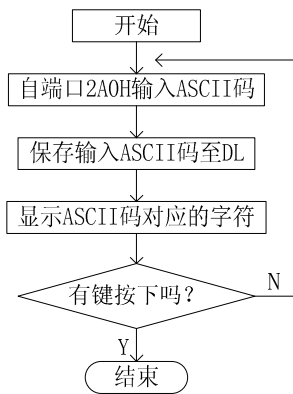


图2-4

## 实验三 可编程定时器 / 计数器（8253）

### 一、实验目的

掌握8253的基本工作原理和编程方法。

### 二、实验内容

1、按图3-1虚线连接电路，将计数器0设置为方式0，计数器初值为N( $N \leq 0FH$ )，用手动逐个输入单脉冲，编程使计数值在屏幕上显示，并同时用逻辑笔观察OUT0电平变化(当输入N+1个脉冲后OUT0变高电平)。

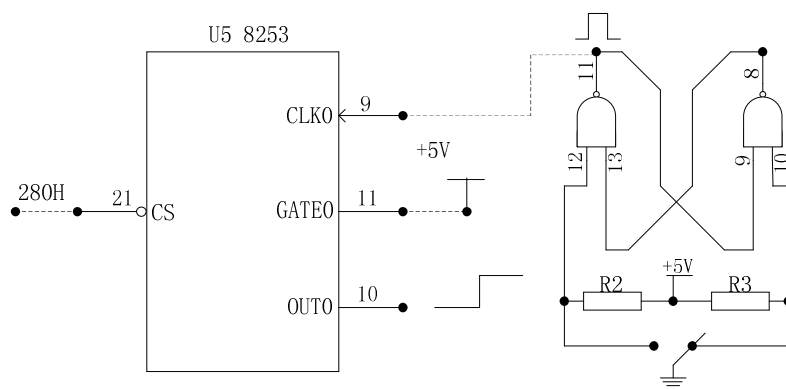


图3-1

2、按图3-2连接电路，将计数器0、计数器1分别设置为方式3，计数初值设为1000，用逻辑笔观察OUT1输出电平的变化(频率1HZ)。

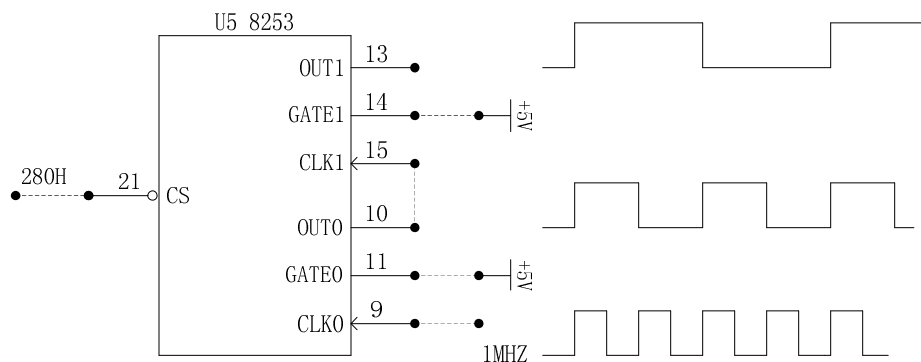


图3-2

### 三、 编程提示

- 1、8253控制寄存器地址      283H
- 计数器0地址                280H
- 计数器1地址                281H
- CLK0连接时钟               1MHZ

2、参考流程图(见图3-3、图3-4)：

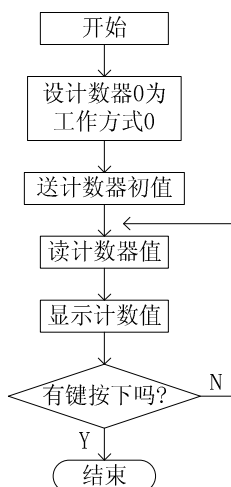


图3-3

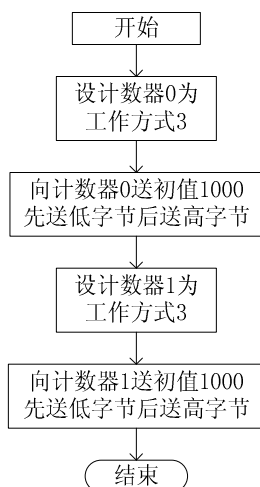


图3-4

## 实验四 可编程并行接口（一）（8255方式0）

### 一、实验目的

掌握8255方式0的工作原理及使用方法。

### 二、实验内容

- 1、实验电路如图4-1，8255C口接逻辑电平开关K0~K7，A口接LED显示电路L0~L7。
- 2、编程从8255C口输入数据，再从A口输出。

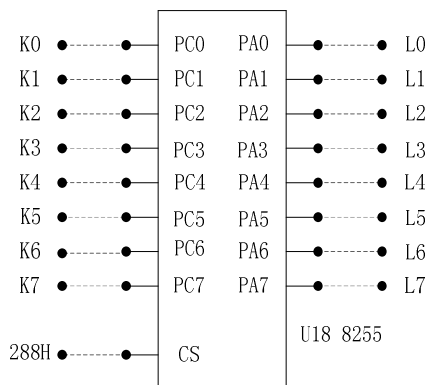


图4-1

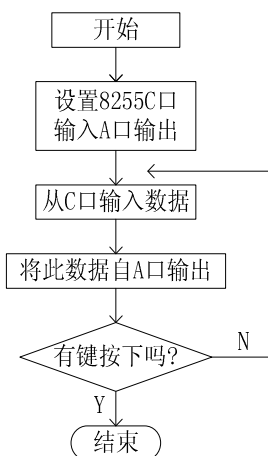


图4-2

### 三、编程提示

- |                 |      |
|-----------------|------|
| 1、8255控制寄存器端口地址 | 28BH |
| A口的地址           | 288H |
| C口的地址           | 28AH |

2、参考流程图(见图4-2)：

## 实验五 七段数码管

### 一、实验目的

掌握数码管显示数字的原理

### 二、实验内容

- 1、静态显示:按图5-1连接好电路,将8255的A口PA0~PA6分别与七段数码管的段码驱动输入端a~g 相连,位码驱动输入端S1接+5V(选中),S0、dp接地(关闭)。编程从键盘输入一位十进制数字(0~9),在七段数码管上显示出来。

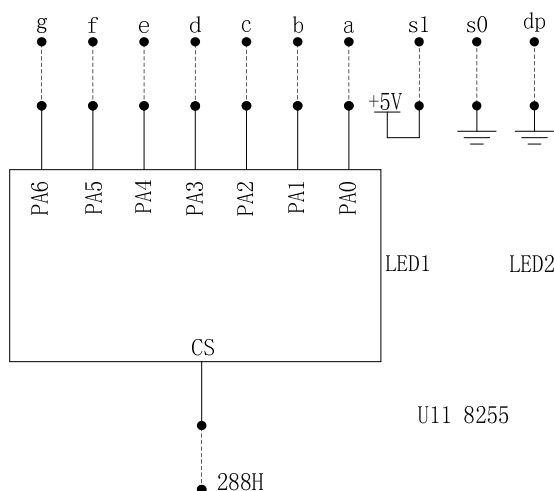


图5-1

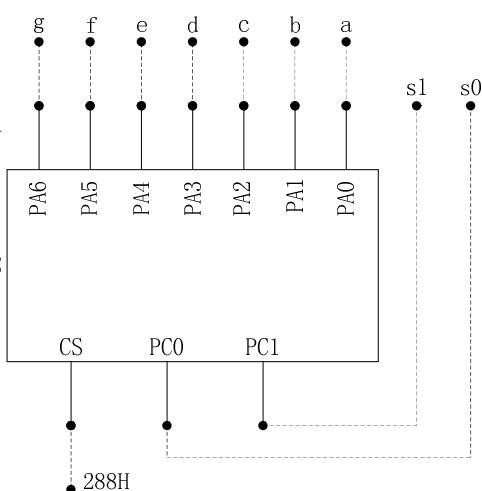


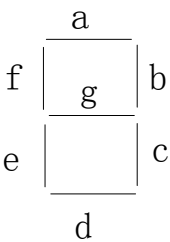
图5-2

- 2、动态显示:按图5-2连接好电路,七段数码管段码连接不变,位码驱动输入端S1, S0接8255 C口的PC1, PC0。编程在两个数码管上显示“56”。
- 3、动态显示(选作):使用图23的电路,编程在两个数码管上循环显示“00-99”。

### 三、编程提示

- 1、实验台上的七段数码管为共阴型,段码采用同相驱动,输入端加高电平,选中的数码管亮,位码加反相驱动器,位码输入端高电平选中。
- 2、七段数码管的字型代码表如下表:

显示字形	g	e	f	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh



3、参考流程图(见图5-3、图5-4)

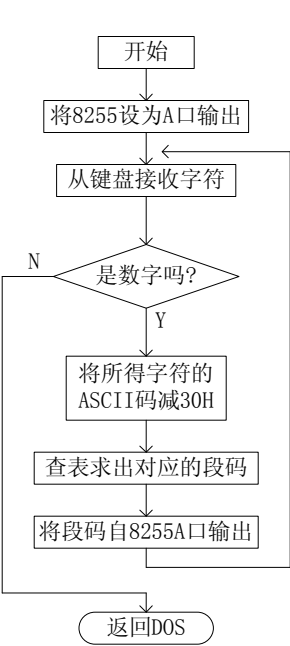


图5-3

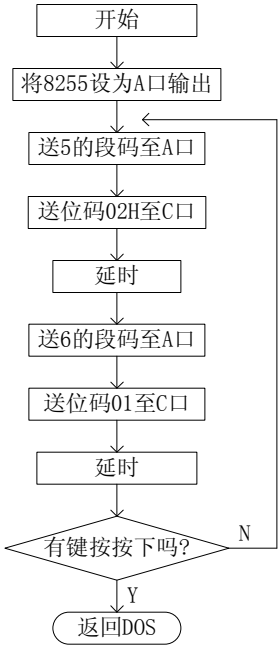


图5-4



## 实验六 继电器控制

### 一、实验目的

- 1、了解微机控制直流继电器的一般方法。
- 2、进一步熟悉使用8255、8253。

### 二、实验内容

实验电路如图6-1，按虚线连接电路：CLK0接1MHZ，GATE0，GATE1，接+5V，OUT0接CLK1，OUT1接PA0，PC0接继电器驱动电路的开关输入端Ik。编程使用8253定时，让继电器周而复始的闭合5秒钟(指示灯灯亮)，断开5秒钟(指示灯灯灭)。

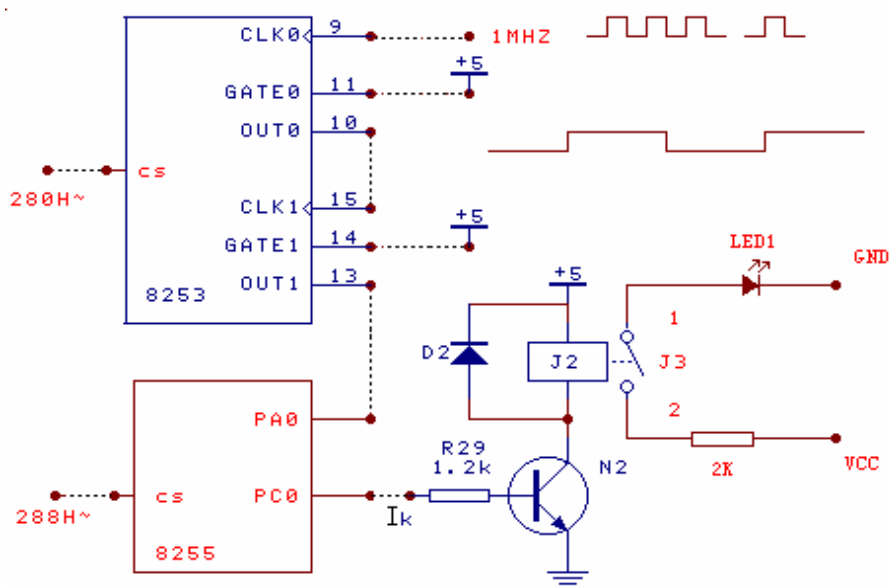


图6-1

### 三、编程提示

1、将8253计数器0设置为方式3、计数器1设置为方式0并联使用，CLK0接1MHZ时钟，设置两个计数器的初值(乘积为5000000)启动计数器工作后，经过5秒钟OUT1输出高电平。通过8255A口查询OUT1的输出电平，用C口PC0输出开关量控制继电器动作。

2、继电器开关量输入端输入“1”时，继电器常开触点闭合，电路接通，指示灯发亮，输入“0”时断开，指示灯熄灭。

3、参考流程图(见图6-2)：

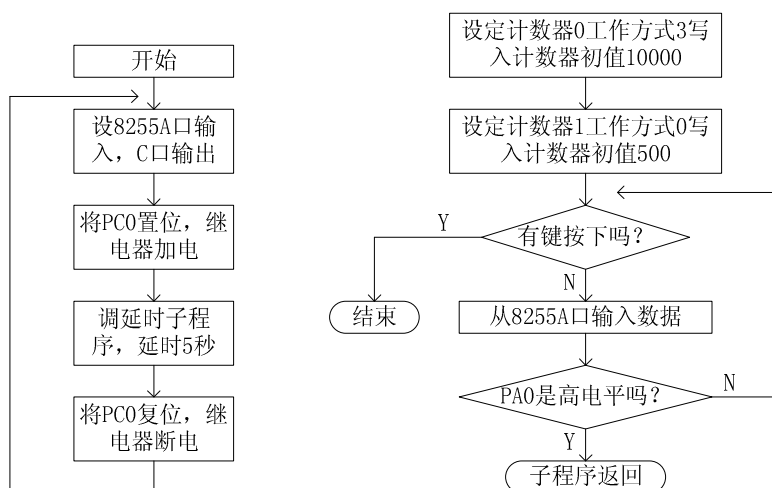


图6-2

## 实验七 竞赛抢答器

### 一、实验目的

- 1、了解微机化竞赛抢答器的基本原理。
- 2、进一步学习使用并行接口。

### 二、实验内容

图7-1为竞赛抢答器(模拟)的原理图, 逻辑开关K0~K7代表竞赛抢答按钮0~7号, 当某个逻辑电平开关置“1”时, 相当某组抢答按钮按下。在七段数码管上将其组号(0~7)显示出来, 并使微机扬声器响一下。

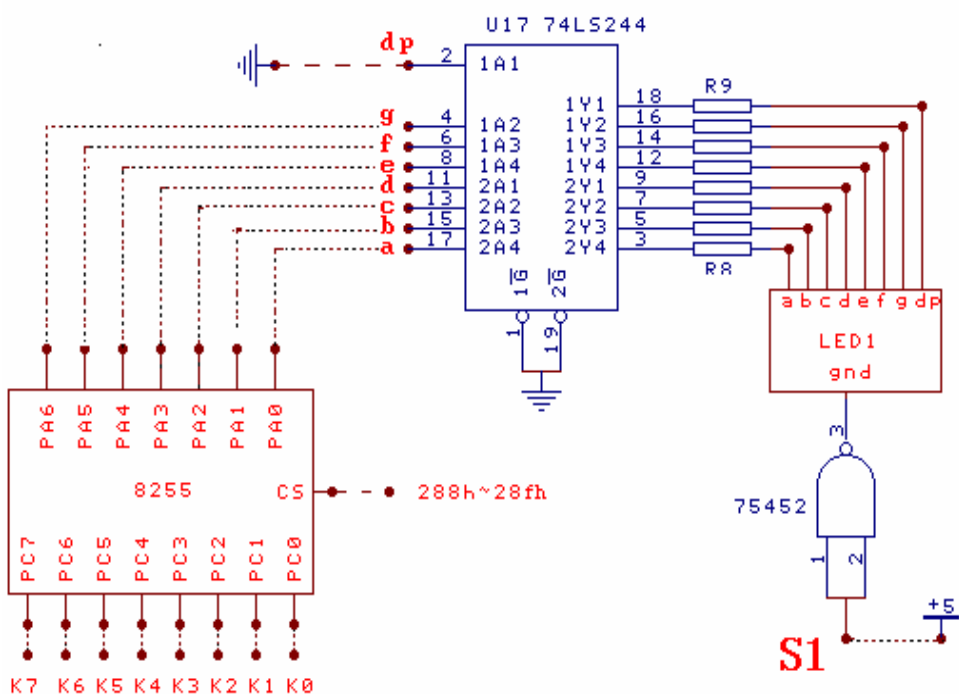


图7-1

### 三、编程提示

设置8255为C口输入、A口输出，读取C口数据，若为0表示无人抢答，若不为0则有人抢答。根据读取数据可判断其组号。从键盘上按空格键开始新一轮抢答，按其它键程序退出。

### 四、参考流程图（见图7-2）

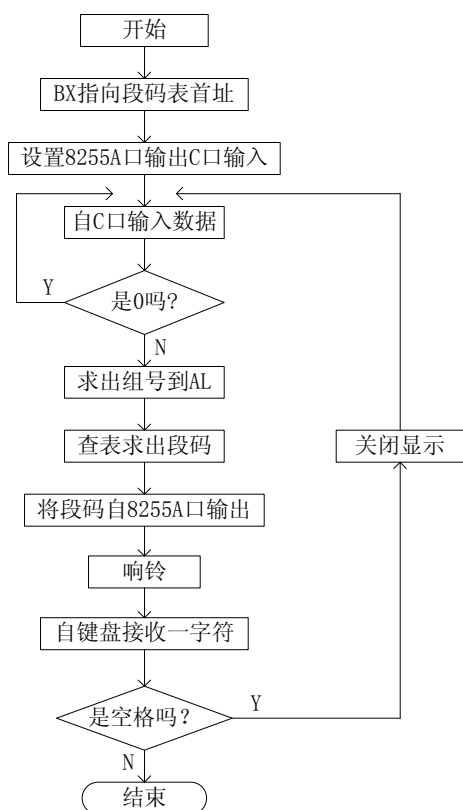


图7-2

## 实验八 交通灯控制实验

### 一、实验目的

通过并行接口8255实现十字路口交通灯的模拟控制, 进一步掌握对并行口的使用。

### 二、实验内容

如图8-1, L7、L6、L5作为南北路口的交通灯与PC7、PC6、PC5相连, L2、L1、L0作为东西路口的交通灯与PC2、PC1、PC0相连。编程使六个灯按交通灯变化规律亮灭。

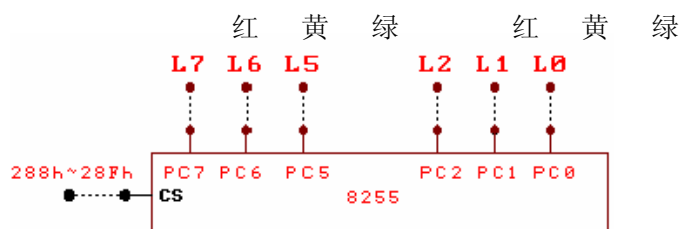


图8-1

### 三、编程提示：十字路口交通灯的变化规律要求：

- (1) 南北路口的绿灯、东西路口的红灯同时亮30秒左右。
- (2) 南北路口的黄灯闪烁若干次，同时东西路口的红灯继续亮。
- (3) 南北路口的红灯、东西路口的绿灯同时亮30秒左右。
- (4) 南北路口的红灯继续亮、同时东西路口的黄灯亮闪烁若干次。
- (5) 转(1)重复。

### 四、参考流程图

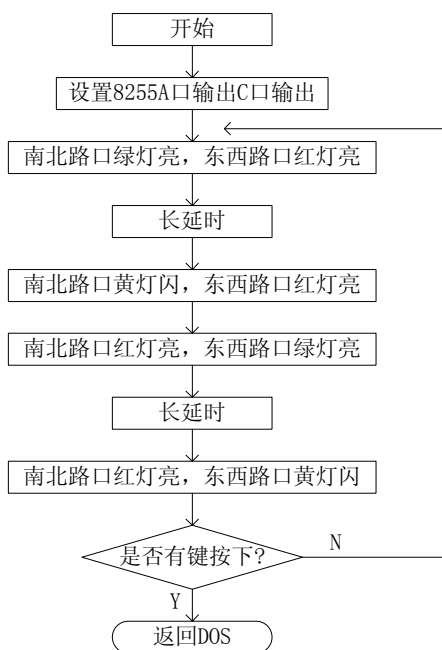


图8-2

## 2、中断-中断函数简介

### 1、EnableIntr;

语法:BOOL EnableIntr();

功能描述:将微机实验装置的中断输入设为有效,执行此函数后,PLX9054将接受微机实验装置上的中断请求,然后根据该请求申请一个PCI中断。

参数:无

返回值:如果成功,则返回True,否则返回False

备注:应用程序在调用该函数之前,必须先调用Startup函数。

### 2、DisableIntr;

语法:BOOL DisableIntr();

功能描述:将微机实验装置的中断输入设为无效,执行此函数后,PLX9054将不相应微机实验装置上的中断请求

参数:无

返回值:如果成功,则返回True,否则返回False

备注:应用程序在调用该函数之前,必须先调用Startup函数。

### 3、RegisterISR;

语法:BOOL RegisterISR(ISR\_ROUTINE pfuncISR);

功能描述:注册中断服务程序,当微机实验箱上的中断输入有效时,且实验箱上的中断输入使能,程序将会执行该中断服务程序。

参数:pfuncISR:该参数即为中断服务函数名

返回值:如果成功,则返回True,否则返回False

备注:应用程序在调用该函数之前,必须先调用Startup函数。

## 实验九 中断

### 一、实验目的

- 1、了解Windows下中断处理过程。
- 2、比较中断和查询两种数据交换方法的效率差别。

### 二、实验内容

用查询和中断方式分别实现控制指示灯,实验电路如图9-1。要求直接用手动产生的单脉冲作为中断请求信号,每按一次单脉冲产生一次中断,让指示灯显示一秒钟的0x55,否则让指示灯显示0xAA。然后在任务栏比较中断和查询方式下CPU利用率的差别。

#### 1、用查询方法

将8255的A口设为输出接指示灯,C口设为输入,将PC0接正脉冲输入,线路连接如下图所示:

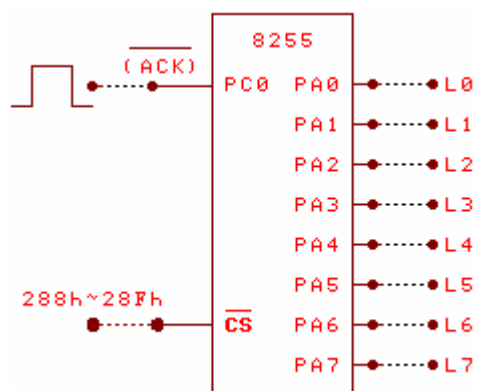


图9-1

## 2、用中断方法

将8255的A口设为输出，IRQ直接接到正脉冲

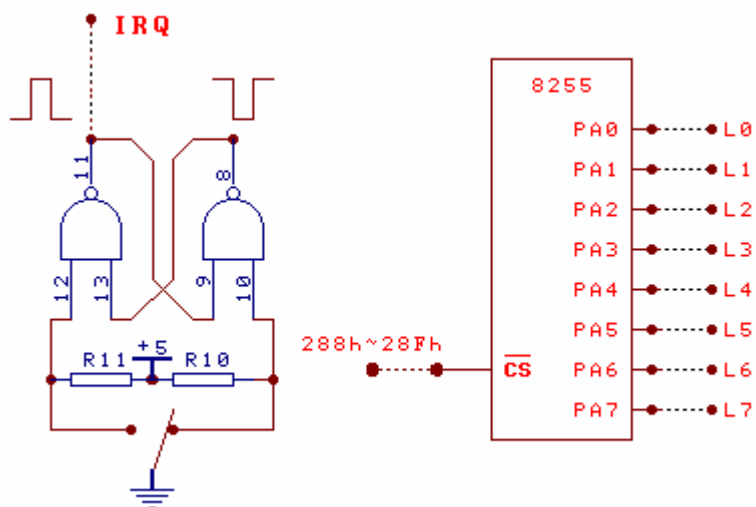


图9-2

## 三、编程提示

### 1、查询方式流程图：

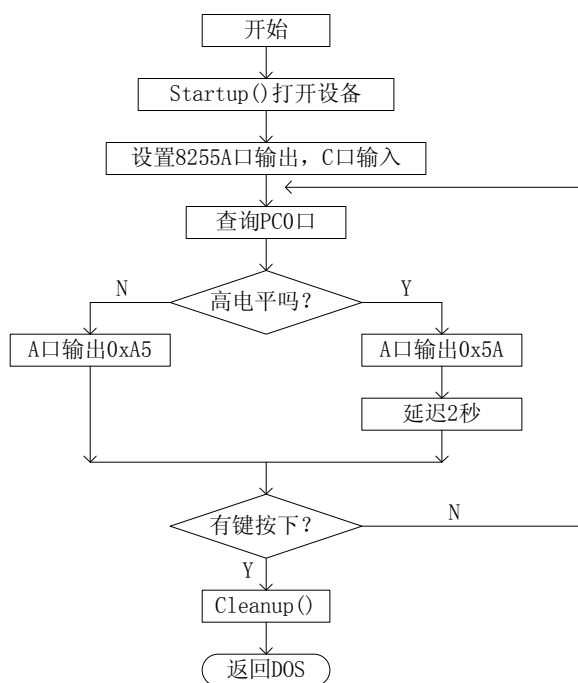


图9-3

## 2、中断方式流程图：

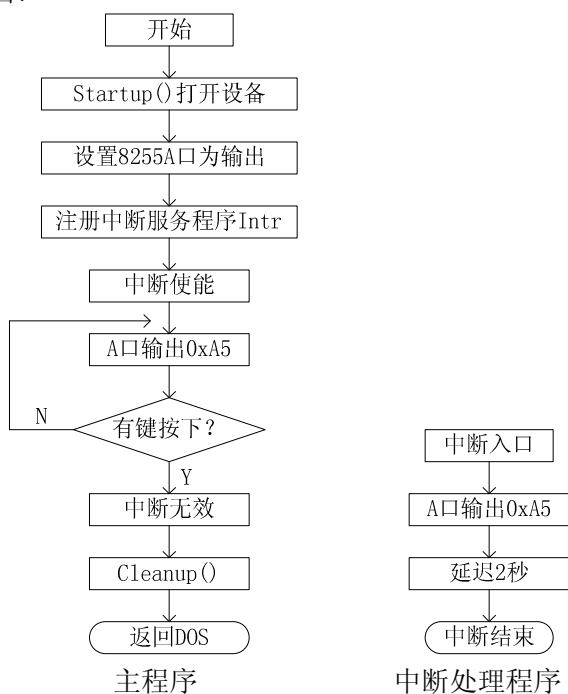


图9-4



## 实验十 可编程并行接口（二）（8255方式1）

### 一、实验目的

- 1、掌握8255工作方式1时的使用及编程。
- 2、进一步掌握中断处理程序的编写。

### 二、实验内容

- 1、按图10-1，8255方式1的输出电路连好线路。
- 2、编程：每按一次单脉冲按钮产生一个正脉冲使8255产生一次中断请求，让CPU进行一次中断服务：依次输出01H、02H、04H、08H、10H、20H、40H、80H使L0~L7依次发光，中断8次结束。
- 3、按图10-2，8255方式1输入电路，连好线路。

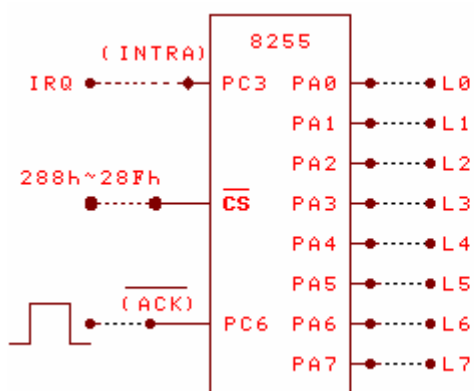


图10-1 输出电路

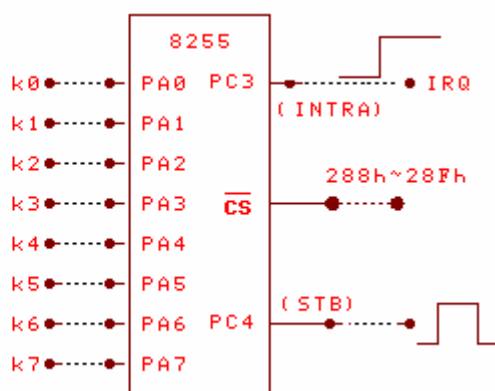


图10-2 输入电路

- 4、编程：每按一次单脉冲按钮产生一个正脉冲使8255产生一次中断请求，让CPU进行一次中断服务：读取逻辑电平开关预置的ASCII码，在屏幕上显示其对应的字符，中断8次结束。

### 三、参考流程图：（如图10-3，图10-4）

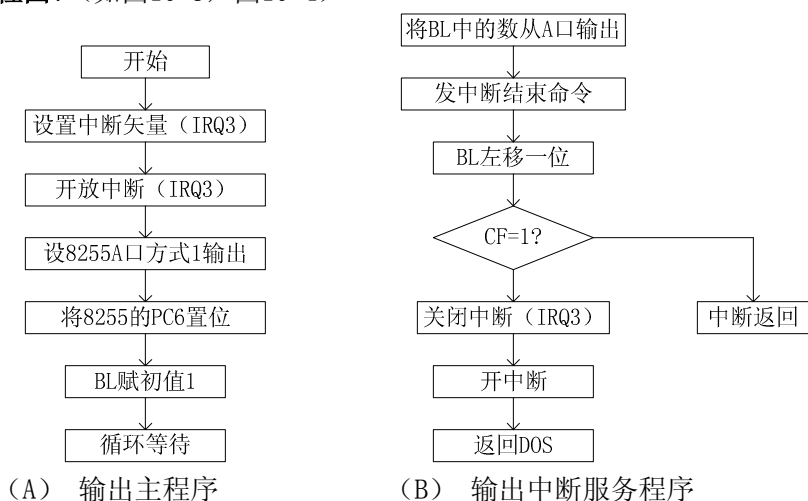


图10-3

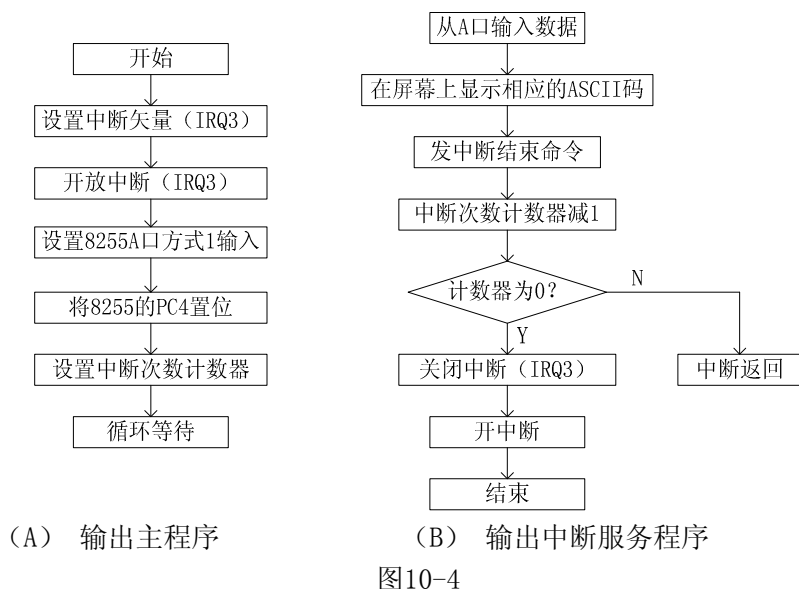


图10-4

## 实验十一 数/模转换器

### 一、实验目的

了解数/模转换器的基本原理，掌握DAC0832芯片的使用方法。

### 二、实验内容

1、实验电路原理如图11-1，DAC0832采用单缓冲方式，具有单双极性输入端(图中的U<sub>a</sub>、U<sub>b</sub>)，利用debug输出命令(Out 290 数据)输出数据给DAC0832，用万用表测量单极性输出端U<sub>a</sub>及双极性输出端U<sub>b</sub>的电压，验证数字与电压之间的线性关系。

2、编程产生以下波形(从U<sub>b</sub>输出，用示波器观察)

- (1) 锯齿波
- (2) 正弦波

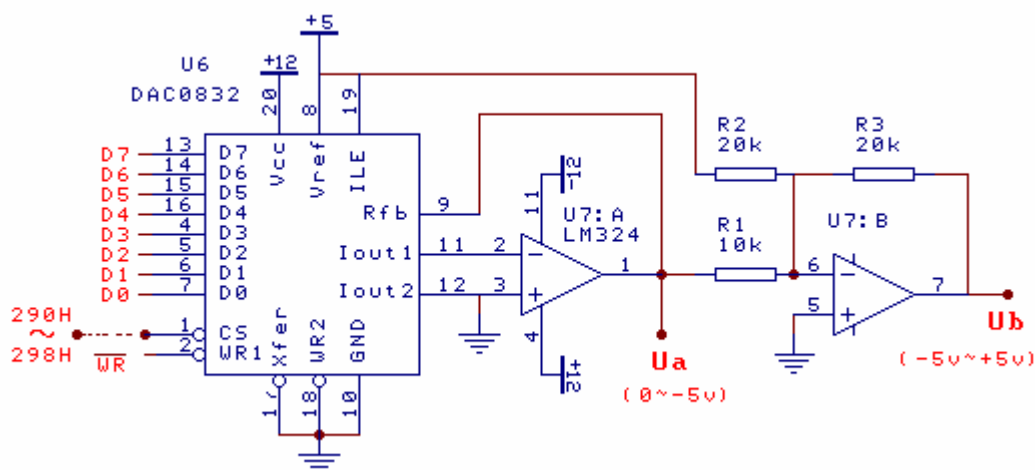


图11-1

### 三、编程提示

1、8位D/A转换器DAC0832的口地址为290H，输入数据与输出电压的关系为：

$$U_a = -\frac{U_{REF}}{256} \times N$$

$$U_b = -\frac{2U_{REF}}{256} \times N - 5$$

(UREF表示参考电压,N表示数数据)，这里的参考电压为P C机的+ 5 V 电源。

2、产生锯齿波只须将输出到DAC0832的数据由0循环递增。产生正弦波可根据正弦函数建一个下弦数字量表，取值范围为一个周期，表中数据个数在16个以上。

### 四、参考流程图：

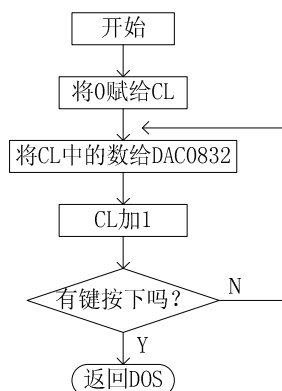


图11-2 锯齿波

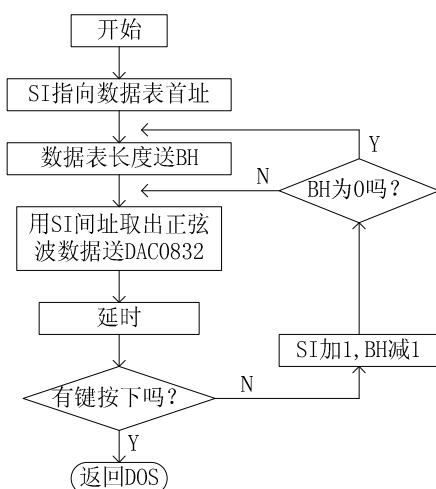
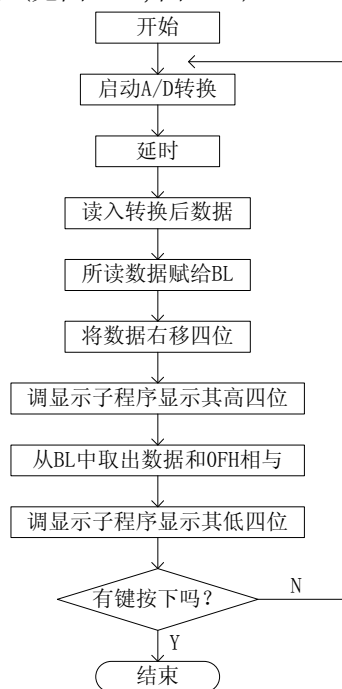


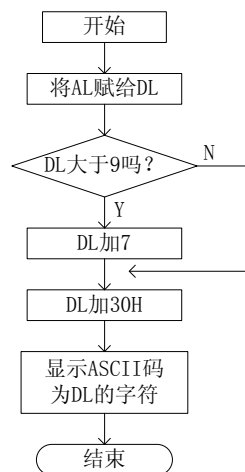
图11-3 正弦波



#### 四、参考流程图（见图12-2, 图12-3）



(A) 主程序



(B) 显示子程序

图12-2

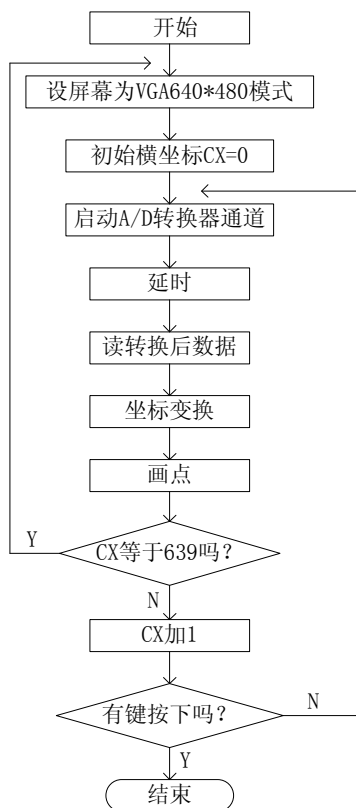


图12-3

## 实验十三 串行通讯

### 一、实验目的

- 1、了解串行通讯的基本原理。
- 2、掌握串行接口芯片8251的工作原理和编程方法。

### 二、实验内容

- 1、按图13-1连接好电路, (8251插通用插座) 其中8253计数器用于产生8251的发送和接收时钟, TXD和RXD连在一起。
- 2、编程: 从键盘输入一个字符, 将其ASCII码加 1 后发送出去, 再接收回来在屏幕上显示, 实现自发自收。

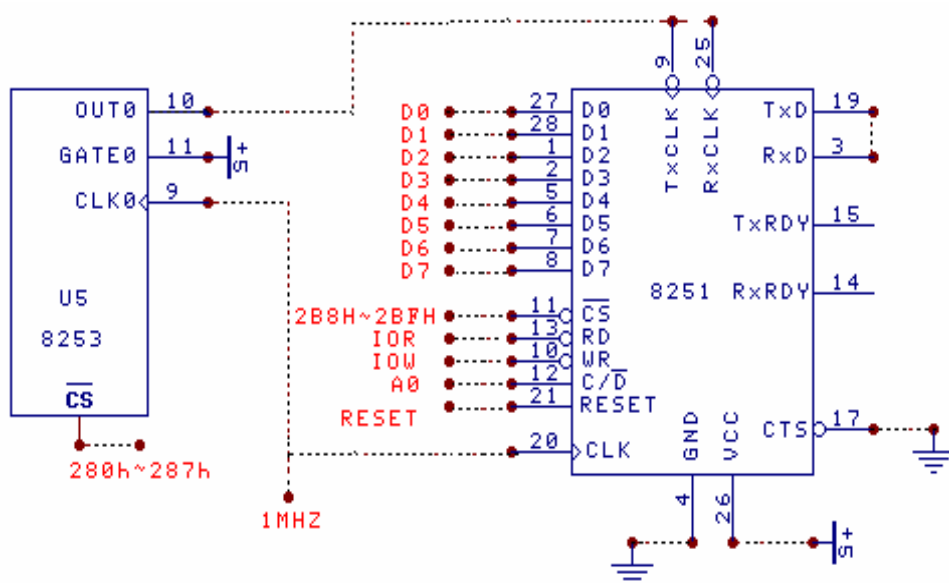


图13-1 串行通讯电路

### 三、实验提示

- 1、图示电路8251的控制口地址为2B9H, 数据口地址为2B8H。
- 2、8253计数器的计数初值=时钟频率/(波特率×波特率因子), 这里的时钟频率接1MHz, 波特率若选1200, 波特率因子若选16, 则计数器初值为52。
- 3、收发采用查询方式。

### 四、参考流程图 (见图13-2)

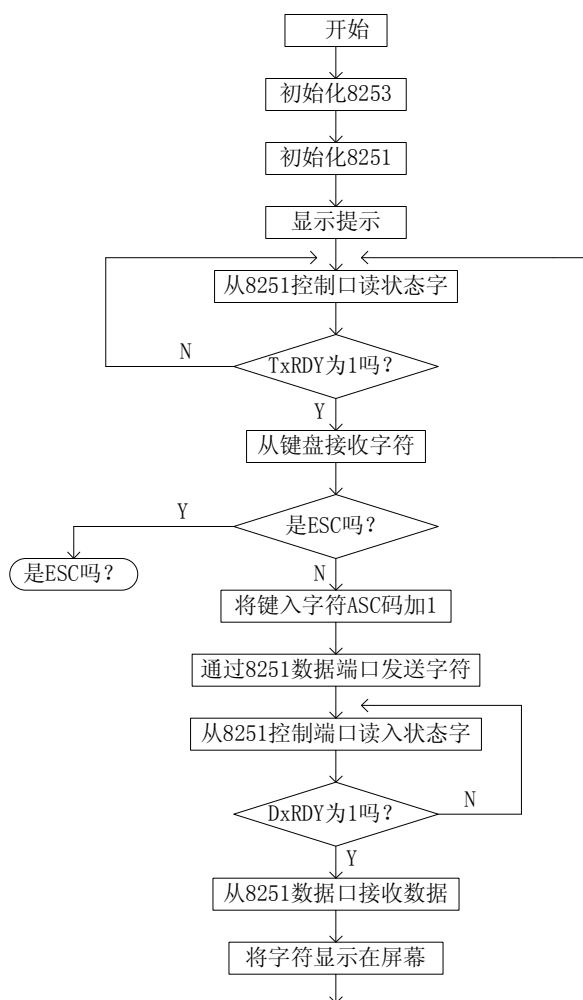


图14-2

### 3、DMA及RAM操作函数简介

#### 1、Write8237

语法: `bool Write8237(WORD address, BYTE data);`

功能描述: 写TPC-USB模块上8237的某个端口。

参数:

address: 指明要写的8237端口地址

data: 该函数执行完后, data将被写入address所指明的8237端口

返回值: 如果读成功, 则返回True, 否则返回False。

备注: 应用程序使用该函数前必须先调用Startup函数。

#### 2、Read8237

语法: `bool Read8237(WORD address, BYTE* pdata);`

功能描述: 读TPC-USB模块上8237某个端口值。

参数:

address: 指明要读的8237端口地址。

pdata: 该函数执行完后, address所指明的端口值被填入该地址。

返回值: 如果读成功, 则返回True, 否则返回False。

备注: 应用程序使用该函数前必须先调用Startup函数。

#### 3、MemReadByte

语法: `bool MemReadByte(WORD address, BYTE *pdata);`

功能描述: 读存储器。

参数:

address: 指明要读的存储器地址。

pdata: 该函数执行完后, address所指明的端口值被填入该地址。

返回值: 如果读成功, 则返回True, 否则返回False。

备注: 应用程序使用该函数前必须先调用Startup函数。

#### 4、MemWriteByte

语法: `bool MemWriteByte(WORD address, BYTE data);`

功能描述: 写存储器。

参数:

address: 指明要写的存储器地址。

pdata: 该函数执行完后, data将被填入address所指明的存储器地址。

返回值: 如果读成功, 则返回True, 否则返回False。

备注: 应用程序使用该函数前必须先调用Startup函数。



## 实验十四 DMA传送

### 一、实验目的

- 1、掌握PC机工作环境下进行DMA方式数据传送(Block MODE和Demand Mode)(块传送、外部请求传送)方法。
- 2、掌握DMA的编程方法。

### 二、实验内容

- 1、用通用插座按图14-1将6116电路连接好。编程将主机内存缓冲区D4000H, 偏移量为0的一块数据循环写入字符A~Z, 用Block MODE DMA方式传送到实验箱上的RAM6116上, 并察看送出的数据是否正确。其中扩展6116首地址为D6000H。

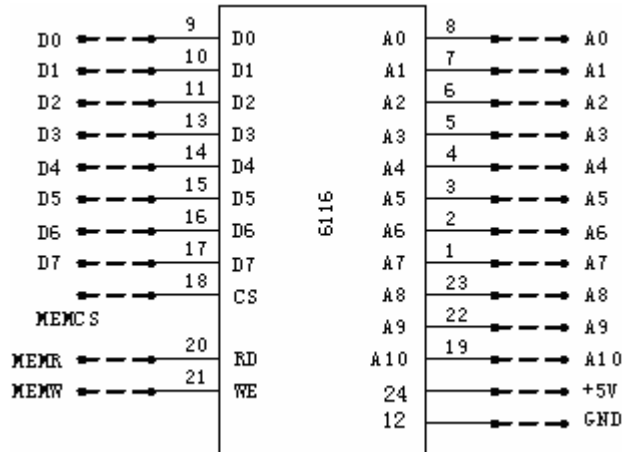


图14-1

- 2、用通用插座按图14-2连接好电路(74LS74利用实验台上的D触发器)。编程将主机内存缓冲区D4000H, 偏移量为0的10个数据, 使用Demand Mode DMA方式从内存向外设传送。

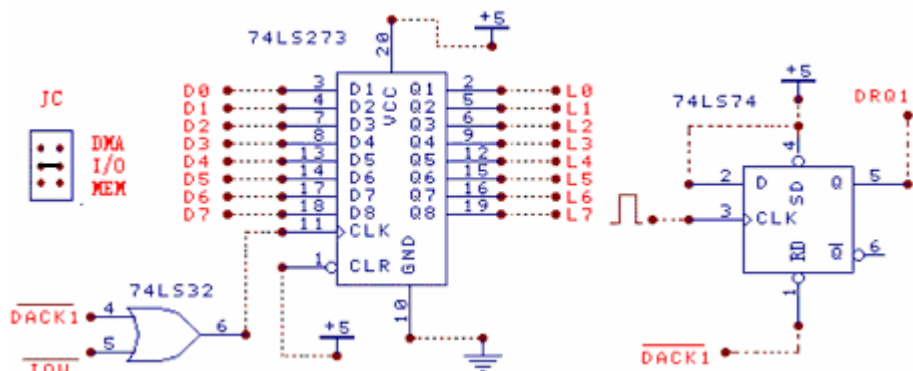


图14-2

- 3、用通用插座按图14-3连接好电路(74LS74利用实验台上的D触发器)。编程在主机内存缓冲区D4000H, 偏移量为0的位置开辟数据缓冲区, 使用Demand Mode DMA方式从外设向内存传送8个数据并存入数据缓冲区, 编程不断显示缓冲区的数据。

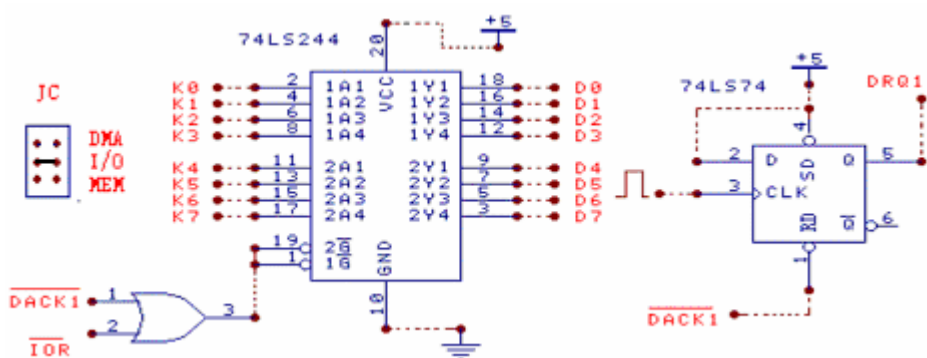


图14-3

### 三、实验提示

- 1、DMA 请求是由单脉冲输入到 D 触发器，由触发器的 Q 端向 DRQ1 发出的。CPU 响应后发出  $\overline{\text{DACK1}}$ ，将触发器 Q 置成低电平以撤消请求。
- 2、TPC-USB 模块上内存范围为 0D4000H-0D7fffH。

### 四、参考流程图（见图14-4、图14-5、图14-6）

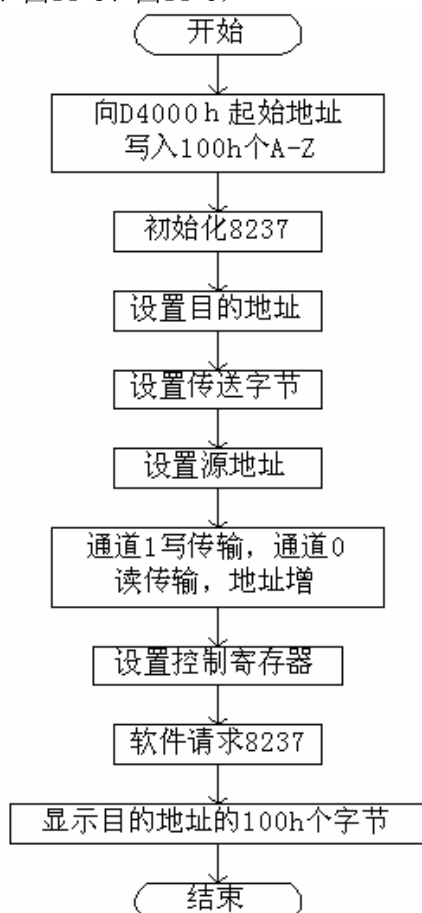


图14-4

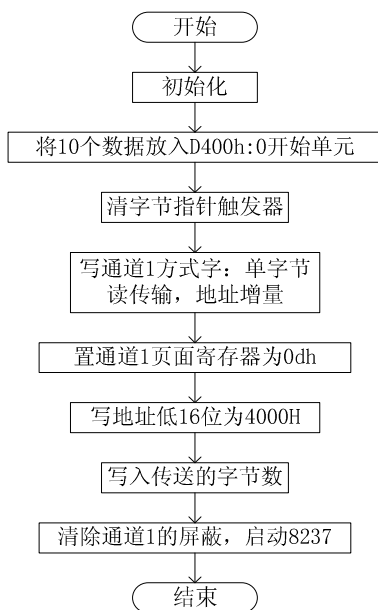


图14-5

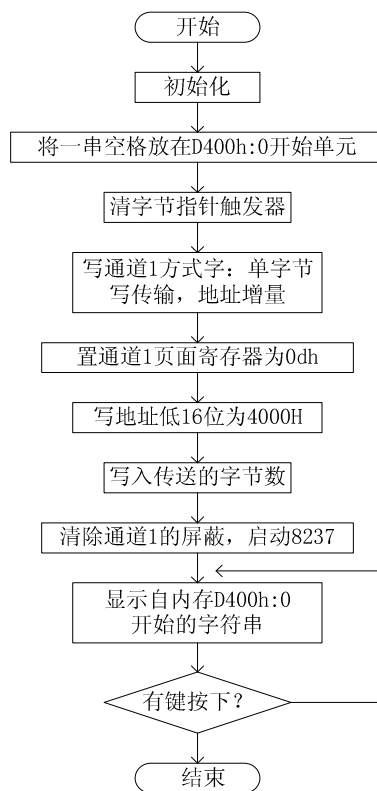


图14-6

## 实验十五 集成电路测试

### 一、实验目的

通过对四与非门(74LS00)集成电路芯片的测试,了解测试一般数字集成电路方法,进一步熟悉可编程并行接口8255 的使用。

### 二、实验内容

1、按图15-1连接电路,其中74LS00插在通用插座上。

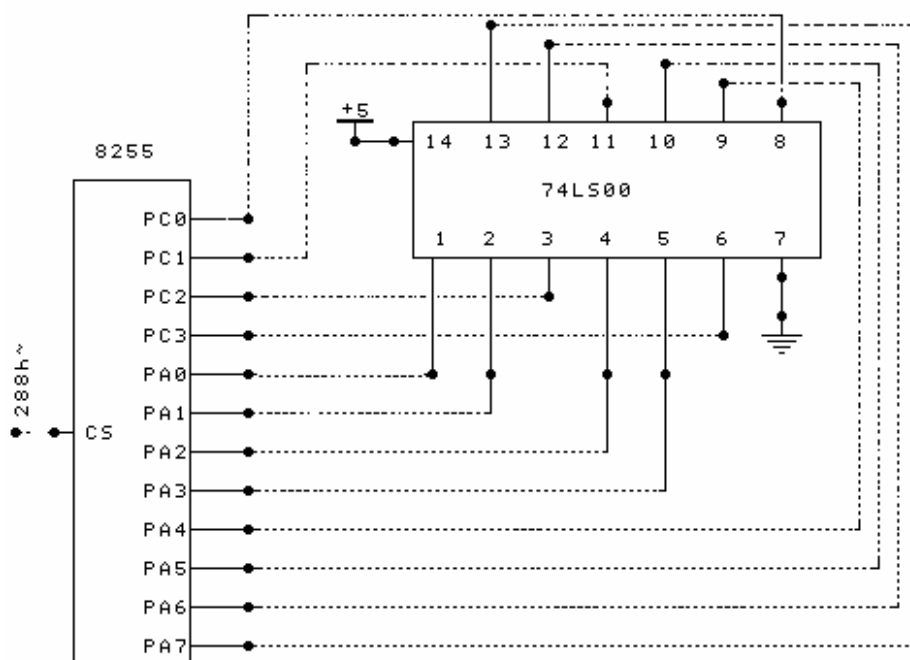


图15-1 集成电路测试电路图

### 2、编程提示:

将8255设置在方式0下工作,使A口输出,C口输入,先后通过A口分别给每一个门电路送入四个测试信号(00、01、10、11),相应从C口读出每一个门电路的输出结果,与正常值(1、1、1、0)进行比较,若相等,则此芯片好,否则为坏。

### 三、参考流程图 (见图15-2)

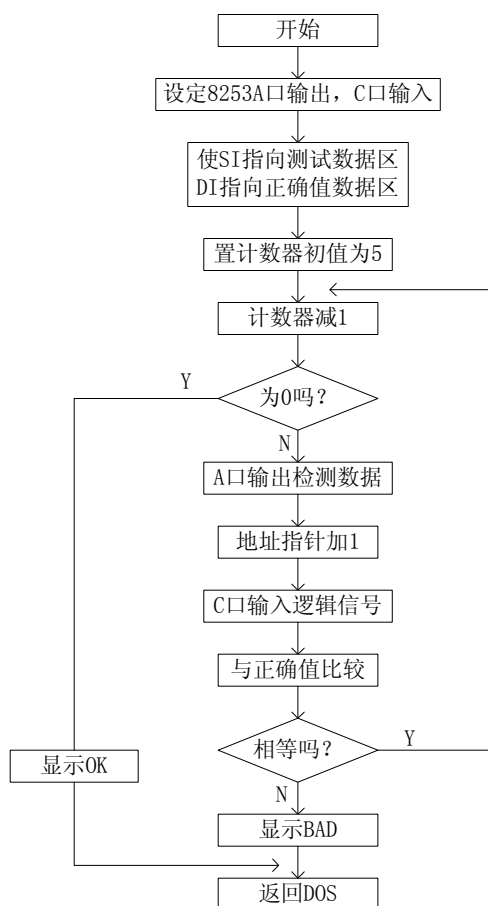


图15-2 集成电路测试流程图

## 实验十六 电子琴

### 一、实验目的

- 1、通过8253产生不同的频率信号，使PC机成为简易电子琴。
- 2、了解利用8255和8253产生音乐的基本方法。

### 二、实验内容

实验电路如图16-1，8253的CLK0接1MHZ时钟，GATE0接8255的PA1，OUT0和8255的PA0接到与门的两个输入端，K8跳线连接喇叭，编程使计算机的数字键1、2、3、4、5、6、7作为电子琴按键，按下即发出相应的音阶。

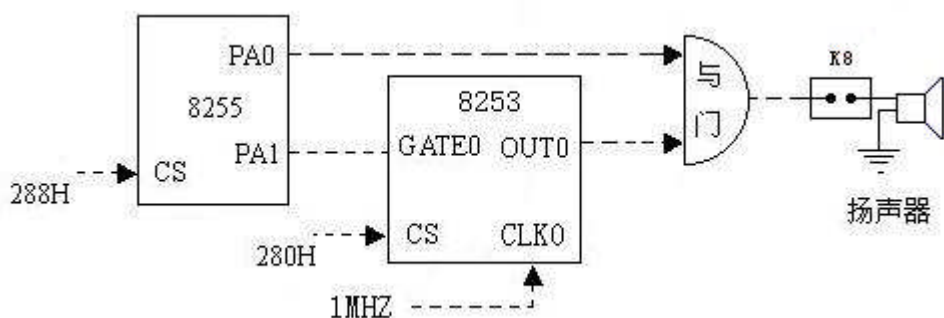


图16-1 电子琴电路

### 三、编程提示：

1、利用8255的PA0口来施加控制信号给与门，用来控制扬声器的开关状态。再利用设置不同的计数值，使8253产生不同频率的波形，使扬声器产生不同频率的音调，达到类似与音阶的高低音变换。对于音乐，每个音阶都有确定的频率。

各音阶标称频率值：

音 阶	1	2	3	4	5	6	7	1 <sub>·</sub>
低频率(单位:Hz)	262	294	330	347	392	440	494	524
高频率(单位:Hz)	524	588	660	698	784	880	988	1048

### 四、参考流程图 （见图16-2）

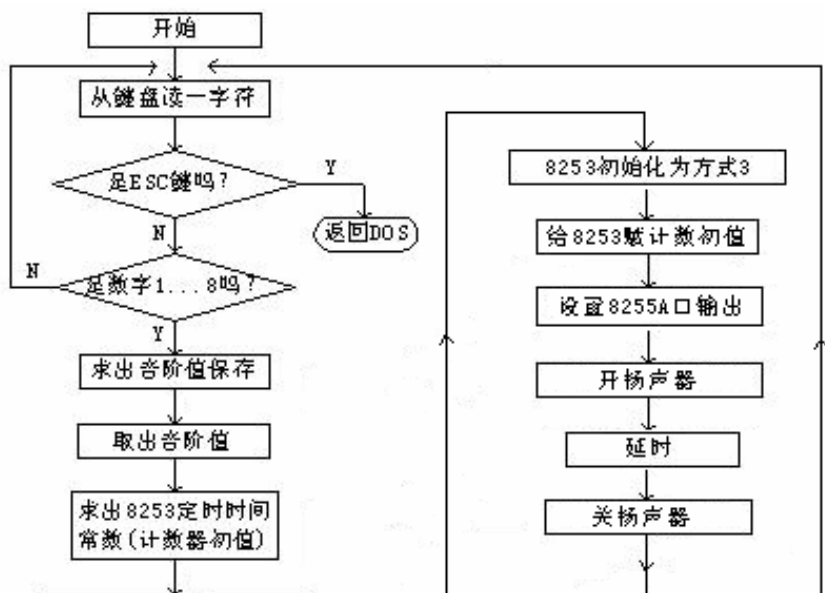


图16-2 主程序

## 实验十七 8250串行通讯实验

### 一、实验目的

- 1、进一步了解串行通信的基本原理。
- 2、掌握串行接口芯片8250的工作原理和编程方法。

### 二、实验内容

- 1、按图17-1连接线路，图中8250芯片插在通用插座上。
- 2、编程:从键盘输入一个字符，将其ASCII码加1后发送出去，再接收回来在屏幕上加1后的字符显示出来，实现自发自收。

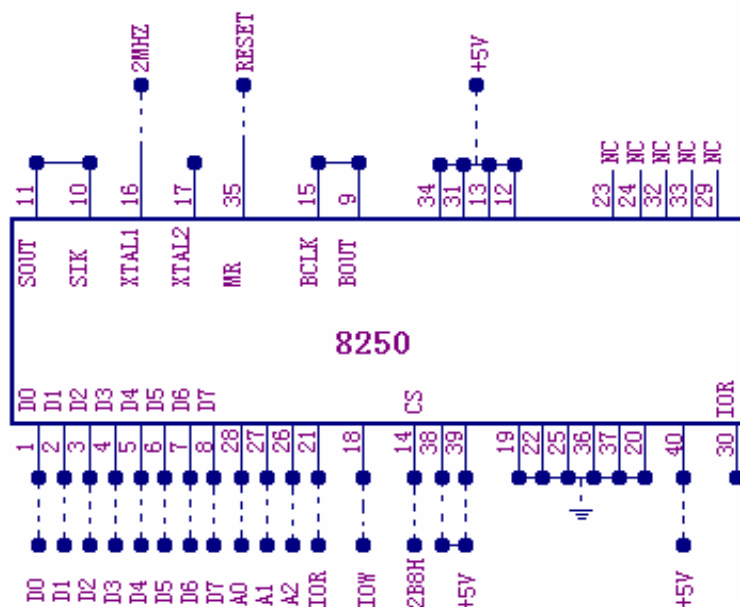


图17-1

### 三、实验提示

#### 1、8250介绍:

INC8250是一个可编程序异步通讯单元芯片，在微机系统中起串行数据的输入输出接口作用。此外，它还包含有可编程序波特率发生器，它可用1~65535的因子对输入时钟进行分频，以产生波特率十六倍的输入输出时钟。

2、8250时钟接2MHZ，若选波特率为9600，波特率因子为16，则因子寄存器中分频数为13。所以因子寄存器低字节送13，高字节为00H。

3、图中CS接02B8H~02BFH:

下表为各寄存器选择地址一览表。表中DLAB为线控制寄存器的最高位，也叫因子寄存器存取位。当DLAB为0时选接收数据缓冲器，发送数据寄存器和中断允许寄存器。当DLAB为1时选因子寄存器的低字节和高字节。

DLAB	A2	A1	A0	选中寄存器
0	0	0	0	接收缓冲器（读）发送保持寄存器（写）
0	0	0	1	中断允许寄存器
X	0	1	0	中断标志寄存器（仅用于读）
X	0	1	1	线控制寄存器
X	1	0	0	MODEM控制寄存器
X	1	0	1	线状态寄存器
X	1	1	0	MODEM状态寄存器
X	1	1	1	无
1	0	0	0	因子寄存器（低字节）
1	0	0	1	因子寄存器（高字节）

4、收发采用查询方式。

#### 四、参考流程

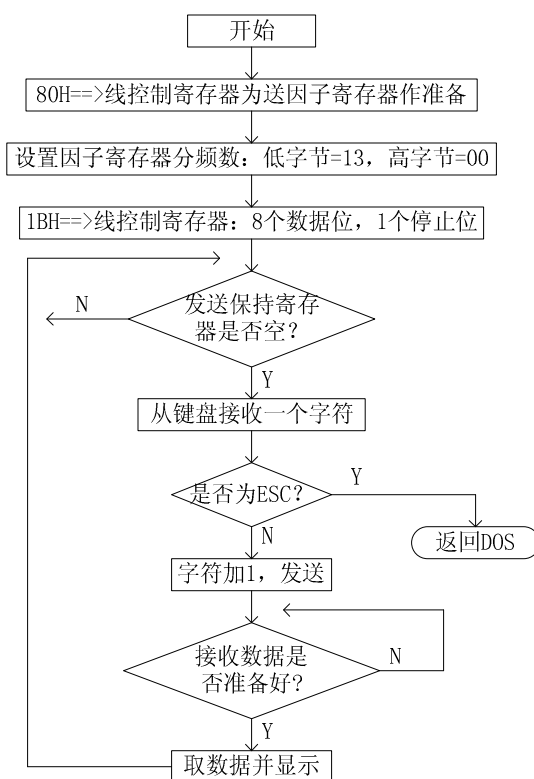


图17-2



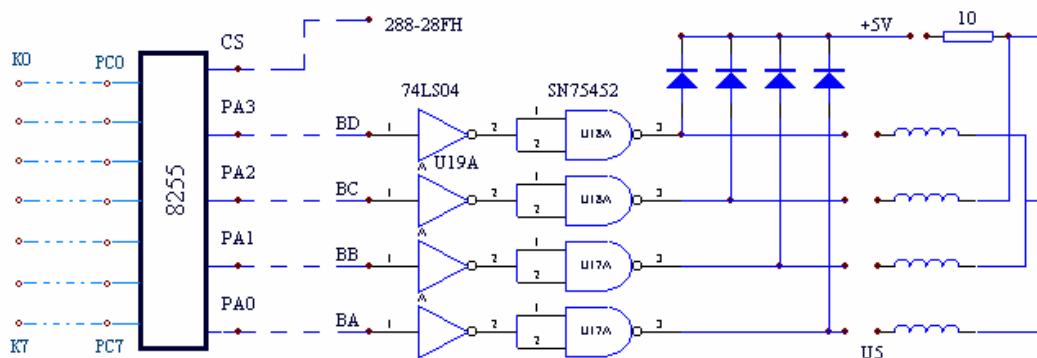
## 实验十八 步进电机控制实验

### 一、实验目的

- 1、了解步进电机控制的基本原理。
- 2、掌握控制步进电机转动的编程方法。

### 二、实验内容

- 1、按图18-1连接线路，利用8255输出脉冲序列，开关K0~K6控制步进电机转速，K7控制步进电机转向。8255 CS接288H~28FH。PA0~PA3接BA~BD；PC0~PC7接K0~K7。
- 2、编程:当K0~K6中某一开关为“1”（向上拨）时步进电机启动。K7向上拨电机正转，向下拨电机反转。



图

18-1

### 三、实验说明

步进电机驱动原理是通过对每相线圈中的电流的顺序切换来使电机作步进式旋转。驱动电路由脉冲信号来控制，所以调节脉冲信号的频率便可改变步进电机的转速。

如图18-2所示：本实验使用的步进电机用直流+5V电压，每相电流为0.16A，电机线圈由四相组成，即： $\phi 1$  (BA)； $\phi 2$  (BB)； $\phi 3$  (BC)； $\phi 4$  (BD)

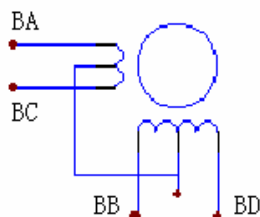
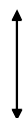


图19-2

驱动方式为二相激励方式，各线圈通电顺序如下表。

顺序 \ 相	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$
0	1	1	0	0
1	0	1	1	0
2	0	0	1	1
3	1	0	0	1

反时针方向回转



正时针方向回转

表中首先向  $\phi 1$  线圈— $\phi 2$  线圈输入驱动电流，接着  $\phi 2-\phi 3$ ， $\phi 3-\phi 4$ ， $\phi 4-\phi 1$ ，又返回到  $\phi 1-\phi 2$ ，按这种顺序切换，电机轴按顺时针方向旋转。

实验可通过不同长度延时来得到不同频率的步进电机输入脉冲，从而得到多种步进速度。

四、参考流程图

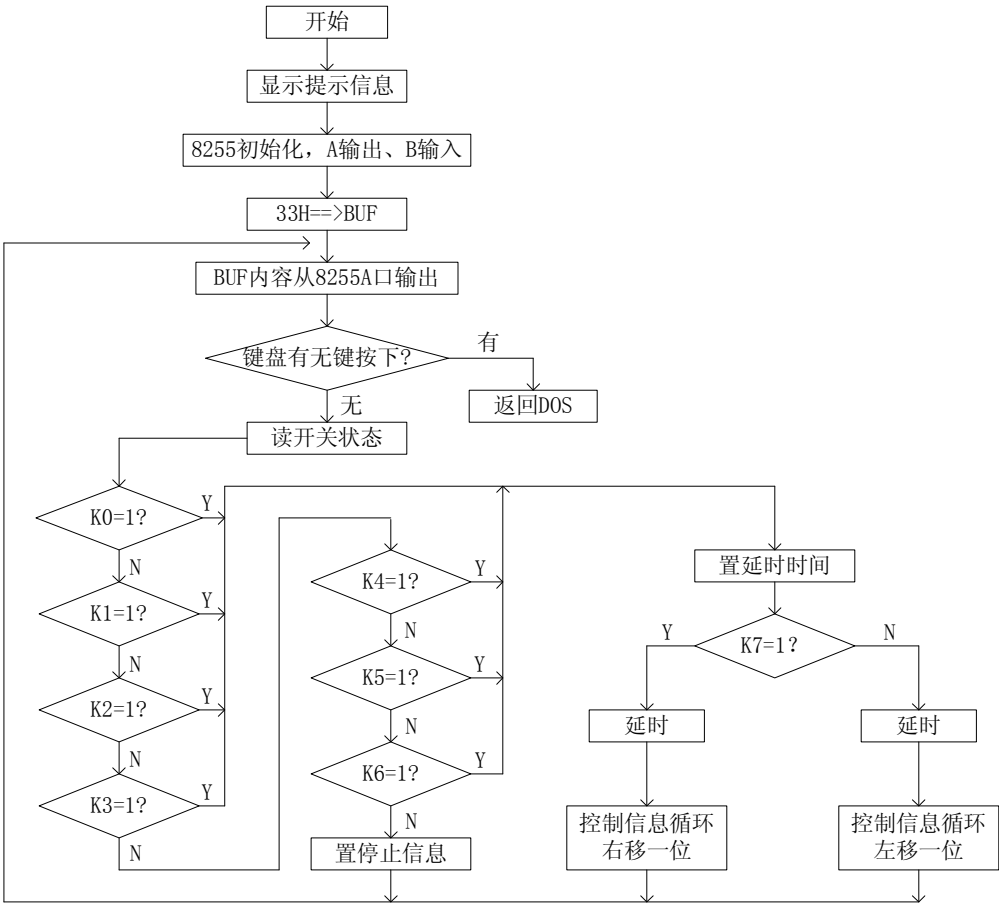


图18-3

## 实验十九 小直流电机转速控制实验

### 一、实验目的

- 1、进一步了解DAC0832的性能及编程方法。
- 2、了解直流电机控制的基本方法。

### 二、实验内容

- 1、按图19-1线路接线。DAC0832的CS接290H~297H，Ub接DJ插孔，8255 CS接288H~28FH。
- 2、编程利用DAC0832输出一串脉冲，经放大后驱动小直流电机，利用开关K0~K5控制改变输出脉冲的电平及持续时间，达到使电机加速，减速之目的。

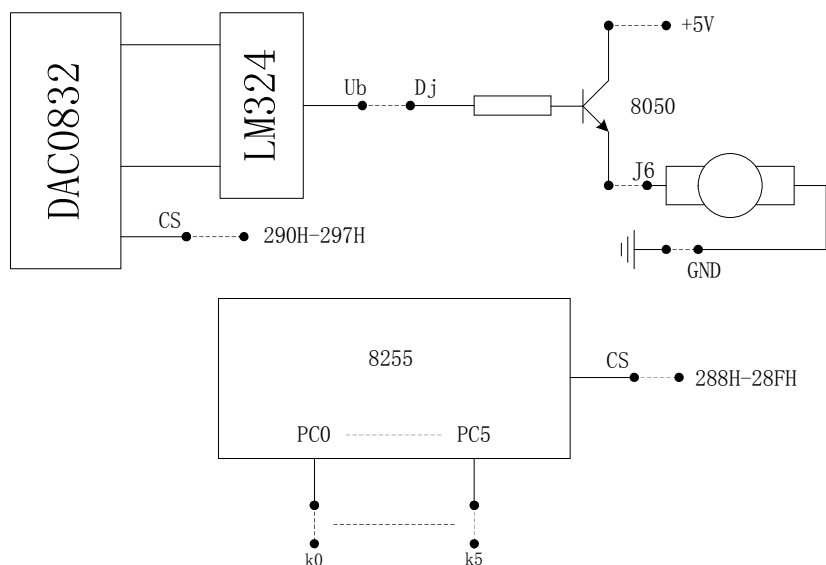


图19-1

### 三、实验原理简述

小直流电机的转速是由Ub输出脉冲的占空比来决定的，正向占空比越大转速越快，反之越慢。见下图19-2：

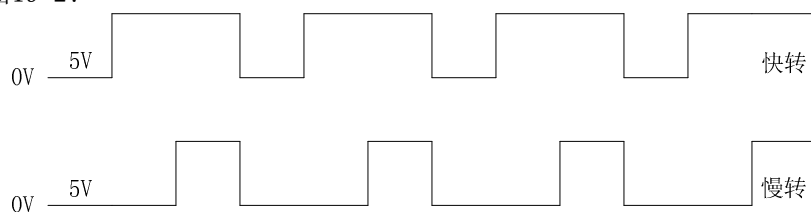


图19-2

在本实验中，模拟量输出Ub为双极性，当输入数字量小于80H时输出为负，输入等于80H时为0V，输入大于80H时输出为正。因而本实验中，DAC0832输入数字量只有2个(80H和FFH)，通过不同的延迟时间达到改变小电机转速的目的。

### 四、参考程序框图

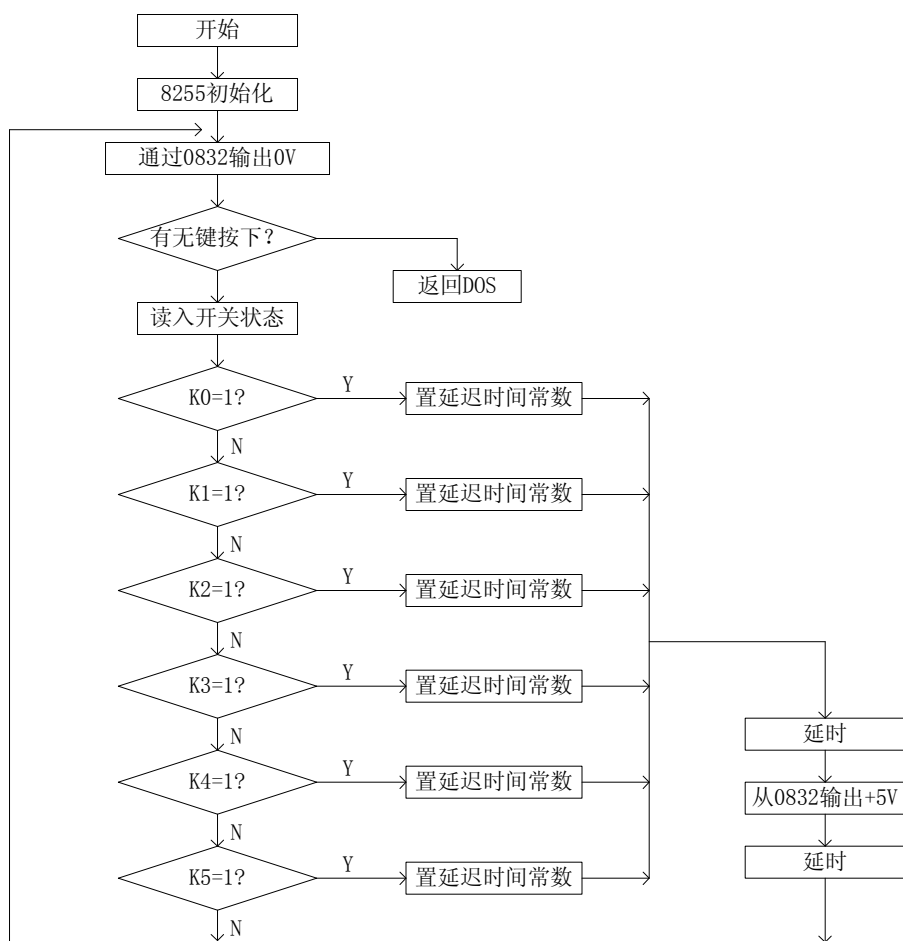


图19-3

## 实验二十 键盘显示控制器实验

### 一、实验目的

- 1、掌握8279键盘显示电路的基本功能及编程方法。
- 2、掌握一般键盘和显示电路的工作原理。
- 3、进一步掌握定时器的使用和中断处理程序的编程方法。

### 二、实验内容

1、本实验的实验电路如图20-1，它做在一块扩展电路板上，用一根20芯扁平电缆与实验台上扩展插头J7相连。

2、编程1:使得在小键盘上每按一个键，6位数码管上显示出相应字符，它们的对应关系如下：

小键盘		显示	小键盘		显示
0	—	0	C	—	C
1	—	1	D	—	d
2	—	2	E	—	E
3	—	3	F	—	F
4	—	4	G	—	q
5	—	5	M	—	□
6	—	6	P	—	p
7	—	7	W	—	U
8	—	8	X	—	
9	—	9	Y	—	4
A	—	⌂	R	—	返回
B	—	b			

### 3、编程2:中断编程

利用实验台上提供的定时器8253和扩展板上提供的8279以及键盘和数码显示电路，设计一个电子钟。由8253中断定时，小键盘控制电子钟的启停及初始值的预置。

电子钟显示格式如下：

XX. XX. XX. 由左向右分别为时、分、秒

要求具有如下功能：

- ①、C键:清除，显示全零。
- ②、G键:启动，电子钟计时。
- ③、D键:停止，电子钟停止计时。
- ④、P键:设置时、分、秒值。输入时依次为时、分、秒，同时应有判断输入错误的能力，若输入有错，则显示:E————。此时敲P键可重新输入预置值。
- ⑤、E键:程序退出。

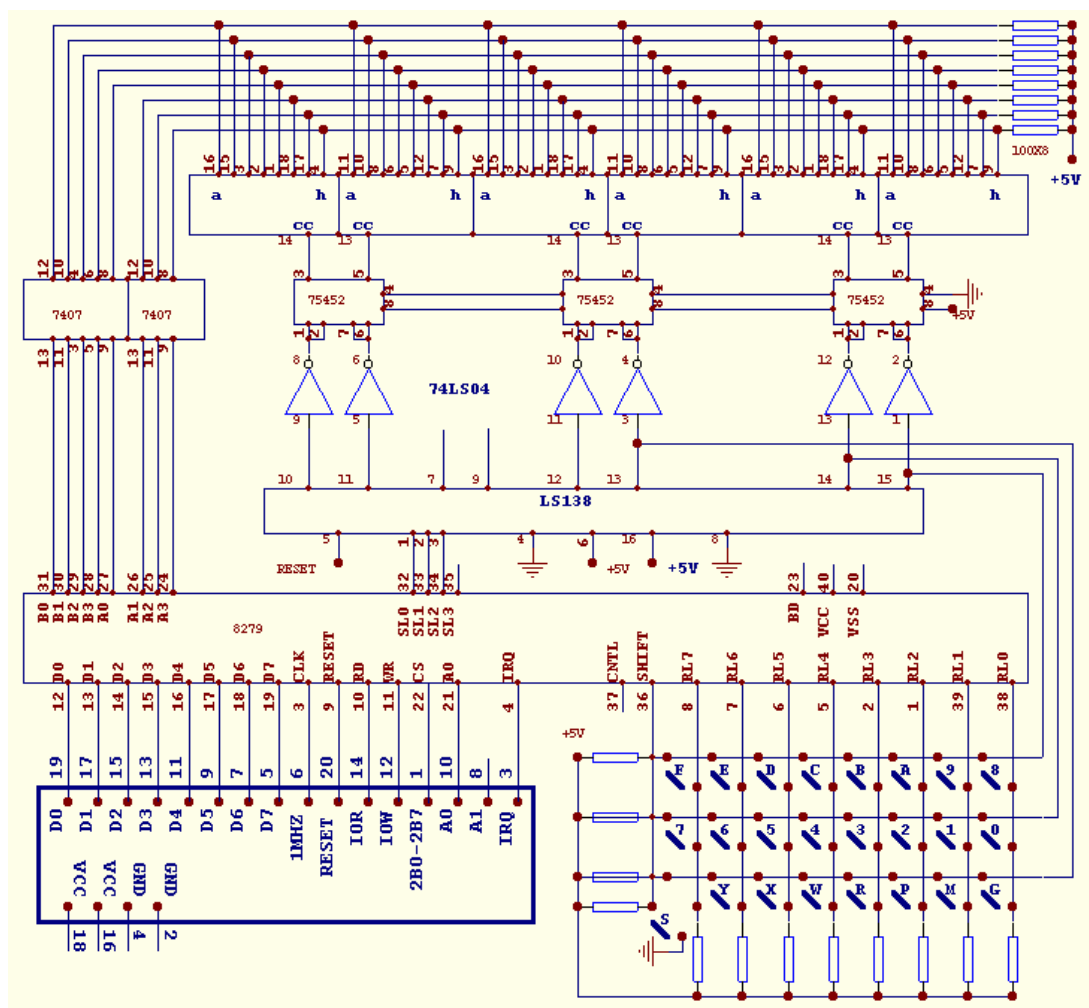


图20-1

### 三、编程1 接线方法

用一根20芯扁平电缆将实验扩展板与实验台上扩展插头J7相连。

### 四、编程2 接线方法

- 1、用一根20芯扁平电缆将实验扩展板与实验台上扩展插头J7相连。
- 2、实验台上8253 CLK0 接 1MHZ, GATE0 和 GATE1接+5V, OUT0 接 CLK1, OUT1 接 IRQ, CS接 280H~287H。

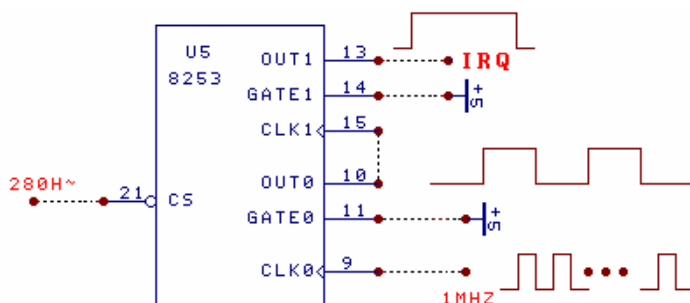
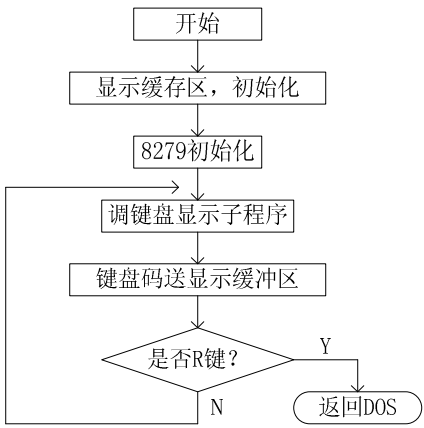
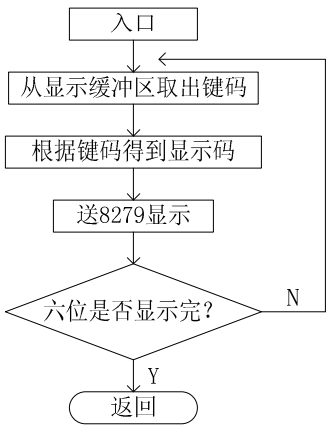


图20-2

五、编程1 参考流程



主程序流程图:



显示子程序流程图:DISP

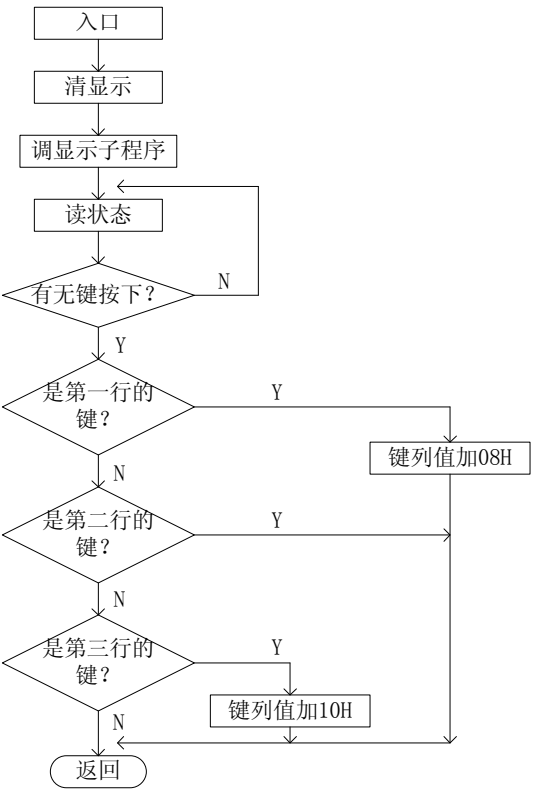
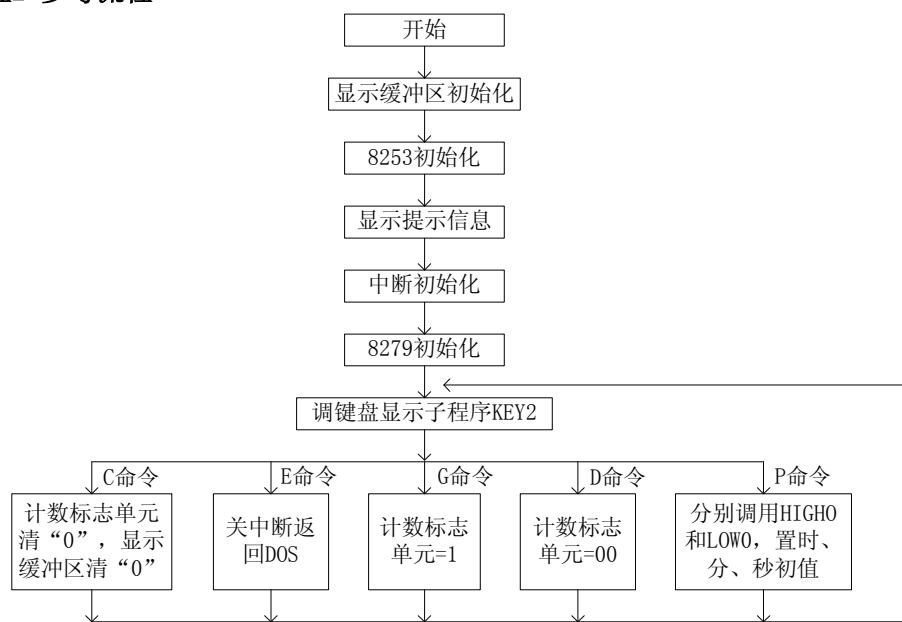


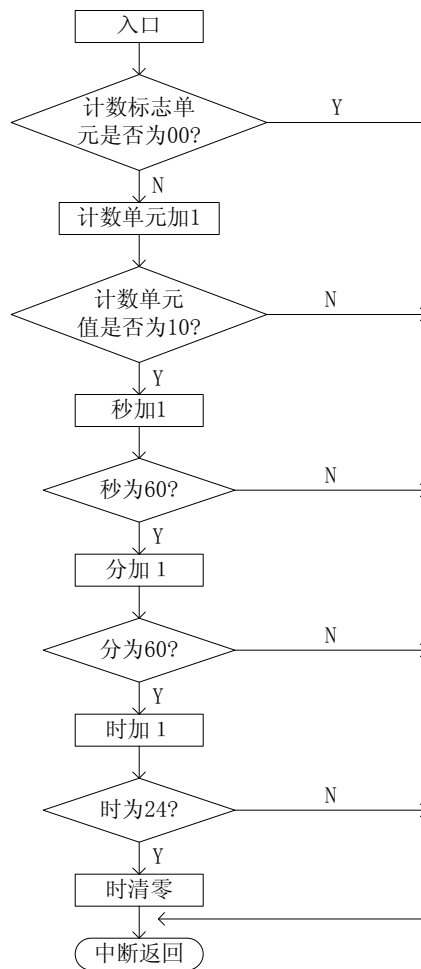
图20-3 键盘显示子程序流程图:KEY2

六、编程2 参考流程

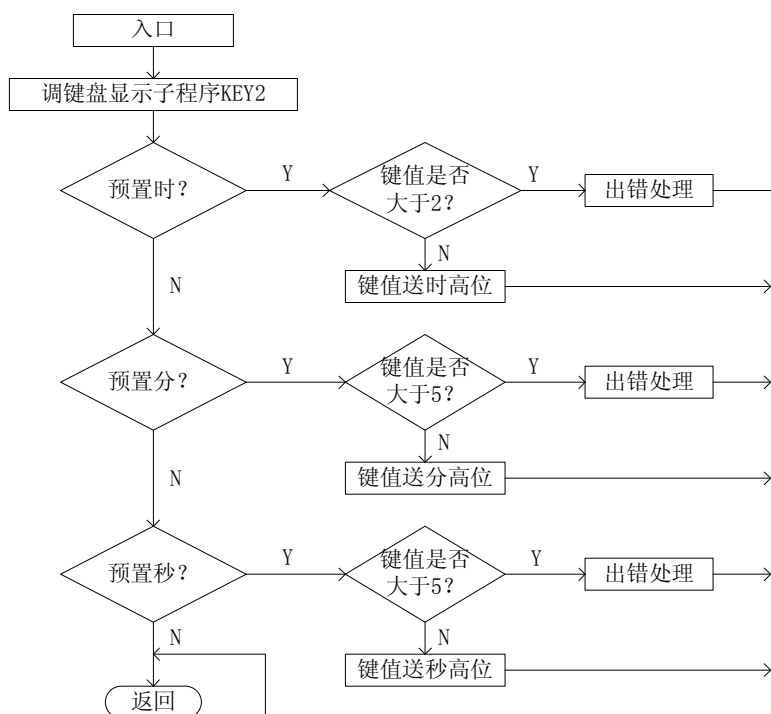


主程序流程图：





中断处理子程序：



预置时、分、秒高位子程序HIGH0:

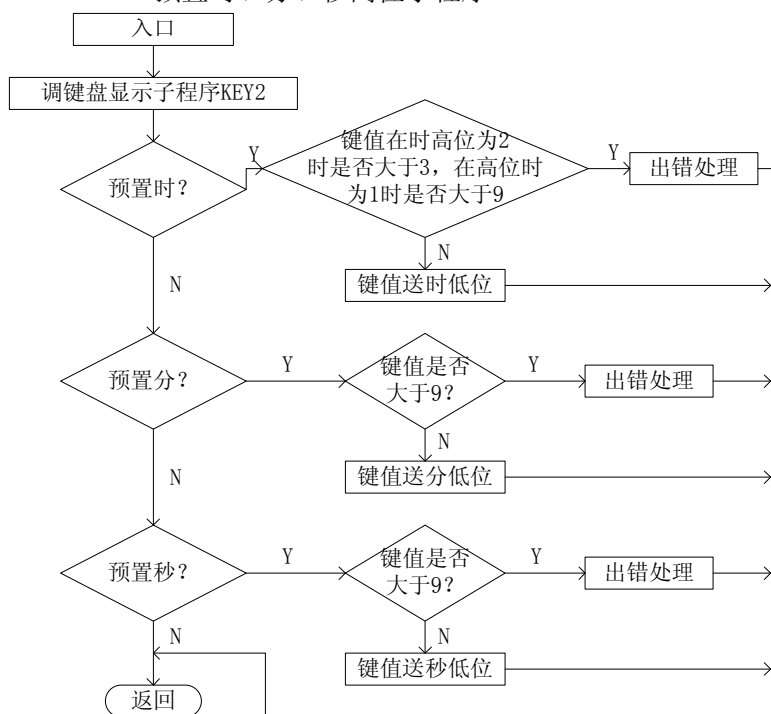


图20-4 预置时、分、秒低位子程序LOW0:

## 实验二十一 存储器读写实验

### 一、实验目的

- 1、熟悉6116静态RAM的使用方法，掌握PC机外存扩充的手段。
- 2、通过对硬件电路的分析，学习了解总线的工作时序。

### 二、实验内容

- 1、硬件电路如下：

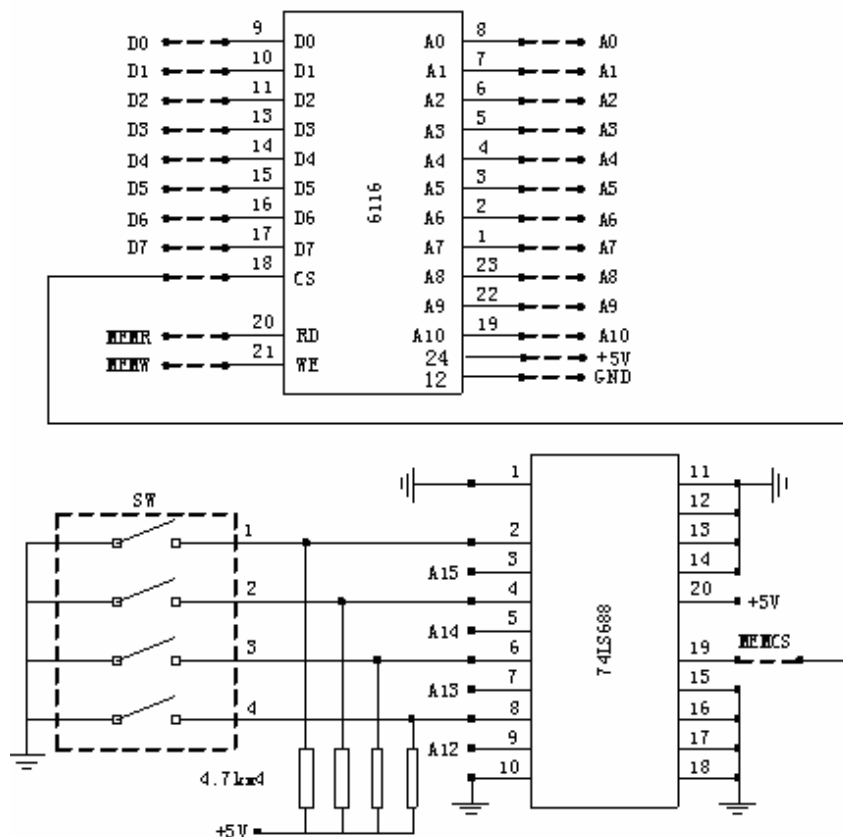


图21-1

2、编制程序，将字符A-Z循环写入扩展的6116RAM中，然后再将6116的内容读出来显示在主机屏幕上。

### 三、编程提示

- 1、注意:TPC-USB已为扩展的6116指定了段地址:0D000H。
- 2、TPC-USB模块外扩储器的默认起始地址为D4000H，即在程序中如访问D6000H，只需通过零地址addr=0，再加上2000H即可，即addr+2000H。（注TPC-USB模块外扩存储器的范围为0D4000H-0D7fffH。

3、通过片选信号的产生方式，确定6116RAM在PC机系统中的地址范围。因为段地址已指定，所以其地址为CS=A15 and A14 and A13 and A12，实验台上设有地址选择微动开关K2，拨动开关，可以选择4000-7fff的地址范围。编制程序，从0d6000H开始循环写入100h个A-Z。开关状态如下：

1	2	3	4	地址
ON	OFF	ON	OFF	d4000h
ON	OFF	OFF	ON	d6000h

#### 四、程序框图

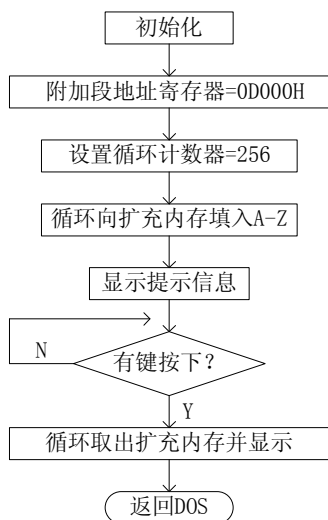


图21-2

## 实验二十二 双色点阵发光二极管显示实验

### 一、实验目的

- 1、了解双色点阵LED显示器的基本原理。
- 2、掌握PC机控制双色点阵LED显示程序的设计方法。

### 二、实验原理

点阵LED显示器是将许多LED类似矩阵一样排列在一起组成的显示器件，双色点阵LED是在每一个点阵的位置上有红绿或红黄或红白两种不同颜色的发光二极管。当微机输出的控制信号使得点阵中有些LED发光，有些不发光，即可显示出特定的信息，包括汉字、图形等。车站广场由微机控制的点阵LED大屏幕广告宣传牌随处可见。

实验仪上设有一个共阳极8×8点阵的红黄两色LED显示器，其点阵结构如图所示。该点阵对外引出24条线，其中8条行线，8条红色列线，8条黄色列线。若使某一种颜色、某一个LED发光，只要将与其相连的行线加高电平，列线加低电平即可。

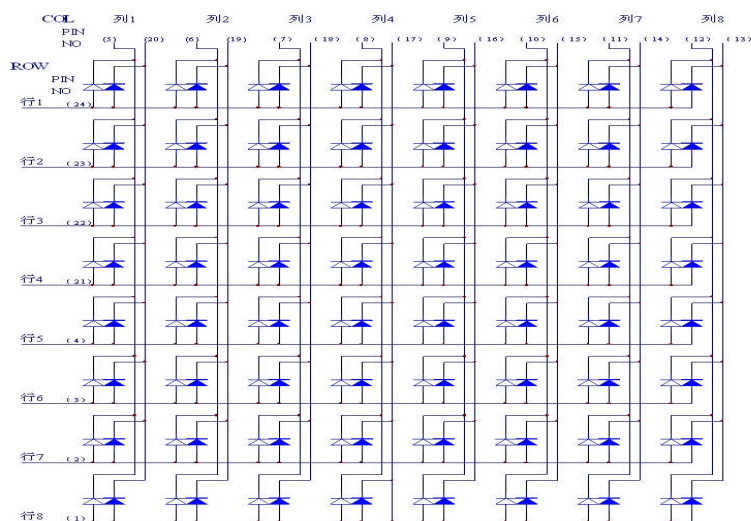


图22-1

例如欲显示汉字“年”，采用逐列循环发光。首先由“年”的点阵轮廓，确定点阵代码(如图所示)根据“年”的点阵代码，确定逐列循环发光的顺序如下：

- ① 行代码输出 44H； 红色列代码输 01H； 第一列2个红色LED发光。
- ② 行代码输出 54H； 红色列代码输 02H； 第二列3个红色LED发光。
- ③ 行代码输出 54H； 红色列代码输 04H； 第三列3个红色LED发光。
- ④ 行代码输出 7FH； 红色列代码输 08H； 第四列7个红色LED发光。
- ⑤ 行代码输出 54H； 红色列代码输 10H； 第五列3个红色LED发光。
- ⑥ 行代码输出 DCH； 红色列代码输 20H； 第六列5个红色LED发光。
- ⑦ 行代码输出 44H； 红色列代码输 40H； 第七列2个红色LED发光。
- ⑧ 行代码输出 24H； 红色列代码输 80H； 第八列2个红色LED发光。

在步骤①～⑧之间可插入几ms的延时，重复进行①～⑧即可在LED上稳定的显示出红色“年”字。若想显示黄色“年”，只需把红色列码改为黄色列码即可。

三、实验内容：

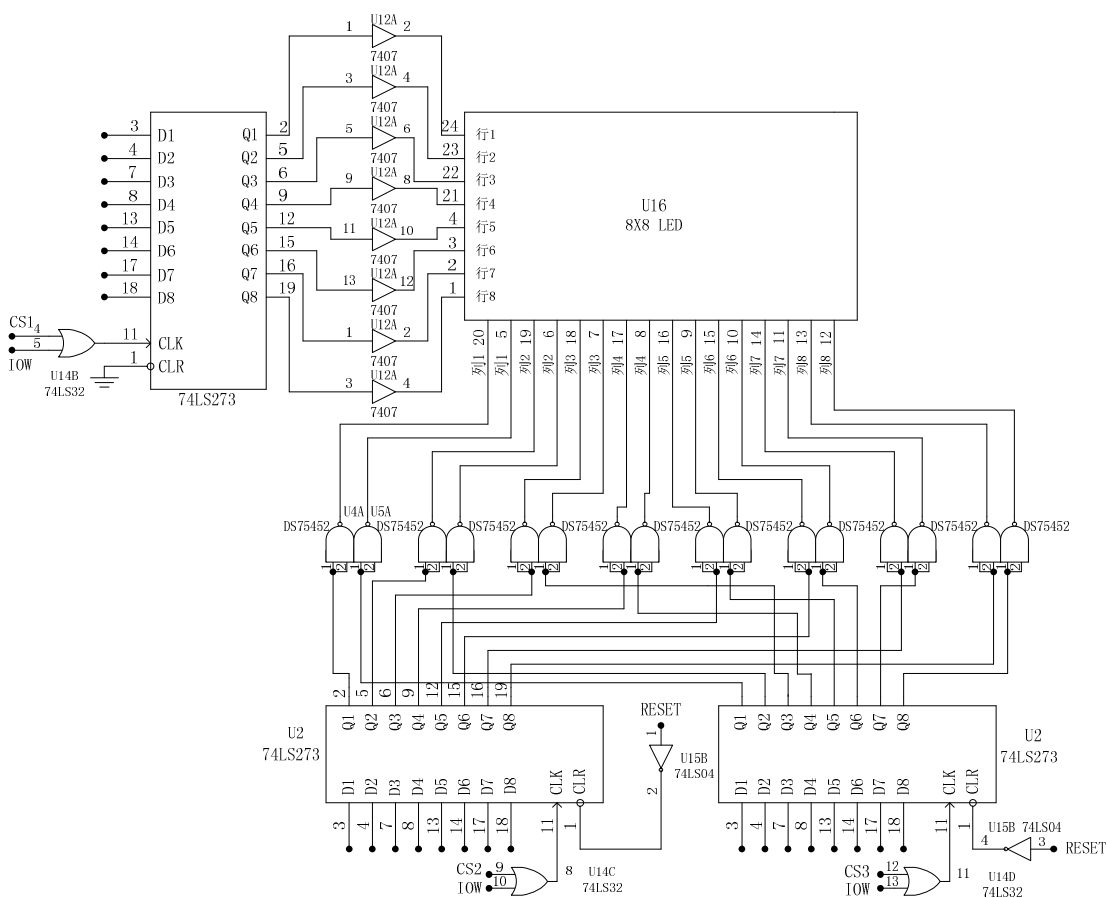


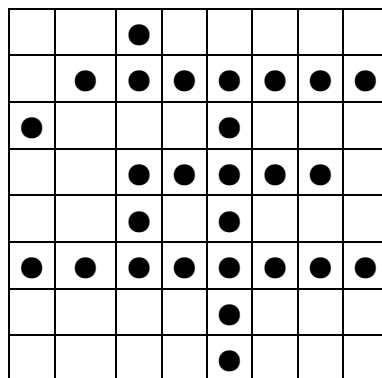
图22-2

1、实验仪上的点阵LED及驱动电路如下图所示，行代码、红色列代码、黄色列代码各用一片74LS273锁存。行代码输出的数据通过行驱动器7407加至点阵的8条行线上，红和黄列代码的输出数据通过驱动器DS75452反相后分别加至红和黄的列线上。行锁存器片选信号为CS1，红色列锁存器片选信号为CS2，黄色列锁存器片选信号为CS3。

2、接线方法:行片选信号 CS1 接 280H；红列片选信号 CS2 接 288H；黄列片选信号 CS3 接 290H。

3、编程重复使LED点阵红色逐列点亮，再黄色逐列点亮，再红色逐行点亮，黄色逐行点亮。

4、编程在LED上重复显示红色“年”和黄色“年”。



四、逐行、逐列显示参考流程图

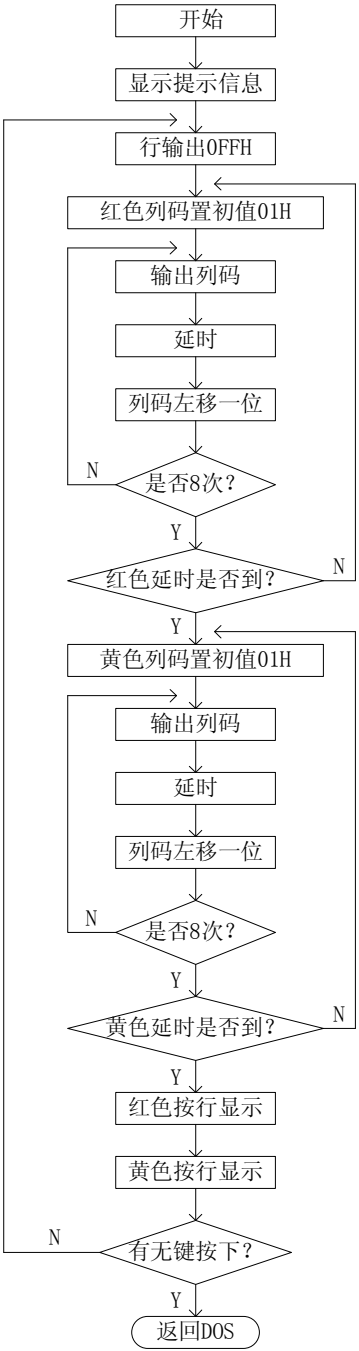


图22-3

## 附录一、随机光盘实验程序名称表

汇编程序与实现相同/类似功能的C程序文件对照

	汇编程序	说明及对应C程序
1	Ymq.asm	io地址译码 Ymq.cpp
2	E244.asm	简单并行口244 E244.cpp
3	E273.asm	简单并行口273 E273.cpp
4	E8253_1.asm	可编程定时器_计数器 E8253_1.cpp
5	E8253_2.asm	可编程定时器_计数器 E8253_2.cpp
6	E8255.asm	可编程并行口1_8255方式0 E8255.cpp
7	Led1.asm	七段数码管1 LED1.cpp
8	Led2.asm	七段数码管2 LED2.cpp
9	Led3.asm	七段数码管3 LED3.cpp
10	Jdq.asm	继电器 JDQ.cpp
11	Qdq.asm	竞赛抢答器 QDQ.cpp
12	Jtd.asm	交通灯控制 JTD.cpp
13	Int.asm	中断 INT.cpp
14	E8255_1o.asm	可编程并行口2_8255方式1_1 E8255_1o.cpp
15	E8255_1i.asm	可编程并行口2_8255方式1_2 E8255_1i.cpp
16	Da_1.asm	数模转换1 DA_1.cpp
17	Da_2.asm	数模转换2 DA_2.cpp
18	Ad_1.asm	模数转换1 AD_1.cpp
19	Ad_2.asm	模数转换2 AD_2.cpp
20		
21	E8251.asm	8251串行通信 E8251.cpp
22	Dma.asm	DMA传送1 DMA.cpp
23	Dma_o.asm	DMA传送2 DMA_0.cpp
24	Dma_i.asm	DMA传送3 DMA_I.cpp
25	Jc.asm	集成电路测试 JC.cpp
26	Dzq.asm	电子琴 DZQ.cpp
27	E8250.asm	8250串行通信 E8250.cpp
28	Bj dj.asm	步进电机 BJDJ.cpp
29	Zl dj.asm	直流电机 ZLDJ.cpp
30	Jpxsh1.asm	8279 键盘显示控制器1 JPXSH1.cpp
31	Jpxsh.asm	8279 键盘显示控制器2 JPXSH.cpp
32	Memrwex.asm	存储器读写 MEMRWEX.cpp
33	11588.ASM	点阵实验一 11588.cpp
34	11588-1.ASM	点阵实验二 11588-1.cpp
35	KEY.ASM	8255A并行口键盘扫描实验一



36	KEY1. ASM	8255A并行口键盘扫描实验二
37	LED. ASM	总线控制LED显示实验一
38	LED4. ASM	总线控制LED显示实验二
39	KL. ASM	微机接口、键盘、LED综合实验—I/O编程
40	KL1. ASM	微机接口、键盘、LED综合实验—中断编程
41	CHLCD. ASM	字符液晶模块实验一
42	CHLCD1. ASM	字符液晶模块实验二
43	GRLCD. ASM	测试液晶模块及屏幕像素程序的编程
44	GRLCD1. ASM	汉字字符显示编程
45	GRLCD2. ASM	图形显示编程
46	GRLCD3. ASM	特效显示编程
47	LCDCTR. ASM	液晶显示屏手动控制对比度实验
48	LCDCTR1. ASM	液晶显示屏程序控制对比度实验
49	LCDBL. ASM	图形液晶程序控制背光实验
51	BMKEY. ASM	薄膜按键开关实验
52	Wuxian. asm	无线通讯实验
53	Hongwai-q. asm	红外模块实验

## 附录二：TPC-USB通用微机接口实验系统硬件实验指导（汇编程序）

汇编语言支持winnt/2000/xp。

参照实验指导书安装TPC-USB模块及其驱动程序后才能正常运行程序。

说明：开关K向上为“1”，向下为“0”。

### 一、I/O地址译码

连线：CD---2A8H-2AFH SD-- +5V D-- +5V Q---L7 CLK---2A0H-2A7H

运行程序：YMQ地址译码

运行结果：发光二极管L7有规律的进行连续闪烁。

### 二、简单并行接口

（1）将74LS273插在相关插座上

连线：D0-D7---74LS273的（3、4、7、8、13、14、17、18）脚 或门输入A---2A8H-2AFH

或门输入B---/IOW 或门输出Y---11脚 10脚---GND 1、20脚-- +5V

L0-L7---74LS273的（2、5、6、9、12、15、16、19）脚

运行程序：E273简单并行口程序

运行结果：拨动开关K，相应置“1”的开关所对应的灯L亮，否则灭。

（2）将74LS244插在相关插座上

连线：K0-K7---74LS244（2、4、6、8、11、13、15、17）脚 或门输入A---/IOR

或门输入B---2A0H-2A7H 或门输出Y---1、19脚 10脚---GND 20脚-- +5V

D0-D7---74LS244（18、16、14、12、9、7、5、3）脚

运行程序：E244简单并行口程序

运行结果：用开关输入字母的ASCII码值，在屏幕上显示对应的字母。

### 三、可编程定时器/计数器

（1）连线：CLK0---正脉冲 CS---280H-287H GATE0-- +5V

运行程序：E8253\_1可编程定时器/计数器程序

运行结果：压单脉冲键，每压一次在屏幕上循环显示“1-9”、“A-F”，把逻辑笔插入

逻辑

孔，用控制端测试OUT0，压15次单脉冲键后，可以看到OUT0的电平变化一次。

（2）连线：CS---280H-287H GATE1-- +5V GATE0-- +5V CLK1---OUT0

CLK0---1MHZ

运行程序：E8253\_2可编程定时器/计数器程序

运行结果：把逻辑笔插入逻辑孔，用探测端测试OUT1，可以看到OUT1的电平有规律的进行高低交替变化。

（3）连线：CS---280H-287H GATE1-- +5V GATE0-- +5V CLK1---OUT0

CLK0---1MHZ

运行程序：E8253\_3可编程定时器/计数器程序

运行结果：如下所示：

counter1:\_\_\_\_

counter2:\_\_\_\_

continue?(y/n)\_\_\_\_

用示波器观察不同方式下的波形，并在纸上画出CLK0、OUT0、OUT1的波形。

### 四、可编程并行接口（一）8255

（1）连线：PC0-PC7---K0-K7 CS---288H-28FH PA0-PA7---L0-L7

运行程序：E8255可编程并行口（一）程序

运行结果：拨动开关K，相应置“1”的开关所对应的灯L亮，否则灭。

（2）连线：利用通用插座插好74LS138芯片，将8255的PC0-PC7---138的（15~7）脚、8255CS---288H-28FH 分别将8255的PA3-PA7---138的（1、2、3、6、4）脚、138的16脚---+5、8脚—GND、4脚—5脚对连。

运行程序：E8255\_1可编程并行口（一）程序

运行结果：

```
+-----+
|  A   B   C   G1  G2A+G2B |
|                               |
|           74LS138           |
|                               |
| Y0  Y1  Y2  Y3  Y4  Y5  Y6  Y7 |
+-----+
```

Test Again ?(Y/N)', '\$'

显示由8255的A口输出，由C口读入的数据。

## 五、七段数码管

（1）连线：PA0-PA6---a-g S1--- +5 S0---GND DP---GND CS---288H-28FH

运行程序：LED1七段数码管程序

运行结果：从电脑键盘上输入0-9，并在七段数码管上直接显示出来。

（2）连线：PA0-PA6---a-g S1---PC1 S0---PC0 DP---GND CS---288H-28FH

运行程序：LED2七段数码管程序

运行结果：在七段数码管上按秒循环显示00-99。

## 六、继电器控制

连线：CLK0---1MHZ GATE0--- +5V OUT0---CLK1 GATE1--- +5V

8253CS---280H-287H OUT1---PA0 8255CS---288H-28FH PC0---IK（右上角）

运行程序：JDQ继电器程序

运行结果：继电器周而复始的闭合5秒，同时指示灯亮，断开5秒，灯灭。

## 七、竞赛抢答器

连线：DP-GND PA0-PA6---a-g PC0-PC7---K0-K7 S1--- +5V

8255CS---288H-28FH

运行程序：QDQ竞赛抢答器程序

运行结果：当某开关置“1”，表示有按键按下，在数码管上显示其组号，同时喇叭响一下。

## 八、交通灯控制实验

连线：DP-GND PC0-PC7---K0-K7 S1--- +5V

8255CS---288H-28FH

运行程序：JTD交通灯控制程序

运行结果：指示灯按交通灯规律进行亮灭。

## 九、中断

连线：单脉冲---IRQ

运行程序：INT中断程序

运行结果：每按一次中断，输出“TPC-USB INTERRUPT！”

## 十、可编程并行接口（二）8255

(1) 连线: PC3---IRQ PC6---单脉冲 PA0-PA7---L0-L7 8255CS---288H-28FH  
运行程序: E8255-1o可编程并行接口程序

运行结果: 按一次单脉冲键, 让CPU进行一次中断, 使L0-L7依次发光, 8次后结束。

(2) 连线: PA0-PA7---K0-K7 PC3---IRQ PC4---单脉冲 8255CS---288H-28FH  
运行程序: E8255-1i可编程并行接口程序

运行结果: 按一次单脉冲键, 让CPU进行一次中断, 读取开关预置的ASCII码, 在屏幕上显示其对应的字母, 中断8次后结束。

#### 十一、数/模转换器

(1) 连线: DAC0832CS---290H-297H

运行程序: DA\_1数模转换程序

运行结果: 用示波器观察, 单极输出端Ua及双极输出端Ub, 均输出刷齿波。

(2) 连线: DAC0832CS---290H-297H

运行程序: DA\_2数模转换程序

运行结果: 用示波器观察, 单极输出端Ua及双极输出端Ub, 均输出正弦波。

#### 十二、模/数转换器

(1) 连线: ADC0809CS---298H-29FH RW1---IN0

运行程序: AD\_1模/数转换程序

运行结果: 采集IN0输入的电压, 在屏幕上显示转换后的数据。

(2) 连线: ADC0809CS---298H-29FH RW1---IN0

运行程序: AD\_2模/数转换程序

运行结果: 将JP3 (左下方) 的1、2短路使IN2处于双极性工作方式, 给IN1输入一个低频交流信号 (幅度为+5 V), 编程采集这个数据并在屏幕上显示对应的幅度。

#### 十三、串行通讯

将8251芯片插在相应的通用插座上

连线: 8253OUT0---8251的9、25脚 GATE0--- +5V 8253CS---280H-287H

CLK0---1MHZ 8251的3脚---19脚 17脚--- GND 26脚--- +5V 4脚---GND

20脚---1MHZ 21脚--- RESET 12脚---A0 10脚---/IOW 13脚---/IOR

11脚---2B8H-2BFH 27、28、1、2脚---D0-D3 5-8脚---D4---D7

运行程序: E8251串行通信程序

运行结果: 从键盘输入一个字符, 将其ASCII码加1后发出去, 再接收回来在计算机屏幕上显示, 实现自收发。(如: 敲a, 显示ab)。

#### 十四、DMA传送

(1) 连线: D0-D7---扩展6116 (9、10、11、13、14、15、16、17) 脚

A0-A10---扩展6116 (8、7、6、5、4、3、2、1、23、22、19) 脚

MEMW--- 6116的21脚、MEMR--- 6116的20脚、MEMCS---6116的18脚

6116的24脚---+5、12脚---GND。

运行程序: DMA存储器到实验箱上扩展6116块传送程序

运行结果: 显示源数据区256个A-Z, 显示目的数据区256个a, 显示传送后目的256个A-Z。

(2) 将芯片74LS273插在相应的通用插座上

连线: D0-D7---74LS273 (3、4、7、8、13、14、17、18) 脚 或门的输入A---/DACK1

或门的输入B---/IOW 或门的输出Y---11脚 1、20脚--- +5V 10脚--- GND

L0-L7---2、5、6、9、12、15、16、19脚 D、SD--- +5V Q--- DRQ1 CD---/DACK1

CLK---正单脉冲

运行程序: DMA\_0写程序

运行结果: 按压单脉冲键, 使发光二极管L0-L7依次发光, 10次后结束。

(3) 将芯片74LS244插在通用插座上

连线: K0-K7---74LS244的2、4、6、8、11、13、15、17脚 或门输入A---/DACK1  
D0-D7---74LS244的18、16、14、12、9、7、5、3脚 或门输入B---/IOR  
20脚---+5V 10脚---GND 或门输出Y---1、19脚 CLK---正单脉冲  
D、SD---+5V CD---/DACK1 Q---DRQ1

运行程序: DMA\_I读程序

运行结果: 每压一次单脉冲键, 便传送一次开关输入的值, 显示开关所对应的ASCII码的字母, 传送8次后结束。

## 十五、集成电路测试

将被测试的芯片74LS00插在通用插座上

连线: 8255CS---288H-28FH PC0~PC3---8、11、3、6脚 14脚---+5V  
PA0-PA7---1、2、4、5、9、10、12、13脚 7脚---GND

运行程序: JC集成电路测试程序

运行结果: 检测芯片的好坏, 好则显示“This chip is ok”, 否则显示“This chip is bad”。

## 十六、电子琴

连线: 8253CS---280H-287H CLK0---1MHZ GATE0---8255 PA1 OUT0---与门一入端口  
8255CS---288H-28FH PA0---与门二入端 与门输出端---K8 喇叭

运行程序: DZQ电子琴程序

运行结果: 在计算机的键盘上按数字键可以象弹钢琴一样发出不同音调。

## 十七、8250串行通讯实验

将8250芯片插在40芯通用插座上

连线: D0-D7---1-8脚 A0-A2---28-26脚 38、39、40脚---+5V  
/IOR---21脚 /IOW---18脚 14脚---2B8H-2BFH  
19、22、25、36、37、20脚---GND 11脚---10脚 15脚---9脚  
16脚---2MHZ 35脚---RESET 34、31、13、12脚---+5V

运行程序: E8250串行通讯程序

运行结果: 从键盘输入一个字符, 将其ASCII码值加1后发出去, 再自动接收回来在计算机的屏幕上显示出来, 实现自收发。

## 十八、步进电机控制实验

连线: 8255CS---288H-28FH PA0-PA3---BA-BD PC0-PC7---K0-K7

运行程序: BJDJ步进电机程序

运行结果: K0-K6控制步进电机的转速, K7控制其转向。

## 十九、直流电机转速控制实验

连线: DAC0832CS---290H-297H Ub---DJ 8255CS---288H-28FH  
K0-K5---PC0-PC5

运行程序: ZLDJ直流电机程序

运行结果: K0-K5控制直流电机的转速。

## 二十、键盘显示控制实验

(1) 连线: 将小键盘上的20芯扁平电缆和实验台上的J7相连(注意接线别接反)。

运行程序: JPXSH1键盘显示程序

运行结果: 按小键盘上键, 数码管上显示相应的字符(如按7, 则显示7)。

(2) 连线: 将小键盘上的20芯扁平电缆和实验台上的J7相连。

8253CS---280H-287H CLK0---1MHZ GATE0、GATE1---+5V OUT0---CLK1  
OUT1---IRQ

运行程序：JPXSH键盘显示程序

运行结果：按电子钟格式显示：XX XX XX 时、分、秒

C键：清除 显示全零 G键：启动 电子钟计时

D键：停止 停止计时 P键：可以自由设置时、分、秒并且可以判断设置的数据的对错，出错显示E-----，此时敲P键可重新输入预置值

E键：程序退出，返回

## 二十一、存储器读写

扩展6116插在40芯通用插座上

连线：6116的8-1脚---A0-A7 23脚---A8 22脚---A9 19脚---A10

9-11脚---D0-D2 13-17脚---D3-D7 24脚---+5V 12、18脚---GND

21脚---/MEMW 20脚---MEMR

运行程序：MEMRWEX

运行结果：显示256个A-Z

## 二十二、双色点阵发光二极管显示实验

连线：行片选信号 CS1 接 280H；红列片选信号 CS2 接 288H；黄列片选信号 CS3 接 290H。

运行程序：11588、11588-1

运行结果：

1、LED点阵红色逐列点亮，再黄色逐列点亮，再红色逐行点亮，黄色逐行点亮。

2、LED上重复显示红色“年”和黄色“年”。

（以上仅供参考，更多内容请参看实验指导书）

## 附录三：TPC-USB通用微机接口实验系统硬件实验指导（C语言程序）

C语言程序支持winnt/2000/xp。

参照实验指导书安装TPC-USB模块及其驱动程序后才能正常运行程序。

说明：开关K向上为“1”，向下为“0”。

### 一、I/O地址译码

连线：CD---2A8H-2AFH SD-- +5V D-- +5V Q---L7 CLK---2A0H-2A7H

运行程序：YMQ地址译码

运行结果：发光二极管L7有规律的进行连续闪烁。

### 二、简单并行接口

（1）将74LS273插在相关插座上

连线：D0-D7---74LS273的（3、4、7、8、13、14、17、18）脚 或门输入A---2A8H-2AFH  
或门输入B---/IOW 或门输出Y---11脚 10脚---GND 1、20脚---+5V

L0-L7---74LS273的（2、5、6、9、12、15、16、19）脚

运行程序：E273简单并行口程序

运行结果：拨动开关K，相应置“1”的开关所对应的灯L亮，否则灭。

（2）将74LS244插在相关插座上

连线：K0-K7---74LS244（2、4、6、8、11、13、15、17）脚 或门输入A---/IOR  
或门输入B---2A0H-2A7H 或门输出Y---1、19脚 10脚---GND 20脚---+5V

D0-D7---74LS244（18、16、14、12、9、7、5、3）脚

运行程序：E244简单并行口程序

运行结果：用开关输入字母的ASCII码值，在屏幕上显示对应的字母。

### 三、可编程定时器/计数器

(1) 连线: CLK0---正脉冲 CS---280H-287H GATE0--- +5V

运行程序: E8253\_1可编程定时器/计数器程序

运行结果: 压单脉冲键, 每压一次在屏幕上循环显示“1-9”、“A-F”, 把逻辑笔插入

逻辑

孔, 用控制端测试OUT0, 压15次单脉冲键后, 可以看到OUT0的电平变化一次。

(2) 连线: CS---280H-287H GATE1--- +5V GATE0--- +5V CLK1---OUT0

CLK0---1MHZ

运行程序: E8253\_2可编程定时器/计数器程序

运行结果: 把逻辑笔插入逻辑孔, 用探测端测试OUT1, 可以看到OUT1的电平有规律的进行高低交替变化。

#### 四、可编程并行接口 (一) 8255

连线: PC0-PC7---K0-K7 CS---288H-28FH PA0-PA7---L0-L7

运行程序: E8255可编程并行口 (一) 程序

运行结果: 拨动开关K, 相应置“1”的开关所对应的灯L亮, 否则灭。

#### 五、七段数码管

(1) 连线: PA0-PA6---a-g S1--- +5 S0---GND DP---GND CS---288H-28FH

运行程序: LED1七段数码管程序

运行结果: 从电脑键盘上输入0-9, 并在七段数码管上直接显示出来。

(2) 连线: PA0-PA6---a-g S1---PC1 S0---PC0 DP---GND CS---288H-28FH

运行程序: LED2七段数码管程序

运行结果: 在七段数码管上按秒循环显示00-99。

#### 六、继电器控制

连线: CLK0---1MHZ GATE0--- +5V OUT0---CLK1 GATE1--- +5V

8253CS---280H-287H OUT1---PA0 8255CS---288H-28FH PC0---IK (右上角)

运行程序: JDQ继电器程序

运行结果: 继电器周而复始的闭合5秒, 同时指示灯亮, 断开5秒, 灯灭。

#### 七、竞赛抢答器

连线: DP-GND PA0-PA6---a-g PC0-PC7---K0-K7 S1--- +5V

8255CS---288H-28FH

运行程序: QDQ竞赛抢答器程序

运行结果: 当某开关置“1”, 表示有按键按下, 在数码管上显示其组号, 同时喇叭响一下。

#### 八、交通灯控制实验

连线: DP-GND PC0-PC7---K0-K7 S1--- +5V

8255CS---288H-28FH

运行程序: JTD交通灯控制程序

运行结果: 指示灯按交通灯规律进行亮灭。

#### 九、中断

(1) 连线: 8255PA0-PA7---L0-L7 PC0---正单脉冲 8255CS---288H-28FH

运行程序: INT0查询法中断程序

运行结果: 单脉冲每按一次, 指示灯显示0x55, 否则指示灯显示0xAA

(2) 连线: 8255PA0-PA7---L0-L7 8255CS---288H-28FH 正单脉冲---IRQ

运行程序: INT1中断法中断程序

运行结果: 单脉冲每按一次, 指示灯显示0x55, 否则指示灯显示0xAA

#### 十、可编程并行接口 (二) 8255

(1) 连线: PC3---IRQ PC6---单脉冲 PA0-PA7---L0-L7 8255CS---288H-28FH  
运行程序: E8255-1o可编程并行接口程序

运行结果: 按一次单脉冲键, 让CPU进行一次中断, 使L0-L7依次发光, 8次后结束。

(2) 连线: PA0-PA7---K0-K7 PC3---IRQ PC4---单脉冲 8255CS---288H-28FH  
运行程序: E8255-1i可编程并行接口程序

运行结果: 按一次单脉冲键, 让CPU进行一次中断, 读取开关预置的ASCII码, 在屏幕上显示其对应的字母, 中断8次后结束。

#### 十一、数/模转换器

(1) 连线: DAC0832CS---290H-297H

运行程序: DA\_1数模转换程序

运行结果: 用示波器观察, 单极输出端Ua及双极输出端Ub, 均输出刷齿波。

(2) 连线: DAC0832CS---290H-297H

运行程序: DA\_2数模转换程序

运行结果: 用示波器观察, 单极输出端Ua及双极输出端Ub, 均输出正弦波。

#### 十二、模/数转换器

(1) 连线: ADC0809CS---298H-29FH RW1---IN0

运行程序: AD\_1模/数转换程序

运行结果: 采集IN0输入的电压, 在屏幕上显示转换后的数据。

(2) 连线: ADC0809CS---298H-29FH RW1---IN0

运行程序: AD\_2模/数转换程序

运行结果: 将JP3 (左下方) 的1、2短路使IN2处于双极性工作方式, 给IN1输入一个低频交流信号 (幅度为+5 V), 编程采集这个数据并在屏幕上显示对应的幅度。

#### 十三、串行通讯

将8251芯片插在相应的通用插座上

连线: 8253OUT0---8251的9、25脚 GATE0--- +5V 8253CS---280H-287H

CLK0---1MHZ 8251的3脚---19脚 17脚--- GND 26脚--- +5V 4脚---GND

20脚---1MHZ 21脚--- RESET 12脚---A0 10脚---/IOW 13脚---/IOR

11脚---2B8H-2BFH 27、28、1、2脚---D0-D3 5-8脚---D4---D7

运行程序: E8251串行通信程序

运行结果: 从键盘输入一个字符, 将其ASCII码加1后发出去, 再接收回来在计算机屏幕上显示, 实现自收发。(如: 敲a, 显示ab)。

#### 十四、DMA传送

(1) 连线: D0-D7---扩展6116 (9、10、11、13、14、15、16、17) 脚

A0-A10---扩展6116 (8、7、6、5、4、3、2、1、23、22、19) 脚

MEMW--- 6116的21脚、MEMR--- 6116的20脚、MEMCS---6116的18脚

6116的24脚---+5、12脚---GND。

运行程序: DMA存储器到实验箱上扩展6116块传送程序

运行结果: 显示源数据区256个A-Z, 显示目的数据区256个a, 显示传送后目的256个A-Z。

(2) 将芯片74LS273插在相应的通用插座上

连线: D0-D7---74LS273 (3、4、7、8、13、14、17、18) 脚 或门的输入A---/DACK1

或门的输入B---/IOW 或门的输出Y---11脚 1、20脚--- +5V 10脚--- GND

L0-L7---2、5、6、9、12、15、16、19脚 D、SD--- +5V Q--- DRQ1 CD---/DACK1

CLK---正单脉冲

运行程序: DMA\_0写程序

运行结果: 按压单脉冲键, 使发光二极管L0-L7依次发光, 10次后结束。



(3) 将芯片74LS244插在通用插座上

连线: K0-K7---74LS244的2、4、6、8、11、13、15、17脚 或门输入A---/DACK1

D0-D7---74LS244的18、16、14、12、9、7、5、3脚 或门输入B---/IOR

20脚---+5V 10脚---GND 或门输出Y---1、19脚 CLK---正单脉冲

D、SD---+5V CD---/DACK1 Q---DRQ1

运行程序: DMA\_I读程序

运行结果: 每压一次单脉冲键, 便传送一次开关输入的值, 显示开关所对应的ASCII码的字母, 传送8次后结束。

## 十五、集成电路测试

将被测试的芯片74LS00插在通用插座上

连线: 8255CS---288H-28FH PC0~PC3---8、11、3、6脚 14脚---+5V

PA0-PA7---1、2、4、5、9、10、12、13脚 7脚---GND

运行程序: JC集成电路测试程序

运行结果: 检测芯片的好坏, 好则显示“This chip is ok”, 否则显示“This chip is bad”。

## 十六、电子琴

连线: 8253CS---280H-287H CLK0---1MHZ GATE0---8255 PA1 OUT0---与门二入端口 8255CS---288H-28FH PA0---与门二入端 与门输出端---K8 喇叭

运行程序: DZQ电子琴程序

运行结果: 在计算机的键盘上按数字键可以象弹钢琴一样发出不同音调。

## 十七、8250串行通讯实验

将8250芯片插在40芯通用插座上

连线: D0-D7---1-8脚 A0-A2---28-26脚 38、39、40脚---+5V

/IOR---21脚 /IOW---18脚 14脚---2B8H-2BFH

19、22、25、36、37、20脚---GND 11脚---10脚 15脚---9脚

16脚---2MHZ 35脚---RESET 34、31、13、12脚---+5V

运行程序: E8250串行通讯程序

运行结果: 从键盘输入一个字符, 将其ASCII码值加1后发出去, 再自动接收回来在计算机的屏幕上显示出来, 实现自收发。

## 十八、步进电机控制实验

连线: 8255CS---288H-28FH PA0-PA3---BA-BD PC0-PC7---K0-K7

运行程序: BJDJ步进电机程序

运行结果: K0-K6控制步进电机的转速, K7控制其转向

## 十九、直流电机转速控制实验

连线: DAC0832CS---290H-297H Ub---DJ 8255CS---288H-28FH

K0-K5---PC0-PC5

运行程序: ZLDJ直流电机程序

运行结果: K0-K5控制直流电机的转速。

## 二十、键盘显示控制实验

(1) 连线: 将小键盘上的20芯扁平电缆和实验台上的J7相连(注意接线别接反)。

运行程序: JPXSH1键盘显示程序

运行结果: 按小键盘上键, 数码管上显示相应的字符(如按7, 则显示7)。

(2) 连线: 将小键盘上的20芯扁平电缆和实验台上的J7相连。

8253CS---280H-287H CLK0---1MHZ GATE0、GATE1---+5V OUT0---CLK1

OUT1---IRQ

运行程序：JPXSH键盘显示程序

运行结果：按电子钟格式显示：XX XX XX 时、分、秒

C键：清除 显示全零 G键：启动 电子钟计时

D键：停止 停止计时 P键：可以自由设置时、分、秒并且可以判断 设置的数据的对错，出错显示E-----，此时敲P键可重新输入预置值E键：程序退出，返回

## 二十一、存储器读写

扩展6116插在40芯通用插座上

连线：6116的8-1脚---A0-A7 23脚---A8 22脚---A9 19脚---A10

9-11脚---D0-D2 13-17脚---D3-D7 24脚--- +5V 12、18脚---GND

21脚---/MEMW 20脚---MEMR

运行程序：MEMRWEX

运行结果：显示256个A-Z

## 二十二：双色点阵发光二极管显示实验

连线：行片选信号 CS1 接 280H；红列片选信号 CS2 接 288H；黄列片选信号 CS3 接 290H。

运行程序：11588、11588-1

运行结果：

3、LED点阵红色逐列点亮，再黄色逐列点亮，再红色逐行点亮，黄色逐行点亮。

4、LED上重复显示红色“年”和黄色“年”。

（以上仅供参考，更多内容请参看实验指导书）